MATHEMATICS OF COMPUTATION
https://doi.org/10.1090/mcom/3487
Article electronically published on December 6, 2019

A GLOBALLY CONVERGENT PRIMAL-DUAL INTERIOR-POINT RELAXATION METHOD FOR NONLINEAR PROGRAMS

XIN-WEI LIU AND YU-HONG DAI

Abstract. We prove that the classic logarithmic barrier problem is equivalent to a particular logarithmic barrier positive relaxation problem with barrier and scaling parameters. Based on the equivalence, a line-search primal-dual interior-point relaxation method for nonlinear programs is presented. Our method does not require any primal or dual iterates to be interior-points, which has similarity to some warmstarting interior-point methods and is different from most of the globally convergent interior-point methods in the literature. A new logarithmic barrier penalty function dependent on both primal and dual variables is used to prompt the global convergence of the method, where the penalty parameter is adaptively updated. Without assuming any regularity condition, it is proved that our method will either terminate at an approximate KKT point of the original problem, an approximate infeasible stationary point, or an approximate singular stationary point of the original problem. Some preliminary numerical results are reported, including the results for a well-posed problem for which many line-search interior-point methods were demonstrated not to be globally convergent, a feasible problem for which the LICQ and the MFCQ fail to hold at the solution and an infeasible problem, and for some standard test problems of the CUTE collection. Correspondingly, for comparison we also report the numerical results obtained by the interior-point solver IPOPT. These results show that our algorithm is not only efficient for well-posed feasible problems, but is also applicable for some feasible problems without LICQ or MFCQ and some infeasible problems.

1. Introduction

We consider the general nonlinear programs with equality and inequality constraints

$$\text{minimize}_{x\in\Re^n} \quad (\min_x) \quad f(x) \tag{1.1}$$

$$\text{subject to} \quad (\text{s.t.}) \quad h_i(x)=0,\ i=1,\dots,m_e, \tag{1.2}$$

$$c_j(x)\le 0,\ j=1,\dots,m, \tag{1.3}$$

Received by the editor December 1, 2017, and, in revised form, August 15, 2018, April 7, 2019, May 29, 2019, and June 20, 2019.

2010 *Mathematics Subject Classification.* Primary 90C30, 90C51; Secondary 90C26.

Key words and phrases. Nonlinear programming, constrained optimization, interior-point method, logarithmic barrier problem, global convergence.

The first author is the corresponding author.

The research was supported in part by the Chinese NSF grants (nos. 11671116 and 11271107), the Major Research Plan of the National Natural Science Foundation of China (no. 91630202), and a grant (no. A2015202365) from the Hebei Natural Science Foundation of China.

The second author was supported in part by the Chinese NSF grants (nos. 11631013, 11331012 and 81173633) and the National Key Basic Research Program of China (no. 2015CB856000).

where m_e and m are positive integer numbers, $x \in \Re^n$, f, h_i $(i = 1, \dots, m_e)$, and c_j $(j = 1, \dots, m)$ are twice continuously differentiable real-valued functions defined on $\Re^n$. Interior-point methods have been among the most efficient methods for nonlinear programs (for example, see [8, 11, 16, 26, 28]). Given parameter $\mu > 0$, interior-point methods for nonlinear program (1.1)–(1.3) usually need to approximately solve the following logarithmic barrier problem:

$$\min_{x,t} \quad f(x) - \mu \sum_{j=1}^{m} \ln t_j \tag{1.4}$$

$$\text{s.t.} \quad h_i(x) = 0, \; i = 1, \dots, m_e, \tag{1.5}$$

$$c_j(x) + t_j = 0, \; j = 1, \dots, m, \tag{1.6}$$

where $t_j > 0$ $(j = 1, \dots, m)$ are slack variables for inequality constraints. Some other interior-point methods approximately solve the KKT system of the above logarithmic barrier problem as follows:

$$\nabla f(x) + \sum_{i=1}^{m_e} \lambda_i \nabla h_i(x) + \sum_{j=1}^{m} s_j \nabla c_j(x) = 0, \tag{1.7}$$

$$t_j > 0, \; s_j > 0, \; t_j s_j - \mu = 0, \; j = 1, \dots, m, \tag{1.8}$$

$$h_i(x) = 0, \; i = 1, \dots, m_e; \quad c_j(x) + t_j = 0, \; j = 1, \dots, m, \tag{1.9}$$

where $\lambda = (\lambda_i, i = 1, \dots, m_e) \in \Re^{m_e}$, $s = (s_j, j = 1, \dots, m) \in \Re^m$ are, respectively, the Lagrange multiplier vectors associated with constraints in (1.5) and (1.6), and $\mu > 0$ is the barrier parameter which can be any number of a decreasing sequence with the limit 0.

During the iterative processes for the logarithmic barrier problem (1.4)–(1.6) and the system (1.7)–(1.9), all primal and dual iterates must be interior points. That is, they should always have $t = (t_j, j = 1, \dots, m) > 0$ and $s > 0$. This requirement for iterates to be interior-points may result in the truncations of the primal and dual steps and can impact the global performance of many existing line-search interior-point methods. An analytical counterexample presented in Wächter and Biegler [27] demonstrated that interior-point methods using linearized constraints and a commonly used interior-point mechanism may fail in converging to any feasible point of a well-posed problem. As $\mu > 0$ is small enough, an approximate KKT point of problem (1.4)–(1.6) is usually expected to be an approximate KKT point of the original problem.

There are already many interior-point methods in the literature which do not suffer from the failure in [27]. These methods either use trust region techniques for new iterates (such as [8, 25]) or change the system (1.7)–(1.9) and/or its linearized system by either penalty or null-space technique (e.g. [2, 11, 14, 16, 18, 20, 28]). Very recently, with the help of the augmented Lagrangian function of the logarithmic barrier problem, Dai, Liu, and Sun [12] presented a new primal-dual interior-point method for nonlinear programs. It was proved that the method had the capability of rapidly detecting the infeasibility of the problem, which is a very useful property in practice.

In this paper, we first prove that the logarithmic barrier problem (1.4)–(1.6) is equivalent to a particular logarithmic barrier positive relaxation problem with barrier and scaling parameters (see problem (2.8)–(2.11) in the next section). That is, we can derive a KKT point of problem (1.4)–(1.6) by solving its equivalent problem (2.8)–(2.11). Based on the equivalence, a primal-dual interior-point relaxation method for nonlinear programs (1.1)–(1.3) is then presented. A remarkable characteristic of our method is that it does not require any primal or dual iterates to be interior-points, which is similar to some warmstarting interior-point methods (for example, see [1,13]) and is different from the method of [12] (where the primal and dual iterates are always interior-points) and most of the globally convergent interior-point methods in the literature. This characteristic of our method makes us free from having to truncate the primal or dual step for the requirement of interior-point iterates.

It is noted that some warmstarting interior-point methods for linear programming have also focused on relaxing the primal and dual interior-point limitations. For example, Benson and Shanno [1] presented a primal-dual penalty method for linear programming, in which primal and dual auxiliary variables were introduced to relax the primal and dual nonnegativity constraints, and these auxiliary variables were penalized with some additional penalty parameters in the objective functions of primal and dual problems. The primal-dual penalty approach was then used for nonlinear programming in [2]. Following a similar idea, Engau et al. [13] considered a simpler scheme in warmstarting interior-point methods for linear programming which only used a set of slack variables to relax the nonnegativity constraints of the original primal-dual variables. Numerical results in [2,13] have shown that the approaches could improve the performance of interior-point methods for linear and nonlinear programming. Our method has some similarity to [13] in dealing with the nonnegativity constraints (see (2.11) of the logarithmic barrier positive relaxation problem), but we use a set of slack functions of the original primal-dual variables which are always positive before finding the solution. Moreover, it is different from warmstarting interior-point methods which attempt to improve the performance of interior-point methods near the KKT solution. Our method is proved to be globally convergent and applicable to solve not only the problem with “jamming” but also the problem for which the KKT conditions do not hold at the solution.

Our method can be thought of as a subsequent work of [12]. However, since our method here is proposed without using the augmented Lagrangian function, it has more flexibility to the selection of penalty parameter. Moreover, our method in this paper does not require any primal or dual iterate to be an interior point (which is essentially quite different from the method in [12] where the primal and dual iterates are always interior-points), and solves general optimization problems with inequality and equality constraints. It is known that the Lagrange multipliers are dependent on the scaling of constraints. Thus, a scaling parameter is incorporated into the method. The incorporation of this parameter makes our method more robust for some degenerate and difficult problems. The barrier and scaling parameters are updated adaptively dependent on the convergence of the inner algorithm of our method. Without assuming any regularity condition, it is proved that our method will terminate at an approximate KKT point of the original problem when the barrier parameter becomes small enough and the scaling parameter is not close to zero. If the scaling parameter becomes small enough, then either an approximate infeasible stationary point or an approximate singular stationary point of the original problem will be found. Some preliminary numerical results are reported, including the results for a well-posed problem for which many line-search interior-point methods were demonstrated not to be globally convergent, a feasible problem for which the LICQ and the MFCQ fail to hold at the solution and an infeasible problem, and for some standard test problems of the CUTE collection. Correspondingly, for comparison we also report the numerical results obtained by the interior-point solver IPOPT. These results show that our algorithm is not only efficient for well-posed feasible problems, but is also applicable for some ill-posed feasible problems and even for some infeasible problems.

This paper is organized as follows. In section 2, we describe a particular logarithmic barrier positive relaxation problem and prove its equivalence to the logarithmic barrier problem (1.4)–(1.6). Our primal-dual interior-point relaxation method for nonlinear program (1.1)–(1.3) is presented in section 3. In section 4, we show the global convergence results of our method. Some preliminary numerical results are reported in section 5. We conclude the paper in section 6.

Throughout the paper, we use standard notation from the literature. A letter with subscript k (l) is related to the kth (lth) iteration, the subscript j (i) indicates the jth (ith) component of a vector, and the subscript kj is the jth component of a vector at the kth iteration. All vectors are column vectors, and $z = (x, u)$ means $z = [x^T,\, u^T]^T$. The expression $\theta_k = O(\tau_k)$ means that there exists a constant M independent of k such that $|\theta_k| \le M|\tau_k|$ for all k large enough, and $\theta_k = o(\tau_k)$ indicates that $|\theta_k| \le \epsilon_k|\tau_k|$ 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 may be showed in the subscript or be clear in the context, and $\|\cdot\|$ is the Euclidean norm.

2. A logarithmic barrier positive relaxation problem

Suppose that $\mu > 0$ and $\tau > 0$ are fixed constants. Let us consider problem (1.4)–(1.6) and its KKT system (1.7)–(1.9). For $x \in \Re^n$, $t \in \Re^m$, and $s \in \Re^m$, define $z \in \Re^m$ and $y \in \Re^m$ by components

$$z_j = (\sqrt{(\tau s_j - t_j)^2 + 4\tau\mu} - (\tau s_j - t_j))/2, \tag{2.1}$$

$$y_j = (\sqrt{(\tau s_j - t_j)^2 + 4\tau\mu} + (\tau s_j - t_j))/2, \tag{2.2}$$

where $j = 1, \dots, m$. That is, both $z : \Re^{2m} \to \Re^m$ and $y : \Re^{2m} \to \Re^m$ are functions on (t, s) and depend on the parameters μ and τ. We have the following simple but important results on z and y.

Lemma 2.1. *For given $\mu > 0$ and $\tau > 0$, let z_j and y_j be defined by* (2.1) *and* (2.2)*. Then*

(1) *$z_j > 0$, $y_j > 0$, $z_j - t_j = y_j - \tau s_j$ and $z_j y_j = \tau\mu$;*
(2) *$t_j > 0$, $s_j > 0$, $t_j s_j = \mu$ if and only if $z_j - t_j = 0$.*

Proof.
(1) z_j and y_j are positive by construction. By (2.1) and (2.2),

$$z_j - t_j = y_j - \tau s_j = (\sqrt{(\tau s_j - t_j)^2 + 4\tau\mu} - (\tau s_j + t_j))/2. \tag{2.3}$$

The result $z_j y_j = \tau\mu$ follows directly from the formula of difference of two squares.

(2) Due to (2.1), $z_j > 0$ for any $t \in \Re^m$ and $s \in \Re^m$. If $z_j - t_j = 0$, then $t_j > 0$ and

$$\sqrt{(\tau s_j - t_j)^2 + 4\tau\mu} = t_j + \tau s_j.$$

The square of the above equation results in $t_js_j = \mu$, which implies $s_j > 0$. Thus, if $z_j - t_j = 0$, one has $t_j > 0$, $s_j > 0$, and $t_js_j = \mu$.

If $t_js_j = \mu$, then $4\tau\mu = 4\tau s_jt_j$ and $\sqrt{(\tau s_j - t_j)^2 + 4\tau\mu} = \sqrt{(\tau s_j + t_j)^2}$. If we further have $t_j > 0$ and $s_j > 0$, there must have $\sqrt{(\tau s_j + t_j)^2} = \tau s_j + t_j$, i.e., $z_j = t_j$. Hence, $z_j - t_j = 0$. □

Lemma 2.2. *For $j = 1, \dots, m$, let z_j and y_j be defined by* (2.1) *and* (2.2), *where $\mu > 0$ and $\tau > 0$ are two scalars.*

(1) *z_j and y_j are differentiable on (t, s), and*

$$\nabla_t z_j = \frac{z_j}{z_j + y_j} e_j, \quad \nabla_t y_j = -\frac{y_j}{z_j + y_j} e_j, \tag{2.4}$$

$$\nabla_s z_j = -\tau \frac{z_j}{z_j + y_j} e_j, \quad \nabla_s y_j = \tau \frac{y_j}{z_j + y_j} e_j, \tag{2.5}$$

where $e_j \in \Re^m$ is the jth coordinate vector.

(2) *z_j is a monotonically increasing function on t_j and is a monotonically decreasing function on s_j.*

(3) *y_j is a monotonically decreasing function on t_j and is a monotonically increasing function on s_j.*

(4) *Both z_j and y_j are decreasing functions of μ as μ decreases.*

Proof.

(1) Due to $z_jy_j = \tau\mu$,

$$y_j \nabla_t z_j + z_j \nabla_t y_j = 0. \tag{2.6}$$

Note that $z_j - y_j = t_j - \tau s_j$. By differentiating the equation, we obtain $\nabla_t z_j - \nabla_t y_j = e_j$. Combining this with (2.6), one immediately has (2.4).

We can similarly prove (2.5).

(2) By (1), $\frac{\partial z_j}{\partial t_j} > 0$, and $\frac{\partial z_j}{\partial s_j} < 0$, the results can be derived straightforward.

(3) The results follow immediately since $\frac{\partial y_j}{\partial t_j} < 0$ and $\frac{\partial y_j}{\partial s_j} > 0$.

(4) Due to Lemma 2.1(1), one has

$$y_j \frac{dz_j}{d\mu} + z_j \frac{dy_j}{d\mu} = \tau \text{ and } \frac{dz_j}{d\mu} - \frac{dy_j}{d\mu} = 0.$$

Thus,

$$\frac{dz_j}{d\mu} = \frac{dy_j}{d\mu} = \frac{\tau}{z_j + y_j} > 0, \tag{2.7}$$

which shows that both z_j and y_j are monotonically increasing on μ. That is, both z_j and y_j are decreasing as μ decreases. □

Throughout this article, z and y are given by (2.1) and (2.2). That is, $z := z(t, s; \mu, \tau)$ and $y := y(t, s; \mu, \tau)$. Both z and y are functions on (t, s) and are dependent on parameters μ and τ. The next theorem provides a firm foundation for the development of our method.

Theorem 2.3. *Suppose $\mu > 0$ and $\tau > 0$. Let $((x^*, t^*), (\lambda^*, s^*))$ be a KKT pair of problem* (1.4)–(1.6), *and let $(x^*, t^*, \lambda^*, s^*)$ satisfy the system* (1.7)–(1.9), *where $\lambda^* \in \Re^{m_e}$ and $s^* \in \Re^m$ are, respectively, the Lagrange multipliers of constraints*

(1.5) and (1.6). Then $((x^*,t^*,s^*),(\lambda^*,s^*,s^*))$ is a KKT pair of the following problem:

$$\min_{x,t,s}\ f(x)-\mu\sum_{j=1}^{m}\ln z_j \tag{2.8}$$

$$\text{s.t. } h_i(x)=0,\ i=1,\dots,m_e, \tag{2.9}$$

$$c_j(x)+t_j=0,\ j=1,\dots,m, \tag{2.10}$$

$$z_j-t_j=0,\ j=1,\dots,m. \tag{2.11}$$

That is, $\lambda^*\in\Re^{m_e}$, $s^*\in\Re^m$, and $s^*\in\Re^m$ are the corresponding Lagrange multipliers of constraints (2.9), (2.10), and (2.11), respectively.

Conversely, if $\mu>0$ and $\tau>0$, $((x^*,t^*,s^*),(\lambda^*,\beta^*,\nu^*))$ is a KKT pair of problem (2.8)–(2.11), where $\lambda^*\in\Re^{m_e}$, $\beta^*\in\Re^m$, and $\nu^*\in\Re^m$ are the associated Lagrange multipliers of constraints (2.9), (2.10), and (2.11), then $\beta^*=\nu^*=s^*$ and (x^*,t^*,λ^*,s^*) satisfies the system (1.7)–(1.9). Thus, $((x^*,t^*),(\lambda^*,s^*))$ is a KKT pair of problem (1.4)–(1.6).

Proof. Due to Lemma 2.1(2), the KKT conditions of problem (1.4)–(1.6) can be written as follows:

$$\nabla f(x^*)+\sum_{i=1}^{m_e}\lambda_i^*\nabla h_i(x^*)+\sum_{j=1}^{m}s_j^*\nabla c_j(x^*)=0, \tag{2.12}$$

$$h_i(x^*)=0,\ i=1,\dots,m_e;\quad c_j(x^*)+t_j^*=0,\ j=1,\dots,m, \tag{2.13}$$

$$z_j^*-t_j^*=0,\ j=1,\dots,m, \tag{2.14}$$

where $z^*=z(t^*,s^*;\mu,\tau)$.

Using Lemma 2.2(1), we can derive the following KKT conditions of problem (2.8)–(2.11):

$$\nabla f(x^*)+\sum_{i=1}^{m_e}\lambda_i^*\nabla h_i(x^*)+\sum_{j=1}^{m}\beta_j^*\nabla c_j(x^*)=0, \tag{2.15}$$

$$\beta^*-\sum_{j=1}^{m}\frac{\mu+y_j^*\nu_j^*}{z_j^*+y_j^*}e_j=0, \tag{2.16}$$

$$\sum_{j=1}^{m}\tau\frac{\mu-z_j^*\nu_j^*}{z_j^*+y_j^*}e_j=0, \tag{2.17}$$

$$h_i(x^*)=0,\ i=1,\dots,m_e, \tag{2.18}$$

$$c_j(x^*)+t_j^*=0,\ j=1,\dots,m, \tag{2.19}$$

$$z_j^*-t_j^*=0,\ j=1,\dots,m, \tag{2.20}$$

where $z^*=z(t^*,s^*;\mu,\tau)$ and $y^*=y(t^*,s^*;\mu,\tau)$, $\lambda_i^*(i=1,\dots,m_e)$, $\beta_j^*(j=1,\dots,m)$, and $\nu_j^*(j=1,\dots,m)$ are the Lagrange multipliers of constraints (2.9), (2.10), and (2.11), respectively.

If $((x^*,t^*),(\lambda^*,s^*))$ is a KKT pair of problem (1.4)–(1.6), conditions (2.12)–(2.14) hold. Thus, by Lemma 2.1(1), $y_j^*-\tau s_j^*=0$ for $j=1,\dots,m$. Since (2.15) is satisfied when $\beta^*=s^*$ due to (2.12), in order to prove the result, we only need to show (2.16) and (2.17) when $\beta^*=s^*$ and $\nu^*=s^*$. Since $z_j^*y_j^*=\tau\mu$, one has

$z_j^* s_j^* = \mu$. Hence,

$$s_j^* - \frac{\mu + y_j^* s_j^*}{z_j^* + y_j^*} = \frac{z_j^* s_j^* - \mu}{z_j^* + y_j^*} = 0,$$

which imply that both (2.16) and (2.17) hold for $\beta^* = s^*$ and $\nu^* = s^*$.

Now we prove the converse result. If $((x^*, t^*, s^*), (\lambda^*, \beta^*, \nu^*))$ is a KKT pair of problem (2.8)–(2.11), then (2.15)–(2.20) hold. For $j = 1, \dots, m$, $z_j^* \nu_j^* = \mu$ and $t_j^* s_j^* = \mu$ due to (2.17) and (2.20) (due to Lemma 2.1(2)). Thus, $\nu^* = s^*$. By (2.16), $\beta^* = \nu^*$. Therefore, we obtain (2.12)–(2.14) directly due to (2.15) and (2.18)–(2.20). □

Since z is a vector function on (t, s), problem (2.8)–(2.11) is a nonlinear programming problem on (x, t, s) with equality constraints. Although the logarithmic barrier positive relaxation problem (2.8)–(2.11) has a similar form to the classic logarithmic barrier problem (1.4)–(1.6), it is essentially distinguished from problem (1.4)–(1.6) in that the relaxation problem does not require t and s to be positive. This important characteristic makes us free from having to truncate the primal and dual steps for the requirement of interior-point iterates, a technique generally used by many existing interior-point methods in the literature.

3. A primal-dual interior-point relaxation algorithm

Our algorithm consists of an inner algorithm and an outer algorithm. In the inner algorithm, we attempt to find an approximate KKT point of the logarithmic barrier positive relaxation problem (2.8)–(2.11) for any given $\mu > 0$ and $\tau > 0$. In the outer algorithm, we update parameters μ and τ according to the information of the solution derived from the inner algorithm.

For the remainder of the paper, we denote $v := (x, t, s) \in \Re^{n+2m}$ and functions $F : \Re^{n+2m} \to \Re$, $C : \Re^{n+2m} \to \Re^{m_e+2m}$,

$$F(v) := f(x) - \mu \sum_{j=1}^{m} \ln z_j, \;\; C(v) := (h(x), c(x) + t, z - t),$$

where we ignore the parameters μ and τ for simplicity of statement, and z is a vector function on (t, s). Then, due to Lemma 2.2,

$$\nabla F(v) = \left(\nabla f(x), -\sum_{j=1}^{m} \frac{\mu}{z_j + y_j} e_j, \sum_{j=1}^{m} \frac{\tau\mu}{z_j + y_j} e_j \right),$$

$$\nabla C(v) = \begin{pmatrix} \nabla h(x) & \nabla c(x) & 0 \\ 0 & I_m & -(Z+Y)^{-1} Y \\ 0 & 0 & -\tau (Z+Y)^{-1} Z \end{pmatrix},$$

where $e_j \in \Re^m$ is the jth coordinate vector in $\Re^m$, I_m is the order-m identity matrix, $Z = \mathrm{diag}(z)$, and $Y = \mathrm{diag}(y)$.

Let

$$L(v, w) = f(x) - \mu \sum_{j=1}^{m} \ln z_j + \sum_{i=1}^{m_e} \lambda_i h_i(x) + \sum_{j=1}^{m} \beta_j (c_j(x) + t_j) + \sum_{j=1}^{m} \nu_j (z_j - t_j)$$

be the Lagrange function of problem (2.8)–(2.11), where $w = (\lambda, \beta, \nu) \in \Re^{m_e+2m}$. By Lemmas 2.1 and 2.2, its Hessian with respect to v has the form

$$\nabla^2_{vv} L(v, w) = \begin{pmatrix} H(x, \lambda, \beta) & 0 \\ 0 & G(v, w) \end{pmatrix},$$

where $H(x, \lambda, \beta) = \nabla^2 f(x) + \sum_{i=1}^{m_e} \lambda_i \nabla^2 h_i(x) + \sum_{j=1}^{m} \beta_j \nabla^2 c_j(x)$ and

$$G(v, w) = \begin{pmatrix} \mu \sum_{j=1}^{m} \frac{(\tau\nu_j + z_j) + (\tau\nu_j - y_j)}{(z_j + y_j)^3} e_j e_j^T & -\tau\mu \sum_{j=1}^{m} \frac{(\tau\nu_j + z_j) + (\tau\nu_j - y_j)}{(z_j + y_j)^3} e_j e_j^T \\ -\tau\mu \sum_{j=1}^{m} \frac{(\tau\nu_j + z_j) + (\tau\nu_j - y_j)}{(z_j + y_j)^3} e_j e_j^T & \tau^2\mu \sum_{j=1}^{m} \frac{(\tau\nu_j + z_j) + (\tau\nu_j - y_j)}{(z_j + y_j)^3} e_j e_j^T \end{pmatrix}.$$

Note that, by Theorem 2.3 and Lemma 2.1, if v^* is a KKT point of problem (2.8)–(2.11), one has $\nu^* = s^* = y^*/\tau$. Thus, by taking $\nu = y/\tau$ in w, we will derive an approximation matrix to $\nabla^2_{vv} L(v, w)$ with the reduced form

$$\begin{pmatrix} H(x, \lambda, \beta) & 0 & 0 \\ 0 & \sum_{j=1}^{m} \frac{\mu}{(z_j + y_j)^2} e_j e_j^T & -\sum_{j=1}^{m} \frac{\tau\mu}{(z_j + y_j)^2} e_j e_j^T \\ 0 & -\sum_{j=1}^{m} \frac{\tau\mu}{(z_j + y_j)^2} e_j e_j^T & \sum_{j=1}^{m} \frac{\tau^2\mu}{(z_j + y_j)^2} e_j e_j^T \end{pmatrix}. \tag{3.1}$$

3.1. **The subproblems for search direction.** Suppose that $v_k := (x_k, t_k, s_k)$ is the current iterate and that $w_k := (\lambda_k, \beta_k, \nu_k)$ is the corresponding estimate vector of the Lagrange multipliers. The classic SQP approach for problem (2.8)–(2.11) solves the quadratic programming (QP) subproblem

$$\min_{d} \quad \nabla F(v_k)^T d + \frac{1}{2} d^T Q(v_k, w_k) d \tag{3.2}$$

$$\text{s.t.} \quad C(v_k) + \nabla C(v_k)^T d = 0, \tag{3.3}$$

where $Q(v_k, w_k)$ is some positive definite approximation to the Hessian $\nabla^2_{vv} L(v_k, w_k)$. In order to overcome the possible inconsistency of the linearized constraints, we introduce the well-behaved null-space technology to the constraint system (for example, see [5, 7] for trust-region methods and [18, 19] for line-search methods). Moreover, $Q(v_k, w_k)$ is selected to have the same form as $\nabla^2_{vv} L(v_k, w_k)$ in (3.1), where $H(x_k, \lambda_k, \beta_k)$ is replaced by a positive definite approximation $B_k \in \Re^{n\times n}$ to $H(x_k, \lambda_k, \beta_k)$. Thus, $Q(v_k, w_k)$ is an approximation to $\nabla^2_{vv} L_(v_k, w_k)$ and

$$Q(v_k, w_k) = \begin{pmatrix} B_k & 0 & 0 \\ 0 & \sum_{j=1}^{m} \frac{\mu}{(z_{kj} + y_{kj})^2} e_j e_j^T & -\sum_{j=1}^{m} \frac{\tau\mu}{(z_{kj} + y_{kj})^2} e_j e_j^T \\ 0 & -\sum_{j=1}^{m} \frac{\tau\mu}{(z_{kj} + y_{kj})^2} e_j e_j^T & \sum_{j=1}^{m} \frac{\tau^2\mu}{(z_{kj} + y_{kj})^2} e_j e_j^T \end{pmatrix}.$$

Motivated by the above arguments, we first approximately solve the subproblem

$$\min_{p} \quad q_k^N(p; \rho) := \frac{1}{2} \rho p^T Q(v_k, w_k) p + \|C(v_k) + \nabla C(v_k)^T p\| \tag{3.4}$$

$$\text{s.t.} \quad \|Rp\| \le \xi \|R^{-1} \nabla C(v_k) C(v_k)\|, \tag{3.5}$$

where $\xi > 1$ is a constant, $\rho > 0$ is a penalty parameter which will be given in the next subsection, and $R = \operatorname{diag}(1, \ldots, 1, \tau, \ldots, \tau) \in \Re^{(n+2m)\times(n+2m)}$ with 1's of number $(n+m)$ and τ's of number m, respectively. Let $p_k \in \Re^{n+2m}$ be the solution. Then our search direction d_k is generated by the null-space QP subproblem

$$\min_d \quad q_k(d) := \nabla F(v_k)^T d + \frac{1}{2} d^T Q(v_k, w_k) d \tag{3.6}$$

$$\text{s.t.} \quad \nabla C(v_k)^T (d - p_k) = 0. \tag{3.7}$$

Apparently, $q_k(d_k) \le q_k(p_k)$ since p_k is a feasible solution of the subproblem (3.6)–(3.7). In particular, if $C(v_k) = 0$, $p_k = 0$ and $q_k(d_k) \le 0$.

The following result paves the way for the selection of ρ and is necessary for proving Lemma 3.3.

Lemma 3.1. *Assume $\nabla C(v_k) C(v_k) \neq 0$. Let p_k be a solution of subproblem* (3.4)–(3.5)*. If $Q(v_k, w_k)$ is positive semidefinite, then*

$$\begin{aligned} \|C(v_k)\| - q_k^N(p_k; \rho) \ge{}& \tfrac{1}{2} \min\{1, \eta_k\} \tfrac{\|R^{-1} \nabla C(v_k) C(v_k)\|^2}{\|C(v_k)\|^2} \\ &\cdot \left\{ \|C(v_k)\| - \rho \|C(v_k)\|^2 \tfrac{C(v_k)^T \nabla C(v_k)^T R^{-2} Q(v_k, w_k) R^{-2} \nabla C(v_k) C(v_k)}{\|R^{-1} \nabla C(v_k) C(v_k)\|^2} \right\}, \end{aligned} \tag{3.8}$$

where $\eta_k = \|R^{-1} \nabla C(v_k) C(v_k)\|^2 / \|\nabla C(v_k)^T R^{-2} \nabla C(v_k) C(v_k)\|^2$. Thus,

$$\|C(v_k)\| - q_k^N(p_k; \rho) \ge \frac{1}{4} \min\{1, \eta_k\} \frac{\|R^{-1} \nabla C(v_k) C(v_k)\|^2}{\|C(v_k)\|}, \tag{3.9}$$

provided $\rho \|C(v_k)\| \le 1/(2\lambda_{\max}(R^{-1} Q(v_k, w_k) R^{-1}))$, where $\lambda_{\max}(R^{-1} Q(v_k, w_k) R^{-1})$ is the maximal eigenvalue of $R^{-1} Q(v_k, w_k) R^{-1}$.

Proof. Let $p_k^C = -\min(1, \eta_k) R^{-2} \nabla C(v_k) C(v_k)$, which is a Cauchy point of subproblem (3.4)–(3.5), that is, a point for minimizing $\|C(v_k) + \nabla C(v_k)^T p\|$ along the Cauchy direction on the region (3.5). Since p_k^C is feasible to the subproblem (3.4)–(3.5),

$$q_k^N(p_k; \rho) \le q_k^N(p_k^C; \rho).$$

Thus, $\|C(v_k)\| - q_k^N(p_k; \rho) \ge \|C(v_k)\| - q_k^N(p_k^C; \rho)$. Due to the positive semidefiniteness of $Q(v_k, w_k)$,

$$\begin{aligned} &\|C(v_k)\| - q_k^N(p_k; \rho) \\ &\ge \|C(v_k)\| - \|C(v_k) + \nabla C(v_k)^T p_k^C\| \\ &\quad - \frac{1}{2} \rho \min\{1, \eta_k\} C(v_k)^T \nabla C(v_k)^T R^{-2} Q(v_k, w_k) R^{-2} \nabla C(v_k) C(v_k). \end{aligned}$$

By Lemma 2.1 of Liu and Yuan [20],

$$\|C(v_k)\| - \|C(v_k) + \nabla C(v_k)^T p_k^C\| \ge \frac{1}{2} \min\{1, \eta_k\} \|R^{-1} \nabla C(v_k) C(v_k)\|^2 / \|C(v_k)\|.$$

Hence, the result (3.8) follows immediately from the above two inequalities.

If $\rho \|C(v_k)\| \le 1/(2\lambda_{\max}(R^{-1} Q(v_k, w_k) R^{-1}))$, then

$$\rho \|C(v_k)\|^2 \frac{C(v_k)^T \nabla C(v_k)^T R^{-2} Q(v_k, w_k) R^{-2} \nabla C(v_k) C(v_k)}{\|R^{-1} \nabla C(v_k) C(v_k)\|^2} \le \frac{1}{2} \|C(v_k)\|.$$

Thus, (3.9) is derived due to (3.8). ∎

3.2. **The merit function.** The merit function plays an important role in prompting the global convergence of the algorithm. For the logarithmic barrier positive relaxation problem (2.8)–(2.11), we introduce the merit function

$$\phi(v;\rho) = \rho F(v) + \|C(v)\| \equiv \rho(f(x) - \mu\sum_{j=1}^{m}\ln z_j) + \|(h(x), c(x)+t, z-t)\|,$$

where $\rho > 0$ is a penalty parameter, and parameters μ and τ are ignored for simplicity of the statement. It is an exact penalty function dependent not only on primal variables $x \in \Re^n$ and $t \in \Re^m$ but also on dual variables $s \in \Re^m$, thus it is essentially different from those merit functions on only primal variables used in some existing primal-dual interior-point methods such as [10,18,20]. It is noted that some methods based on augmented Lagrangian have introduced merit functions with primal and dual variables; for example, see [12, 14, 15, 29]. However, they generated their search directions by different systems and subproblems and shared different motivations with our algorithm.

We first describe a basic property of the merit function.

Lemma 3.2. *Given parameters $\mu > 0$ and $\tau > 0$, for any $\rho \geq 0$, $v \in \Re^{n+2m}$, $d \in \Re^{n+2m}$, and $d \neq 0$, the directional derivative $\phi'(v,d;\rho)$ of function $\phi(v;\rho)$ at v along d exists, and*

$$\phi'(v,d;\rho) \leq \pi(v,d;\rho), \tag{3.10}$$

where $\pi(v,d;\rho) = \rho\nabla F(v)^T d + \chi(v,d)$ and $\chi(v,d) = \|C(v) + \nabla C(v)^T d\| - \|C(v)\|$.

Proof. If $C(v) \neq 0$, $\phi(v;\rho)$ is differentiable at v. Thus, $\phi'(v,d;\rho)$ exists for any $d \neq 0$ whenever $C(v) \neq 0$. For v such that $C(v) = 0$, using the definition of the directional derivative (for example, see (A.14) of [22]), one has $\phi'(v,d;\rho) = \rho\nabla F(v)^T d + \|\nabla C(v)^T d\|$. Therefore, $\phi'(v,d;\rho)$ exists for every $v \in \Re^{n+2m}$, $d \in \Re^{n+2m}$, and $d \neq 0$.

Since

$$\begin{aligned}
\phi'(v,d;\rho) &= \lim_{\alpha\downarrow 0}\frac{\phi(v+\alpha d;\rho) - \phi(v;\rho)}{\alpha} \\
&= \rho\nabla F(v)^T d + \lim_{\alpha\downarrow 0}\frac{\|C(v) + \alpha\nabla C(v)^T d + o(\alpha)\| - \|C(v)\|}{\alpha} \\
&\leq \rho\nabla F(v)^T d + \|C(v) + \nabla C(v)^T d\| - \|C(v)\| + \lim_{\alpha\downarrow 0}\frac{\|o(\alpha)\|}{\alpha} \\
&= \rho\nabla F(v)^T d + \|C(v) + \nabla C(v)^T d\| - \|C(v)\|,
\end{aligned}$$

(3.10) follows immediately. ∎

The following result shows that d_k generated by subproblem (3.6)–(3.7) can be a descent direction of the merit function $\phi(v;\rho)$, provided ρ is suitably selected.

Lemma 3.3. *For given $\mu > 0$ and $\tau > 0$, suppose that d_k is a solution of the QP subproblem* (3.6)–(3.7) *at v_k. Let $\phi_k'(d_k;\rho) = \phi'(v_k,d_k;\rho)$, and let $\delta \in (0,1)$ be a constant.*

(1) *If $\nabla C(v_k)C(v_k) \neq 0$, $\hat\rho_k > 0$ is selected such that*

$$\hat\rho_k\|C(v_k)\| \leq 1/(2\lambda_{\max}(R^{-1}Q(v_k,w_k)R^{-1}))$$

and $\hat{\rho}_k q_k(p_k) + \delta(q_k^N(p_k;\hat{\rho}_k) - \|C(v_k)\|) \le 0$*, then, for all* $0 < \rho \le \hat{\rho}_k$*, it holds that*

$$\phi_k'(d_k;\rho) \le -\tfrac{1}{4}(1-\delta)\min\{1,\eta_k\}\tfrac{\|R^{-1}\nabla C(v_k)C(v_k)\|^2}{\|C(v_k)\|} - \tfrac{1}{2}\rho d_k^T Q(v_k,w_k)d_k. \tag{3.11}$$

(2) $\phi_k'(d_k;\rho) \le -\frac{1}{2}\rho d_k^T Q(v_k,w_k)d_k$ *for any* $\rho > 0$ *whenever* $\|C(v_k)\| = 0$.

Proof. Due to Lemma 3.2, $\phi_k'(d_k;\rho) \le \rho\nabla F(v_k)^T d_k + \chi(v_k,d_k)$. Thus,

$$\phi_k'(d_k;\rho) \le \rho\nabla F(v_k)^T p_k - \tfrac{1}{2}\rho d_k^T Q(v_k,w_k)d_k + q_k^N(p_k;\rho) - \|C(v_k)\| \tag{3.12}$$

since $\nabla F(v_k)^T d_k + \frac{1}{2}d_k^T Q(v_k,w_k)d_k \le \nabla F(v_k)^T p_k + \frac{1}{2}p_k^T Q(v_k,w_k)p_k$. If $\|C(v_k)\| = 0$, $p_k = 0$ and $q_k^N(p_k;\rho) = 0$. Thus, $\phi_k'(d_k;\rho) \le -\frac{1}{2}\rho d_k^T Q(v_k,w_k)d_k$. That is, the result (2) is obtained. In order to prove (1), note that

$$\begin{aligned}&\pi(v_k,d_k;\rho) - (1-\delta)(q_k^N(p_k;\rho) - \|C(v_k)\|) + \frac{1}{2}\rho d_k^T Q(v_k,w_k)d_k\\ &= \rho q_k(d_k) + \delta(q_k^N(p_k;\rho) - \|C(v_k)\|) - \frac{1}{2}\rho p_k^T Q(v_k,w_k)p_k\end{aligned}$$

and $p_k^T Q(v_k,w_k)p_k \ge 0$. If $\rho\|C(v_k)\| \le 1/(2\lambda_{\max}(R^{-1}Q(v_k,w_k)R^{-1}))$, by Lemma 3.1, one has $q_k^N(p_k;\rho) - \|C(v_k)\| < 0$. Furthermore, if

$$\rho q_k(p_k) + \delta(q_k^N(p_k;\rho) - \|C(v_k)\|) \le 0,$$

then

$$\pi(v_k,d_k;\rho) \le (1-\delta)(q_k^N(p_k;\rho) - \|C(v_k)\|) - \frac{1}{2}\rho d_k^T Q(v_k,w_k)d_k.$$

By Lemma 3.2, (3.11) follows immediately. □

The next lemma shows us that we can maintain the vector inequality $z_{k+1} \ge t_{k+1}$ over all k if only we appropriately update s_{k+1}. The inequality plays an important role in proving the global convergence of our algorithm.

Lemma 3.4. *Given* $\mu > 0$ *and* $\tau > 0$*, for any* $j = 1,\dots,m$*,* $z_{k+1,j} \ge t_{k+1,j}$ *if either* $t_{k+1,j} \le 0$*, or* $t_{k+1,j} > 0$ *and* $s_{k+1,j} \le \frac{\mu}{t_{k+1,j}}$*, where* $z_{k+1,j} = z_j(t_{k+1},s_{k+1};\mu,\tau)$ *is given by* (2.2).

Proof. If $t_{k+1,j} \le 0$, then $z_{k+1,j} - t_{k+1,j} > 0$ since $z_{k+1,j} > 0$. Now we consider the cases $t_{k+1,j} > 0$ and $s_{k+1,j} \le \frac{\mu}{t_{k+1,j}}$. We prove the result by contradiction. Suppose $z_{k+1,j} < t_{k+1,j}$. Then, due to (2.2),

$$\sqrt{(\tau s_{k+1,j} - t_{k+1,j})^2 + 4\tau\mu} < \tau s_{k+1,j} + t_{k+1,j}.$$

By taking the square on both sides of the above inequality, one has $t_{k+1,j}s_{k+1,j} > \mu$, which contradicts $s_{k+1,j} \le \frac{\mu}{t_{k+1,j}}$ since $t_{k+1,j} > 0$. The contradiction shows that the desired result is obtained. □

3.3. **Our algorithm.** Similar to the existing primal-dual interior-point methods, our algorithm consists of an inner algorithm and an outer algorithm, where the inner algorithm terminates to satisfy the prescribed conditions in a finite number of iterations, while the outer algorithm updates the parameters by the terminating information of the inner algorithm.

For convenience of statement, we denote

$$r(v_k,\lambda_k;\mu,\tau)=\begin{pmatrix}\nabla f(x_k)+\nabla h(x_k)\lambda_k+\nabla c(x_k)s_k\\ C(v_k)\end{pmatrix},$$

$$g(v_k;\mu,\tau)=\frac{1}{\|C(v_k)\|}\begin{pmatrix}\nabla h(x_k)h(x_k)+\nabla c(x_k)(z_k-t_k)\\ c(x_k)+t_k-(z_k-t_k)\\ Z_k(z_k-t_k)\end{pmatrix},$$

where $v_k=(x_k,t_k,s_k)$, $z_k=z(t_k,s_k;\mu,\tau)$, $C(v_k)=(h(x_k),c(x_k)+t_k,z_k-t_k)$, $Z_k=\operatorname{diag}(z_k)$. By Theorem 2.3, $r(v_k,\lambda_k;\mu,\tau)$ is a vector of residuals of the KKT conditions of problem (1.4)–(1.6). Meanwhile, $g(v_k;\mu,\tau)$ is used to measure the regularity of constraints.

The following proposition provides the terminating conditions for the inner algorithm of our algorithm. Suppose that the inner algorithm generates an infinite sequence $\{v_k\}$ satisfying those very general assumptions in Assumption 4.1; if $\liminf_{k\to\infty}R^{-1}\nabla C(v_k)C(v_k)/\|C(v_k)\|>0$, then any limit point of the sequence will be a KKT point of problem (2.8)–(2.11) (see Theorem 4.8), which will trigger case (1) of the proposition. Otherwise, the case (2) of the proposition will be met.

Proposition 3.5. *For any given $\mu>0$ and $\tau>0$, let $\{v_k\}$ be an infinite sequence with $v_k=(x_k,t_k,s_k)\in\Re^n\times\Re^m\times\Re^m$, and let $\epsilon>0$ be any given small scalar.*

(1) *If $\{v_k\}$ is convergent and its limit point is a KKT point of problem (2.8)–(2.11), then there exists a $\lambda_k\in\Re^{m_e}$ such that $\|r(v_k,\lambda_k;\mu,\tau)\|_\infty\le\epsilon$ for every sufficiently large k.*

(2) *If $\{x_k\}$, $\{t_k\}$, and $\{z_k\}$ are bounded, and*

$$\lim_{k\to\infty}R^{-1}\nabla C(v_k)C(v_k)/\|C(v_k)\|=0, \tag{3.13}$$

then $\|g(v_k;\mu,\tau)\|_\infty\le\epsilon$ for every sufficiently large k.

Proof.

(1) Without loss of generality, suppose that $v^*=(x^*,t^*,s^*)$ is the limit point of $\{v_k\}$ which is a KKT point of problem (2.8)–(2.11), and $w^*=(\lambda^*,\beta^*,\nu^*)\in\Re^{m_e}\times\Re^m\times\Re^m$ is the corresponding Lagrange multiplier. By the second part of Theorem 2.3, $\beta^*=\nu^*=s^*$, and (x^*,t^*,λ^*,s^*) satisfies (1.7)–(1.9). Hence, $\|r(v^*,\lambda^*;\mu,\tau)\|_\infty=0$. Then the result follows immediately from the continuity of $\nabla f(x)$, $\nabla h(x)$, $\nabla c(x)$, and $C(v)$.

(2) Due to (3.13), one has

$$\lim_{k\to\infty}\nabla h(x_k)h(x_k)/\|C(v_k)\|+\nabla c(x_k)(c(x_k)+t_k)/\|C(v_k)\|=0, \tag{3.14}$$

$$\lim_{k\to\infty}(c(x_k)+t_k)/\|C(v_k)\|-(Z_k+Y_k)^{-1}Y_k(z_k-t_k)/\|C(v_k)\|=0, \tag{3.15}$$

$$\lim_{k\to\infty}-(Z_k+Y_k)^{-1}Z_k(z_k-t_k)/\|C(v_k)\|=0. \tag{3.16}$$

By (3.15) and (3.16), $\lim_{k\to\infty}(c(x_k)+t_k)/\|C(v_k)\|-(z_k-t_k)/\|C(v_k)\|=0$. Thus, (3.14) implies

$$\lim_{k\to\infty}\frac{1}{\|C(v_k)\|}\left[\nabla h(x_k)h(x_k)+\nabla c(x_k)(z_k-t_k)\right]=0.$$

In order to obtain the result, we need to prove the limit

$$\lim_{k\to\infty}Z_k(z_k-t_k)/\|C(v_k)\|=0. \tag{3.17}$$

Note that, by the definition of z_k, if $\{x_k\}$, $\{t_k\}$, and $\{z_k\}$ are bounded, then $s_{kj} \not\to -\infty$ for every $j = 1, \dots, m$. If $\{s_k\}$ is bounded, then, by the definition of y_k, y_k is bounded. Thus, $(z_k + y_k)$ is bounded, $(Z_k + Y_k)^{-1}$ does not converge to zero, and (3.17) immediately follows from (3.16). Now we suppose $s_{kj} \to \infty$ for some $j = 1, \dots, m$ as $k \to \infty$, which can induce $y_{kj} \to \infty$ and $z_{kj} \to 0$ as $k \to \infty$. Finally, one has $z_{kj}(z_{kj} - t_{kj})/\|C(v_k)\| \to 0$ as $k \to \infty$ since $(z_k - t_k)/\|C(v_k)\|$ is bounded. Hence, (3.17) always holds and the desired result is proved. $\square$

Now we are ready to present our primal-dual interior-point relaxation algorithm for problem (1.1)–(1.3).

Algorithm 3.6 A primal-dual interior-point relaxation algorithm

Step 1 Given $(x_0, t_0, s_0) \in \Re^n \times \Re^m \times \Re^m$, $B_0 \in \Re^{n\times n}$, $\mu_0 > 0$, $\tau_0 > 0$, $\rho_0 > 0$, $\delta \in (0, 1)$, $\sigma \in (0, \frac{1}{2})$, $\epsilon > 0$. Set $l := 0$.

Step 2 While $\mu_l > \epsilon$ and $\tau_l > \epsilon$, start the inner algorithm. Otherwise, stop the algorithm.

Step 2.0 Let $(x_0, t_0, s_0) = (x_l, t_l, s_l)$, $B_0 = B_l$, $\rho_0 = \rho_l$, $\mu = \mu_l$, and $\tau = \tau_l$. Evaluate z_0 and y_0 by (2.2). Set $k := 0$.

Step 2.1 Find first an approximate solution p_k of subproblem (3.4)–(3.5) such that (3.8) holds, then solve the QP subproblem (3.6)–(3.7) to derive (d_{xk}, d_{tk}, d_{sk}).

Step 2.2 Choose ρ_{k+1} with either $\rho_{k+1} = \rho_k$ or $\rho_{k+1} \le 0.5\rho_k$ such that (3.11) holds.

Step 2.3 Choose the step-size $\alpha_k \in (0, 1]$ to be the maximal in $\{1, \delta, \delta^2, \dots\}$ such that

$$\begin{aligned}&\phi(x_k + \alpha_k d_{xk}, t_k + \alpha_k d_{tk}, s_k + \alpha_k d_{sk}; \rho_{k+1}) - \phi(x_k, t_k, s_k; \rho_{k+1})\\ &\le \sigma\alpha_k\pi(v_k, d_k; \rho_{k+1}).\end{aligned} \tag{3.18}$$

Step 2.4 Set $x_{k+1} = x_k + \alpha_k d_{xk}$, $t_{k+1} = t_k + \alpha_k d_{tk}$, and $\hat{s}_{k+1} = s_k + \alpha_k d_{sk}$.

Step 2.5 For $j = 1, \dots, m$, set

$$s_{k+1,j} = \begin{cases} \hat{s}_{k+1,j}, & \text{if} \quad t_{k+1,j} \le 0; \\ \min\{\hat{s}_{k+1,j}, \frac{\mu}{t_{k+1,j}}\}, & \text{otherwise.} \end{cases} \tag{3.19}$$

Step 2.6 Compute an estimate λ_{k+1} of Lagrange multiplier vector corresponding to the equality constraints in (1.2).

Step 2.7 If $\|r(v_{k+1}, \lambda_{k+1}; \mu_l, \tau_l)\|_\infty > 10\mu_l$ and $\|g(v_{k+1}; \mu_l, \tau_l)\|_\infty > \tau_l$,
update B_k to B_{k+1}, set $k := k + 1$ and go to Step 2.1;
else if $\|r(v_{k+1}, \lambda_{k+1}; \mu_l, \tau_l)\|_\infty \le 10\mu_l$
set $\mu_{l+1} = \min\{0.5\mu_l, \|r(v_{k+1}, \lambda_{k+1}; \mu_l, \tau_l)\|_\infty^{1.8}\}$, $\tau_{l+1} = \tau_l$;
else $\|g(v_{k+1}; \mu_l, \tau_l)\|_\infty \le \tau_l$,
set $\mu_{l+1} = \mu_l$, $\tau_{l+1} \le 0.6\tau_l$,
end.
Stop the inner algorithm, set $(x_{l+1}, t_{l+1}, s_{l+1}) = (x_{k+1}, t_{k+1}, s_{k+1})$, $l := l + 1$.

End (while)

In Step 2.2, by virtue of Lemma 3.3, if $\nabla C(v_k)C(v_k) \ne 0$, $\hat{\rho}_k$ can be selected as ρ_{k+1} provided $\hat{\rho}_k \le 0.5\rho_k$. If $\nabla C(v_k)C(v_k) = 0$ and $C(v_k) \ne 0$, then $g(v_k; \mu, \tau) = 0$. If $C(v_k) = 0$ and $Q(v_k, w_k)d_k = 0$, then $p_k = 0$ and there exists a u_k such that $\nabla F(v_k) + \nabla C(v_k)u_k = 0$, which implies that there is a λ_k such that $r(v_k, \lambda_k; \mu, \tau) = 0$. In both cases, the inner algorithm of Algorithm 3.6 will terminate at v_k.

Due to Lemma 3.4, the update (3.19) guarantees $z_{k+1} - t_{k+1} \geq 0$ for all k. Moreover, it will further reduce the value of $\phi(x_k + \alpha_k d_{xk}, t_k + \alpha_k d_{tk}, s_k + \alpha_k d_{sk}; \rho_{k+1})$. Thus, one has

$$\phi(x_{k+1}, t_{k+1}, s_{k+1}; \rho_{k+1}) - \phi(x_k, t_k, s_k; \rho_{k+1}) \leq \sigma \alpha_k \pi(v_k, d_k; \rho_{k+1}) < 0 \tag{3.20}$$

for all $k \geq 0$.

For λ_{k+1} in Step 2.6, theoretically we may take it to be the vector of the multipliers corresponding to the first m_e constraints of subproblem (3.6)–(3.7). Since we do not assume any regularity condition, the multipliers of subproblem (3.6)–(3.7) may not be unique. Thus, in our implementation, we compute $\lambda_{k+1} = \operatorname{argmin}_\lambda \|\nabla f(x_{k+1}) + \nabla c(x_{k+1}) s_{k+1} + \nabla h(x_{k+1}) \lambda\|$.

4. Global convergence

Without requiring any regularity of constraints, problem (1.1)–(1.3) can even be infeasible. If the problem is feasible, its local solution may be either a KKT point or a singular stationary point (see Definition 4.9). For an infeasible problem, we usually want to find an infeasible stationary point which is a stationary point for minimizing some measure of constraint violations (see [4, 6, 21]).

In order to solve the original problem (1.1)–(1.3), Algorithm 3.6 approximately solves a sequence of logarithmic barrier positive relaxation problem (2.8)–(2.11). In this section, we first prove that, for any given $\mu > 0$ and $\tau > 0$, the inner algorithm of Algorithm 3.6 will terminate in a finite number of iterations. Thus, either $\mu_l \to 0$ or $\tau_l \to 0$ as $l \to \infty$. After that, we consider the global convergence of the whole algorithm. It is proved that our algorithm will terminate at an approximate KKT point of the original problem provided the scaling parameter τ_l is far from zero. Otherwise, either an approximate infeasible stationary point or an approximate singular stationary point of the original problem will be found.

4.1. **Global convergence of the inner algorithm.**

We consider the global convergence of the inner algorithm. Suppose that, for some given $\mu > 0$ and $\tau > 0$, the inner algorithm of Algorithm 3.6 does not terminate in a finite number of iterations and $\{(x_k, t_k, s_k)\}$ is an infinite sequence generated by the algorithm. We need the following blanket assumptions for our global convergence analysis.

Assumption 4.1.

(1) The functions f and $c_i (i \in \mathcal{I})$ are twice continuously differentiable on $\Re^n$.
(2) The iterative sequence $\{x_k\}$ is in an open bounded set.
(3) The sequence $\{B_k\}$ is bounded, and for all $k \geq 0$ and $d_x \in \Re^n$, $d_x^T B_k d_x \geq \gamma \|d_x\|^2$, where $\gamma > 0$ is a constant.
(4) For all $k \geq 0$, p_k is an approximate solution of subproblem (3.4)–(3.5) satisfying (3.8).

Parts (1) and (2) of Assumption 4.1 are commonly used in global convergence analysis of algorithms for nonlinear programming. It should be noted that Assumption 4.1(3) is also an assumption extensively used in the literature, and it can be replaced by a weaker assumption that B_k is positive semidefinite on the whole space and is positive definite on the null-space of $\nabla C(v_k)^T$. We use it for simplicity of the statement. Since $p_k = 0$ as $\nabla C(v_k) C(v_k) = 0$, part (4) of Assumption 4.1 only makes sense when $\nabla C(v_k) C(v_k) \neq 0$. Lemma 3.1 shows that part (4) of Assumption 4.1 is not a difficult condition.

We have the following results which are proved similar to Lemma 5 of [7] and Lemma 4.2 of [18].

Lemma 4.2. *Suppose that Assumption* 4.1 *holds. Then* $\{z_k\}$ *and* $\{t_k\}$ *are bounded,* $\{y_k\}$ *is componentwise bounded away from zero, and* $\{s_k\}$ *is componentwise bounded away from* $-\infty$*. Furthermore, if the penalty parameter* ρ_k *is bounded away from zero as* $k \to \infty$*, then* $\{z_k\}$ *is componentwise bounded away from zero, and* $\{y_k\}$ *and* $\{s_k\}$ *are bounded above.*

Proof. Assumption 4.1 implies that there exists a scalar $\chi > 0$ such that $\|f(x_k)\| \leq \chi$ and $\|c(x_k)\| \leq \chi$ for all $k \geq 0$. Due to (3.20), $\phi(v_{k+1}; \rho_{k+1}) \leq \phi(v_k; \rho_{k+1})$ for all $k \geq 0$. Thus, for every $k \geq 0$,

$$\phi(v_{k+1}; \rho_{k+1}) - \phi(v_k; \rho_k) \leq (\rho_k - \rho_{k+1})(\chi + m\mu \ln \|z_k\|).$$

Therefore,

$$\phi(v_{k+1}; \rho_{k+1}) \leq \phi(v_0; \rho_0) + (\rho_0 - \rho_{k+1})(\chi + m\mu \max_{0 \leq l \leq k+1} \ln \|z_l\|). \tag{4.1}$$

Note that by the inequalities

$$\phi(v_{k+1}; \rho_{k+1}) \geq -\rho_{k+1}(\chi + m\mu \max_{0 \leq l \leq k+1} \ln \|z_l\|) + \|t_{k+1}\| - \|c(x_{k+1})\| \tag{4.2}$$

and $\phi(v_{k+1}; \rho_{k+1}) \geq -\rho_{k+1}(\chi + m\mu \max_{0 \leq l \leq k+1} \ln \|z_l\|) + \|z_{k+1}\| - \|t_{k+1}\|$, one has

$$\phi(v_{k+1}; \rho_{k+1}) \geq -\rho_{k+1}(\chi + m\mu \max_{0 \leq l \leq k+1} \ln \|z_l\|) + \|z_{k+1}\| - \|c(x_{k+1})\|. \tag{4.3}$$

Hence, it follows from (4.1), (4.2), and (4.3) that, for all $k \geq 0$,

$$\phi(v_0; \rho_0) + (1 + \rho_0)\chi + \rho_0 m\mu \max_{0 \leq l \leq k+1} \ln \|z_l\|) \geq \max(\|z_{k+1}\|, \|t_{k+1}\|),$$

which implies that $\{z_k\}$ is bounded. Furthermore, $\{t_k\}$ is bounded since $\{z_k\}$ is bounded. Due to Lemma 2.1(1), the results on $\{y_k\}$ follow immediately.

For given $\mu > 0$ and $\tau > 0$, if $\{z_k\}$ is bounded, by (2.2), $s_{kj} > -\infty$ for all $k \geq 0$ and $j = 1, \ldots, m$. Otherwise, if $s_{kj} \to -\infty$ for some j, then $z_{kj} \to \infty$, which is a contradiction. If $\{z_k\}$ is componentwise bounded away from zero, then again by (2.2), $s_{kj} < +\infty$ for all $k \geq 0$ and $j = 1, \ldots, m$. Thus, the results on $\{s_k\}$ are proved. □

The following corollary shows that the sequences of the values of function $C(v)$, the derivatives $\nabla C(v)$ and $\nabla F(v)$, and the approximate Lagrange Hessian $Q(v, w)$ over k are all bounded.

Corollary 4.3. *Suppose that Assumption* 4.1 *holds. Then, for any given* $\mu > 0$ *and* $\tau > 0$*, all sequences* $\{C(v_k)\}$*,* $\{\nabla F(v_k)\}$*,* $\{\nabla C(v_k)\}$*, and* $\{Q(v_k, w_k)\}$ *are bounded. Thus, there is a constant* $\chi_0 > 0$ *such that* $\|C(v_k)\| \leq \chi_0$*,* $\|\nabla C(v_k)\| \leq \chi_0$*, and* $\lambda_{\max}(Q(v_k, w_k)) \leq \chi_0$.

Proof. Due to Lemma 2.1(1),

$$\frac{1}{z_{kj} + y_{kj}} = \frac{z_{kj}}{z_{kj}^2 + z_{kj} y_{kj}} \leq \frac{1}{\tau\mu} z_{kj}.$$

Thus, by Lemma 4.2, $\{1/(z_{kj} + y_{kj})\}$ is bounded. The boundednesses of $\{C(v_k)\}$, $\{\nabla F(v_k)\}$, $\{\nabla C(v_k)\}$, and $\{Q(v_k, w_k)\}$ follow immediately from their expressions in the previous section. □

The next result provides a sufficient condition for keeping the sequence of the penalty parameter bounded away from zero. The condition does not require $\nabla C(v_k)$ to be of full column rank, which is a weaker condition and admits the convergence of our algorithm to a singular stationary point, a candidate of the minimizer of the original nonlinear programming problems.

Lemma 4.4. *Suppose that Assumption* 4.1 *holds. If*

$$\|R^{-1}\nabla C(v_k)C(v_k)\| \geq \chi_1\|C(v_k)\| \tag{4.4}$$

for some constant $\chi_1 > 0$ and for all $k \geq 0$, then there is a constant $\hat{\rho} > 0$ such that $\rho_{k+1} = \hat{\rho}$ for all sufficiently large k.

Proof. Due to Corollary 4.3, $\rho_{k+1}\|C(v_k)\| \leq 1/(2\lambda_{\max}(R^{-1}Q(v_k, w_k))R^{-1})$ provided $\rho_{k+1} \leq \tau^2/(2\chi_0^2)$. Thus, if $\hat{\rho} \leq \tau^2/(2\chi_0^2)$, it follows from (3.9) and (4.4) that

$$\|C(v_k)\| - \|C(v_k) + \nabla C(v_k)^T p_k\| \geq \frac{1}{4}\min\{1, \frac{1}{\chi_0^2}\}\chi_1^2\|C(v_k)\|. \tag{4.5}$$

We achieve the result by proving that (3.11) holds with $\rho_{k+1} \leq \hat{\rho}$ for some scalar $\hat{\rho} > 0$. Since $q_k(d_k) \leq q_k(p_k)$, one has

$$\begin{aligned}
&\pi(v_k, d_k; \rho_{k+1}) - (1-\delta)(q_k^N(p_k; \rho_{k+1}) - \|C(v_k)\|) + \frac{1}{2}\rho_{k+1}d_k^T Q(v_k, w_k)d_k \\
&\leq \rho_{k+1}(\nabla F(v_k)^T p_k + \frac{1}{2}p_k^T Q(v_k, w_k)p_k) + \delta(\|C(v_k) + \nabla C(v_k)^T p_k\| - \|C(v_k)\|) \\
&\leq (\rho_{k+1}\xi_1 - \delta\xi_2)\|C(v_k)\|,
\end{aligned}$$

where ξ_1 and ξ_2 are positive constants. Hence, (3.11) holds with $\rho_{k+1} \leq \hat{\rho}$ provided $\hat{\rho} = \min\{\tau^2/(2\chi_0^2), \delta\xi_2/\xi_1\}$. □

Based on the above result that the penalty parameter is bounded away from zero, we can prove that the sequence of search directions is bounded.

Lemma 4.5. *Suppose that Assumption* 4.1 *holds and that $d_k = (d_{xk}, d_{tk}, d_{sk}) \in \Re^{n+2m}$ is a solution of QP* (3.6)–(3.7)*. If* (4.4) *holds for all sufficiently large k, the sequence $\{\|d_k\|\}$ is bounded.*

Proof. If (4.4) holds for all sufficiently large k, by Lemma 4.4, ρ_k is bounded away from zero. It follows from Lemma 4.2 that z_k and y_k are bounded. Thus, there exists a constant $\chi_2 > 0$ such that

$$\frac{\mu}{(z_{kj} + y_{kj})^2} \geq \chi_2 \quad \text{for} \quad j = 1, \ldots, m.$$

Due to

$$\begin{aligned}
d^T Q(v_k, w_k)d &= d_x^T B_k d_x + \sum_{j=1}^{n} \frac{\mu}{(z_{kj} + y_{kj})^2}(d_{tj} - \tau d_{sj})^2 \\
&\geq \gamma\|d_x\|^2 + \chi_2\|d_t - \tau d_s\|^2 \qquad (4.6)
\end{aligned}$$

for every $d = (d_x, d_t, d_s)$, $d_k^T Q(v_k, w_k)d_k \geq \xi_3\|(d_{xk}, d_{tk} - \tau d_{sk})\|^2$ for some scalar $\xi_3 > 0$.

Since $q_k(d_k) \leq q_k(p_k)$, $q_k(p_k) \leq \chi_3$, and $q_k(d_k) \geq -\chi_3\|(d_{xk}, d_{tk} - \tau d_{sk})\| + \xi_3\|(d_{xk}, d_{tk} - \tau d_{sk})\|^2$ for some scalars $\chi_3 > 0$ and $\xi_3 > 0$, one can deduce that $\|(d_{xk}, d_{tk} - \tau d_{sk})\|$ is bounded. Otherwise, if $\|(d_{xk}, d_{tk} - \tau d_{sk})\|$ is unbounded, then $\xi_3 \le 0$, which is a contradiction. Thus, $\|d_{tk}\|$ is bounded due to $d_{tk} = p_{tk} - \nabla c(x_k)^T(d_{xk} - p_{xk})$. It implies that $\|d_{sk}\|$ is bounded. □

The boundedness of search directions implies that we can always find a step-size not too small for the new iterate under suitable conditions.

Lemma 4.6. *Suppose that Assumption* 4.1 *holds and that* $\{\alpha_k\}$ *is the sequence of step-sizes derived from* (3.18) *of Algorithm* 3.6. *If the inequality* (4.4) *holds for all sufficiently large* k, *then* $\{\alpha_k\}$ *is bounded away from zero.*

Proof. Due to Lemmas 4.2 and 4.5, for every $j = 1, \dots, m$, one has

$$-\ln z_j(v_k + \alpha d_k; \mu, \tau) + \ln z_{kj} - \alpha \tfrac{1}{z_{kj}+y_{kj}} e_j^T(d_{tk} - \tau d_{sk}) = o(\alpha), \tag{4.7}$$

$$\|C(v_k + \alpha d_k)\| = \|C(v_k) + \alpha \nabla C(v_k)^T d_k\| + o(\alpha) \tag{4.8}$$

for all $\alpha > 0$ sufficiently small. Hence,

$$\phi(v_k + \alpha d_k; \rho_{k+1}) - \phi(v_k; \rho_{k+1}) = \alpha \pi(v_k, d_k; \rho_{k+1}) + o(\alpha) \tag{4.9}$$

for all $\alpha \in [0, \tilde{\alpha}]$, where $\tilde{\alpha} > 0$ is a sufficiently small constant. Due to

$$\begin{aligned} (1-\sigma)\alpha\pi(v_k, d_k; \rho_{k+1}) &\le \alpha(1-\sigma)(1-\delta)(q_k^n(p_k; \rho_{k+1}) - \|C(v_k)\|) \\ &\le -\alpha\xi_4\|C(v_k)\| \end{aligned} \tag{4.10}$$

(where $\xi_4 = \xi_2(1-\sigma)(1-\delta)$), it follows from (4.9) and (4.10) that there exists a scalar $\hat{\alpha} \in (0, \tilde{\alpha}]$ such that

$$\phi(v_k + \alpha d_k; \rho_{k+1}) - \phi(v_k; \rho_{k+1}) \le \sigma\alpha\pi(v_k, d_k; \rho_{k+1})$$

for all $\alpha \in (0, \hat{\alpha}]$ and all $k \ge 0$. Thus, by Step 2.3 of Algorithm 3.6, $\alpha_k \ge \delta\hat{\alpha}$ for all $k \ge 0$. □

The following result paves the way for proving our global convergence results. It shows that, under suitable conditions, the sequence $\{v_k\}$ generated by our inner algorithm will converge to a feasible point of problem (2.8)–(2.11) and that the search direction will vanish.

Lemma 4.7. *Suppose that Assumption* 4.1 *holds. If the inequality* (4.4) *holds for all sufficiently large* k, *then*

$$\lim_{k\to\infty} \|C(v_k)\| = 0 \quad \textit{and} \quad \lim_{k\to\infty} \|d_k\| = 0. \tag{4.11}$$

Proof. According to Lemma 4.4, without loss of generality, we suppose that $\rho_k = \hat{\rho}$ for all $k \ge 0$. Then, by (3.20), $\{\phi(v_k; \rho_{k+1})\}$ is a monotonically nonincreasing sequence. Note that it is also a bounded sequence. Thus,

$$\lim_{k\to\infty} \pi(v_k, d_k; \rho_{k+1}) = 0 \tag{4.12}$$

since α_k is bounded away from zero. Using the last inequality of (4.10), one has

$$\lim_{k\to\infty} \|C(v_k)\| = 0,$$

which implies $\lim_{k\to\infty} \|p_k\| = 0$. Thus, by (3.11) and (4.6), $\lim_{k\to\infty} \|d_k\| = 0$. □

Now we are ready to present our global convergence result on the inner algorithm.

Theorem 4.8. *Given* $\mu > 0$ *and* $\tau > 0$, *suppose that Assumption* 4.1 *holds. Let* $\{v_k\}$ *and* $\{\lambda_k\}$ *be two sequences generated by the inner algorithm of Algorithm* 3.6. *Let* $\epsilon > 0$ *be any given small scalar. Then there is an integer* $k > 0$ *such that either* $\|r(v_{k+1}, \lambda_{k+1}; \mu, \tau)\|_\infty \le \epsilon$ *or* $\|g(v_{k+1}; \mu, \tau)\|_\infty \le \epsilon$.

Proof. We prove the result in two cases.

Case 1. Condition (4.4) holds for all $k \geq 0$. By Lemma 4.4, ρ_k remains a positive constant after a finite number of iterations. Thus, $\{z_k\}$ and $\{y_k\}$ are bounded above and componentwise bounded away from zero, and $\{s_k\}$ is bounded.

We prove the result in Case 1 by contradiction. If the result does not hold, then the inner algorithm will not terminate in a finite number of iterations. Thus, $\{v_k\}$ is an infinite and bounded sequence since $\{x_k\}$ is bounded by Assumption 4.1(2), $\{t_k\}$ is bounded due to Lemma 4.2, and $\{s_k\}$ is bounded. Let $v^* = (x^*, t^*, s^*)$ be any limit point of $\{v_k\}$. Without loss of generality, suppose that $\lim_{k\to\infty} v_k = v^*$. Due to Lemma 4.7, $\lim_{k\to\infty} \|C(v_k)\| = 0$ and $\lim_{k\to\infty} \|d_k\| = 0$. Thus,

$$\lim_{k\to\infty} z_k = t^* > 0, \quad \lim_{k\to\infty} y_k = \tau s^* > 0$$

since $z_k - t_k = y_k - \tau s_k$. Moreover, by taking the limit on $k \to \infty$ on both sides of the KKT condition

$$\nabla F(v_k) + Q(v_k, w_k) d_k + \nabla C(v_k) u_k = 0$$

of subproblem (3.6)–(3.7), due to Corollary 4.3, one has

$$\lim_{k\to\infty} (\nabla F(v_k) + \nabla C(v_k) u_k) = 0. \tag{4.13}$$

Note that $\lim_{k\to\infty} \|C(v_k)\| = 0$. Therefore, every limit point of $\{v_k\}$ is a KKT point of (2.8)–(2.11). In view of Proposition 3.5, there exists a $\lambda_{k+1} \in \Re^{m_e}$ such that the inequality $\|r(v_{k+1}, \lambda_{k+1}; \mu, \tau)\|_\infty \leq \epsilon$ holds for every sufficiently large k.

Case 2. Condition (4.4) does not hold for all sufficiently large k. Then there exists some infinite index subset $\mathcal{K}$ such that the condition (4.4) does not hold for all $k \in \mathcal{K}$. That is,

$$\lim_{k\in\mathcal{K}, k\to\infty} \|R^{-1} \nabla C(v_k) C(v_k)\| / \|C(v_k)\| = 0. \tag{4.14}$$

Due to Proposition 3.5, one has $\|g(v_{k+1}; \mu, \tau)\|_\infty \leq \epsilon$ for all sufficiently large $k \in \mathcal{K}$.

The above arguments show that the sequence $\{v_k\}$ cannot be an infinite sequence. This contradiction implies the result of the theorem. □

4.2. **Convergence results of the whole algorithm.** Now we consider the global convergence of the whole algorithm. It is well known that, without assuming any constraint qualification, a local solution of the general nonlinear program can be either a KKT point or a Fritz John point which is not a KKT point of the problem. For those nonlinear programs arising from a practical situation, whether or not they are feasible is not known before solving them. Thus, some robust methods for nonlinear programs not only focus on convergence to KKT points of problems under some assumptions on constraint regularities, but are also concerned about convergence to Fritz John points which are not KKT points but possibly local solutions and infeasible stationary points of problems without assuming any regularity on constraints (for example, see [4–6, 9, 10, 12, 18, 20, 31]).

Definition 4.9. $x^* \in \Re^n$ is called a Fritz John point or a singular stationary point of problem (1.1)–(1.3) if there exist $\lambda^* \in \Re^{m_e}$ and $\beta^* \in \Re^m$ such that

$$\begin{aligned}
&\nabla h(x^*)\lambda^* + \nabla c_i(x^*)\beta^* = 0,\\
&h_i(x^*) = 0,\ i = 1, \ldots, m_e,\\
&\beta_j^* \geq 0,\ c_j(x^*) \leq 0,\ \beta_j^* c_j(x^*) = 0,\ j = 1, \ldots, m.
\end{aligned}$$

Definition 4.10. $x^* \in \Re^n$ is called an infeasible stationary point of problem (1.1)–(1.3) if x^* is an infeasible point and

$$\nabla h(x^*)h(x^*) + \nabla c(x^*)\max(0, c(x^*) = 0.$$

Now we are ready to present our convergence results on the whole algorithm.

Theorem 4.11. *Suppose that Assumption* 4.1 *holds for every given parameter* $\mu_l > 0$ *and* $\tau_l > 0$ *and that sequences* $\{x_l\}$ *and* $\{B_l\}$ *are bounded. Let* $\epsilon > 0$ *be any given small scalar. Then there exists an integer* $\bar{l} > 0$ *such that either* $\mu_{\bar{l}} \le \epsilon$ *or* $\tau_{\bar{l}} \le \epsilon$ *and one of the following statements is true:*

(1) *The parameter* $\tau_{\bar{l}} > \epsilon$ *and the inner algorithm terminates at a point* $v_{\bar{l}+1}$ *where the terminating condition* $\|r(v_{\bar{l}+1}, \lambda_{\bar{l}+1}; \mu_{\bar{l}}, \tau_{\bar{l}})\|_\infty \le 10\epsilon$ *holds. Algorithm* 3.6 *terminates at an approximate KKT point of the original problem* (1.1)–(1.3)*.*

(2) *The parameter* $\tau_{\bar{l}} \le \epsilon$*, and the inner algorithm terminates at a point* $v_{\bar{l}+1}$ *at which the condition* $\|g(v_{\bar{l}+1}; \mu_{\bar{l}}, \tau_{\bar{l}})\|_\infty \le \epsilon$ *is satisfied and* $\|C(v_{\bar{l}+1})\| \le \epsilon$*. Algorithm* 3.6 *terminates at a point* $x_{\bar{l}+1}$ *which is an approximate singular stationary point of the problem* (1.1)–(1.3)*.*

(3) *The parameter* $\tau_{\bar{l}} \le \epsilon$*, the inner algorithm terminates due to* $\|g(v_{\bar{l}+1}; \mu_{\bar{l}}, \tau_{\bar{l}})\|_\infty \le \epsilon$ *and* $\|C(v_{\bar{l}+1})\|$ *is bounded away from zero. Algorithm* 3.6 *terminates at a point* $x_{\bar{l}+1}$ *which is an approximate infeasible stationary point of the problem* (1.1)–(1.3)*.*

Proof. For every $l \ge 0$, Theorem 4.8 shows that, in Algorithm 3.6, either μ_l or τ_l will be reduced. Finally, there must be an integer $\bar{l} > 0$ such that either $\mu_{\bar{l}} \le \epsilon$ or $\tau_{\bar{l}} \le \epsilon$. Furthermore, the argument of Lemma 4.2 shows that $\{z_l\}$ and $\{t_l\}$ are bounded.

The condition $\|r(v_{\bar{l}+1}, \lambda_{\bar{l}+1}; \mu_{\bar{l}}, \tau_{\bar{l}})\|_\infty \le 10\epsilon$ implies that $(x_{\bar{l}+1}, t_{\bar{l}+1})$ is an approximate KKT point of the logarithmic barrier positive relaxation problem (1.4)–(1.6). Comparing it with the KKT conditions of problem (1.1)–(1.3), it can be seen that, as $\mu_{\bar{l}}$ is small enough, $x_{\bar{l}+1}$ is also an approximate KKT point of the original problem (1.1)–(1.3).

If it is other than the above case, then one has $\tau_{\bar{l}} \le \epsilon$ and $\|g(v_{\bar{l}+1}; \mu_{\bar{l}}, \tau_{\bar{l}})\|_\infty \le \epsilon$. That is,

$$\|\nabla h(x_{\bar{l}+1})h(x_{\bar{l}+1}) + \nabla c(x_{\bar{l}+1})(z_{\bar{l}+1} - t_{\bar{l}+1})\|_\infty / \|C(v_{\bar{l}+1})\| \le \epsilon, \tag{4.15}$$

$$\|Z_{\bar{l}+1}(z_{\bar{l}+1} - t_{\bar{l}+1})\|_\infty / \|C(v_{\bar{l}+1})\| \le \epsilon. \tag{4.16}$$

In order to demonstrate that the point $x_{\bar{l}+1}$ satisfying (4.15)–(4.16) is an approximate singular stationary point of the problem (1.1)–(1.3) when $\|C(v_{\bar{l}+1})\|$ is small enough, we take the limit $\epsilon \to 0$ on both sides of (4.15)–(4.16) (in which we substitute l for $\bar{l}$). The limit $\epsilon \to 0$ implies $\tau_l \to 0$. Without loss of generality, suppose $z_{l+1} \to z^*$ and $x_{l+1} \to x^*$, $t_{l+1} \to t^*$ as $\tau_l \to 0$, and $h(x_{l+1})/\|C(v_{l+1})\| \to \lambda^*$, $(z_{l+1} - t_{l+1})/\|C(v_{l+1})\| \to \beta^*$. Then x^*, t^*, z^* satisfy $h(x^*) = 0$, $c(x^*) + t^* = 0$, $z^* - t^* = 0$ due to $\|C(v_{\bar{l}+1})\| \to 0$, and $\beta^* \ge 0$ since $z_{l+1} - x_{l+1} \ge 0$ for all l. Thus, $c(x^*) = -t^* = -z^* \le 0$. Furthermore, it follows from (4.15) and (4.16) that

$$\nabla h(x^*)\lambda^* + \nabla c(x^*)\beta^* = 0,$$
$$\beta_j^* c_j(x^*) = 0.$$

That is, x^* is a singular stationary point of problem (1.1)–(1.3). Therefore, $x_{\bar{l}+1}$ is an approximate singular stationary point of the problem.

If $\|g(v_{\bar{l}+1};\mu_{\bar{l}},\tau_{\bar{l}})\|_\infty \le \epsilon$ but $C(v_{\bar{l}+1})$ is bounded away from zero, then there exist scalars $\sigma_1 > \sigma_2 > 0$ independent of ϵ such that $\sigma_1 \ge \|C(v_{\bar{l}+1})\| \ge \sigma_2$, and

$$\|\nabla h(x_{\bar{l}+1})h(x_{\bar{l}+1}) + \nabla c(x_{\bar{l}+1})(z_{\bar{l}+1} - t_{\bar{l}+1})\|_\infty \le \sigma_1\epsilon, \tag{4.17}$$

$$\|c(x_{\bar{l}+1}) + t_{\bar{l}+1} - (z_{\bar{l}+1} - t_{\bar{l}+1})\|_\infty \le \sigma_1\epsilon, \tag{4.18}$$

$$\|Z_{\bar{l}+1}(z_{\bar{l}+1} - t_{\bar{l}+1})\|_\infty \le \sigma_1\epsilon. \tag{4.19}$$

Similar to the preceding arguments, we consider the above inequalities (4.17)–(4.19) by substituting l for $\bar{l}$. Suppose $z_{l+1} \to z^*$ and $x_{l+1} \to x^*$, $t_{l+1} \to t^*$ as $\tau_l \to 0$; then $t^* = \frac{1}{2}(z^* - c(x^*))$ due to (4.18). Thus, by (4.19), $z_j^*(z_j^* + c_j(x^*)) = 0$ for $j = 1, \dots, m$. This fact implies

$$c_j(x^*) + t_j^* = z_j^* - t_j^* = \max(0, c_j(x^*), \quad j = 1, \dots, m.$$

Therefore, it follows from (4.17) that

$$\nabla h(x^*)h(x^*) + \nabla c(x^*)\max(0, c(x^*)) = 0.$$

That is, x^* is a stationary point to minimize $\frac{1}{2}\|(h(x), \max\{0, c(x)\})\|^2$ and thus is an infeasible stationary point of problem (1.1)–(1.3). It shows that $x_{\bar{l}+1}$ is an approximate infeasible stationary point of problem (1.1)–(1.3). □

5. Numerical experiments

The numerical experiments were conducted on a Lenovo laptop with the LINUX operating system (Fedora 11). The algorithm was implemented in MATLAB (version R2008a). Two kinds of test problems that originated from the literature were solved, including some simple but hard problems, which may be an infeasible problem, a problem feasible but LICQ and MFCQ failing to hold at the solution, or a well-posed one but some class of interior-point methods was proved not to be globally convergent, and some standard test problems of the CUTE collection [3].

We use the standard initial point x_0 for all test problems, and set $t_0 = -c(x_0)$ and $s_{0j} = \min\{1, 0.95\mu/\max(0, t_{0j})\}$. The initial parameters are selected as follows: $\mu_0 = 0.1$, $\tau_0 = 1$, $\delta = 0.5$, $\sigma = 10^{-4}$, and $\epsilon = 10^{-8}$. The initial penalty is selected as $\rho_0 = \min\{100, \max(1, \|(\max(0, c(x_0)), h(x_0))\|/|f(x_0)|)\}$ (that is, it is dependent on the initial point x_0), B_0 is simply taken as the identity matrix, and B_k is updated by the well-known Powell's damped BFGS update formula (for example, see [22]). The subproblems are solved by similar techniques as those used in [12], while subproblem (3.4)–(3.5) in this article should be treated more carefully since $Q(v_k, w_k)$ is always positive semidefinite.

The vector p_k is derived similar to Algorithm 6.1 of [18], where subproblem (3.4)–(3.5) in this paper should be treated more carefully since $Q(v_k, w_k)$ is always positive semidefinite. For solving the QP subproblem (3.6)–(3.7), we have no freedom to use the MATLAB's built-in "backslash" command since the coefficient matrix can be singular without assuming LICQ. In our implementation, we first compute the null-space matrix W_k of R_k^T by the MATLAB null-space routine, and then obtain the solution of the QP by forming the reduced Hessian explicitly and using the MATLAB routine of biconjugate gradients method with preconditioner generated by the sparse incomplete Cholesky-Infinity factorization for avoiding numerically zero pivots in the sparse incomplete Cholesky factorization. The whole algorithm is terminated as either $\mu_l \le \epsilon$ or $\tau_l \le \epsilon$, or the total number of iterations (that is, the number of solving QP (3.6)–(3.7)) is larger than 500 (which is the default setting of IPOPT).

5.1. **Numerical results on three simple but hard problems.** In this subsection, we report our numerical results on three simple but hard examples taken from the literature. In Tables 1, 2, and 3, the number in column l means that the data in the row are taken from the corresponding outer iterate at which either μ_l or τ_l is reduced, $f_l = f(x_l)$, $v_l = \|(h(x_l), \max(0, c(x_l))\|$, $\|r_l\|_\infty = \|r(x_{l+1}, s_{l+1}, \lambda_{l+1}; \mu_l, \tau_l)\|_\infty$, $\|g_l\|_\infty = \|g(x_{l+1}, s_{l+1}; \mu_l, \tau_l)\|_\infty$, k is the number of inner iterations needed from (μ_{l-1}, τ_{l-1}) to (μ_l, τ_l).

The first example was presented by Wächter and Biegler [27] and further discussed by Byrd, Marazzi and Nocedal [9]:

$$
\text{(TP1)} \qquad \begin{array}{ll} \min\limits_{(x_1,x_2,x_3)} & x_1 \\ \text{s.t.} & x_1^2 - x_2 - 1 = 0, \\ & x_1 - x_3 - 2 = 0, \\ & x_2 \geq 0, \quad x_3 \geq 0. \end{array}
$$

The standard initial point is $x_0 = (-4, 1, 1)$. This problem is a well-posed problem. It has a unique global minimizer $(2, 3, 0)$, at which gradients of the active constraints are linearly independent, and MFCQ holds. However, [27] showed that many line search interior-point methods might fail to find the solution.

The implementation of our algorithm terminates at $x^* =$(2.0000,3.0000,0.0000) together with $t^* = (3.0000, 0.0000)$ and $s^* = (0.0000, 1.0000)$ in 19 total iterations. The numbers of function and gradient evaluations are 20 and 20, respectively. See Table 1 for more details on iterations; from there one can observe the superlinear convergence of v_l and $\|r_l\|_\infty$.

Table 1. Output for test problem (TP1)

l	f_l	v_l	$\|r_l\|_\infty$	$\|g_l\|_\infty$	μ_l	τ_l	k
0	-4	14	14	7.6026	0.1000	1	-
1	-1.5235	3.5562	3.5477	0.9974	0.1000	0.6000	2
2	-1.2344	2.2938	0.9737	0.5926	0.1000	0.3600	2
3	-1.0298	2.2408	0.4768	0.3133	0.1000	0.2160	1
4	-0.4257	1.9679	0.1957	0.6123	0.0500	0.2160	5
5	2.0245	1.6851e-04	0.0049	4.4642	0.0011	0.2160	6
6	2.0011	5.4789e-04	1.1834e-04	3.9978	1.3479e-06	0.2160	1
7	2.0000	1.1711e-06	2.5296e-07	4.0000	1.0000e-09	0.2160	1
8	2.0000	1.8137e-12	5.6019e-11	4.0000	-	-	1

The second example is a standard test problem taken from [17, Problem 13]:

$$
\text{(TP2)} \qquad \begin{array}{ll} \min\limits_{(x_1,x_2)} & (x_1 - 2)^2 + x_2^2 \\ \text{s.t.} & (1 - x_1)^3 - x_2 \geq 0, \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{array}
$$

The standard initial point $x_0 = (-2, -2)$ for problem (TP2) is an infeasible point. Note that its optimal solution $x^* = (1, 0)$ is not a KKT point but a singular stationary point at which LICQ and MFCQ fail to hold.

This problem has not been solved in [23, 30], but has been solved in [8, 24]. The implementation of Algorithm 3.6 terminated at an approximate point to the solution $x_l = (0.9905, -0.0000)$. The associated vectors are $t_l = (0.0000, 0.9905, 0.0000)$, $s_l = (7.3986 \cdot 10^3, 0.0000 \cdot 10^3, 7.3989 \cdot 10^3)$. The numbers of iterations, function evaluations, and gradient evaluations are 28, 76, and 29, respectively (see Table 2). It can be observed from Table 2 that our implementation terminates due to $\mu_l \le \epsilon$ instead of $\tau_l \le \epsilon$ since the gradients of active constraints at the iterates when $\tau_l = 1.5502 \cdot 10^{-5}$ are numerically treated to be linearly independent (please see the row $l = 2$ where $v_l = 0$ and $\|g_l\|_\infty = 1.0000$), which results in μ_l being reduced successively from $\mu_1 = 0.1$ to $\mu_3 = 10^{-9}$ such that the approximate solution $x_l = (0.9905, -0.0000)$ cannot be improved since $\tau_l \mu_l$ is very small.

TABLE 2. Output for test problem (TP2)

l	f_l	v_l	$\|r_l\|_\infty$	$\|g_l\|_\infty$	μ_l	τ_l	k
0	20	2	8.9116	0.7071	0.1000	1	-
1	19.6742	2.7923	1.6715	0.7592	0.1000	1.5502e-05	1
2	3.9492	0	6.2765e-06	1.0000	1.7127e-07	1.5502e-05	9
3	1.0192	9.1401e-11	1.3450e-11	1.0000	1.0000e-09	1.5502e-05	17
4	1.0192	9.1390e-11	1.3450e-11	0.0203	-	-	1

The third example is an infeasible problem named *isolated* presented by Byrd, Curtis, and Nocedal [6]:

$$\text{(TP3)} \qquad \begin{aligned} \min_{(x_1,x_2)} \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 - x_2 + 1 \le 0, \\ & x_1^2 + x_2 + 1 \le 0, \\ & -x_1 + x_2^2 + 1 \le 0, \\ & x_1 + x_2^2 + 1 \le 0. \end{aligned}$$

The standard initial point is $x_0 = (3, 2)$, and its solution $x^* = (0, 0)$ is a strict minimizer of the Euclidean norm of constraint infeasibility measures. The algorithm presented in [6] found this point. Our implementation terminates at an approximate point to it with $x_l = (-0.1547 \cdot 10^{-4}, -0.6259 \cdot 10^{-4})$, $t_l = (-0.5000, -0.5000, -0.5000, -0.5000)$, $s_l = (5.3752 \cdot 10^4, -5.4001 \cdot 10^4, 3.6816, -3.6977 \cdot 10^4)$ in 17 iterations. Note that t_l is not positive, as it should always be in existing interior-point methods based on solving logarithmic barrier problem (1.4)–(1.6). The numbers of function and gradient evaluations are 20 and 18, respectively. For more details please refer to Table 3.

5.2. **Numerical results on test problems of the CUTE collection.** A set of small- and medium-sized test problems ($n \le 100$ and $m + m_e \le 200$) with general inequality constraints taken from the CUTE collection [3] were solved. In selecting

TABLE 3. Output for test problem (TP3)

l	f_l	v_l	$\|r_l\|_\infty$	$\|g_l\|_\infty$	μ_l	τ_l	k
0	5	12	43.1944	7.5805	0.1000	1	-
1	2.5509	5.8735	4.3125	3.1566	0.1000	0.6000	1
2	0.7102	2.3657	0.5663	0.5946	0.0500	0.6000	1
3	-0.1702	2.0217	0.4846	0.3160	0.0500	0.3600	1
4	0.0606	2.0029	0.2179	0.1233	0.0500	0.2160	1
5	-0.0633	2.0042	0.1209	0.0903	0.0500	2.0359e-04	1
6	1.0010e-04	2.0000	1.0183e-04	1.8054e-04	0.0500	1.0000e-09	11
7	-7.8060e-05	2.0000	5.0005e-10	2.3597e-07	-	-	1

these problems, we eliminated those problems whose inequalities were only simple bounds on the variables, and for simplicity we did not include those problems whose inequality constraints had a nonzero on the right-hand side. We claim that our choices were not made for any biased reason but rather for preprocessing convenience. We note that a lot of test problems in the CUTE collection have bound constraints, many of which only have nonlinear equality constraints and do not have any nonlinear inequality constraints. Since we mainly focus on the interior-point approach for problems with nonlinear inequality constraints in this paper, we only consider those test problems with general nonlinear inequality constraints in current preliminary implementation. Besides general nonlinear inequality constraints, some test problems may also have equality constraints.

We found 55 test problems in the CUTE collection satisfying the above restrictions of selection. For comparison, these test problems were also solved by the interior-point solver IPOPT [28] (Version 3.0.0). It should be noted that both our algorithm and IPOPT need to find an approximate KKT point of the logarithmic barrier problem with given barrier parameter μ_l. They differ in details such as the linear system for search directions, the step-length selection and control, and the use of an exact or approximate Lagrangian Hessian. Since the two codes used different terminating conditions, the accuracy differences between them were observed (see also [2] for the same observation).

In summary, the implementations of our algorithm and IPOPT were terminated in a finite number of iterations for 53 test problems, where there were five problems (that is, problems POLAK6, TFI2, VANDERM1, VANDERM2, VANDERM3) for which the implementation of Algorithm 3.6 was terminated before approaching the feasible points of test problems. In contrast there were only three problems (problems VANDERM1, VANDERM2, VANDERM3) for IPOPT that failed in feasibility restoration. In addition, there were two problems (problems MINMAXBD and MINMAXRB) for which the implementation of our algorithm was terminated since it reached the restriction of the maximum 500 of the total number of iterations before our terminating conditions were satisfied, but IPOPT found a solution in, respectively, 27 ($N_f = 38$) and eight ($N_f = 10$) iterations (see Table 6), while there were 2 problems (problems POLAK3 and SPIRAL) for which IPOPT reached the

TABLE 4. Outputs of Algorithm 3.6 and IPOPT for CUTE problems

	Prob. data.			Algorithm 3.6			IPOPT		
Problem	n	m	f	iter	N_f	N_g	iter	N_f	N_g
CB2	3	3	1.9522	8	9	9	8	9	9
CB3	3	3	2.0000	9	10	10	9	10	10
CHACONN1	3	3	1.9522	8	9	9	8	9	9
CHACONN2	3	3	2.0000	9	10	10	8	9	9
CONGIGMZ	3	5	28.0000	36	69	37	34	42	35
DEMYMALO	3	3	-3.0000	17	25	18	13	14	14
DIPIGRI	7	4	680.6301	24	88	25	12	24	13
EXPFITA	5	22	0.0011	30	106	31	18	19	19
GIGOMEZ1	3	3	-3.0000	16	43	17	16	17	17
GIGOMEZ2	3	3	1.9522	8	10	9	9	10	10
GIGOMEZ3	3	3	2.0000	13	28	14	8	9	9
HAIFAS	13	9	-0.4500	12	21	13	12	13	13
HS10	2	1	-1.0000	11	12	12	13	14	14
HS11	2	1	-8.4985	7	8	8	8	9	9
HS12	2	1	-30.0000	12	36	13	9	10	10
HS14	2	2	1.3935	6	7	7	7	8	8
HS22	2	2	1.0000	8	10	9	6	7	7
HS29	3	1	-22.6274	35	142	36	8	9	9
HS43	4	3	-44.0000	14	18	15	9	10	10
HS100	7	4	680.6301	24	88	25	12	24	13
HS113	10	8	24.3062	54	150	55	11	12	12
KIWCRESC	3	2	2.0000e-9	13	27	14	9	11	10
MADSEN	3	6	0.6164	16	19	17	19	20	20
MAKELA1	3	2	-1.4142	10	22	11	14	15	15
MAKELA2	3	3	7.2000	16	41	17	7	8	8
MAKELA3	21	20	1.9895e-8	331	365	332	17	19	18
MAKELA4	21	40	5.4155e-8	58	354	59	9	11	10
MIFFLIN1	3	2	-1.0000	7	8	8	6	7	7
POLAK1	3	2	2.7183	165	1569	166	7	8	8
POLAK5	3	2	50.0000	7	9	8	31	32	32
ROSENMMX	5	4	-44.0000	31	79	32	16	18	17
TFI3	3	101	4.3012	147	554	148	16	18	17

restriction of the maximum 500 of the total number of iterations before the terminating conditions were satisfied, but the implementation of our algorithm found a solution in 22 ($N_f = 26$) and 146 ($N_f = 319$) iterations (see Table 5), respectively.

TABLE 5. Results for some problems obtained by Algorithm 3.6

Problem	n	m	f	v	iter	N_f	N_g	R_{KKT}
EXPFITB	5	102	0.3704	0	46	68	47	2.8463e-7
GOFFIN	51	50	3.9795e-5	0	106	297	107	3.1029e-7
HALDMADS	6	42	0.4163	0	15	34	16	1.1128e-9
HS88	2	1	1.3627	1.4230e-7	21	40	22	4.5241e-7
HS89	3	1	1.3627	0	27	33	28	1.1073e-9
HS90	4	1	1.3627	0	23	40	24	1.4424e-9
HS91	5	1	1.3627	0	27	40	28	1.0819e-9
HS92	6	1	1.3627	0	36	38	27	1.9847e-9
HS100MOD	7	4	678.6798	0	54	263	55	2.0270e-8
MIFFLIN2	3	2	-0.9993	0	47	227	48	7.1572e-9
PENTAGON	6	15	1.3685e-4	0	17	26	18	1.1987e-9
POLAK3	12	10	5.9410	0	22	26	23	9.9531e-10
S268	5	5	2.3837e-6	0	21	106	22	4.1462e-9
SPIRAL	3	2	1.9984e-9	0	146	319	147	4.3318e-7
TFI1	3	101	21.6437	0	34	60	35	7.7655e-10
WOMFLET	3	3	3.0732e-3	0	23	116	24	9.9111e-10

There are 32 problems for which the difference between the optimal function values achieved by the implementation of our algorithm and that obtained by IPOPT was within the termination tolerance. The numerical results on these problems are reported in Table 4, where the columns "n" and "m" are the numbers of variables and inequality constraints of test problems, the column "f" is the objective value, "iter", and "N_f" and "N_g" represent the total number of iterations, the total numbers of function and gradient evaluations, respectively, needed by our algorithm and IPOPT. The preliminary results show that the implementation of our algorithm needs many more iterations, evaluations of functions, and gradients than IPOPT for many of the test problems, such as problems EXPFITA, HS29, HS100, HS113, MAKELA3, MAKELA4, POLAK1, ROSENMMX, TFI3.

The implementations of both Algorithm 3.6 and IPOPT have successfully solved the other 14 problems where the difference between the optimal function values achieved was out of the termination tolerance. The numerical results for these problems are reported, respectively, in Tables 5 and 6, where v and R_{KKT} represent the measures of constraint violations and residuals of KKT equations. We also report the numerical results on the problems POLAK3 and SPIRAL which IPOPT failed to solve in Table 5 and the numerical results on the problems MINMAXBD, MINMAXRB, POLAK6, and TFI2 which our implementation of Algorithm 3.6 failed in Table 6.

It is not surprising that our implementation of Algorithm 3.6 is not very convincing in comparison to IPOPT, since our method is still at an early stage of development and IPOPT is a fairly mature code that has been tuned over almost

TABLE 6. Results for some problems obtained by IPOPT.

Problem	n	m	f	v	iter	N_f	N_g	R_{KKT}
EXPFITB	5	102	5.0193e-3	0	23	24	24	9.0914e-10
GOFFIN	51	50	3.5455e-8	0	6	7	7	9.0909e-10
HALDMADS	6	42	3.3683e-2	0	40	54	41	9.0910e-10
HS100MOD	7	4	678.6796	0	11	29	12	9.0909e-10
HS88	2	1	1.3626	0	19	24	20	9.0934e-10
HS89	3	1	1.3626	0	21	25	22	9.0933e-10
HS90	4	1	1.3626	0	22	26	23	9.0935e-10
HS91	5	1	1.3626	0	52	114	53	1.1514e-9
HS92	6	1	1.3626	0	19	23	20	9.0931e-10
MIFFLIN2	3	2	-1.0000	0	15	16	16	9.0921e-10
MINMAXBD	5	20	115.7064	0	27	38	28	9.0911e-10
MINMAXRB	3	4	-6.3636e-9	0	8	10	9	9.8694e-10
PENTAGON	6	15	1.3653e-4	0	14	15	15	9.0911e-10
POLAK6	5	4	-44.0000	0	109	315	110	9.0918e-10
S268	5	5	1.6880e-9	0	19	20	20	3.3819e-9
TFI1	3	101	5.3347	0	81	210	82	9.1234e-10
TFI2	3	101	0.6490	0	18	23	19	9.0922e-10
WOMFLET	3	3	-7.2727e-9	0	9	10	10	9.0914e-10

two decades. By taking particular care to the tests with obviously inferior performance, we observe the slow convergence of the estimates of multipliers as iterates are close to the solution, which may further affect the BFGS approximation to the Lagrangian Hessian and result in the slow convergence of barrier to the accuracy. Because the slow convergence is resulted from small step sizes, we guess that the merit function dependent on primal and dual iterates may be partially responsible for the inferior performance, and more investigation on selecting the merit function may be needed the following time. It is also noted that in our implementation the updates of parameters in the outer algorithm of Algorithm 3.6 can greatly impact the performance of the whole algorithm. Thus, it is important for improving the performance of Algorithm 3.6 in implementation to find a better way to update parameters μ_l and τ_l. Moreover, selecting a suitable initial t_0 and s_0 in the inner algorithm can be useful to reduce the number of evaluations of functions. Since our MATLAB implementation only uses MATLAB routines simply, our codes still have many computational details such as how to solve the subproblems more efficiently and how to deal with the simple and bound constraints waiting for improvements. Motivated by Benson and Shanno [2], it is worthwhile to investigate whether a hybrid approach similar to [2] can result in better performances.

6. Conclusion

We present a globally convergent primal-dual interior-point relaxation method for nonlinear programs in this article. The method is based on an equivalence between the classic logarithmic barrier problem and a particular logarithmic barrier positive relaxation problem. Different from the current globally convergent primal-dual interior-point methods in the literature, our method does not require any primal or dual iterates to be interior-point points. Thus, the interior-point restriction in existing line-search methods can be removed. A new logarithmic barrier penalty function dependent on both primal and dual variables was used to prompt the global convergence of the method, where the penalty parameter is updated adaptively. Without assuming any regularity of constraints such as feasibility or constraint qualification, the method is proved to be of strong global convergence. Preliminary numerical results demonstrates that the method can not only be efficient for well-posed feasible problems, but also is applicable for some feasible problems without LICQ or MFCQ and some even infeasible problems.

Acknowledgments

We are grateful to two anonymous referees for their very careful reading on our manuscript and revisions and for their patience in reviewing our paper. Their valuable comments and the reminding on warmstarting references help us to improve our manuscript greatly.

References

[1] H. Y. Benson and D. F. Shanno, *An exact primal-dual penalty method approach to warmstarting interior-point methods for linear programming*, Comput. Optim. Appl. **38** (2007), no. 3, 371–399, DOI 10.1007/s10589-007-9048-6. MR2354201

[2] H. Y. Benson and D. F. Shanno, *Interior-point methods for nonconvex nonlinear programming: regularization and warmstarts*, Comput. Optim. Appl. **40** (2008), no. 2, 143–189, DOI 10.1007/s10589-007-9089-x. MR2390823

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, *CUTE: Constrained and Unconstrained Testing Environment*, ACM Tran. Math. Software **21** (1995), 123–160.

[4] J. V. Burke, F. E. Curtis, and H. Wang, *A sequential quadratic optimization algorithm with rapid infeasibility detection*, SIAM J. Optim. **24** (2014), no. 2, 839–872, DOI 10.1137/120880045. MR3215068

[5] R. H. Byrd, *Robust trust-region method for constrained optimization*, Paper presented at the SIAM Conference on Optimization, Houston, TX, 1987.

[6] R. H. Byrd, F. E. Curtis, and J. Nocedal, *Infeasibility detection and SQP methods for nonlinear optimization*, SIAM J. Optim. **20** (2010), no. 5, 2281–2299, DOI 10.1137/080738222. MR2650848

[7] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program. **89** (2000), no. 1, Ser. A, 149–185, DOI 10.1007/PL00011391. MR1795061

[8] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim. **9** (1999), no. 4, 877–900, DOI 10.1137/S1052623497325107. Dedicated to John E. Dennis, Jr., on his 60th birthday. MR1724768

[9] R. H. Byrd, M. Marazzi, and J. Nocedal, *On the convergence of Newton iterations to non-stationary points*, Math. Program. **99** (2004), no. 1, Ser. A, 127–148, DOI 10.1007/s10107-003-0376-8. MR2031780

[10] L. Chen and D. Goldfarb, *Interior-point l_2-penalty methods for nonlinear programming with strong global convergence properties*, Math. Program. **108** (2006), no. 1, Ser. A, 1–36, DOI 10.1007/s10107-005-0701-5. MR2229450

[11] F. E. Curtis, *A penalty-interior-point algorithm for nonlinear constrained optimization*, Math. Program. Comput. **4** (2012), no. 2, 181–209, DOI 10.1007/s12532-012-0041-4. MR2934972

[12] Y.-H. Dai, X.-W. Liu, and J. Sun, *A primal-dual interior-point method capable of rapidly detecting infeasibility for nonlinear programs*, J. Ind. Manag. Optim., 2018. DOI 10.3934/jimo.2018190.

[13] A. Engau, M. F. Anjos and A. Vannelli, *A primal-dual slack approach to warmstarting interior-point methods for linear programming*, In: Operations Research and Cyber-Infrastructure, J.W. Chinneck, B. Kristjansson, M.J. Saltzman, eds., Springer US, 2009, 195–217.

[14] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim. **8** (1998), no. 4, 1132–1152, DOI 10.1137/S1052623496305560. MR1646122

[15] P. E. Gill and D. P. Robinson, *A primal-dual augmented Lagrangian*, Comput. Optim. Appl. **51** (2012), no. 1, 1–25, DOI 10.1007/s10589-010-9339-1. MR2872489

[16] N. I. M. Gould, D. Orban, and P. L. Toint, *An Interior-point ℓ_1-penalty Method for Nonlinear Optimization*, Numerical analysis and optimization, Springer Proc. Math. Stat., vol. 134, Springer, Cham, 2015, pp. 117–150, DOI 10.1007/978-3-319-17689-5_6. MR3446845

[17] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, vol. 187, Springer-Verlag, Berlin-New York, 1981. MR611512

[18] X. Liu and J. Sun, *A robust primal-dual interior-point algorithm for nonlinear programs*, SIAM J. Optim. **14** (2004), no. 4, 1163–1186, DOI 10.1137/S1052623402400641. MR2112969

[19] X.-W. Liu and Y.-X. Yuan, *A robust algorithm for optimization with general equality and inequality constraints*, SIAM J. Sci. Comput. **22** (2000), no. 2, 517–534, DOI 10.1137/S1064827598334861. MR1780612

[20] X. Liu and Y. Yuan, *A null-space primal-dual interior-point algorithm for nonlinear optimization with nice convergence properties*, Math. Program. **125** (2010), no. 1, Ser. A, 163–193, DOI 10.1007/s10107-009-0272-y. MR2718699

[21] J. Nocedal, F. Öztoprak, and R. A. Waltz, *An interior point method for nonlinear programming with infeasibility detection capabilities*, Optim. Methods Softw. **29** (2014), no. 4, 837–854, DOI 10.1080/10556788.2013.858156. MR3175518

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999. MR1713114

[23] D. F. Shanno and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods*, Math. Program. **87** (2000), no. 2, Ser. B, 303–316, DOI 10.1007/s101070050116. Studies in algorithmic optimization. MR1763854

[24] P. Tseng, *Convergent infeasible interior-point trust-region methods for constrained minimization*, SIAM J. Optim. **13** (2002), no. 2, 432–469, DOI 10.1137/S1052623499357945. MR1951029

[25] M. Ulbrich, S. Ulbrich, and L. N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*, Math. Program. **100** (2004), no. 2, Ser. A, 379–410, DOI 10.1007/s10107-003-0477-4. MR2062934

[26] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl. **13** (1999), no. 1-3, 231–252, DOI 10.1023/A:1008677427361. Computational optimization—a tribute to Olvi Mangasarian, Part II. MR1704122

[27] A. Wächter and L. T. Biegler, *Failure of global convergence for a class of interior point methods for nonlinear programming*, Math. Program. **88** (2000), no. 3, Ser. A, 565–574, DOI 10.1007/PL00011386. MR1782156

[28] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program. **106** (2006), no. 1, Ser. A, 25–57, DOI 10.1007/s10107-004-0559-y. MR2195616

[29] M. H. Wright, *Why a pure primal Newton barrier step may be infeasible?*, SIAM J. Optim. **5** (1995), no. 1, 1–12, DOI 10.1137/0805001. MR1315702

[30] H. Yamashita, *A globally convergent primal-dual interior point method for constrained optimization*, Optim. Methods Softw. **10** (1998), no. 2, 443–469, DOI

10.1080/10556789808805723. Dedicated to Professor Masao Iri on the occasion of his 65th birthday. MR1754750

[31] Y. X. Yuan, *On the convergence of a new trust region algorithm*, Numer. Math. **70** (1995), no. 4, 515–539, DOI 10.1007/s002110050133. MR1337231

Institute of Mathematics, Hebei University of Technology, Tianjin 300401, People's Republic of China

Email address: mathlxw@hebut.edu.cn

Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, People's Republic of China; and School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

Email address: dyh@lsec.cc.ac.cn