A primal-dual interior-point relaxation method with global and rapidly local convergence for nonlinear programs

Xin-Wei Liu¹ · Yu-Hong Dai² · Ya-Kui Huang¹

Received: 20 February 2021 / Revised: 22 June 2022 / Accepted: 4 August 2022 / Published online: 30 August 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
Based on solving an equivalent parametric equality constrained mini-max problem of the classic logarithmic-barrier subproblem, we present a novel primal-dual interior-point relaxation method for nonlinear programs with general equality and nonnegative constraints. In each iteration, our method approximately solves the KKT system of a parametric equality constrained mini-max subproblem, which avoids the requirement that any primal or dual iterate is an interior-point. The method has some similarities to the warmstarting interior-point methods in relaxing the interior-point requirement and is easily extended for solving problems with general inequality constraints. In particular, it has the potential to circumvent the jamming difficulty that appears with many interior-point methods for nonlinear programs and improve the ill conditioning of existing primal-dual interior-point methods as the barrier parameter is small. A new smoothing approach is introduced to develop our relaxation method and promote convergence of the method. Under suitable conditions, it is proved that our method can be globally convergent and locally quadratically convergent to the KKT point of the original problem. The preliminary numerical results on a well-posed problem for which many interior-point methods fail to find the minimizer and a set of test problems from the CUTEr collection show that our method is efficient.

Xin-Wei Liu
mathlxw@hebut.edu.cn
Yu-Hong Dai
dyh@lsec.cc.ac.cn
Ya-Kui Huang
hyk@hebut.edu.cn

¹ Institute of Mathematics, Hebei University of Technology, Xiping Road No. 5340, Beichen District, 300401 Tianjin, China
² LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road No. 55, Haidian District, 100190 Beijing, China
Keywords  Nonlinear programming · Interior-point relaxation method · Smoothing method · Logarithmic-barrier problem · Mini-max problem · Global and local convergence

Mathematics Subject Classification  90C26 · 90C30 · 90C51

1 Introduction

We consider the nonlinear programs with the form

\[
\begin{align*}
\text{minimize} \quad & (\min) \ f(x) \\
\text{subject to} \quad & (\text{s.t.}) \ h(x) = 0, \ x \geq 0,
\end{align*}
\]

(1) (2)

where \( x \in \mathbb{R}^n, f : \mathbb{R}^n \to \mathbb{R} \) and \( h : \mathbb{R}^n \to \mathbb{R}^m \) are twice continuously differentiable real-valued functions defined on \( \mathbb{R}^n \). If all functions \( f \) and \( h_i (i = 1, \ldots, m) \) are linear functions, problem (1)–(2) is a standard form linear programming problem (for examples, see Nocedal and Wright 1999; Wright 1997; Ye 1997). In this paper, we mainly focus on the nonlinear programs that at least one of functions \( f \) and \( h_i \) \((i = 1, \ldots, m)\) is a nonlinear (and possibly nonconvex) function in problem (1)–(2). Our method can be easily extended to cope with nonlinear programs with general nonlinear inequality constraints (see Sect. 6 for details).

There are already many efficient algorithms and several efficient solvers for nonlinear program (1)–(2), among them is the state-of-the-art and well known solver LANCELOT (see Conn et al. 1992). Using the augmented Lagrangian function on equality constraints, Conn et al. (1988) solves the relaxed subproblem

\[
\begin{align*}
\min_x \quad & L_A(x, \lambda; \rho) \equiv f(x) - \lambda^T h(x) + \frac{1}{2} \rho \| h(x) \|^2 \\
\text{s.t.} \quad & x \geq 0,
\end{align*}
\]

(3)

where \( \lambda \in \mathbb{R}^m \) is an estimate of the multiplier vector, \( \rho > 0 \) is a penalty parameter. Both \( \lambda \) and \( \rho \) are held fixed during the solution of each subproblem and are updated adaptively in virtue of the convergence and feasibility of the approximate solution of the subproblem. Problem (3) is a nonlinear program with nonnegative constraints, and many algorithms in the literature can be used to solve this problem (see Conn et al. 1988).

Primal-dual interior-point methods have been demonstrated to be a class of very efficient methods for solving problem (1)–(2). For example, for nonlinear programs, the readers can consult (Byrd et al. 2000, 1999; Chen and Goldfarb 2006; Curtis 2012; Curtis et al. 2017; Forsgren and Gill 1998; Gertz and Gill 2004; Gould et al. 2015; Nocedal et al. 2014; Shanno and Vanderbei 2000; Ulbrich et al. 2004; Vanderbei and Shanno 1999; Wächter and Biegler 2006) and the references there in. Generally, by requiring \( x \) to be an interior-point, primal-dual interior-point methods solve the logarithmic-barrier subproblem
or its corresponding parametric Karush-Kuhn-Tucker (KKT) system, where $\mu > 0$ is a barrier parameter which is held fixed when solving the subproblem (4) or its parametric KKT system. Different from subproblem (3) in the form, problem (4) is an equality constrained nonlinear program with logarithmic-barrier terms. Although all those effective algorithms for equality constrained nonlinear programming seem to be applicable to the subproblem, their convergence to a KKT point of the original problem may fail even for a well-posed problem (that is, a problem with a unique solution at which the second-order sufficient optimality conditions hold, see Benson et al. 2004; Wächter and Biegler 2000).

Improving the jamming difficulty (i.e., the failure of global convergence to a KKT point), the rapid convergence and the numerical performance of interior-point methods has been one of the main topics of the optimization research in recent years. For example, some warm-starting interior-point methods for linear programming have focused on relaxing the primal and dual interior-point limitations (see Benson and Shanno 2007; Engau et al. 2009) when the iterate is close to the solution. These methods were also extended to solve nonlinear programming in Benson and Shanno (2008). Numerical results in Benson and Shanno (2008); Engau et al. (2009) have shown that the warm-starting technique could improve the performance of interior-point methods for linear and nonlinear programming. Most recently, Haeser et al. (2019) investigated how the update of the barrier parameter affects the convergence of classic interior-point methods for convex and nonconvex optimization. Furthermore, Hinder and Ye (2018) proposed a one-phase interior-point method for nonconvex optimization with general inequality constraints, and showed that, by careful initialization and updates of the slack variables, the proposed method can be guaranteed to have more robust global convergence properties and will closely resemble successful algorithms from linear programming.

With the help of a logarithmic barrier augmented Lagrangian function, Dai et al. (2020) proposed a bi-parametric primal-dual nonlinear system which corresponds to a KKT point and an infeasible stationary point of the original problem, respectively, as one of two parameters is zero. The method in Dai et al. (2020) always generated interior-point iterates without any truncation of the step. Based on the equivalence of a positive relaxation problem to the logarithmic-barrier subproblem, Liu and Dai (2020) presented a globally convergent primal-dual interior-point relaxation method for nonlinear programs, which did not require any primal or dual iterate to be an interior point. The method has similarity to the warmstarting interior-point methods (Benson and Shanno 2007; Engau et al. 2009) and is different from most of the globally convergent interior-point methods in the literature. Without assuming any regularity condition, the method either terminates at an approximate KKT point of the original problem, an approximate infeasible stationary point, or an approximate singular stationary point of the original problem. The preliminary numerical results show that the algorithm is not only efficient for well-posed feasible problems, but also is applicable for some feasible problems without LICQ or MFCQ and some infeasible problems.
In this paper, we first prove that, under suitable conditions, any solution of a parametric equality constrained mini-max problem is a KKT point of the logarithmic-barrier subproblem. Based on this observation, we present a novel primal-dual interior-point relaxation method with iteratively updating barrier for nonlinear programs subject to general equality and nonnegative constraints. Our method is established on approximately solving a sequence of KKT systems of the parametric equality constrained mini-max subproblems, which avoids the requirement that any primal or dual iterate is an interior-point. The barrier parameter is updated with the iteration point as we did for linear programming, which is distinct from the newly proposed primal-dual interior-point relaxation method (see Liu and Dai 2020) for nonlinear programming where the parameter is only updated in outer iterations when, for a fixed barrier, the inner iterations have found some approximate solutions of the logarithmic-barrier subproblems satisfying the given accuracy. In particular, our update for the barrier parameter is autonomous and iterative, allowing our method to potentially avoid the possible difficulties caused by the inappropriate initial selection of the barrier parameter and to speed up convergence to the solution.

The method is easily extended for solving problems with general inequality constraints without incorporating any additional slack variables. It has the potential to circumvent the jamming difficulty that appears with many interior-point methods for nonlinear programs and improve the ill conditioning of the existing primal-dual interior-point methods as the barrier parameter is small (see Nocedal and Wright 1999). Furthermore, a new smoothing approach, which is totally different from the techniques used in Qi et al. (2000), is introduced to develop our relaxation method and promote convergence of the method. Under suitable conditions, it is proved that our method can be globally convergent and locally quadratically convergent to the KKT point of the original problem. The preliminary numerical results on a well-posed problem for which many interior-point methods fail to find the minimizer and a set of test problems from the CUTEr collection show that our method is efficient.

Our paper is organized as follows. In Sect. 2, we prove that the classic logarithmic-barrier subproblem can be equivalently converted into an equality constrained mini-max problem. Based on this equivalence, we present the framework of our primal-dual interior-point relaxation method for nonlinear programs in Sect. 3. In this section, we also figure out why our method can be expected to be efficient in improving the classic interior-point methods. We analyze and prove the global and local convergence results of our method for nonlinear programs in Sects. 4 and 5, respectively. Some preliminary numerical results on nonlinear programming test problems are reported in Sect. 6. We conclude our paper in the last section.

Throughout the paper, we use standard notations from the literature. A letter with subscript \( k \) is related to the \( k \)th iteration, the subscript \( j \) indicates the \( j \)th component of a vector, and the subscript \( kj \) is the \( j \)th component of a vector at the \( k \)th iteration. All vectors are column vectors, and \( z = (x, u) \) means \( z = [x^T, u^T]^T \). The expression \( \theta_k = O(t_k) \) means that there exists a constant \( M \) independent of \( k \) such that \( |\theta_k| \leq M|t_k| \) for all \( k \) large enough, and \( \theta_k = o(t_k) \) indicates that \( |\theta_k| \leq \epsilon_k|t_k| \) for all \( k \) large enough with \( \lim_{k \to 0} \epsilon_k = 0 \). If it is not specified, \( I \) is an identity matrix whose order is either marked in the subscript or is clear in the context, and \( \| \cdot \| \) is the Euclidean norm. Some unspecified notations may be identified from the context.
2 An equality constrained mini-max problem

Before presenting our main results, we review an equivalent problem of the logarithmic-barrier subproblem proposed in Liu and Dai (2020).

For any given parameters $\mu \geq 0$ and $\rho > 0$, and any $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^n$, Liu and Dai (2020) defined $z : \mathbb{R}^{2n} \to \mathbb{R}^n$, $z = z(x, s; \mu, \rho)$ and $y : \mathbb{R}^{2n} \to \mathbb{R}^n$, $y = y(x, s; \mu, \rho)$ by components to be functions on $(x, s)$ as follows,

$$z_j(x_j, s_j; \mu, \rho) \equiv \frac{1}{2\rho} \left( \sqrt{(s_j - \rho x_j)^2 + 4\rho \mu} - (s_j - \rho x_j) \right), \quad (5)$$

$$y_j(x_j, s_j; \mu, \rho) \equiv \frac{1}{2\rho} \left( \sqrt{(s_j - \rho x_j)^2 + 4\rho \mu} + (s_j - \rho x_j) \right), \quad (6)$$

where $j = 1, \ldots, n$, $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^n$ are variables. Based on definitions (5) and (6), Liu and Dai (2020) proposed to solve an equivalent positive relaxation problem to the logarithmic-barrier subproblem (4) (see Theorem 2.3 of Liu and Dai 2020) in the form

$$\min_{x, s} f(x) - \mu \sum_{j=1}^n \ln z_j(x, s; \mu, \rho) \quad (7)$$

s.t.

$$h(x) = 0, \quad (8)$$

$$z(x, s; \mu, \rho) - x = 0. \quad (9)$$

For convenience of readers and our subsequent discussions, we list some preliminary results in the following lemmas. These results have some similarities to Lemmas 2.1 and 2.2 and Theorem 2.3 of Liu and Dai (2020).

**Lemma 2.1** For given $\mu \geq 0$ and $\rho > 0$, $z_j$ and $y_j$ are defined by (5) and (6). Then

(1) $z_j \geq 0$, $y_j \geq 0$, $z_j - x_j = y_j - (s_j/\rho)$, and $z_j y_j = \mu/\rho$;

(2) $x_j \geq 0$, $s_j \geq 0$, $x_j s_j = \mu$ if and only if $z_j - x_j = 0$;

(3) $z_j - x_j = \frac{\mu - x_j s_j}{\rho(y_j + x_j)}$ and $\rho(z_j + y_j) = \sqrt{(s_j - \rho x_j)^2 + 4\rho \mu}$.

**Proof** Results (1) and (2) can be proved in the same way as Lemma 2.1 of Liu and Dai (2020). We are left to prove the result (3). Note that

$$z_j - x_j = \frac{1}{2\rho} \left( \sqrt{(s_j - \rho x_j)^2 + 4\rho \mu} - (s_j + \rho x_j) \right)$$

$$= \frac{2\mu - 2x_j s_j}{\sqrt{(s_j - \rho x_j)^2 + 4\rho \mu} + (s_j + \rho x_j)}$$

$$= \frac{\mu - x_j s_j}{\rho(y_j + x_j)}.$$

---

1 A little change is that both $z$ and $y$ are divided by $\rho$ in this paper.
and the last equality in Lemma 2.1 (3) follows from the definitions (5) and (6). All results are derived.

By Lemma 2.1, we always have \( \rho(y_j + x_j - z_j) = s_j \) and \( \mu = \rho z_j y_j \). Moreover, it follows from Lemma 2.1 (3), \( \rho(z - x)^T(y + z) = (n\mu - x^T s) + \rho \|z - x\|^2 \).

**Lemma 2.2** Given \( \mu > 0 \) and \( \rho > 0 \). Let \( z_j \) and \( y_j \) be defined by (5) and (6). Then

1. \( z_j \) and \( y_j \) are differentiable, respectively, on \( x \) and \( s \), and

\[
\nabla_x z_j = \frac{z_j}{z_j + y_j} e_j, \quad \nabla_x y_j = -\frac{y_j}{z_j + y_j} e_j, \quad (10)
\]

\[
\nabla_s z_j = -\frac{1}{\rho} \frac{z_j}{z_j + y_j} e_j, \quad \nabla_s y_j = \frac{1}{\rho} \frac{y_j}{z_j + y_j} e_j, \quad (11)
\]

where \( e_j \in \mathbb{R}^n \) is the \( j \)-th coordinate vector;

2. \( z_j \) and \( y_j \) are differentiable on \( \mu \), and

\[
\frac{\partial z_j}{\partial \mu} = \frac{\partial y_j}{\partial \mu} = \frac{1}{\rho} \frac{1}{z_j + y_j}, \quad (12)
\]

3. \( z_j \) and \( y_j \) are differentiable on \( \rho \), and

\[
\frac{\partial z_j}{\partial \rho} = \frac{1}{\rho} \frac{z_j}{z_j + y_j} (x_j - z_j), \quad \frac{\partial y_j}{\partial \rho} = -\frac{1}{\rho} \frac{y_j}{z_j + y_j} (y_j + x_j). \]

Thus,

\[
\frac{\partial (z_j - x_j)^2}{\partial \rho} = -\frac{2}{\rho} \frac{z_j}{z_j + y_j} (z_j - x_j)^2. \quad (13)
\]

**Proof** By the result (1) of Lemma 2.2 of Liu and Dai (2020), one has

\[
\nabla_x (\rho z_j) = \rho \frac{\rho z_j}{\rho z_j + \rho y_j} e_j, \quad \nabla_x (\rho y_j) = -\rho \frac{\rho y_j}{\rho z_j + \rho y_j} e_j,
\]

\[
\nabla_s (\rho z_j) = -\frac{\rho z_j}{\rho z_j + \rho y_j} e_j, \quad \nabla_s (\rho y_j) = \frac{\rho y_j}{\rho z_j + \rho y_j} e_j.
\]

Thus, (10) and (11) follow immediately.

Due to

\[
\frac{\partial (\rho z_j)}{\partial \mu} = \frac{\partial (\rho y_j)}{\partial \mu} = \frac{1}{2} \frac{4\rho}{\sqrt{(s_j - \rho x_j)^2 + 4\rho \mu}},
\]

the result (12) is derived from Lemma 2.1 (3).
Since $\rho(z_j + y_j) = \sqrt{(s_j - \rho x_j)^2 + 4\rho \mu}$ and $\rho(z_j - y_j) = \rho x_j - s_j$, one has

$$\frac{\partial \rho(z_j + y_j)}{\partial \rho} = \frac{1}{\rho} \left( \frac{\rho x_j - s_j}{z_j + y_j} + 2\mu \right), \quad \frac{\partial \rho(z_j - y_j)}{\partial \rho} = x_j.$$

Thus,

$$\frac{\partial z_j}{\partial \rho} = \frac{1}{\rho} \left( \frac{\partial \rho(z_j + y_j)}{\partial \rho} - z_j \right) = \frac{1}{\rho} \left( \frac{1}{2} \left( \frac{\partial \rho(z_j + y_j)}{\partial \rho} + \frac{\partial \rho(z_j - y_j)}{\partial \rho} \right) - z_j \right) = \frac{1}{\rho z_j + y_j} (x_j - z_j),$$

$$\frac{\partial y_j}{\partial \rho} = \frac{\partial z_j}{\partial \rho} + \frac{1}{\rho} \left( z_j - y_j - x_j \right) = -\frac{1}{\rho z_j + y_j} (y_j + x_j),$$

$$\frac{\partial (z_j - x_j)^2}{\partial \rho} = 2(z_j - x_j) \frac{\partial z_j}{\partial \rho} = \frac{2}{\rho z_j + y_j} (z_j - x_j)^2.$$

This result implies that $\|z - x\|^2$ is a monotonically nonincreasing function on $\rho$. □

The following result is the foundation of development of the primal-dual interior-point relaxation method in Liu and Dai (2020).

**Lemma 2.3** Given $\mu > 0$ and $\rho > 0$. Let $(x^*, \lambda^*)$ be a KKT pair of the logarithmic-barrier subproblem (4) and $(x^*, \lambda^*, s^*)$ satisfies its KKT system

\begin{align*}
\nabla f(x^*) - \nabla h(x^*) \lambda^* - s^* &= 0, \\
h(x^*) &= 0, \\
x_j^* > 0, \ s_j^* > 0, \ x_j^* s_j^* &= \mu, \ j = 1, \ldots, n,
\end{align*}

where $\lambda^* \in \Re^m$ is the Lagrange multiplier vector. Then $((x^*, s^*), (\lambda^*, v^*))$ is a KKT pair of the relaxation problem (7)–(9).

Conversely, if $\mu > 0$ and $\rho > 0$, $((x^*, s^*), (\lambda^*, v^*))$ is a KKT pair of problem (7)–(9), where $\lambda^* \in \Re^m$ and $v^* \in \Re^n$ are, respectively, the associated Lagrange multipliers of constraints (8) and (9), then $v^* = s^*$ and $(x^*, \lambda^*, s^*)$ satisfies the system (14)–(16). Thus, $(x^*, \lambda^*)$ is a KKT pair of the logarithmic-barrier subproblem (4).

**Proof** Please refer to the proof of Theorem 2.3 of Liu and Dai (2020). □
Throughout the paper, we take \( z \) and \( y \) to be functions on \((x, s)\) dependent on parameters \((\mu, \rho)\). When it is thought to be clear in the context, we may ignore the variables and parameters in writing functions \( z \) and \( y \) for simplicity.

Now we consider the relaxation problem (7)–(9). By incorporating the “similar” augmented Lagrangian terms on constraints of (9) into the objective function, and taking the maximum with respect to \( s \), we obtain a particular mini-max problem

\[
\min_{x \in \mathbb{R}^n | h(x) = 0} \left\{ f(x) + \sum_{j=1}^{n} \max_{s_j \in \mathbb{R}} G(x_j, s_j; \mu, \rho) \right\},
\]

or its equivalent form

\[
\min_{x \in \mathbb{R}^n | h(x) = 0} \max_{s \in \mathbb{R}^n} F(x, s; \mu, \rho),
\]

where \( F: \mathbb{R}^{2n} \to \mathbb{R} \), \( F(x, s; \mu, \rho) \equiv f(x) + \sum_{j=1}^{n} G(x_j, s_j; \mu, \rho) \) and \( G : \mathbb{R} \to \mathbb{R} \),

\[
G(x_j, s_j; \mu, \rho) \equiv -\mu \ln z_j(x_j, s_j; \mu, \rho) + s_j(z_j(x_j, s_j; \mu, \rho) - x_j) + \frac{1}{2} \rho \|z_j(x_j, s_j; \mu, \rho) - x_j\|^2.
\]

It should be noticed that the extra two terms \( s^T(z(x, s; \mu, \rho) - x) + \frac{1}{2} \rho \|z(x, s; \mu, \rho) - x\|^2 \) in \( F(x, s; \mu, \rho) \) (comparing to (7)) are not the usual augmented Lagrangian terms, since they definitely use the variables of \( s \) of the function \( z \) as the estimates of Lagrange multipliers, and take the parameter \( \rho \) in \( z \) as the penalty parameter. Moreover, the barrier parameter \( \mu \) is used not only in the logarithmic-barrier terms but also in the other terms.

Using the previous preliminary results, we can derive some properties on \( F(x, s; \mu, \rho) \).

**Lemma 2.4** Given \( \mu > 0 \) and \( \rho > 0 \). Let \( z = z(x, s; \mu, \rho) \) and \( y = y(x, s; \mu, \rho) \) be defined by (5) and (6), \( Z = \text{diag}(z), Y = \text{diag}(y) \).

(1) If \( f \) is twice differentiable, then \( F \) is twice differentiable with respect to \( x \) and \( s \). Moreover,

\[
\nabla_x F(x, s; \mu, \rho) = \nabla f(x) - \rho y, \quad \nabla_x^2 F(x, s; \mu, \rho) = \nabla^2 f(x) + \rho(Z + Y)^{-1}Y,
\]

\[
\nabla_s F(x, s; \mu, \rho) = z - x, \quad \nabla_s^2 F(x, s; \mu, \rho) = -\frac{1}{\rho} (Z + Y)^{-1}Z.
\]

(2) Function \( F(x, s; \mu, \rho) \) is a strictly concave function with respect to \( s \), and \( F(x, s; \mu, \rho) - f(x) \) is a strictly convex function with respect to \( x \).

(3) There holds

\[
\frac{\partial F(x, s; \mu, \rho)}{\partial \rho} = \frac{(\rho - 1)}{\rho} (z - x)^T(Z + Y)^{-1}Z(z - x).
\]
**Proof** Due to Lemmas 2.1 and 2.2, one has the derivatives

\[
\frac{\partial G(x_j, s_j; \mu, \rho)}{\partial x_j} = -\mu - y_j (s_j + \rho z_j - \rho x_j) = -\rho y_j, \\
\frac{\partial G(x_j, s_j; \mu, \rho)}{\partial s_j} = z_j - x_j + \frac{\mu - z_j s_j - \rho (z_j - x_j)}{\rho (z_j + y_j)} = z_j - x_j.
\]

Again by Lemma 2.2, the second-order derivatives in (1) follow immediately.

The results in (2) are straightforward since \(\nabla^2 F(x, s; \mu, \rho)\) is always negative definite and \(\nabla^2 s F(x, s; \mu, \rho)\) is always positive definite.

Noting that \(\mu = \rho z_j y_j\), Lemma 2.2 (3), (13), and \(\partial \ln z_j / \partial \rho = z_j - 1\), the result (3) follows immediately due to \(\rho (z_j - x_j) = \rho y_j - s_j\) and

\[
\frac{\partial G(x_j, s_j; \mu, \rho)}{\partial \rho} = \frac{\rho - 1}{\rho} \frac{z_j}{z_j + y_j} (z_j - x_j)^2.
\]

\(\Box\)

In the following, we prove our main result of this section, which is the foundation of our novel primal-dual interior-point relaxation method in this paper.

**Theorem 2.5** Let \(\mu > 0\) and \(\rho > 0\). The following two results can be obtained.

1. The pair \((x^*, s^*) \in \mathbb{R}^n \times \mathbb{R}^n\) is a local solution of the mini-max problem (17) if and only if \(x^* > 0\) is a local solution of the logarithmic-barrier subproblem (4) and \(s_j^* = \mu / x_j^*\) for all \(j = 1, \ldots, n\).

2. If \((x^*, s^*) \in \mathbb{R}^n \times \mathbb{R}^n\) is a local solution of the mini-max problem (17) and \(\nabla h(x^*)\) is of full column rank, then there exists a \(\lambda^* \in \mathbb{R}^m\) such that

\[
\nabla f(x^*) - \nabla h(x^*) \lambda^* - s^* = 0, \\
h(x^*) = 0, \\
z^* - x^* = 0,
\]

where \(z^* = z(x^*, s^*; \mu, \rho)\). Thus, \((x^*, \lambda^*\) is a KKT pair of the logarithmic-barrier subproblem (4).

**Proof** (1) In light of Lemma 2.4, for any \(x_j > 0\), \(G(x_j, s_j; \mu, \rho)\) reaches its maximum at \(s_j^* = \mu / x_j\) since \(z_j (x_j, s_j^*; \mu, \rho) - x_j = 0\). If \(x_j \leq 0\), then \(\frac{\partial G(x_j, s_j; \mu, \rho)}{\partial s_j} > 0\), which means that \(G(x_j, s_j; \mu, \rho)\) is strictly monotonically increasing to \(\infty\) as \(s_j \to \infty\). Thus,

\[
\max_{s_j \in \mathbb{R}} G(x_j, s_j; \mu, \rho) = \begin{cases} -\mu \ln x_j, & \text{if } x_j > 0; \\ \infty, & \text{otherwise,} \end{cases}
\]

and

\[
\arg\max_{s_j \in \mathbb{R}} G(x_j, s_j; \mu, \rho) = \begin{cases} \mu / x_j, & \text{if } x_j > 0; \\ \infty, & \text{otherwise.} \end{cases}
\]
The result follows immediately from the above two equations.

(2) If \((x^*, s^*)\) is a solution of the mini-max problem (17), then \(z^* - x^* = 0\) by (1) and \(x^*\) is a local solution of the subproblem

\[
\min_x F(x, s^*; \mu, \rho) \quad \text{s.t.} \quad h(x) = 0.
\]  

Thus, if \(\nabla h(x^*)\) is of full column rank, by the first-order necessary conditions of optimality (for example, see Nocedal and Wright 1999; Sun and Yuan 2006), there exists a \(\lambda^* \in \mathbb{R}^m\) such that \((x^*, \lambda^*)\) is a KKT pair of subproblem (23), i.e., there exists a \(\lambda^* \in \mathbb{R}^m\) such that

\[
\nabla f(x^*) - \nabla h(x^*)\lambda^* - \rho y^* = 0,
\]
\[
h(x^*) = 0,
\]
\[
z^* - x^* = 0,
\]

where \(y^* = y(x^*, s^*; \mu, \rho)\) and \(z^* = z(x^*, s^*; \mu, \rho)\). Then the equations (18)–(20) are attained immediately since \(z^* - x^* = 0\) if and only if \(y^* - s^*/\rho = 0\) due to Lemma 2.1 (1). 

Although the logarithmic-barrier subproblem (4), its relaxation subproblem (7)–(9), and the mini-max subproblem (17) are equivalent in some sense, they provide us insightful views on the existing methods and possibilities for developing different and possibly robust methods for the original problem (1)–(2). For example, by using the relaxation subproblem (7)–(9), we can remove the interior-point restrictions on primal and dual variables in Liu and Dai (2020). In this paper, we note that, \((x^*, s^*)\) is a solution of a mini-max subproblem if \(x^*\) is a local solution of the logarithmic-barrier subproblem. Thus, the residual function on the system (18)–(20) is reasonable to be chosen as the merit function. In addition, by solving the system (18)–(20), we are capable of improving the ill conditioning often observed during the final stages of the classic primal-dual algorithms based on solving the subproblem (4) or its corresponding KKT system (please refer to Sect. 3 for details).

As a special example, when \(f\) and \(h\) are linear functions, that is, program (1)–(2) is a linear programming problem, the mini-max problem is a particular saddle-point problem. The next result is a corollary of Theorem 2.5.

**Corollary 2.6** Assume \(\mu > 0\) and \(\rho > 0\), \(f\) and \(h_i\) \((i = 1, \ldots, m)\) are linear functions on \(\mathbb{R}^n\). The primal-dual pair \((x^*, s^*)\) is a solution of the mini-max problem (17) if and only if there exists a \(\lambda^* \in \mathbb{R}^m\) such that \((x^*, \lambda^*)\) is a KKT pair of the logarithmic-barrier subproblem (4).

### 3 A novel primal-dual interior-point relaxation method

Based on solving the mini-max subproblem (17), we develop a novel primal-dual interior-point relaxation method for solving the nonlinear constrained optimization
problem (1)–(2). Since problem (17) originates from the logarithmic-barrier subproblem, our method can be thought of as a variant of classic primal-dual interior-point methods. The method updates the barrier parameter $\mu$ in every iteration, which resembles some successful interior-point methods for linear and nonlinear programming (such as Hinder and Ye 2018; Mehrotra 1992; Nocedal et al. 2009), and is different from those based on the Fiacco-McCormick approach (Fiacco and McCormick 1990) for nonlinear programming in which they often attempt to find an approximate solution for a fixed parameter $\mu$ in an inner algorithm and then reduce the barrier parameter $\mu$ by the residual of the solution in an outer algorithm. In particular, our update for the barrier parameter is autonomous and iterative, which makes our method capable of avoiding the possible difficulties caused by inappropriate initial selection of the barrier parameter and makes our method have the potential of speeding up the convergence to the solution.

Instead of solving the subproblem (17) directly, we solve the associated system (18)–(20) and consider the extended system of equations of (18)–(20) in the form

$$
\mu = 0, \quad \nabla f(x) - \nabla h(x)\lambda - s = 0, \quad h(x) = 0, \quad z - x = 0,
$$

where $z = z(x, s; \mu, \rho)$ and $y = y(x, s; \mu, \rho)$ are functions on $x$ and $s$ defined by (5) and (6). Distinct from our recent work (Dai et al. 2020; Liu and Dai 2020) and many interior-point methods for nonlinear programs, we also take $\mu$ as a variable in the system (24)–(27) instead of only a parameter in the system (18)–(20) so that $\mu$ is updated with the iteration point. This approach has been used successfully in smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities (see Qi et al. 2000), where $\mu$ is a vector of smoothing parameters. Note that, for $j = 1, \ldots, n$,

$$
z_j(x_j, s_j; 0, \rho) = \frac{1}{2\rho}(|s_j - \rho x_j| - (s_j - \rho x_j)) = \max\{0, x_j - s_j/\rho\},
$$

$$
y_j(x_j, s_j; 0, \rho) = \frac{1}{2\rho}(|s_j - \rho x_j| + (s_j - \rho x_j)) = \max\{0, s_j/\rho - x_j\}.
$$

Thus, for any $j = 1, \ldots, n$, the equality $z_j = x_j$ implies that one has either $x_j = 0$, $s_j \geq 0$, $\rho y_j = s_j$, or $x_j \geq 0$, $s_j = 0$, $\rho y_j = 0$. Therefore, any $(x^*, \lambda^*, s^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ satisfying the extended system of equations (24)–(27) is a KKT triple of the original problem (1)–(2).

Denote the residual function of the system (18)–(20) as follows,

$$
\phi(\mu, \rho)(x, \lambda, s) = \frac{1}{2}||\psi(\mu, \rho)(x, \lambda, s)||^2,
$$

$$
\text{Springer}$$
where

\[
\psi(\mu, \rho)(x, \lambda, s) = \left[ \begin{array}{c}
\nabla f(x) - \nabla h(x) \lambda - s \\
h(x) \\
z - x
\end{array} \right].
\]

Using this notation, the system (24)–(27) can be further reformulated as

\[
\begin{align*}
\mu + \gamma \phi(\mu, \rho)(x, \lambda, s) &= 0, \\
\psi(\mu, \rho)(x, \lambda, s) &= 0,
\end{align*}
\]

(29) (30)

where \( \mu \) is supposed to be nonnegative, \( \phi(\mu, \rho)(x, \lambda, s) \) is defined by (28), and \( \gamma \in (0, 1] \) is a given parameter.

In order to solve the system (24)–(27) efficiently, \( \mu \) should not approach zero too quickly. Thus it is important to balance the reduction of \( \mu \) and the associated KKT residual of the mini-max subproblem. The methods in Qi et al. (2000) were established on solving the system with elaborately constructed perturbation of the Newton system, and the residual function of the whole system was taken as the merit function. In contrast, instead of solving the system (24)–(27) directly, we develop our relaxation method by solving the reformulation (29)–(30) and promote convergence of our method by reducing the residual function \( \phi(\mu, \rho)(x, \lambda, s) \).

Suppose that \((x_k, \lambda_k, s_k)\) is the current primal and dual iterates, \( \mu = \mu_k \) and \( \rho = \rho_k \) are current values of the barrier and penalty parameters. Let \( r^d_k = \nabla f(x_k) - \nabla h(x_k) \lambda_k - s_k, r^e_k = z_k - x_k, r^h_k = h(x_k) \), and correspondingly \( \psi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \) be the residual of the equation (30) at iterate \( k \). The Newton’s equations corresponding to equations (29)–(30) are as follows,

\[
\Delta \mu = -\mu_k + \gamma \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k),
\]

(31)

\[
\nabla_{(\mu, x, \lambda, s)} \psi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)^T p = -\psi(\mu_k, \rho_k)(x_k, \lambda_k, s_k),
\]

(32)

where \( p = (\Delta \mu, dx, d\lambda, ds) \). (31) is derived from \( \nabla_{(\mu, x, \lambda, s)} \psi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) = \nabla_{(\mu, x, \lambda, s)} \psi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \psi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \) and (32). Let \( \Delta \mu_k \) be calculated by (31), then the second equation (32) can be written in detail as

\[
\begin{bmatrix}
B_k & -\nabla h(x_k) & -I \\
\nabla h(x_k)^T & 0 & 0 \\
(Z_k + Y_k)^{-1}Y_k & 0 & \frac{1}{\rho_k}(Z_k + Y_k)^{-1}Z_k
\end{bmatrix}
\begin{bmatrix}
d_x \\
d\lambda \\
ds
\end{bmatrix}
= \begin{bmatrix}
-r^d_k \\
0 \\
r^e_k + \frac{1}{\rho_k} \Delta \mu_k (Z_k + Y_k)^{-1}e
\end{bmatrix},
\]

(33)

where the term on the variation \( \Delta \mu_k \) of \( \mu \) is moved to the right-hand-side of the linearized equation.
Our proposed method generates the new value of parameter \( \mu_{k+1} \) by

\[
\mu_{k+1} = (1 - \alpha_k)\mu_k + \gamma \alpha_k \phi_{(\mu_k, \rho_k)}(x_k, \lambda_k, s_k)
\]

and the new primal and dual iterates by a line search procedure

\[
x_{k+1} = x_k + \alpha_k d_{xk}, \quad \lambda_{k+1} = \lambda_k + \alpha_k d_{\lambda k}, \quad s_{k+1} = x_k + \alpha_k d_{sk},
\]

where \( (d_{xk}, d_{\lambda k}, d_{sk}) \) is the search direction derived from the system (33), and \( \alpha_k \in (0, 1] \) is the step-size.

The preceding system can also be equivalently written as the linear system with a symmetric coefficient matrix in the form

\[
\begin{bmatrix}
B_k + \rho_k (Z_k + Y_k)^{-1} Y_k & -\nabla h(x_k) & -(Z_k + Y_k)^{-1} Y_k \\
-\nabla h(x_k)^T & 0 & 0 \\
-(Z_k + Y_k)^{-1} Y_k & 0 & -\frac{1}{\rho_k} (Z_k + Y_k)^{-1} Z_k
\end{bmatrix}
\begin{bmatrix}
d_x \\
d_\lambda \\
d_s
\end{bmatrix}
\]

\[
= - \begin{bmatrix}
\hat{r}_k^d - \Delta \mu_k (Z_k + Y_k)^{-1} e \\
\tilde{r}_k^e - r_k^h \\
r_k^e + \frac{1}{\rho_k} \Delta \mu_k (Z_k + Y_k)^{-1} e
\end{bmatrix},
\]

where \( B_k \) is the Hessian of the Lagrangian \( L(x, \lambda, s) = f(x) - \lambda^T h(x) - s^T x \) or its approximation at \( (x_k, \lambda_k, s_k) \), \( z_k = z(x_k, s_k; \mu_k, \rho_k) \), \( y_k = y(x_k, s_k; \mu_k, \rho_k) \), \( Z_k = \text{diag} (z_k) \), \( Y_k = \text{diag} (y_k) \), \( \Delta \mu_k = -\mu_k + \gamma \phi_{(\mu_k, \rho_k)}(x_k, \lambda_k, s_k) \), \( \tilde{r}_k^d = \nabla f(x_k) - \nabla h(x_k) \lambda_k - \rho_k y_k \).

Since we are facing a mini-max subproblem, taking the residual function \( \phi_{(\mu, \rho)}(x, \lambda, s) \) defined by (28) as the merit function is a natural and reasonable selection. The step-size \( \alpha_k \) is selected such that the value of \( \phi_{(\mu, \rho)}(x, \lambda, s) \) is sufficiently decreased when the iterate moves from point \((x_k, \lambda_k, s_k)\) to \((x_{k+1}, \lambda_{k+1}, s_{k+1})\) and the barrier parameter varies from \( \mu_k \) to \( \mu_{k+1} \), while the penalty parameter \( \rho_k \) holds fixed. Then \( \rho_k \) is updated adaptively to \( \rho_{k+1} \) such that \( \rho_{k+1} \geq \rho_k \).

In the following, we describe our algorithm for problem (1)–(2), in which the parameter \( \mu \) is updated with the iteration point. In our algorithm, scalars \( \gamma \), \( \gamma_0 \) and \( \eta \) are parameters used to balance the reduction of \( \mu_k \) and \( \phi_{(\mu_k, \rho_k)}(x_k, \lambda_k, s_k) \). That is, for given \( \gamma_0 \in (0, 1) \) and \( \eta > 1 \), \( \mu_k \in [\gamma_0 \phi_{(\mu_k, \rho_k)}(x_k, \lambda_k, s_k), \eta \phi_{(\mu_k, \rho_k)}(x_k, \lambda_k, s_k)] \) is thought to be in a good balance and will be updated normally by the Newton’s step; otherwise, it will be reduced provided it is larger or fixed if it is smaller before proceeding to a new iteration. The scalar \( \gamma \in (0, \gamma_0] \) is a balance parameter introduced in (3.6). Scalars \( \delta \) and \( \tau \) are parameters necessary for Armijo’s line search procedure in (34) and scalar \( \sigma \) is a given factor for the update of the penalty parameter.

For Algorithm 1, the initial point can be any point which is either an interior or other point. Our algorithm does not also require any primal or dual iterate to be interior during the iterative process, which is distinct from most of the classic interior-point methods. Steps 0.1 and 5.1 are used to prevent \( \mu_0 \) and \( \mu_{k+1} \) from being too large in comparison with the residuals of KKT system \( \phi_{(\mu_0, \rho_0)}(x_0, \lambda_0, s_0) \) and \( \phi_{(\mu_{k+1}, \rho_{k+1})}(x_{k+1}, \lambda_{k+1}, s_{k+1}) \), respectively. If \( \mu_{k+1} \leq
Algorithm 1 A novel primal-dual interior-point relaxation method for problem (1)–(2)

Given \((x_0, \lambda_0, s_0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n, B_0 \in \mathbb{R}^{n \times n}, \mu_0 > 0, \rho_0 > 0, \eta > 1, \gamma_0, \delta, \tau, \sigma \in (0, 1)\).

Evaluate \(z_0\) and \(y_0\) by (5) and (6), compute \(\phi(\mu_0, \rho_0)(x_0, \lambda_0, s_0)\). Given \(\epsilon \in (0, \mu_0)\), set \(k := 0\).

Set \(\ell := 0, \mu_{k, \ell} = \mu_k\).

Step 0.1 While \(\mu_{k, \ell} > \max\{\eta \phi(\mu_{k, \ell}, \rho_{k}) (x_k, \lambda_k, s_k), \epsilon\}\), set \(\mu_{k, \ell+1} = \mu_{k, \ell}/\eta\); evaluate \(z_k\) and \(y_k\) by (5) and (6) with \(\mu = \mu_{k, \ell+1}\), compute \(\phi(\mu_{k, \ell+1}, \rho_{k})(x_k, \lambda_k, s_k)\), set \(\ell = \ell + 1\), end.

Set \(\mu_k = \mu_{k, \ell}, \gamma = \min\{\gamma_0, \mu_k/\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)\}\).

While \(\mu_k \leq \epsilon\) and \(\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \leq \epsilon\), stop the algorithm.

Step 1. Calculate \(\Delta \mu_k\) by \(\Delta \mu_k = -\mu_k + \gamma \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)\).

Step 2. Solve the linear system (33) to obtain \(d_k \equiv (d_{sk}, d_{s^k}, d_{jk})\).

Step 3. Select the step-size \(\alpha_k \in (0, 1]\) to be the maximal number in \([1, \delta, \delta^2, \ldots]\) such that the inequality

\[
\phi(\mu_k + \alpha_k \Delta \mu_k, \rho_k)(x_k + \alpha_k d_{sk}, \lambda_k + \alpha_k d_{s^k}, s_k + \alpha_k d_{jk}) \leq (1 - 2\alpha_k) \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)
\]  

is satisfied.

Step 4. Set \(\mu_{k+1} = \mu_k + \alpha_k \Delta \mu_k, x_{k+1} = x_k + \alpha_k d_{sk}, s_{k+1} = s_k + \alpha_k d_{s^k}, \) and \(\lambda_{k+1} = \lambda_k + \alpha_k d_{jk}\).

Step 5. Update \(\rho_k\) to \(\rho_{k+1} = \max\{\mu_k, \sigma, ||s_{k+1}||/\max(||x_{k+1}||,1)\}\). Evaluate by (5) and (6)

\[
z_{k+1} = z(x_{k+1}, s_{k+1}^1, \mu_{k+1}, \rho_{k+1}) \text{ and } y_{k+1} = y(x_{k+1}, s_{k+1}^1; \mu_{k+1}, \rho_{k+1}),
\]

compute \(\phi(\mu_{k+1}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\).

Set \(\ell := 0, \mu_{k+1, \ell} = \mu_{k+1}\).

Step 5.1 While \(\mu_{k+1, \ell} > \max\{\eta \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1}), \epsilon\}\), set \(\mu_{k+1, \ell+1} = \mu_{k+1, \ell}/\eta\); evaluate \(z_{k+1}\) and \(y_{k+1}\) by (5) and (6), compute \(\phi(\mu_{k+1, \ell+1}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\), set \(\ell = \ell + 1\), end.

Set \(\mu_{k+1} = \mu_{k+1, \ell}, \gamma = \min\{\gamma_0, \mu_{k+1}/\phi(\mu_{k+1}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\}\).

Step 6. Update \(B_k\) to \(B_{k+1}\), set \(k := k + 1\).

End (while)

max\(\{\eta \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1}), \epsilon\}\), then one of the following three kinds of results will arise:

1. \(\epsilon < \mu_{k+1} \leq \eta \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\);
2. \(\mu_{k+1} \leq \epsilon \leq \eta \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\);
3. \(\mu_{k+1} \leq \epsilon \) and \(\phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1}) \leq \epsilon/\eta < \epsilon\).

Note that, if the case (3) happens, Algorithm 1 will be terminated; otherwise, one will have either case (1) or case (2), and in both cases,

\[
\phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1}) \geq \epsilon/\eta.
\]  

Moreover, for cases (1) and (2), the parameter \(\gamma\) is selected such that either \(\gamma = \gamma_0\) and \(\mu_{k+1} > \gamma_0 \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\) or \(\Delta \mu_{k+1} = 0\). If \(\mu_{k+1} > \gamma_0 \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\), then

\[
\mu_{k+2, 0} = (1 - \alpha_{k+1}) \mu_{k+1} + \alpha_{k+1} \gamma_0 \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1}) < \mu_{k+1}
\]  

\(\Delta \mu_{k+1} = 0\) if \(\mu_{k+1} = \gamma_0 \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\). If \(\mu_{k+1} > \gamma_0 \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1})\), then

\[
\mu_{k+2, 0} = (1 - \alpha_{k+1}) \mu_{k+1} + \alpha_{k+1} \gamma_0 \phi(\mu_{k+1, \ell}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}^1, s_{k+1}) < \mu_{k+1}
\]
A primal-dual interior-point relaxation method

\[ \mu_{k+2,0} > \gamma_0 \phi(\mu_{k+1}, \rho_{k+1}) (x_{k+1}, \lambda_{k+1}, s_{k+1}) \geq (\gamma_0 / \eta) \epsilon; \]  
(37)

otherwise,

\[ \mu_{k+1} = \gamma \phi(\mu_{k+1}, \rho_{k+1}) (x_{k+1}, \lambda_{k+1}, s_{k+1}) \leq \gamma_0 \phi(\mu_{k+1}, \rho_{k+1}) (x_{k+1}, \lambda_{k+1}, s_{k+1}), \]

in this case \( \mu_{k+1} \) is viewed as to be too small in comparison with \( \phi(\mu_{k+1}, \rho_{k+1}) (x_{k+1}, \lambda_{k+1}, s_{k+1}) \) and we set \( \mu_{k+2,0} = \mu_{k+1} \). Thus, there is always \( (\gamma_0 / \eta) \epsilon \leq \mu_{k+1,0} \leq \mu_k \) for all \( k \geq 0 \).

In order to have a deep understanding on the significance of Algorithm 1, let us consider its application to the linear programs with the standard form

\[ \min \ c^T x \quad \text{s.t.} \ Ax = b, \ x \geq 0. \]  
(38)

Corresponding to the original problem (1)–(2), \( f(x) = c^T x, \ h(x) = Ax - b \). In this case, \( \nabla f(x) = c \) and \( \nabla h(x) = A^T \). Without loss of generality, we suppose that \( A \) has full row rank. Since the Lagrangian Hessian is null, (33) is reduced to the following system

\[
\begin{bmatrix}
0 & -A^T & -I \\
A & 0 & 0 \\
(Z_k + Y_k)^{-1} Y_k & 0 & \frac{1}{\rho_k} (Z_k + Y_k)^{-1} Z_k \\
\end{bmatrix}
\begin{bmatrix}
d_x \\
d_{\lambda} \\
d_s \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-(c - A^T \lambda_k - s_k) \\
-(A x_k - b) \\
(z_k - x_k) + \frac{1}{\rho_k} \Delta \mu_k (Z_k + Y_k)^{-1} e \\
\end{bmatrix},
\]

which, due to Lemma 2.1 (3), can be further written as

\[
\begin{bmatrix}
0 & A^T & I \\
A & 0 & 0 \\
\rho_k Y_k & 0 & Z_k \\
\end{bmatrix}
\begin{bmatrix}
d_x \\
d_{\lambda} \\
d_s \\
\end{bmatrix}
\]

\[= \begin{bmatrix}
c - A^T \lambda_k - s_k \\
b - A x_k \\
\mu_k e - X_k S_k e \\
\end{bmatrix}
\]

\[
+ \begin{bmatrix}
0 \\
0 \\
\rho_k (Z_k - X_k) (z_k - x_k) + \Delta \mu_k e \\
\end{bmatrix},
\]

(39)

where the minus signs in the first row are changed by left multiplying a negative identity matrix and the last row in the system is derived by left multiplying \( \rho_k (Z_k + Y_k) \), respectively, on both sides of the equations.

Comparing with the system in classic primal-dual interior-point methods for linear programming (for example, see (14.12) of Nocedal and Wright (1999)), our system (39) is different in that both \( S_k \) and \( X_k \) in the last row of the Jacobian have been substituted with \( \rho Y_k \) and \( Z_k \) and the associated right-hand-side term has also been changed (i.e., some additional correction terms have been incorporated). As we will
Lemma 3.1 There always holds conditioning of primal-dual interior-point methods for linear programming.

Note that (39) can be formalized as

\[(AY_k^{-1}Z_kA^T)d_\lambda = \rho_k(b - Ax_k) + AZ_k^{-1}Z_k(c - A^T\lambda_k - s_k)\]
\[-\rho_kA(I + Y_k^{-1}Z_k)(z_k - x_k) - \Delta \mu_k AY_k^{-1}e,\]
\[d_s = (c - A^T\lambda_k - s_k) - A^Td_\lambda,\]
\[d_x = (I + Y_k^{-1}Z_k)(z_k - x_k) + \frac{1}{\rho_k}(\Delta \mu_k Y_k^{-1}e - Y_k^{-1}Z_kd_s).\]

Due to \(\rho_k Y_k Z_k = \mu_k I\), one has \(Y_k^{-1} = (\rho_k/\mu_k)Z_k\), and

\[(AZ_k^2A^T)d_\lambda = \mu_k(b - Ax_k) + AZ_k^2(c - A^T\lambda_k - s_k)\]
\[-A(\mu_k I + \rho_k Z_k^2)(z_k - x_k) - \Delta \mu_k AZ_k e,\]  \hspace{1cm} (40)
\[d_s = (c - A^T\lambda_k - s_k) - A^Td_\lambda,\]  \hspace{1cm} (41)
\[d_x = (I + \frac{\rho_k}{\mu_k} Z_k^2)(z_k - x_k) + \frac{1}{\mu_k}(\Delta \mu_k Z_k e - Z_k^2d_s).\]  \hspace{1cm} (42)

If \(z_k \rightarrow x^*\) and \(\mu_k \rightarrow 0\) as \(k \rightarrow \infty\), where \(x^*\) is an optimal solution of the non-degenerate linear program, then \(AZ_k\) should be of full rank and (40) is capable of escaping from the ill-conditioning trap often observed during the final stages of the existing primal-dual algorithms for linear programming (see, for example, page 409 of Nocedal and Wright 1999).

Subsequently, we will show that Algorithm 1 is well-defined. Firstly, it is easy to note that Steps 0.1 and 5.1 will always be terminated finitely for any given \(\epsilon > 0\).

Lemma 3.1 There always holds \(\mu_k \geq (\gamma_0/\eta)\epsilon\) for all \(k \geq 0\).

Proof We firstly prove that, if \(\mu_{k+1,0} > \max\{\eta\phi(\mu_{k+1,0},\rho_{k+1}) (x_{k+1},\lambda_{k+1},s_{k+1}),\epsilon\}\), then

\[\mu_{k+1} \geq \epsilon/\eta.\]  \hspace{1cm} (43)

By Step 5.1, \(\mu_{k+\ell} = \mu_{k+1,\ell}\) for some \(\ell \geq 1\). Thus, \(\mu_{k+1,\ell-1} > \epsilon\) and \(\mu_{k+1,\ell} = \mu_{k+1,\ell-1}/\eta\), which implies \(\mu_{k+1,\ell} > \epsilon/\eta\). If \(\mu_{k+1,0} \leq \max\{\eta\phi(\mu_{k+1,0},\rho_{k+1}) (x_{k+1},\lambda_{k+1},s_{k+1}),\epsilon\}\), \(\mu_{k+1} = \mu_{k+1,0}\).

We have already known that \(\mu_{k+1,0} \geq (\gamma_0/\eta)\epsilon\). Note that \(\mu_0 > \epsilon\), the result follows immediately from (43). \(\square\)

In view of (5) and (6), \(\mu_k > 0\) implies \(y_k > 0\) and \(z_k > 0\). The following result asserts that the linear system (33) has a unique solution.

Lemma 3.2 Let \(x_k\) be the current iterate generated by Algorithm 1. If \(\nabla h(x_k)\) has full column rank and \(v^T (B_k + \rho_k Z_k^{-1}Y_k) v > 0\) for all \(v \neq 0 \in \mathbb{R}^n\) with \(\nabla h(x_k)^T v = 0\), then the coefficient matrix of the linear system (33) is nonsingular.
Proof In order to obtain our desired result, we need prove that the system of equations

\begin{align}
B_k d_x - \nabla h(x_k)d_\lambda - d_s &= 0, \\
\nabla h(x_k)^T d_x &= 0, \\
\rho_k Y_k d_x + Z_k d_s &= 0
\end{align}

has only zero solution. Left-multiplying \( d_x^T \) on the two-sides of (44), one has \( d_x^T B_k d_x = d_x^T d_s \) due to (45). Thus, by (46),

\[ d_x^T (B_k + \rho_k Z_k^{-1} Y_k) d_x = 0. \tag{47} \]

Note that the conditions of the lemma suggest \( d_x^T (B_k + \rho_k Z_k^{-1} Y_k) d_x \neq 0 \) for all \( d_x \neq 0 \) satisfying (45), thus \( d_x = 0 \). Therefore, \( d_x = 0 \) and \( \nabla h(x_k)d_\lambda = 0 \) due to the last and the first equations of the preceding system. Since \( \nabla h(x_k) \) has full column rank, the equation \( \nabla h(x_k)d_\lambda = 0 \) implies \( d_\lambda = 0 \), which completes our proof. \( \square \)

If Algorithm 1 does not terminate at \( x_k \), then \( \phi(\mu_k, p_k)(x_k, \lambda_k, s_k) \geq \epsilon/\eta > 0 \) due to (35). This fact shows that there will be \( (\Delta \mu_k, d_k) \neq 0 \) for all \( k \geq 0 \). Otherwise, by Lemma 3.2, the right-hand-side of (33) will be zero for some integer \( k \), which implies \( r_k^x = r_k^\lambda = r_k^h = 0 \). Thus, \( \phi(\mu_k, p_k)(x_k, \lambda_k, s_k) = 0 \), a contradiction to (35). The next result shows that, at the \( k \)-th iteration, a new iterate can be generated, thus Algorithm 1 is well-defined.

Lemma 3.3 Suppose that \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( h : \mathbb{R}^n \rightarrow \mathbb{R}^m \) are twice continuously differentiable on \( \mathbb{R}^n \). There always exists an \( \alpha_k \in (0, 1) \) such that (34) holds.

Proof The supposition implies that \( \phi(\mu, p_k)(x, \lambda, s) \) is differentiable with respect to \((\mu, x, \lambda, s)\), thus it is directionally differentiable. Due to (33), its directional derivative along \((\Delta \mu_k, d_k)\) at \((x_k, \lambda_k, s_k)\) with \( \mu = \mu_k \) is

\[
\phi'(\mu_k, p_k)(x_k, \lambda_k, s_k; \Delta \mu_k, d_k) = \left[ \frac{D\phi(\mu, p_k)(x, \lambda, s)}{D\mu} \nabla_{(x, \lambda, s)} \phi(\mu, p_k)(x, \lambda, s)^T \right]_{(\mu_k, x_k, \lambda_k, s_k) = (\mu_k, x_k, \lambda_k, s_k)} \left[ \begin{array}{c} \Delta \mu_k \\ d_k \end{array} \right] 
\]

\[ = -2\phi(\mu_k, p_k)(x_k, \lambda_k, s_k). \tag{48} \]

The Taylor’s expansion of \( \phi(\mu_k + \alpha \Delta \mu_k, p_k)(x_k + \alpha d_{x_k}, s_k + \alpha d_{s_k}, \lambda_k + \alpha d_{\lambda_k}) \) regarding \( \alpha \) at \( \alpha = 0 \) shows that

\[
\phi(\mu_k + \alpha \Delta \mu_k, p_k)(x_k + \alpha d_{x_k}, s_k + \alpha d_{s_k}, \lambda_k + \alpha d_{\lambda_k}) - \phi(\mu_k, p_k)(x_k, \lambda_k, s_k) \\
= \alpha \phi'(\mu_k, p_k)(x_k, \lambda_k, s_k; \Delta \mu_k, d_k) + o(\alpha) \\
= -2\alpha \phi(\mu_k, p_k)(x_k, \lambda_k, s_k) - 2(1 - \tau)\alpha \phi(\mu_k, p_k)(x_k, \lambda_k, s_k) + o(\alpha). \tag{49} \]

Thus, (34) holds for all sufficiently small \( \alpha > 0 \) since \( \tau < 1 \) and \( \phi(\mu_k, p_k)(x_k, \lambda_k, s_k) > 0 \). \( \square \)
The preceding result suggests that sequences \( \{(x_k, \lambda_k, s_k)\} \) and \( \{\mu_k\}, \{\rho_k\} \) will be derived from Algorithm 1 before the terminating conditions are satisfied. Moreover, (36) has shown that the barrier sequence \( \{\mu_k\} \) is monotonically nonincreasing. It will be proved that the sequence of merit function values \( \{\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)\} \) is monotonically decreasing.

**Lemma 3.4** Let \( z_{k+1}^{(\rho)} = z(x_{k+1}, s_{k+1}; \mu_{k+1}, \rho) \) and \( \hat{z}_{k+1} = z_{k+1}^{(\rho_k)} \). Suppose that \( \|\hat{z}_{k+1} - x_{k+1}\| \neq 0 \) and \( \phi(\mu_{k+1}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}, s_{k+1}) \leq \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \). If \( \rho_{k+1} \geq \rho_k > 0 \), one has

\[
\phi(\mu_{k+1}, \rho_{k+1})(x_{k+1}, \lambda_{k+1}, s_{k+1}) \leq \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k).
\]

**Proof** Note that

\[
\frac{D\phi(\mu_{k+1}, \rho)(x_{k+1}, \lambda_{k+1}, s_{k+1})}{D\rho} \bigg|_{\rho=\rho_k} = -\frac{1}{\rho_k} (\hat{z}_{k+1} - x_{k+1})^T (\hat{Z}_{k+1} + \hat{Y}_{k+1})^{-1} \hat{Z}_{k+1} (\hat{z}_{k+1} - x_{k+1}) < 0,
\]

where \( \hat{Z}_{k+1} = \text{diag} (\hat{z}_{k+1}), \hat{Y}_{k+1} = \text{diag} (\hat{y}_{k+1}) \) with \( \hat{y}_{k+1} = z(x_{k+1}, s_{k+1}; \mu_{k+1}, \rho_k) \). The above equation shows that \( \phi(\mu_{k+1}, \rho)(x_{k+1}, \lambda_{k+1}, s_{k+1}) \) is a monotonically decreasing function on \( \rho \) over \( \rho > 0 \), which implies the desired result. \( \Box \)

By Algorithm 1, the sequence \( \{\rho_k\} \) of penalty parameters is a monotonically nondecreasing sequence. The following result follows from Steps 0.1 and 5.1 immediately.

**Lemma 3.5** There hold

\[ 0 < \mu_{k+1} \leq \mu_k \leq \mu_0 \quad \text{and} \quad \mu_k \leq \eta\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \]

for all \( k > 0 \).

**Proof** The result follows from (36) and \( \mu_{k+2} \leq \mu_{k+2,0} \) immediately. \( \Box \)

## 4 Global Convergence

For global and local convergence analysis, we set \( \epsilon = 0 \). In this situation, Algorithm 1 may have infinite loop in either Step 0.1 for the initial iteration \( k = 0 \) or in Step 5.1 for some iteration \( k > 0 \). In any of these two trivial cases, one will have \( \ell \to \infty \), \( \lim_{\ell \to \infty} \mu_k, \ell = 0 \) and \( \lim_{\ell \to \infty} \phi(\mu_k, \ell, \rho_k)(x_k, \lambda_k, s_k) = 0 \), thus \( (x_k, \lambda_k, s_k) \) is a KKT triple of the problem (1)–(2). Otherwise, Algorithm 1 will generate an infinite sequence of vectors \( \{(x_k, \lambda_k, s_k)\} \). We consider this nontrivial case and prove in this section that, under suitable assumptions, there are some cluster points of the iterative sequence \( \{(x_k, \lambda_k, s_k)\} \) which will be KKT triples of the problem (1)–(2), i.e., the cluster points together with \( \mu^* = 0 \) are solutions of the system of equations (24)–(27).
We need the following blanket assumptions for our global convergence analysis.

**Assumption 4.1**
(1) The functions \( f \) and \( h_i \) \((i \in I)\) are twice continuously differentiable on \( \mathbb{R}^n \);
(2) The iterative sequence \( \{x_k\} \) is in an open bounded set of \( \mathbb{R}^n \);
(3) The sequence \( \{B_k\} \) is bounded, and for all \( k \geq 0 \) and all \( dx \in \mathbb{R}^n \):
   \[ dx \neq 0, \quad \nabla h(x_k)^T dx = 0, \quad d_x^T (B_k + \rho_k Z_k^{-1} Y_k) d_x \geq \chi \|d_x\|^2, \]
   where \( \chi > 0 \) is a constant;
(4) For all \( k \geq 0 \), \( \nabla h(x_k) \) has full column rank.

The above assumptions are commonly used in global convergence analysis for nonlinear programs. Some milder assumptions can be used by incorporating some additional optimization techniques, such as the null-space technology (see Burke et al. 2014; Byrd 1987; Byrd et al. 2000; Liu and Sun 2004; Liu and Yuan 2010) for weakening Assumption 4.1 (3) and (4), and the line search procedure without using a penalty function or a filter (see Gould and Toint 2009; Liu and Yuan 2011) for replacing Assumption 4.1 (2) on the requirement of the boundedness of the iterative sequence by some assumptions on bounded level sets. For simplicity of statement, we leave these concerns outside our scope. The following lemma shows that some related sequences are bounded.

**Lemma 4.2** Under Assumption 4.1, \( \{z_k\} \) is bounded and \( \{s_k\} \) is bounded below. Furthermore, if \( [\nabla h(x_k) I_{A_k}] \) has full column rank for all \( k \), where \( A_k = \{ j \in \{1, 2, \ldots, n\} \mid s_{kj} \neq 0 \} \), \( I_{A_k} \in \mathbb{R}^{n \times |A_k|} \) is a submatrix of \( I_n \) with indices of the columns in \( A_k \), then \( \rho_k \) keeps constant after a finite number of iterations, \( \{y_k\}, \{s_k\} \) and \( \{\lambda_k\} \) are bounded, and there exists a scalar \( \hat{\tau} > 0 \) such that, for \( j = 1, \ldots, n \),

\[ y_{kj} \geq \hat{\tau} \mu_k, \quad z_{kj} \geq \hat{\tau} \mu_k. \]

**Proof** Note that \( z_k \geq 0 \) for all \( k \geq 0 \) and

\[ \phi_{(\mu, k+1, \lambda, k+1, s, k+1)}(x_{k+1}, \lambda_{k+1}, s_{k+1}) \leq \phi_{(\mu, k, k)}(x_k, \lambda_k, s_k) \leq \ldots \leq \phi_{(\mu_0, \rho_0)}(x_0, \lambda_0, s_0). \]

By the definition (28) of \( \phi_{(\mu, \rho)}(x, \lambda, s) \), one has

\[ \frac{1}{2} \|z_k - x_k\|^2 \leq \phi_{(\mu_0, \rho_0)}(x_0, \lambda_0, s_0), \]

which together with Assumption 4.1 (2) implies that \( \{z_k\} \) is bounded. Thus, due to (5), for every \( j = 1, \ldots, n \),

\[ \sqrt{(s_{kj}/\rho_k - x_{kj})^2 + 4 \mu_k \rho_k - (s_{kj}/\rho_k - x_{kj})^2} \]

is bounded. That is, \( s_{kj}/\rho_k \to -\infty \) as \( k \to \infty \), which implies that \( \{s_k\} \) is bounded below.

Note that

\[ \frac{1}{2} \|\nabla f(x_k) - \nabla h(x_k)\lambda_k - s_k\|^2 \leq \phi_{(\mu_0, \rho_0)}(x_0, s_0, \lambda_0). \]  

(50)

If there is a subsequence \( \{s_{k_i}\} \) such that \( \|s_{k_i}\|_\infty \to \infty \) as \( k_i \to \infty \), then, due to (50), one should have \( \|\lambda_{k_i}\|_\infty \to \infty \) as \( k_i \to \infty \). Dividing by \( \|(\lambda, s_k)\|_\infty \) and taking the
Thus, by the boundedness of \( \{s_{ki}\} \) and \( \{\lambda_{ki}\} \) implies that limit on the two sides of (50) as \( k_i \to \infty \), it follows

\[
\lim_{k_i \to \infty} \left\| \nabla h(x_{ki}) \frac{\lambda_{ki}}{\| (\lambda_{ki}, s_{ki}) \|_\infty} + \frac{s_{ki}}{\| s_{ki} \|} \right\| = 0, \tag{51}
\]

which contradicts the condition that \( \{\nabla h(x_k) = I_A_k\} \) is of full column rank. The contradiction shows that \( \{s_{ki}\} \) and \( \{\lambda_{ki}\} \) are bounded. Furthermore, the update rule of \( \rho_k \) implies that \( \{\rho_k\} \) is bounded above. Thus, by (6), \( \{y_k\} \) is bounded.

The relation \( \rho_k y_{kj} z_{kj} = \mu_k \) together with the fact that both \( \{y_k\} \) and \( \{z_k\} \) are bounded implies the desired inequalities.

The preceding results show that, under suitable conditions, \( \rho_k \) will keep constant after a finite number of iterations. In other words, there exists a scalar \( \rho^* > 0 \), such that \( \rho_k = \rho^* \) for all sufficiently large \( k \). In this situation, the sequence \( \{\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)\} \) and the second derivatives of \( \phi(\mu, \rho)(x, \lambda, s) \) for all iterates are bounded. In the following, we prove that there holds \( \mu_k \to 0 \) and \( \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \to 0 \).

**Lemma 4.3** Under Assumption 4.1, suppose that \( \rho_k = \rho^* \) for all sufficiently large \( k \), where \( \rho^* > 0 \) is a scalar. If \( \mu_k \leq \eta \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \) for all sufficiently large \( k \), then

\[
\lim_{k \to \infty} \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) = 0 \quad \text{and} \quad \lim_{k \to \infty} \mu_k = 0.
\]

**Proof** Note that \( \{\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)\} \) is a monotonically nonincreasing sequence. Thus, by the boundedness of \( \{\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)\} \), there is a scalar \( \phi^* \geq 0 \) such that

\[
\lim_{k \to \infty} \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) = \phi^*, \quad \lim_{k \to \infty} \alpha_k \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) = 0.
\]

We prove the result by contradiction. Assume that \( \phi^* > 0 \). Then the preceding equations imply \( \lim_{k \to \infty} \alpha_k = 0 \) and \( \liminf_{k \to \infty} \mu_k > 0 \) since \( \mu_k \) keeps constant provided \( \mu_k \leq \gamma_0 \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \). Hence, by Lemma 4.2, \( z_k \) and \( y_k \) are bounded away from zero. Similar to Lemma 3.2, we can prove that the matrix

\[
\begin{bmatrix}
B_k & -A_k^T & -I \\
A_k & 0 & 0 \\
\rho_k Y_k & 0 & Z_k
\end{bmatrix}
\]

is nonsingular for all \( k \), where \( A_k = \nabla h(x_k)^T \). Therefore, \( \|d_k\| \) is bounded. In this case Assumption 4.1 asserts that \( \alpha_k \) is bounded away from zero since, by (49),

\[
\begin{align*}
\phi(\mu_k + \alpha \Delta \mu_k, \rho_k)(x_k + \alpha d_{sk}, \lambda_k + \alpha d_{sk}, s_k + \alpha d_{sk}) & - (1 - 2 \tau \alpha) \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \\
& = -2(1 - \tau) \alpha \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) + o(\alpha) \\
& \leq -2(1 - \tau) \phi^* \alpha + o(\alpha),
\end{align*}
\]

which suggests that there exists an \( \alpha^* \in (0, 1) \) such that (34) holds for all \( \alpha \in (0, \alpha^*) \). It is contrary to \( \lim_{k \to \infty} \alpha_k = 0 \). This contradiction shows \( \phi^* = 0 \). The desired results are obtained accordingly. \( \square \)

Springer
Now we are ready for presenting our global convergence results on Algorithm 1.

**Theorem 4.4** Under Assumption 4.1, suppose that $\rho_k = \rho^*$ for all sufficiently large $k$, where $\rho^* > 0$ is a scalar. Then one of the following three cases will arise.

1. For all sufficiently large $k$, $\mu_k \leq \eta \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)$. In this case, $\phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k) \to 0$ and $\mu_k \to 0$ as $k \to \infty$. That is, every cluster point of sequence $\{(x_k, \lambda_k, s_k)\}$ is a KKT triple of the original problem.

2. For some iteration $k \geq 0$, $\mu_k > \eta \phi(\mu_k, \rho_k)(x_k, \lambda_k, s_k)$, either Step 0.1 or Step 5.1 of Algorithm 1 has an infinite loop, $\lim_{\ell \to 0} \mu_{k, \ell} = 0$ and $\lim_{\ell \to 0} \phi(\mu_{k, \ell}, \rho_k)(x_k, \lambda_k, s_k) = 0$, i.e., $(x_k, \lambda_k, s_k)$ is a KKT triple of the original problem.

3. Both Step 0.1 and Step 5.1 of Algorithm 1 have finite loops and Step 5.1 of Algorithm 1 is started over infinitely many times. Then $\lim_{k \to \infty} \mu_k = 0$, and there is an infinite subsequence $\{(x_k, \lambda_k, s_k)\}$ of sequence $\{(x_k, \lambda_k, s_k)\}$ such that

$$
\lim_{i \to \infty} \phi(\mu_{k_i}, \rho_k)(x_{k_i}, \lambda_{k_i}, s_{k_i}) = 0.
$$

That is, there is a cluster point of sequence $\{(x_k, \lambda_k, s_k)\}$ is a KKT triple of the original problem.

**Proof** The result in case (1) has been obtained in the preceding Lemma 4.3. In case (2), let $\mu_{k,0} = \mu_k$ and $\mu_{k, \ell} = \mu_{k, \ell-1}/\eta$, where $\ell = 1, 2, \ldots$ is the number of the cycle of while in Step 5.1 of Algorithm 1. Thus, $\lim_{\ell \to \infty} \mu_{k, \ell} = 0$ and $\lim_{\ell \to \infty} \mu_{k, \ell} \geq \lim_{\ell \to \infty} \eta \phi(\mu_{k, \ell}, \rho_k)(x_k, \lambda_k, s_k) \geq 0$ which implies $\lim_{\ell \to \infty} \phi(\mu_{k, \ell}, \rho_k)(x_k, \lambda_k, s_k) = 0$.

Now we prove the result in case (3). Suppose that $k_i$ and $k_{i+1}$ are the indices of two adjoining iterations such that

$$
\mu_{k_i} > \eta \phi(\mu_{k_i}, \rho_k)(x_{k_i}, \lambda_{k_i}, s_{k_i}), \quad \mu_{k_{i+1}} > \eta \phi(\mu_{k_{i+1}}, \rho_{k_{i+1}})(x_{k_{i+1}}, \lambda_{k_{i+1}}, s_{k_{i+1}}),
$$

(52)

$\ell_i$ is the number of loops in Step 5.1 of Algorithm 1 such that

$$
\mu_{k_i, \ell_i} \leq \eta \phi(\mu_{k_i}, \ell_i, \rho_k)(x_{k_i}, \lambda_{k_i}, s_{k_i}).
$$

Since $\mu_{k_i, \ell_i} \geq \gamma \phi(\mu_{k_i, \ell_i}, \rho_k)(x_{k_i}, \lambda_{k_i}, s_{k_i})$, one has

$$
\mu_{k_{i+1}} = (1 - \alpha_{k_i})\mu_{k_i, \ell_i} + \alpha_{k_i} \gamma \phi(\mu_{k_i, \ell_i}, \rho_k)(x_{k_i}, \lambda_{k_i}, s_{k_i}) \leq \mu_{k_i, \ell_i} \leq \mu_{k_i}/\eta,
$$

and $\mu_{k_{i+1}} \leq \mu_{k_{i+1}} \leq \mu_{k_i}/\eta$. Thus, a strictly monotonically decreasing infinite subsequence $\{\mu_{k_i}\}$ satisfying (52) is derived. Therefore,

$$
\lim_{i \to \infty} \mu_{k_i} = 0, \quad \lim_{i \to \infty} \phi(\mu_{k_i}, \rho_k)(x_{k_i}, \lambda_{k_i}, s_{k_i}) = 0.
$$

Note that $\{\mu_{k_i}\}$ is a monotonically nonincreasing sequence, the desired result is straightforward by the preceding equations. □
5 Local convergence

In this section, we prove that, under suitable conditions, our algorithm with global convergence result (1) of Theorem 4.4 can be quadratically convergent to the KKT point of the original problem. For convenience of statement, we denote \( w^* = (x^*, \lambda^*, s^*) \) and \( w_k = (x_k, \lambda_k, s_k) \in \mathbb{R}^{2n+m} \) for all \( k \geq 0 \). The following blanket assumptions are requested for local convergence analysis.

**Assumption 5.1**

1. \( w_k \to w^* \) and \( \mu_k \to 0 \) as \( k \to \infty \);
2. The functions \( f \) and \( h_i (i = 1, \ldots, m) \) are twice differentiable on \( \mathbb{R}^n \), and their second derivatives are Lipschitz continuous at some neighborhood of \( x^* \);
3. The gradients \( \nabla h_i (x^*) (i = 1, \ldots, m) \) are linearly independent;
4. There holds \( x^* + s^* > 0 \);
5. \( d^T B^* d > 0 \) for all \( d \neq 0 \) such that \( \nabla h(x^*)^T d = 0 \) and \( d_j = 0 \) for \( j \in \{ j | x_j^* = 0, j = 1, \ldots, n \} \), where \( B^* = \nabla^2 f (x^*) - \sum_{i=1}^m \lambda_i^* \nabla^2 h_i (x^*) \) and \( \lambda^* \in \mathbb{R}^m \) is the associated Lagrange multiplier vector at \( x^* \) for all equality constraints, \( d_j \) is the \( j \)-th component of \( d \).

The convergence \( w_k \to w^* \) as \( k \to \infty \) in Assumption 5.1 (1) may be replaced by assuming \( \| w_k - w^* \| \leq \epsilon \) for all \( k \geq 0 \) and for \( \epsilon \) sufficiently small without affecting our proofs and results. Under Assumption 5.1, \( \{ s_k \} \) is bounded, thus \( \rho_k \) will keep constant after a finite number of iterations. By Theorem 4.4, \( (x^*, \lambda^*, s^*) \) is a KKT triple of the original problem. Without loss of generality, let \( \rho_k = \rho^* \) for all \( k \geq 0 \), and, correspondingly, \( y_k \to y^* \) and \( z_k \to z^* \) as \( k \to \infty \). It follows from (5) and (6) that \( z^* = x^* \) and \( y^* = s^*/\rho^* \). Thus, \( z_j^* + y_j^* > 0 \) for all \( j = 1, \ldots, n \).

**Lemma 5.2**

Suppose that Assumption 5.1 hold. Let \( Y^* = \text{diag} (y^*) \) and \( Z^* = \text{diag} (z^*) \). Then the matrix

\[
G^* = \begin{bmatrix}
1 + \gamma \frac{D \phi_{(0,0^*),(w^*)}}{D \mu} & \gamma (\nabla_w \phi_{(0,0^*)}(w^*))^T \\
0 & 0 \\
-\frac{1}{\rho^*} (Z^* + Y^*)^{-1} e & H^*
\end{bmatrix}
\]

is nonsingular, where \( \frac{D \phi_{(0,0^*),(w^*)}}{D \mu} = \frac{D \phi_{(\mu,0^*)}(w)}{D \mu}_{|(\mu, \rho)=(0,0^*), w=w^*} \) and \( \nabla_w \phi_{(0,0^*)}(w^*) = \nabla_w \phi_{(\mu,0^*)}(w)_{|(\mu, \rho)=(0,0^*), w=w^*} \) and

\[
H^* = \begin{bmatrix}
B^* & -\nabla h(x^*) & -I \\
\nabla h(x^*)^T & 0 & 0 \\
(Z^* + Y^*)^{-1} & 0 & \frac{1}{\rho^*} (Z^* + Y^*)^{-1} Z^*
\end{bmatrix}.
\]

**Proof**

In order to derive the result, we need only to prove that the system

\[
G^* d = 0
\]
has a unique solution \( d^* = 0 \). Corresponding to the partition of \( G^* \), \( d \in \mathbb{R}^{2n+m+1} \) has a partition \( d = (d_\mu, d_w) \), where \( d_\mu \in \mathbb{R}, d_w = (d_x, d_\lambda, d_s) \) with \( d_x \in \mathbb{R}^n, d_\lambda \in \mathbb{R}^m \), and \( d_s \in \mathbb{R}^n \). Thus,

\[
(1 + \gamma \frac{D\phi_{(0, \rho^*)}(w^*)}{D\mu})d_\mu + \gamma (\nabla_w \phi_{(0, \rho^*)}(w^*))^T d_w = 0,
\]

(53)  

\[
B^* d_x - \nabla h(x^*)d_\lambda - d_s = 0,
\]

(54)  

\[
\nabla h(x^*)^T d_x = 0,
\]

(55)  

\[
- \frac{1}{\rho^*}(Z^* + Y^*)^{-1}ed_\mu + (Z^* + Y^*)^{-1}Y^* d_x + \frac{1}{\rho^*}(Z^* + Y^*)^{-1}Z^* d_s = 0.
\]

(56)  

Note that \( \frac{D\phi_{(0, \rho^*)}(w^*)}{D\mu} = 0 \) and \( \nabla_w \phi_{(0, \rho^*)}(w^*) \) is small since \( z^* - x^* = 0 \) and \( w^* \) is a KKT triple of the original problem. Thus, due to (53), \( d_\mu^* = 0 \). Furthermore, since \( y_j^* z_j^* = 0 \) for all \( j = 1, \ldots, n \), (56) implies \( (d_\mu^*)^T d_s^* = 0 \), and \( (d_\mu^*)^T (d_s^*) = 0 \) when \( x_j^* = 0 \), \( (d_\mu^*) = 0 \) as \( s_j^* = 0 \), where \( (d_\mu^*)_j \) and \( (d_s^*)_j \) are, respectively, the \( j \)-th components of \( d_\mu^* \) and \( d_s^* \). Hence,

\[
(d_\mu^*)^T B^* d_s^* = 0, \quad \nabla h(x^*)^T d_\lambda^* = 0, \quad (d_\mu^*)_j = 0 \text{ for } j \in \{ j|x_j^* = 0, j = 1, \ldots, n \},
\]

which, due to Assumption 5.1 (5), implies \( d_\mu^* = 0 \). Finally, \( d_\lambda^* = 0 \) follows from Assumption 5.1 (3) since \( \nabla h(x^*)d_\lambda = 0 \).

The preceding proof also shows that \( H^* p = 0 \) implies \( p = 0 \). Thus, \( H^* \) is also nonsingular. Let \( w = (x, \lambda, s) \) and

\[
\Phi(\mu, w) = \begin{bmatrix}
\mu + \gamma \phi_{(\mu, \rho^*)}(w) & \nabla f(x) - \nabla h(x) \lambda - s \\
\nabla h(x) & z - x
\end{bmatrix}.
\]

Then \( \Phi(0, w^*) = 0 \). The following lemma can be obtained in a way similar to Lemma 2.1 in Byrd et al. (1997). We will not give its proof for brevity.

**Lemma 5.3** Suppose that Assumption 5.1 holds. Then there are sufficiently small scalar \( \epsilon > 0 \) and positive constants \( M_0 \) and \( L_0 \), such that, for all \( (\mu, w) \in \mathbb{R}_{++} \times \mathbb{R}^{2n+m+1} \) \( ||(\mu, w) - (0, w^*)|| < \epsilon \), \( \nabla_{(\mu, w)} \Phi(\mu, w) \) is invertible, \( \| [\nabla_{(\mu, w)} \Phi(\mu, w)]^{-1} \| \leq M_0 \), and

\[
\|(\nabla_{(\mu, w)} \Phi(\mu, w))^T ((\mu, w) - (0, w^*)) - \Phi(\mu, w)\| \leq L_0 \| (\mu, w) - (0, w^*) \|^2,
\]

(57)  

where \( \nabla_{(\mu, w)} \Phi(0, w^*) = \nabla_{(\mu, w)} \Phi(\mu, w)|_{\mu=0, w=w^*} \).

Using Lemma 5.3, the following result shows that the step \( (\Delta \mu_k, d_k) \) can be a quadratically or superlinearly convergent step.
Theorem 5.4 Suppose that Assumption 5.1 holds. Then there is a sufficiently small scalar $\epsilon > 0$, such that, for all $(\mu_k, w_k) \in \{(\mu, w) \in \mathbb{R}^{n+n} | ((\mu, w) - (0, w^*)) < \epsilon \}$, one has the following results.

(1) If $\| (B_k - B^*)d_x \| = O(\|d_x\|^2)$ for every $d_x \in \mathbb{R}^n$, then

$$
\| (\mu_k, w_k) + (\Delta \mu_k, d_k) - (0, w^*) \| = O(\| (\mu_k, w_k) - (0, w^*) \|^2).
$$

That is, $(\Delta \mu_k, d_k)$ is a quadratically convergent step.

(2) If $\| (B_k - B^*)d_x \| = o(\|d_x\|)$ for every $d_x \in \mathbb{R}^n$, then

$$
\| (\mu_k, w_k) + (\Delta \mu_k, d_k) - (0, w^*) \| = o(\| (\mu_k, w_k) - (0, w^*) \|),
$$

i.e., $(\Delta \mu_k, d_k)$ is a superlinearly convergent step.

Proof In order to prove the result (1), we show

$$
\limsup_{k \to \infty} \| (\mu_k, w_k) + (\Delta \mu_k, d_k) - (0, w^*) \| / \| (\mu_k, w_k) - (0, w^*) \|^2 \leq \xi,
$$

where $\xi > 0$ is a constant.

Let $\Phi_k = \Phi(\mu_k, w_k)$, $J_k = \nabla_{(\mu, w)} \Phi(\mu_k, w_k)^T$, $G_k$ is a matrix which has the same components as $J_k$ except that the Lagrangian Hessian $\nabla_{xx}^2 L(w_k) = \nabla^2 f(x_k) - \sum_{i=1}^m \lambda_i \nabla^2 h_i(x_k)$ in $J_k$ is replaced by $B_k$. Then $G_k(\Delta \mu_k, d_k) = -\Phi_k$.

By Lemma 5.3, $J_k$ is invertible. Note that

$$
G_k = J_k + G_k - J_k = J_k + \begin{bmatrix} 0 & 0 & 0 \\ 0 & B_k - B^* & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \nabla_{xx}^2 L(w_k) - B^* & 0 \\ 0 & 0 & 0 \end{bmatrix},
$$

it follows from the condition $\| (B_k - B^*)d_x \| = O(\|d_x\|^2)$ and Assumption 5.1 (2) that $G_k$ is invertible and $\| G_k^{-1} \| \leq M_0$ for some scalar $M_0 > 0$ and for all sufficiently large $k$. Thus, $\| (\Delta \mu_k, d_k) \| = O(\| (\mu_k, w_k) - (0, w^*) \|)$. Moreover,

$$
G_k(\Delta \mu_k, d_k) = J_k(\Delta \mu_k, d_k) + (B_k - B^*)d_{xk} + (\nabla_{xx}^2 L(w_k) - B^*)d_{xk} = -\Phi_k.
$$

Therefore,

$$
\| (\mu_k, w_k) + (\Delta \mu_k, d_k) - (0, w^*) \|
= \| J_k^{-1}(J_k((\mu_k, w_k) - (0, w^*)) - \Phi_k - (B_k - B^*)d_{xk} + (\nabla_{xx}^2 L(w_k) - B^*)d_{xk}) \|
\leq M_0[\| (\mu_k, w_k) - (0, w^*) \|^2 + O(\| (\mu_k, w_k) - (0, w^*) \|^2)],
$$

where the last inequality follows from (57) of Lemma 5.3. Thus, (60) follows immediately from (61).

If $\| (B_k - B^*)d_x \| = o(\|d_x\|)$, then the last inequality (61) should be

$$
\| (\mu_k, w_k) + (\Delta \mu_k, d_k) - (0, w^*) \|
$$

 Springer
\[ \leq M_0 L_0 \|(\mu_k, w_k) - (0, w^*)\|^2 + o(\|(\mu_k, w_k) - (0, w^*)\|). \] (62)

Hence, the result (2) follows immediately. \qed

Now we prove that, under suitable conditions, our algorithm can be quadratically convergent to the KKT triple of the original problem.

**Theorem 5.5** Suppose that Assumption 5.1 holds. If \( \|(B_k - B^*) d_x\| = O(\|d_x\|^2) \) for every \( d_x \in \mathfrak{H}^n \), \( \tau < 1/2 \), then either \( \mu_{k+1} = \mu_k \) or \( \mu_{k+1} = \gamma \phi(\mu_k, \rho^*)(w_k) \), \( x_{k+1} = x_k + d_{x,k} \), \( s_{k+1} = s_k + d_{s,k} \), and \( \lambda_{k+1} = \lambda_k + d_{\lambda,k} \) for all sufficiently large \( k \).

**Proof** We need to prove that, for all sufficiently large \( k \), \( \alpha_k = 1 \) will be accepted by the line search procedure (34). By Theorem 5.4, \( \phi(\mu_k + \Delta \mu_k, \rho^*)(w_k + d_k) = \phi(\mu_k + \Delta \mu_k, \rho^*)(w_k + d_k) - \phi(0, \rho^*)(w^*) + O(\|(\mu_k, w_k) + (\Delta \mu_k, d_k) - (0, w^*)\|^2) \) and \( \tau < \frac{1}{2} \). Thus,

\[ (1 - 2\tau) \phi(\mu_k, \rho^*)(w_k) = O(\|(\mu_k, w_k) - (0, w^*)\|), \]

and the full step will be accepted by (34).

By Theorem 5.4 (1),

\[ \|(\mu_{k+1}, w_{k+1}) - (0, w^*)\| = O(\|(\mu_k, w_k) - (0, w^*)\|^2). \] (63)

Due to \( \mu_k \leq \eta \phi(\mu_k, \rho^*)(w_k) \), \( \mu_k = O(\|w_k - w^*\|) \). The desired result follows from (63) immediately. \qed

**6 Numerical experiments**

Our method can be easily extended to solve the nonlinear programs with general equality and inequality constraints

\[ \min f(x) \] (64)
\[ \text{s.t. } h(x) = 0, \quad g(x) \geq 0, \] (65)

by substituting (5) and (6), respectively, with

\[ z_j(x, s; \mu, \rho) \equiv \frac{1}{2\rho} \left( (s_j - \rho g_j(x))^2 + 4\rho \mu - (s_j - \rho g_j(x)) \right), \]
\[
y_j(x, s; \mu, \rho) \equiv \frac{1}{2\rho} \left( \sqrt{(s_j - \rho g_j(x))^2 + 4\rho \mu} + (s_j - \rho g_j(x)) \right),
\]

where \( g : \mathbb{R}^n \to \mathbb{R}^m \) is a twice continuously differentiable real-valued function on \( \mathbb{R}^n \), \( j = 1, \ldots, m \). No slack variables are introduced to cope with the general inequality constraints, which is different from the technique commonly used in interior-point methods for nonlinear programs (64)–(65). Our numerical experiments are conducted on a Lenovo laptop with the LINUX operating system (Fedora 11). Algorithm 1 is implemented in MATLAB (version R2008a).

The algorithm is firstly used to solve a well-posed nonlinear program from the literature. The test problem was presented by Wächter and Biegler (2000) and further discussed by Byrd et al. (2004):

\[
\begin{align*}
\min & \quad x \\
\text{s.t.} & \quad x^2 - 1 \geq 0, \quad x - 2 \geq 0.
\end{align*}
\]

This problem is well-posed since it has a unique global minimizer \( x^* = 2 \), at which both the linear independence constraint qualification (short for LICQ) and the Mangasarian-Fromowitz constraint qualification (short for MFCQ) hold. However, starting from \( x_0 = -4 \), Wächter and Biegler (2000) showed that many line-search infeasible interior-point methods may be jammed and fail to find the solution.

Algorithm 1 is then used to find the solutions for a set of nonlinear programming test problems of the CUTEr collection (Bongartz et al. 1995). Since the code is very elementary, we restricted our test problems to the 122 HS problems, where problems HS101–103 were excluded since they are only defined on positive variables. These test problems include not only the problems with general equality and inequality constraints, but also the problems with bound constraints and the problems with only equality constraints (Hock and Schittkowski 1981).

In our implementation, the initial parameters are selected as follows: \( \mu_0 = 0.1 \), \( \rho_0 = 1 \), \( \eta = 10 \), \( \gamma_0 = 0.001 \), \( \delta = 0.5 \), \( \tau = 0.01 \), \( \sigma = 0.01 \), and \( \epsilon = 10^{-8} \). For all \( k \geq 0 \), we take \( B_k \) to be the exact Lagrangian Hessian provided that it is positive semi-definite (where the gradient and Hessian are provided by the test sets). Otherwise, we modify \( B_k \) to \( B_k + \xi I \) with \( \xi > 0 \) being as small as possible so that the modified Hessian is positive semi-definite.

For comparison, these test problems are also solved by the well regarded and recognized interior-point solver IPOPT (Wächter and Biegler 2006) (Version 3.0.0). In implementation, Algorithm 1 can use the KKT residuals of the original problem directly as the measure of our terminating conditions:

\[
E(x_k, \lambda_k, s_k) \leq \epsilon,
\]

where

\[
E(x_k, \lambda_k, s_k) = \max \{ \| \nabla f(x_k) - \nabla h(x_k) \lambda_k - s_k \|_\infty, \| h(x_k) \|_\infty, \| \max \{ - (x_k + s_k), 0 \} \|_\infty, \| x_k \circ s_k \|_\infty \},
\]

\( \subseteq \) Springer
$x_k \circ s_k$ is the Hadamard product of $x_k$ and $s_k$. If one has the scaling parameters $s_d = 1$ and $s_c = 1$ in the terminating conditions of Wächter and Biegler (2006), then the accuracy differences between Algorithm 1 and IPOPT should be in the range of the tolerance.

For test problem (66)–(67), we use the standard initial point $x_0$ as the starting point, and set $s_0$ to be the all-one vector. The implementation of our algorithm terminates at $x^* = 2$ together with $s^*_1 = -1.1972 \times 10^{-16}$, $s^*_2 = 1.0000$ in 4 iterations. Both the numbers of function and gradient evaluations are 5. See Table 1 for more details on iterations. From there one can observe the rapid convergence of $\mu_k$, $\phi_k(x_k, \lambda_k, s_k)$ and $E(x_k, \lambda_k, s_k)$, where $\mu_k$ is the current value of the parameter, $x_k$ and $s_k$ are the estimates of the primal and dual variables, respectively, $f_k = f(x_k)$, $v_k$ is the $\ell_\infty$ norm of violations of constraints, $\phi_k = \phi_k(x_k, \lambda_k, s_k)$, $E_k = E_k(x_k, \lambda_k, s_k)$. As a comparison, IPOPT fails to find the solution and terminates at $x^* = -1.0000$ in 13 iterations. In interior-point framework, this problem has been solved by the recently developed methods of Dai et al. (2020) and Liu and Dai (2020) in totally 16 and 19 iterations, respectively.

When solving the HS test problems of the CUTEr collection, Algorithm 1 was terminated as either $E(x_k, \lambda_k, s_k) \leq \epsilon$, or the number of iterations is larger than 500 (which is the default setting of IPOPT), the step-size is too small ($\alpha_k \leq \delta^{40}$), the coefficient matrix of the system (33) is degenerate. The latter three cases of termination can be resulted from that the Hessian does not satisfy Assumption 4.1 (3), the condition (4) of Assumption 4.1 does not hold, and some test problems are only defined on strictly positive variables.

Since we do not require the iterates to be interior points, our algorithm has the freedom to use the standard initial points for all HS test problems. However, for the purpose of comparison, we have modified the initial points in line with the initialization of IPOPT (Wächter and Biegler 2006). In our implementation, Algorithm 1 successfully solved 79 problems and terminated with (68), while IPOPT found the approximate solutions of 121 problems satisfying its default terminating conditions, where only for problem HS87 IPOPT reached its restriction of the maximum of the total number of iterations.

In order to further observe how Algorithm 1 performs in solving nonlinear programming test problems, we provide 4 figures Figs. 1, 2, 3 and 4 to show log scaling performance profiles (see Dolan and More 2002) of our algorithm in comparison with IPOPT on both solved 79 problems with respect to iteration count, function evaluations, gradient evaluations, and the CPU time, where IPRM represents our primal-dual interior-point relaxation method (Algorithm 1), respectively. Figs. 1, 2 and 3 show that, under the measures on the former three items, IPRM performs approximate but inferior to IPOPT. However, Fig. 4 shows that IPRM needs less CPU time than IPOPT, which may be partially resulted from that the system (33) in IPRM is solved by the MATLAB’s built-in “backslash” command and that our algorithm does not incorporate any sophisticated techniques such as inertia correction, feasibility restoration, and so on.

Since our method is currently at a very early stage of development, and we note that a nonmonotone line search variant of our algorithm can successfully solve more than 100 HS test problems, it is not surprising that our implementation of Algorithm 1
Table 1  Output of Algorithm 1 for test problem (66)–(67)

| $k$ | $\mu_k$ | $x_k$ | $s_k$    | $f_k$ | $v_k$ | $\phi_k$ | $E_k$ |
|-----|---------|-------|---------|-------|-------|----------|-------|
| 0   | 0.1     | −4    | (1, 1)  | −4    | 6     | 50.5785  | 15    |
| 1   | 0.0506  | 2.0190| (0.0276, 1.2212) | 2.0190 | 0     | 0.0557   | 0.3328|
| 2   | 5.5681e−05 | 2.0080 | (0.0002, 0.9992) | 2.0080 | 0     | 3.1754e−05 | 0.0080|
| 3   | 3.1754e−13 | 2.0000 | (0.0000, 1.0000) | 2.0000 | 4.6437e−07 | 1.2875e−13 | 6.1372e−07|
| 4   | 1.2875e−16 | 2     | (−0.0000, 1.0000) | 2     | 0     | 6.1630e−33 | 3.5916e−16|
is not very convincing in comparison to the very regarded and recognized IPOPT. However, it is still encouraging by the numerical experiments since Algorithm 1 has still much space for improvement such as incorporating some scaling and inertial control techniques and using some robust subroutine and solver for solving the system (33) more efficiently.
7 Conclusion

We present a novel primal-dual interior-point relaxation method for nonlinear programs with general equality and nonnegative constraints in this paper. The method can be easily extended to solve the problems with general inequality constraints. It is based on solving a parametric equality constrained mini-max subproblem. Our method is of the interior-point variety, but does not require any primal or dual iterates to be interior. A new smoothing approach is introduced. Our method is capable of circumventing the jamming difficulty which results in that many interior-point methods failed.
A primal-dual interior-point relaxation method

to converge to the solution and improving the ill conditioning of the classic primal-dual interior-point methods as the barrier is small. Under suitable conditions, our method is proved to be globally convergent and locally quadratically convergent to the KKT triple of the original problem. Preliminary numerical results on a well-posed problem for which many line-search interior-point methods fail to find the minimizer and a set of test problems from CUTEr collection show that our method is efficient.

Acknowledgements The research is supported by the NSFC grants (nos. 12071108, 11671116, 12021001, 11991021, 11991020, 11971372, and 11701137), National Key R&D Program of China (nos. 2021YFA1000300 and 2021YFA1000301), the Strategic Priority Research Program of Chinese Academy of Sciences (no. XDA27000000), and the Natural Science Foundation of Hebei Province (no. A2021202010).

References

Benson HY, Shanno DF (2007) An exact primal-dual penalty method approach to warmstarting interior-point methods for linear programming. Comput Optim Appl 38:371–399
Benson HY, Shanno DF (2008) Interior-point methods for nonconvex nonlinear programming: regularization and warmstarts. Comput Optim Appl 40:143–189
Benson HY, Shanno DF, Vanderbei RJ (2004) Interior-point methods for nonconvex nonlinear programming: jamming and comparative numerical testing. Math Program 99:35–48
Bongartz I, Conn AR, Gould NIM, Toint PL (1995) CUTEr: Constrained and Unconstrained Testing Environment. ACM Tran Math Software 21:123–160
Burke JV, Curtis FE, Wang H (2014) A sequential quadratic optimization algorithm with rapid infeasibility detection. SIAM J Optim 24:839–872
Byrd RH (1987) Robust trust-region method for constrained optimization. Paper presented at the SIAM Conference on Optimization, Houston, TX
Byrd RH, Gilbert JC, Nocedal J (2000) A trust region method based on interior point techniques for nonlinear programming. Math Program 89:149–185
Byrd RH, Hribar ME, Nocedal J (1999) An interior point algorithm for large-scale nonlinear programming. SIAM J Optim 9:877–900
Byrd RH, Liu G, Nocedal J (1997) On the local behaviour of an interior point method for nonlinear programming. In: Griffiths DF, Higham DJ (eds) Numerical Analysis. Addison-Wesley Longman, Reading, MA, pp. 37–56
Byrd RH, Marazzi M, Nocedal J (2004) On the convergence of Newton iterations to non-stationary points. Math Program 99:127–148
Chen LF, Goldfarb D (2006) Interior-point $\ell_2$-penalty methods for nonlinear programming with strong global convergence properties. Math Program 108:1–36
Conn AR, Gould NIM, Toint PhL (1988) Testing a class of algorithms for solving minimization problems with simple bounds on the variables. Math Comput 50:399–430
Conn AR, Gould NIM, Toint PhL (1992) LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A). Springer-Verlag, Berlin/Heidelberg,
Curtis FE (2012) A penalty-interior-point algorithm for nonlinear constrained optimization. Math Program Comput 4:181–209
Curtis FE, Gould NIM, Robinson DP (2017) An interior-point trust-funnel algorithm for nonlinear optimization. Math Program 161:73–134
Dai YH, Liu XW, Sun J (2020) A primal-dual interior-point method capable of rapidly detecting infeasibility for nonlinear programs. J Ind Manag Optim 16:1009–1035
Dolan ED, Mofe JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91:201–213
Engau A, Anjos MF, Vannelli A (2009) A primal-dual slack approach to warmstarting interior-point methods for linear programming. In: Chinneck JW, Kristjansson B, Saltzman MJ (eds) Operations Research and Cyber-Infrastructure. Springer, US, pp. 195–217
Fiacco AV, McCormick GP (1990) Nonlinear Programming: Sequential Unconstrained Minimization Techniques. SIAM, Philadelphia, PA
Forsgren A, Gill PE (1998) Primal-dual interior methods for nonconvex nonlinear programming. SIAM J Optim 8:1132–1152
Gertz EM, Gill PhE (2004) A primal-dual trust region algorithm for nonlinear optimization. Math Program 100:49–94
Gould NIM, Orban D, Toint PhL (2015) An interior-point $l_1$-penalty method for nonlinear optimization. Recent Developments in Numerical Analysis and Optimization, Proceedings of NAOIII 2014, Springer, Verlag, 134:117–150
Gould NIM, Toint PhL (2009) Nonlinear programming without a penalty function or a filter. Math Program 122:155–196
Haeser G, Hinder O, Ye Y (2019) On the behavior of Lagrange multipliers in convex and nonconvex infeasible interior point methods. Math Program. https://doi.org/10.1007/s10107-019-01454-4
Hinder O, Ye Y (2018) A one-phase interior point method for nonconvex optimization. arXiv: 1801.03072
Hock W, Schittkowski K (1981) Test Examples for Nonlinear Programming Codes. Lecture Notes in Econ. and Math. Systems 187, Springer-Verlag, Berlin, New York
Liu XW, Dai YH (2020) A globally convergent primal-dual interior-point relaxation method for nonlinear programs. Math Comput 89:1301–1329
Liu XW, Sun J (2004) A robust primal-dual interior-point algorithm for nonlinear programs. SIAM J Optim 14:1163–1186
Liu XW, Yuan YX (2010) A null-space primal-dual interior-point algorithm for nonlinear optimization with nice convergence properties. Math Program 125:163–193
Liu XW, Yuan YX (2011) A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization. SIAM J Optim 21:545–571
Mehrotra S (1992) On the implementation of a primal-dual interior point method. SIAM J Optim 2:575–601
Nocedal J, Öztoprak F, Waltz RA (2014) An interior point method for nonlinear programming with infeasibility detection capabilities. Optim Methods Softw 4:837–854
Nocedal J, Wächter A, Waltz RA (2009) Adaptive barrier update strategies for nonlinear interior methods. SIAM J Optim 19:1674–1693
Nocedal J, Wright S (1999) Numerical Optimization. Springer-Verlag, New York
Qi LQ, Sun DF, Zhou GL (2000) A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities. Math Program 87:1–35
Shanno DF, Vanderbei RJ (2000) Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods. Math Program 87:303–316
Sun WY, Yuan YX (2006) Optimization Theory and Methods: Nonlinear Programming. Springer, New York
Ulbrich M, Ulbrich S, Vicente LN (2004) A globally convergent primal-dual interior-point filter method for nonlinear programming. Math Program 100:379–410
Vanderbei RJ, Shanno DF (1999) An interior-point algorithm for nonconvex nonlinear programming. Comput Optim Appl 13:231–252
Wächter A, Biegler LT (2000) Failure of global convergence for a class of interior point methods for nonlinear programming. Math Program 88:565–574
Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106:25–57
Wright SJ (1997) Primal-Dual Interior-Point Methods. SIAM Publications, Philadelphia, PA
Ye Y (1997) Interior Point Algorithms: Theory and Analysis. John Wiley & Sons, Inc, Hoboken, New Jersey