A primal-dual interior-point relaxation method with adaptively updating barrier for nonlinear programs

Xin-Wei Liu,* Yu-Hong Dai† and Ya-Kui Huang‡

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. In the proposed method, the barrier parameter is updated in every step as done in interior-point methods for linear programs, which is prominently different from the existing interior-point methods and the relaxation methods for nonlinear programs. Since our update for the barrier parameter is autonomous and adaptive, the method has potential of avoiding the possible difficulties caused by the unappropriate initial selection of the barrier parameter and speeding up the convergence to the solution. Moreover, it can circumvent the jamming difficulty of global convergence caused by the interior-point restriction for nonlinear programs and improve the ill conditioning of the existing primal-dual interior-point methods as the barrier parameter is small. Under suitable assumptions, our method is proved to be globally convergent and locally quadratically convergent. The 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 the CUTE collection show that our method is efficient.

Key words: Nonlinear programming, interior-point method, logarithmic-barrier problem, mini-max problem, global and local convergence.

AMS subject classifications. 90C26, 90C30, 90C51.

*Institute of Mathematics, Hebei University of Technology, Tianjin 300401, China. E-mail: mathlxw@hebut.edu.cn. The research is supported by the NSFC grants (nos. 11671116 and 11271107) and a grant of Hebei Province of China (A2020202025).

†Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China. This author is supported by the NSFC grants (nos. 11631013, 11331012 and 81173633) and the National Key Basic Research Program of China (no. 2015CB856000).

‡Institute of Mathematics, Hebei University of Technology, Tianjin 300401, China. This author is supported by the NSFC grant (no. 11701137).

1. Introduction

We consider the nonlinear programs with the form

$$\text{minimize} \quad (\min) \ f(x) \tag{1.1}$$

$$\text{subject to} \quad (\text{s.t.}) \ h(x) = 0, \quad x \geq 0, \tag{1.2}$$

where $x \in \Re^n$, $f : \Re^n \to \Re$ and $h : \Re^n \to \Re^m$ are twice continuously differentiable real-valued functions defined on $\Re^n$. If all functions f and $h_i (i = 1, \ldots, m)$ are linear functions, problem (1.1)–(1.2) is a standard form linear programming problem (for examples, see [29, 36, 37]). 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 (1.1)–(1.2). Our method can be easily extended to cope with nonlinear programs with general nonlinear inequality constraints (see section 6 for details).

There are already many efficient algorithms and several efficient solvers for nonlinear programming problem (1.1)–(1.2), among them well known is the state-of-the-art solver LANCELOT (see [13]). Using the augmented Lagrangian function on equality constraints, Conn, Gould and Toint [13] solves the relaxed subproblem

$$\min_x \quad L_A(x, \lambda; \rho) \equiv f(x) - \lambda^T h(x) + \frac{1}{2}\rho \|h(x)\|^2 \quad \text{s.t.} \quad x \geq 0, \tag{1.3}$$

where $\lambda \in \Re^m$ is an estimate of the multiplier vector, $\rho > 0$ is a penalty parameter. Both λ and ρ 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 (1.3) is a nonlinear program with nonnegative constraints, many algorithms in the literature can be used to solve this problem (see [12]).

Primal-dual interior-point methods have been demonstrated to be a class of very efficient methods for solving problem (1.1)–(1.2). For example, for nonlinear programs, the readers can consult [7, 8, 11, 14, 15, 19, 20, 21, 25, 26, 28, 30, 32, 33, 35] and the references there in. Generally, by requiring x to be an interior-point, primal-dual interior-point methods solve the logarithmic-barrier subproblem

$$\min_x \quad f(x) - \mu \sum_{j=1}^{n} \ln x_j \quad \text{s.t.} \quad h(x) = 0 \tag{1.4}$$

or its corresponding parametric Karush-Kuhn-Tucker (KKT) system, where $\mu > 0$ is a barrier parameter which is held fixed when solving the subproblem (1.4) or its parametric KKT system. Different from subproblem (1.3) in the form, problem (1.4) 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 be jammed by the interior-point requirement (see [3, 34]).

Relieving the interior-point affect on the convergence, the numerical performance and the jamming difficulty of line-search interior-point methods has been one of 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 [1, 18]). These methods were also extended to solve nonlinear programming in [2]. Numerical results in [2, 18] have shown that the warm-starting technique could improve the performance of interior-point methods for linear and nonlinear programming.

With the help of a logarithmic barrier augmented Lagrangian function, [16] 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 [16] 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 [24] 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. However, the preliminary numerical results reported in [24] seem to be out of the initial expectation and does not show the superiority of the algorithm in comparing to IPOPT, a well known and very popular interior-point solver for nonlinear programming in recent almost two decades (see [35]).

In the interior point methods for nonlinear programming, how to select a suitable initial value of the barrier parameter is one of the important topics which worths to be concerned but has been overlooking so far in the literature. Since the update for the barrier parameter is required to be monotonically nonincreasing and going to zero, the unappropriate initial selection of the barrier parameter can bring up difficulties for the efficient solution of the logarithmic-barrier subproblems, and finally affect the efficiency of the interior point methods for the original problem. Although this is not an issue in the interior point methods for linear programming, the update of the barrier parameter is not autonomous and it has been shown that some correction technique is essential for improving the performance of the interior point methods.

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 adaptively updating barrier for nonlinear programs. Our method is established on solving a sequence of KKT systems of the parametric equality constrained mini-max subproblems, and the barrier parameter is updated adaptively in each iteration as we did for linear programming. This feature is remarkably different from the classic primal-dual interior-point methods and the newly proposed primal-dual interior-point relaxation method (see [24]) for nonlinear programming, in which 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 adaptive, which makes our method to be potential of avoiding the possible difficulties caused by the unappropriate initial selection of the barrier parameter and speeding up the convergence to the solution.

Alike that in [24], our primal-dual interior-point relaxation method in this paper does not require any primal or dual iterate to be an interior point. Thus, our method is expected to be able to circumvent some difficulties including the jamming phenomenon (see [3, 34]) caused by the interior-point restriction for nonlinear programs and improve the ill conditioning of primal-dual interior-point methods as the barrier parameter is small (see [29]). Under suitable assumptions, we proved that our method can be globally convergent to the KKT point and is of locally quadratic convergence. Some preliminary numerical results on a well-posed problem for which many line-search infeasible interior-point methods fail to find the minimizer and a set of test problems from CUTE collection show that our method is efficient.

Our paper is organized as follows. In section 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 section 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 sections 4 and 5, respectively. Some preliminary numerical results on nonlinear programming test problems are reported in section 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 kth iteration, the subscript j indicates the jth 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 $\theta_k = O(t_k)$ means that there exists a constant M independent of k such that $|\theta_k| \le M|t_k|$ for all k large enough, and $\theta_k = o(t_k)$ indicates that $|\theta_k| \le \epsilon_k|t_k|$ 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, $\|\cdot\|$ 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 [24].

For any given parameters $\mu \ge 0$ and $\rho > 0$, and any $x \in \Re^n$ and $s \in \Re^n$, Liu and Dai [24] defined $z : \Re^{2n} \to \Re^n$, $z = z(x, s; \mu, \rho)$ and $y : \Re^{2n} \to \Re^n$, $y = y(x, s; \mu, \rho)$ by components to be functions on (x, s) as follows,

$$z_j(x_j, s_j; \mu, \rho) \equiv \frac{1}{2\rho}(\sqrt{(s_j - \rho x_j)^2 + 4\rho\mu} - (s_j - \rho x_j)), \tag{2.1}$$

$$y_j(x_j, s_j; \mu, \rho) \equiv \frac{1}{2\rho}(\sqrt{(s_j - \rho x_j)^2 + 4\rho\mu} + (s_j - \rho x_j)), \tag{2.2}$$

where $j = 1, \dots, n$, $x \in \Re^n$ and $s \in \Re^n$ are variables[1]. Based on definitions (2.1) and (2.2), Liu and Dai [24] proposed to solve an equivalent positive relaxation problem to the logarithmic-

[1]A little change is that both z and y are divided by ρ in this paper.

barrier subproblem (1.4) (see Theorem 2.3 of [24]) in the form

$$\min_{x,s} \; f(x) - \mu \sum_{j=1}^{n} \ln z_j(x, s; \mu, \rho) \tag{2.3}$$

$$\text{s.t.} \; h(x) = 0, \tag{2.4}$$

$$z(x, s; \mu, \rho) - x = 0. \tag{2.5}$$

For convenience of readers and our subsequent discussions, we list some preliminary results in the following lemmas. These results have some similarity to Lemmas 2.1 and 2.2 and Theorem 2.3 of [24].

Lemma 2.1 *For given $\mu \geq 0$ and $\rho > 0$, z_j and y_j are defined by (2.1) and (2.2). 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 [24]. We are left to prove the result (3). Note that

$$\begin{aligned} z_j - x_j &= \frac{1}{2\rho}(\sqrt{(s_j - \rho x_j)^2 + 4\rho\mu} - (s_j + \rho x_j)) \\ &= \frac{2\mu - 2x_j s_j}{\sqrt{(s_j - \rho x_j)^2 + 4\rho\mu} + (s_j + \rho x_j)} \\ &= \frac{\mu - x_j s_j}{\rho(y_j + x_j)}, \end{aligned}$$

and the last equality in Lemma 2.1 (3) follows from the definitions (2.1) and (2.2). All results are derived. □

By Lemma 2.1, there 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 (2.1) and (2.2). Then*
(1) z_j and y_j are differentiable, respectively, on x and s, and

$$\nabla_x z_j = \frac{z_j}{z_j + y_j} e_j, \quad \nabla_x y_j = -\frac{y_j}{z_j + y_j} e_j, \tag{2.6}$$

$$\nabla_s z_j = -\frac{1}{\rho} \frac{z_j}{z_j + y_j} e_j, \quad \nabla_s y_j = \frac{1}{\rho} \frac{y_j}{z_j + y_j} e_j, \tag{2.7}$$

where $e_j \in \Re^n$ is the j-th coordinate vector;
(2) z_j and y_j are differentiable on μ, and

$$\frac{\partial z_j}{\partial \mu} = \frac{\partial y_j}{\partial \mu} = \frac{1}{\rho} \frac{1}{z_j + y_j}; \tag{2.8}$$

(3) z_j and y_j are differentiable on ρ, 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,

$$\frac{\partial (z_j - x_j)^2}{\partial \rho} = -\frac{2}{\rho}\frac{z_j}{z_j + y_j}(z_j - x_j)^2. \tag{2.9}$$

Proof. By the result (1) of Lemma 2.2 of Liu and Dai [24], one has

$$\begin{aligned} \nabla_x(\rho z_j) &= \rho \frac{\rho z_j}{\rho z_j + \rho y_j} e_j, \quad \nabla_x(\rho y_j) = -\rho \frac{\rho y_j}{\rho z_j + \rho y_j} e_j, \\ \nabla_s(\rho z_j) &= -\frac{\rho z_j}{\rho z_j + \rho y_j} e_j, \quad \nabla_s(\rho y_j) = \frac{\rho y_j}{\rho z_j + \rho y_j} e_j. \end{aligned}$$

Thus, (2.6) and (2.7) follow immediately.

Due to

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

the result (2.8) 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

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

Thus,

$$\begin{aligned} \frac{\partial z_j}{\partial \rho} &= \frac{1}{\rho}\left(\frac{\partial \rho z_j}{\partial \rho} - z_j\right) \\ &= \frac{1}{\rho}\left(\frac{1}{2}(\frac{\partial \rho(z_j + y_j)}{\partial \rho} + \frac{\partial \rho(z_j - y_j)}{\partial \rho}) - z_j\right) \\ &= \frac{1}{\rho}\left(\frac{1}{2}(\frac{2\mu - x_j(s_j - \rho x_j)}{\rho(z_j + y_j)} + x_j) - z_j\right) \\ &= \frac{1}{\rho}\frac{z_j}{z_j + y_j}(x_j - z_j), \\ \frac{\partial y_j}{\partial \rho} &= \frac{\partial z_j}{\partial \rho} + \frac{1}{\rho}(z_j - y_j - x_j) \\ &= -\frac{1}{\rho}\frac{y_j}{z_j + y_j}(y_j + x_j), \end{aligned}$$

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

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

The following result is the foundation of development of the primal-dual interior-point relaxation method in [24].

Lemma 2.3 *Given $\mu > 0$ and $\rho > 0$. Let (x^*, λ^*) be a KKT pair of the logarithmic-barrier subproblem (1.4) and (x^*, λ^*, s^*) satisfies its KKT system*

$$\nabla f(x^*) - \nabla h(x^*)\lambda^* - s^* = 0, \tag{2.10}$$

$$h(x^*) = 0, \tag{2.11}$$

$$x_j^* > 0,\ s_j^* > 0,\ x_j^* s_j^* = \mu,\ j = 1, \dots, n, \tag{2.12}$$

where $\lambda^* \in \Re^m$ is the Lagrange multiplier vector. Then $((x^*, s^*), (\lambda^*, s^*))$ is a KKT pair of the relaxation problem (2.3)–(2.5).

Conversely, if $\mu > 0$ and $\rho > 0$, $((x^*, s^*), (\lambda^*, \nu^*))$ is a KKT pair of problem (2.3)–(2.5), where $\lambda^* \in \Re^m$ and $\nu^* \in \Re^n$ are, respectively, the associated Lagrange multipliers of constraints (2.4) and (2.5), then $\nu^* = s^*$ and (x^*, λ^*, s^*) satisfies the system (2.10)–(2.12). Thus, (x^*, λ^*) is a KKT pair of the logarithmic-barrier subproblem (1.4).

Proof. Please refer to the proof of Theorem 2.3 of [24]. □

Throughout the paper, we take z and y to be functions on (x, s) dependent on parameters (μ, ρ). Sometimes, we may ignore the variables and parameters in writing functions z and y for simplicity.

Now we consider the relaxation problem (2.3)–(2.5). By incorporating the "similar" augmented Lagrangian terms on constraints of (2.5) into the objective function, and taking the maximum with respect to s, we obtain a particular minimax problem

$$\min_{x \in \{x \in \Re^n | h(x) = 0\}} \left\{ f(x) + \sum_{j=1}^{n} \max_{s_j \in \Re^n} G(x_j, s_j; \mu, \rho) \right\}, \tag{2.13}$$

or its equivalent form

$$\min_{x \in \{x \in \Re^n | h(x) = 0\}} \max_{s \in \Re^n} F(x, s; \mu, \rho),$$

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

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

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 (2.3)) 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 ρ in z as the penalty parameter. Moreover, the barrier parameter μ exists 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 (2.1) and (2.2).*

(1) If f is twice differentiable, then F is twice differentiable on x and s. Moreover,

$$\nabla_x F(x,s;\mu,\rho) = \nabla f(x) - \rho y, \quad \nabla_x^2 F(x,s;\mu,\rho) = \nabla^2 f(x) + \rho(Z+Y)^{-1}Y,$$
$$\nabla_s F(x,s;\mu,\rho) = z - x, \qquad \nabla_s^2 F(x,s;\mu,\rho) = -\frac{1}{\rho}(Z+Y)^{-1}Z.$$

(2) Function $F(x,s;\mu,\rho)$ is a strictly concave function on s, and $F(x,s;\mu,\rho) - f(x)$ is a strictly convex function on x.
(3) For all $j = 1,\dots,n$, 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

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

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

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

Note that $\mu = \rho z_j y_j$, Lemma 2.2 (3), (2.9), and $\partial \ln z_j/\partial\rho = z_j^{-1}\partial z_j/\partial\rho$, $\partial s_j(z_j - x_j)/\partial\rho = s_j \partial z_j/\partial\rho$, the result (3) follows immediately due to $\rho(z_j - x_j) = \rho y_j - s_j$ and

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

□

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 $(x^*, s^*) \in \Re^n \times \Re^n$ is a local solution of the mini-max problem (2.13) if and only if $x^* > 0$ is a local solution of the logarithmic-barrier subproblem (1.4) and $s_j^* = \mu/x_j^*$ for all $j = 1,\cdots,n$.
(2) If $(x^*, s^*) \in \Re^n \times \Re^n$ is a local solution of the mini-max problem (2.13) and $\nabla h(x^*)$ is of full column rank, then there exists a $\lambda^* \in \Re^m$ such that

$$\nabla f(x^*) - \nabla h(x^*)\lambda - s^* = 0, \tag{2.14}$$
$$h(x^*) = 0, \tag{2.15}$$
$$z^* - x^* = 0, \tag{2.16}$$

where $z^* = z(x^*, s^*;\mu,\rho)$. Thus, (x^*, λ^*) is a KKT pair of the logarithmic-barrier subproblem (1.4).

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 ∞ as $s_j \to \infty$. Thus,

$$\max_{s_j \in \Re^n} G(x_j, s_j; \mu, \rho) = \begin{cases} -\mu \ln x_j, & \text{if} \quad x_j > 0; \\ \infty, & \text{otherwise}, \end{cases} \tag{2.17}$$

and

$$\text{argmax}_{s_j \in \Re^n} G(x_j, s_j; \mu, \rho) = \begin{cases} \mu/x_j, & \text{if} \quad x_j > 0; \\ \infty, & \text{otherwise}. \end{cases} \tag{2.18}$$

The result follows immediately from the above two equations.

(2) If (x^*, s^*) is a solution of the mini-max problem (2.13), then $z^* - x^* = 0$ by (1) and x^* is a local solution of the subproblem

$$\min_x \; F(x, s^*; \mu, \rho) \tag{2.19}$$

$$\text{s.t.} \; h(x) = 0. \tag{2.20}$$

Thus, if $\nabla h(x^*)$ is of full column rank, by the first-order necessary conditions of optimality (for example, see [29, 31]), there exists a $\lambda^* \in \Re^m$ such that (x^*, λ^*) is a KKT pair of subproblem (2.19)–(2.20), i.e., there exists a $\lambda^* \in \Re^m$ such that

$$\begin{aligned} &\nabla f(x^*) - \nabla h(x^*)\lambda^* - \rho y^* = 0, \\ &h(x^*) = 0, \\ &z^* - x^* = 0, \end{aligned}$$

where $y^* = y(x^*, s^*; \mu, \rho)$ and $z^* = z(x^*, s^*; \mu, \rho)$. Then the equations (2.14)–(2.16) are attained immediately since $z^* - x^* = 0$ implies $\rho y^* - s^* = 0$ and $x_j^* s_j^* = \mu$ for all $j = 1, \ldots, n$. □

Although the logarithmic-barrier subproblem (1.4), its relaxation subproblem (2.3)–(2.5), and the mini-max subproblem (2.13) 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 (1.1)–(1.2). For example, by using the relaxation subproblem (2.3)–(2.5), we can remove the interior-point restrictions on primal and dual variables in [24]. 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 (2.14)–(2.16) is reasonable to be chosen as the merit function. In addition, by solving the system (2.14)–(2.16), 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 (1.4) or its corresponding KKT system (please refer to Section 3 for details).

As a special example, when f and h are linear functions, that is, program (1.1)–(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 $\Re^n$. The primal-dual pair (x^*, s^*) is a solution of the mini-max problem (2.13) if and only if there exists a $\lambda^* \in \Re^m$ such that (x^*, λ^*) is a KKT pair of the logarithmic-barrier subproblem (1.4).*

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

Based on solving the mini-max subproblem (2.13), we develop a novel primal-dual interior-point relaxation method for solving the nonlinear constrained optimization problem (1.1)–(1.2). Since problem (2.13) is originated 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 μ in every iteration, which shares a similar algorithmic framework to the interior-point methods for linear programming, and is remarkably different from those for nonlinear programming in which they often attempt to find an approximate solution for a fixed parameter μ in an inner algorithm and then reduce the barrier parameter μ by the residual of the solution in an outer algorithm. In particular, our update for the barrier parameter is autonomous and adaptive, which makes our method capable of avoiding the possible difficulties caused by unappropriate initial selection of the barrier parameter and have the potential of speeding up the convergence to the solution.

Instead of solving the subproblem (2.13) directly, we solve the associated system (2.14)–(2.16) and consider the extended system of equations of (2.14)–(2.16) in the form

$$\mu = 0, \tag{3.1}$$

$$\nabla f(x) - \nabla h(x)\lambda - s = 0, \tag{3.2}$$

$$h(x) = 0, \tag{3.3}$$

$$z - x = 0, \tag{3.4}$$

where $z = z(x, s; \mu, \rho)$ and $y = y(x, s; \mu, \rho)$ are functions on x and s defined by (2.1) and (2.2). Distinct from our recent work [16, 24] and the existing interior-point methods for nonlinear programs, we take μ as a variable in the system (3.1)–(3.4) instead of a parameter in that of (2.14)–(2.16). 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 $(x^*, \lambda^*, s^*) \in \Re^n \times \Re^n$ satisfying the extended system of equations (3.1)–(3.4) is a KKT triple of the original problem (1.1)–(1.2).

Denote the residual function of the system (2.14)–(2.16) as follows,

$$\phi_{(\mu,\rho)}(x, \lambda, s) = \frac{1}{2}\|\nabla f(x) - \nabla h(x)\lambda - s\|^2 + \frac{1}{2}\|h(x)\|^2 + \frac{1}{2}\|z - x\|^2. \tag{3.5}$$

Using this notation, the system (3.1)–(3.4) can be further reformulated as

$$\mu + \gamma\phi_{(\mu,\rho)}(x, \lambda, s) = 0, \tag{3.6}$$

$$\nabla f(x) - \nabla h(x)\lambda - s = 0, \tag{3.7}$$

$$h(x) = 0, \tag{3.8}$$

$$z - x = 0, \tag{3.9}$$

where μ is supposed to be nonnegative, $\phi_{(\mu,\rho)}(x, \lambda, s)$ is defined by (3.5), $\gamma \in (0, 1]$ is a given parameter.

Suppose that (x_k, λ_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 $r_k^d = \nabla f(x_k) - \nabla h(x_k)\lambda_k - s_k$, $r_k^e = z_k - x_k$, and $r_k^h = h(x_k)$ be the residuals of equations in (3.7)–(3.9) at iterate k. Our proposed method generates the new value of parameter μ_{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 Newton's equations of system (3.7)–(3.9), $\alpha_k \in (0, 1]$ is the step-size. At iterate (x_k, λ_k, s_k) with $\mu = \mu_k$ and $\rho = \rho_k$, $(d_{xk}, d_{\lambda k}, d_{sk})$ is derived from solving the linearized system with the form

$$\begin{bmatrix} B_k & -\nabla h_k & -I \\ \nabla h_k^T & 0 & 0 \\ (Z_k + Y_k)^{-1}Y_k & 0 & \frac{1}{\rho_k}(Z_k + Y_k)^{-1}Z_k \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_s \end{bmatrix} = \begin{bmatrix} -r_k^d \\ -r_k^h \\ r_k^e + \frac{1}{\rho_k}\Delta\mu_k(Z_k + Y_k)^{-1}e \end{bmatrix}, \tag{3.10}$$

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

$$\begin{bmatrix} B_k + \rho_k(Z_k + Y_k)^{-1}Y_k & -\nabla h_k & -(Z_k + Y_k)^{-1}Y_k \\ -\nabla h_k^T & 0 & 0 \\ -(Z_k + Y_k)^{-1}Y_k & 0 & -\frac{1}{\rho_k}(Z_k + Y_k)^{-1}Z_k \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_s \end{bmatrix} = -\begin{bmatrix} \hat{r}_k^d - \Delta\mu_k(Z_k + Y_k)^{-1}e \\ -r_k^h \\ r_k^e + \frac{1}{\rho_k}\Delta\mu_k(Z_k + Y_k)^{-1}e \end{bmatrix},$$

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, λ_k, s_k), $z_k = z(x_k, s_k; \mu_k, \rho_k)$, $y_k = y(x_k, s_k; \mu_k, \rho_k)$, $Z_k = \mathrm{diag}\,(z_k)$, $Y_k = \mathrm{diag}\,(y_k)$, $\Delta\mu_k = -\mu_k + \gamma\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$, $\hat{r}_k^d = \nabla f(x_k) - \nabla h(x_k)\lambda_k - \rho_k y_k$.

Since we are facing a mini-max subproblem, taking the residual function $\phi_{(\mu,\rho)}(x,\lambda,s)$ defined by (3.5) as the merit function is a natural and reasonable selection. The step-size α_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,λ_k,s_k) to $(x_{k+1},\lambda_{k+1},s_{k+1})$ and the barrier parameter varies from μ_k to μ_{k+1}, while the penalty parameter ρ_k holds fixed. Then ρ_k is updated adaptively to ρ_{k+1} such that $\rho_{k+1} \ge \rho_k$.

In the following, we describe our algorithm for problem (1.1)–(1.2), in which the parameter μ is updated adaptively in every iteration.

Algorithm 1 A novel primal-dual interior-point relaxation method for problem (1.1)–(1.2)

Given $(x_0,\lambda_0,s_0)\in\Re^n\times\Re^m\times\Re^n$, $B_0\in\Re^{n\times n}$, $\mu_0>0$, $\rho_0>0$, $\eta>1$, $\gamma_0,\delta,\tau,\sigma\in(0,1)$. Evaluate z_0 and y_0 by (2.1) and (2.2), 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 (2.1) and (2.2) 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 $\mu_k\le\epsilon$ and $\phi_{(\mu_k,\rho_k)}(x_k,\lambda_k,s_k)\le\epsilon$, 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 (3.10) to obtain $d_k\equiv(d_{xk},d_{\lambda k},d_{sk})$.

Step 3. Select the step-size $\alpha_k\in(0,1]$ to be the maximal in $\{1,\delta,\delta^2,\dots\}$ such that the inequality

$$\phi_{(\mu_k+\alpha_k\Delta\mu_k,\rho_k)}(x_k+\alpha_k d_{xk},\lambda_k+\alpha_k d_{\lambda k},s_k+\alpha_k d_{sk})\le(1-2\tau\alpha_k)\phi_{(\mu_k,\rho_k)}(x_k,\lambda_k,s_k) \quad (3.11)$$

is satisfied.

Step 4. Set $\mu_{k+1}=\mu_k+\alpha_k\Delta\mu_k$, $x_{k+1}=x_k+\alpha_k d_{xk}$, $s_{k+1}=s_k+\alpha_k d_{sk}$, and $\lambda_{k+1}=\lambda_k+\alpha_k d_{\lambda k}$.

Step 5. Update ρ_k to $\rho_{k+1}=\max\{\rho_k,\sigma\|s_{k+1}\|_\infty/\max(\|x_{k+1}\|,1)\}$. Evaluate by (2.1) and (2.2)

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

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

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

Step 5.1 While $\mu_{k+1,\ell}>\max\{\eta\phi_{(\mu_{k+1,\ell},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}),\epsilon\}$, set $\mu_{k+1,\ell+1}=\mu_{k+1,\ell}/\eta$;
evaluate z_{k+1} and y_{k+1} by (2.1) and (2.2) with $\mu=\mu_{k+1,\ell+1}$,
compute $\phi_{(\mu_{k+1,\ell+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1})$, set $\ell=\ell+1$, end.

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

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

End (while)

In Algorithm 1, the initial point can be an arbitrary point. Our algorithm does not also require any primal or dual iterate to be interior during the iterative process, which is prominently distinct from the existing interior-point methods. Furthermore, we do not reduce the barrier parameter coercively as we have done in the interior-point methods for nonlinear programs. In those methods, the barrier parameter is reduced as the approximate solution of the logarithmic-barrier subproblem or its KKT residuals satisfied the given accuracy.

Steps 0.1 and 5.1 are used to prevent μ_0 and μ_{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 $\mu_{k+1} \le \max\{\eta\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}),\epsilon\}$, then one of the following three kinds of results will arise:

(1) $\epsilon < \mu_{k+1} \le \eta\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1})$;

(2) $\mu_{k+1} \le \epsilon \le \eta\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1})$;

(3) $\mu_{k+1} \le \epsilon$ and $\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}) \le \epsilon/\eta < \epsilon$.

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,

$$\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}) \ge \epsilon/\eta. \tag{3.12}$$

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

$$\mu_{k+2,0} = (1-\alpha_{k+1})\mu_{k+1} + \alpha_{k+1}\gamma_0\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}) < \mu_{k+1} \tag{3.13}$$

and

$$\mu_{k+2,0} > \gamma_0\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}) \ge (\gamma_0/\eta)\epsilon; \tag{3.14}$$

otherwise, $\mu_{k+1} = \gamma\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1}) < \gamma_0\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1},\lambda_{k+1},s_{k+1})$, μ_{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 set $\mu_{k+2,0} = \mu_{k+1}$. Thus, there is always $(\gamma_0/\eta)\epsilon \le \mu_{k+1,0} \le \mu_k$ for all $k \ge 0$.

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

$$\min\ c^T x \quad \text{s.t. } Ax = b,\ x \ge 0. \tag{3.15}$$

That is, $f(x) = c^T x$, $h(x) = Ax - b$. 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, (3.10) is reduced to the following system

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ (Z_k+Y_k)^{-1}Y_k & 0 & \frac{1}{\rho_k}(Z_k+Y_k)^{-1}Z_k \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_s \end{bmatrix}$$
$$= -\begin{bmatrix} (A^T\lambda_k + s_k - c) \\ (Ax_k - b) \\ (x_k - z_k) - \frac{1}{\rho_k}\Delta\mu_k(Z_k+Y_k)^{-1}e \end{bmatrix},$$

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

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \rho_k Y_k & 0 & Z_k \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_s \end{bmatrix} = \begin{bmatrix} c - A^T\lambda_k - s_k \\ b - Ax_k \\ \mu_k e - X_k S_k e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \rho_k(Z_k - X_k)(z_k - x_k) + \Delta\mu_k e \end{bmatrix}. \quad (3.16)$$

Comparing with the system for search direction in classic primal-dual interior-point methods for linear programming (for example, see (14.12) of Nocedal and Wright [29]), our system (3.16) is different in that both S_k and X_k in the last row of the Jacobian have been substituted with ρY_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 note from what follows, these changes make our method capable of improving the ill conditioning of primal-dual interior-point methods for linear programming.

Note that (3.16) can be formalized as

$$\begin{aligned} (AY_k^{-1}Z_kA^T)d_\lambda &= \rho_k(b - Ax_k) + AY_k^{-1}Z_k(c - A^T\lambda_k - s_k) \\ &\quad -\rho_k A(I + Y_k^{-1}Z_k)(z_k - x_k) - \Delta\mu_k AY_k^{-1}e, \\ d_s &= (c - A^T\lambda_k - s_k) - A^T d_\lambda, \\ d_x &= (I + Y_k^{-1}Z_k)(z_k - x_k) + \frac{1}{\rho_k}(\Delta\mu_k Y_k^{-1}e - Y_k^{-1}Z_k d_s). \end{aligned}$$

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

$$\begin{aligned} (AZ_k^2A^T)d_\lambda &= \mu_k(b - Ax_k) + AZ_k^2(c - A^T\lambda_k - s_k) - A(\mu_k I + \rho_k Z_k^2)(z_k - x_k) \\ &\quad -\Delta\mu_k AZ_k e, && (3.17) \\ d_s &= (c - A^T\lambda_k - s_k) - A^T d_\lambda, && (3.18) \\ d_x &= (I + \frac{\rho_k}{\mu_k}Z_k^2)(z_k - x_k) + \frac{1}{\mu_k}(\Delta\mu_k Z_k e - Z_k^2 d_s). && (3.19) \end{aligned}$$

If $z_k \to x^*$ and $\mu_k \to 0$ as $k \to \infty$, where x^* is an optimal solution of the nondegenerate linear program, then AZ_k should be of full rank and (3.17) 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 [29]). One may note that (3.19) can be numerically difficult as $\mu_k \to 0$. However, in contrast to the implicit trap of the existing primal-dual algorithms, this difficulty of (3.19) is explicit and singlet. Theoretically, under suitable conditions, we have proved that, for all $j = 1, \dots, n$, $\frac{1}{\mu_k}z_{kj}$ is bounded away from zero (see Lemma 4.2 for details).

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 $\mu_k \ge (\gamma_0/\eta)\epsilon$ for all $k \ge 0$.*

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

$$\mu_{k+1} \ge \epsilon/\eta. \quad (3.20)$$

By Step 5.1, $\mu_{k+1} = \mu_{k+1,\ell}$ 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 (3.20). ☐

In view of (2.1) and (2.2), $\mu_k > 0$ implies $y_k > 0$ and $z_k > 0$. The following result asserts that the linear system (3.10) has a unique solution.

Lemma 3.2 *If Algorithm 1 does not terminate at x_k, ∇h_k has full column rank and $v^T B_k v \geq 0$ for all $v \in \Re^n$ with $\nabla h_k^T v = 0$, then the coefficient matrix of the linear system (3.10) is nonsingular.*

Proof. In order to obtain our desired result, we need prove that the system of equations

$$B_k d_x - \nabla h_k d_\lambda - d_s = 0, \tag{3.21}$$

$$\nabla h_k^T d_x = 0, \tag{3.22}$$

$$\rho_k Y_k d_x + Z_k d_s = 0 \tag{3.23}$$

has only zero solution. Left-multiplying d_x^T on the two-sides of (3.21), one has $d_x^T B_k d_x = d_x^T d_s$ due to (3.22). Thus, by (3.23),

$$d_x^T B_k d_x = -\rho_k d_x^T Z_k^{-1} Y_k d_x \leq 0. \tag{3.24}$$

Note that the conditions of the lemma suggest $d_x^T B_k d_x \geq 0$ for all d_x satisfying (3.22). Hence,

$$\rho_k d_x^T Z_k^{-1} Y_k d_x = 0,$$

which implies $d_x = 0$. Therefore, $d_s = 0$ and $\nabla h_k d_\lambda = 0$ due to the last and the first equations of the preceding system. Since ∇h_k has full column rank, the equation $\nabla h_k d_\lambda = 0$ implies $d_\lambda = 0$. Hence, our proof is completed. ☐

If Algorithm 1 does not terminate at x_k, then $\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) \geq \epsilon/\eta > 0$ due to (3.12). 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 (3.10) will be zero for some integer k, which implies $\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) = 0$, a contradiction to (3.12). 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 $f : \Re^n \to \Re$ and $h : \Re^n \to \Re^m$ are twice continuously differentiable on $\Re^n$. There always exists an $\alpha_k \in (0, 1]$ such that (3.11) holds.*

Proof. The supposition implies that $\phi_{(\mu,\rho_k)}(x, \lambda, s)$ is differentiable on (μ, x, λ, s), thus it is directionally differentiable and, due to (3.10), its directional derivative along $(\Delta\mu_k, d_k)$ at (x_k, λ_k, s_k) with $\mu = \mu_k$ is

$$\begin{aligned}
&\phi'_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k; \Delta\mu_k, d_k) \\
&= \begin{pmatrix} \dfrac{D\phi_{(\mu,\rho_k)}(x, \lambda, s)}{D\mu} & \nabla_{(x,\lambda,s)}\phi_{(\mu,\rho_k)}(x, \lambda, s)^T \end{pmatrix} |_{(\mu,x,\lambda,s)=(\mu_k,x_k,\lambda_k,s_k)} \begin{pmatrix} \Delta\mu_k \\ d_k \end{pmatrix} \\
&= -2\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k).
\end{aligned} \tag{3.25}$$

The Taylor's expansion of $\phi_{(\mu_k+\alpha\Delta\mu_k,\rho)}(x_k+\alpha d_{xk}, s_k+\alpha d_{\lambda k}, \lambda_k+\alpha d_{sk})$ regarding α at $\alpha = 0$ shows that

$$\begin{aligned}
&\phi_{(\mu_k+\alpha\Delta\mu_k,\rho)}(x_k+\alpha d_{xk}, s_k+\alpha d_{\lambda k}, \lambda_k+\alpha d_{sk}) - \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) \\
&= \alpha\phi'_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k; \Delta\mu_k, d_k) + o(\alpha) \\
&= -2\tau\alpha\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) - 2(1-\tau)\alpha\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) + o(\alpha).
\end{aligned} \tag{3.26}$$

Thus, (3.11) holds for all sufficiently small $\alpha > 0$ since $\tau < 1$ and $\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) > 0$. □

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 condition is satisfied. Moreover, (3.13) 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 $\phi_{(\mu_{k+1},\rho_k)}(x_{k+1}, \lambda_{k+1}, s_{k+1}) \le \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$. If $\rho_{k+1} \ge \rho_k > 0$, one has*

$$\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1}, \lambda_{k+1}, s_{k+1}) \le \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k).$$

Proof. Note that

$$\begin{aligned}
&\frac{D\phi_{(\mu_{k+1},\rho)}(x_{k+1}, \lambda_{k+1}, s_{k+1})}{D\rho}|_{\rho=\rho_k} \\
&= -\frac{1}{\rho_k}(\hat{z}_{k+1} - x_{k+1})^T(\hat{Z}_{k+1} + \hat{Y}_{k+1})^{-1}\hat{Z}_{k+1}(\hat{z}_{k+1} - x_{k+1}) \\
&< 0,
\end{aligned}$$

where $\hat{Z}_{k+1} = \mathrm{diag}\,(\hat{z}_{k+1})$, $\hat{Y}_{k+1} = \mathrm{diag}\,(\hat{y}_{k+1})$ 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 ρ over $\rho > 0$, which implies the desired result. □

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*

$$0 < \mu_{k+1} \le \mu_k \le \mu_0 \quad and \quad \mu_k \le \eta\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$$

for all $k > 0$.

Proof. The result follows from (3.13) and $\mu_{k+2} \le \mu_{k+2,0}$ immediately. □

4. Global convergence

For doing 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 $\lim_{\ell\to\infty} \phi_{(\mu_{k,\ell},\rho_\ell)}(x_k, \lambda_k, s_k) = 0$, thus (x_k, λ_k, s_k) is a KKT triple of the problem (1.1)–(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 (1.1)–(1.2), i.e., the cluster points together with $\mu^* = 0$ are solutions of the system of equations (3.1)–(3.4).

We need the following blanket assumptions for our global convergence analysis.

Assumption 4.1
(1) The functions f and $h_i (i \in \mathcal{I})$ are twice continuously differentiable on $\Re^n$;
(2) The iterative sequence $\{x_k\}$ is in an open bounded set of $\Re^n$;
(3) The sequence $\{B_k\}$ is bounded, and for all $k \ge 0$ and $d_x \in \Re^n$, $d_x^T B_k d_x \ge \chi \|d_x\|^2$, where $\chi > 0$ is a constant;
(4) For all $k \ge 0$, $\nabla h(x_k)$ 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 [5, 6, 7, 26]) for weakening Assumption 4.1 (3) and (4), and the line search procedure without using a penalty function or a filter (see [22, 27]) 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 ρ_k keeps constant after a finite number of iterations, then $\{y_k\}$, $\{s_k\}$ and $\{\lambda_k\}$ are bounded, and there exists a scalar $\hat{\tau} > 0$ such that, for $j = 1, \ldots, n$,*

$$y_{kj} \ge \hat{\tau}\mu_k, \;\; z_{kj} \ge \hat{\tau}\mu_k.$$

Proof. Note that $z_k \ge 0$ for all $k \ge 0$ and

$$\phi_{(\mu_{k+1},\rho_{k+1})}(x_{k+1}, \lambda_{k+1}, s_{k+1}) \le \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) \le \ldots \le \phi_{(\mu_0,\rho_0)}(x_0, \lambda_0, s_0).$$

By the definition (3.5) of $\phi_{(\mu,\rho)}(x, \lambda, s)$, one has

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

which together with Assumption 4.1 (2) implies that $\{z_k\}$ is bounded. Thus, due to (2.1), for every $j = 1, \ldots, n$, $\sqrt{(s_{kj}/\rho_k - x_{kj})^2 + 4\mu_k/\rho_k} - (s_{kj}/\rho_k - x_{kj})$ is bounded. That is, $s_{kj}/\rho_k \not\to -\infty$ as $k \to \infty$, which implies that $\{s_k\}$ is bounded below.

If ρ_k is bounded above, then $\{s_k\}$ is bounded since $\|s_k\|_\infty \le \rho_k \max\{\|x_k\|, 1\}/\sigma$ for all $k \ge 0$. Thus, by (2.2), $\{y_k\}$ is bounded. Due to

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

and note that $\nabla h(x_k)$ has full column rank, one can deduce that $\{\lambda_k\}$ is bounded.

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

The preceding results show that, if ρ_k keeps constant after a finite number of iterations, 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 $\mu_k \le \eta\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$ for all sufficiently large k, then*

$$\lim_{k\to\infty} \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) = 0, \text{ and } \lim_{k\to\infty} \mu_k = 0.$$

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 $\phi^* \ge 0$ 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 μ_k keeps constant provided $\mu_k \le \gamma_0\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$. 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

$$\begin{pmatrix} B_k & -A_k^T & -I \\ A_k & 0 & 0 \\ \rho_k Y_k & 0 & Z_k \end{pmatrix}$$

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 α_k is bounded away from zero since, by (3.26),

$$\begin{aligned}
&\phi_{(\mu_k+\alpha\Delta\mu_k,\rho_k)}(x_k + \alpha d_{xk}, \lambda_k + \alpha d_{\lambda k}, s_k + \alpha d_{sk}) - (1 - 2\tau\alpha)\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) \\
&= -2(1-\tau)\alpha\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) + o(\alpha) \\
&\le -2(1-\tau)\sigma\alpha + o(\alpha),
\end{aligned}$$

which suggests that there exists an $\alpha^* \in (0, 1)$ such that (3.11) 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. □

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, $\mu_k \le \eta\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$. 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, λ_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

$$\lim_{k\to\infty} \mu_k = 0, \quad \lim_{k\to\infty} \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k) = 0.$$

That is, any 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

$$\mu_{k_i} > \eta\phi_{(\mu_{k_i},\rho_{k_i})}(x_{k_i}, \lambda_{k_i}, s_{k_i}), \quad \mu_{k_{i+1}} > \eta\phi_{(\mu_{k_{i+1}},\rho_{k_{i+1}})}(x_{k_{i+1}}, \lambda_{k_{i+1}}, s_{k_{i+1}}), \tag{4.1}$$

ℓ_i is the number of loops in Step 5.1 of Algorithm 1 such that

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

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

$$\mu_{k_i+1} = (1 - \alpha_{k_i})\mu_{k_i,\ell_i} + \alpha_{k_i}\gamma\phi_{(\mu_{k_i,\ell_i},\rho_{k_i})}(x_{k_i}, \lambda_{k_i}, s_{k_i}) \leq \mu_{k_i,\ell_i} \leq \mu_{k_i}/\eta,$$

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 (4.1) is derived. Therefore,

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

Note that both $\{\mu_k\}$ and $\{\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)\}$ are monotonically nonincreasing sequences. 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 results (1) and (3) 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 $w_k = (x_k, \lambda_k, s_k) \in \Re^{2n+m}$ 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 $\Re^n$, 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 $\nabla h(x^*)^T d = 0$, where $B^* = \nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 h_i(x^*)$ and $\lambda^* \in \Re^m$ is the Lagrange multiplier vector associated with at x^* for all equality constraints.

Under Assumption 5.1, $\{s_k\}$ is bounded, thus ρ_k will keep constant after a finite number of iterations. By Theorem 4.4, (x^*, λ^*, 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 (2.1) and (2.2) 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 $Y^* = diag(y^*)$ and $Z^* = diag(z^*)$. Then the matrix*

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

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

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

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^*, $d \in \Re^{2n+m+1}$ has a partition $d = (d_\mu, d_w)$, where $d_\mu \in \Re$, $d_w = (d_x, d_\lambda, d_s)$ with $d_x \in \Re^n$, $d_\lambda \in \Re^m$, and $d_s \in \Re^n$. Thus,

$$(1 + \tfrac{D\phi_{(0,\rho^*)}(w^*)}{D\mu}) d_\mu + (\nabla_w \phi_{(0,\rho^*)}(w^*))^T d_w = 0, \tag{5.1}$$

$$B^* d_x - \nabla h(x^*) d_\lambda - d_s = 0, \tag{5.2}$$

$$\nabla h(x^*)^T d_x = 0, \tag{5.3}$$

$$-\tfrac{1}{\rho^*}(Z^* + Y^*)^{-1} e d_\mu + (Z^* + Y^*)^{-1} Y^* d_x + \tfrac{1}{\rho^*}(Z^* + Y^*)^{-1} Z^* d_s = 0. \tag{5.4}$$

Note that, by (5.2)–(5.4),

$$\frac{D\phi_{(0,\rho^*)}(w^*)}{D\mu} d_\mu^* + (\nabla_w \phi_{(0,\rho^*)}(w^*))^T d_w^* = 0.$$

Thus, due to (5.1), $d_\mu^* = 0$. Furthermore, since $y_j^* z_j^* = 0$ for all $j = 1, \ldots, n$, (5.4) implies $(d_x^*)^T d_s^* = 0$. Hence,

$$(d_x^*)^T B^* d_x^* = 0, \quad \nabla h(x^*)^T d_x^* = 0,$$

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

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

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

Then $\Phi(0, w^*) = 0$. The following lemma can be obtained in a way similar to Lemma 2.1 in [9]. 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 $(\mu, w) \in \{(\mu, w) \in \Re_{++} \times \Re^{2n+m} | \|(\mu, w) - (0, w^*)\| < \epsilon\}$, $\nabla_{(\mu,w)}\Phi(\mu, w)$ is invertible, $\|[\nabla_{(\mu,w)}\Phi(\mu, w)]^{-1}\| \le M_0$, and*

$$\|(\nabla_{(\mu,w)}\Phi(\mu, w))^T((\mu, w) - (0, w^*)) - \Phi(\mu, w)\| \le L_0 \|(\mu, w) - (0, w^*)\|^2, \tag{5.5}$$

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.*
(1) If $\|(B_k - B^*)d_x\| = O(\|d_x\|^2)$ for every $d_x \in \Re^n$, then

$$\|(\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)\| = O(\|(\mu_k, w_k) - (0, w^*)\|^2). \tag{5.6}$$

That is, $(\Delta\mu_k, d_k)$ is a quadratically convergent step.
(2) If $\|(B_k - B^*)d_x\| = o(\|d_x\|)$ for every $d_x \in \Re^n$, then

$$\|(\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)\| = o(\|(\mu_k, w_k) - (0, w^*)\|), \tag{5.7}$$

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

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

$$\limsup_{k\to\infty} \|(\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)\| / \|(\mu_k, w_k) - (0, w^*)\|^2 \le \xi, \tag{5.8}$$

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 $\nabla^2_{xx}L(w_k) = \nabla^2 f(x_k) - \sum_{i=1}^m \lambda_{ki}\nabla^2 h_i(x_k)$ 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

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

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 $\|G_k^{-1}\| \le M_0$ 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,

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

Therefore,

$$\begin{aligned}
&\|(\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)\| \\
&= \|J_k^{-1}(J_k((\mu_k, w_k) - (0, w^*)) - \Phi_k - (B_k - B^*)d_{xk} + (\nabla^2_{xx}L(w_k) - B^*)d_{xk})\| \\
&\le M_0[L_0\|(\mu_k, w_k) - (0, w^*)\|^2 + O(\|(\mu_k, w_k) - (0, w^*)\|^2)],
\end{aligned} \tag{5.9}$$

where the last inequality follows from (5.5) of Lemma 5.3. Thus, (5.8) follows immediately from (5.9).

If $\|(B_k - B^*)d_x\| = o(\|d_x\|)$, then the last inequality (5.9) should be

$$\begin{aligned}
&\|(\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)\| \\
&\le M_0[L_0\|(\mu_k, w_k) - (0, w^*)\|^2 + o(\|(\mu_k, w_k) - (0, w^*)\|)].
\end{aligned} \tag{5.10}$$

Hence, the result (2) follows immediately. □

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 $d_x \in \Re^n$, $\sigma < 1/2$, then either $\mu_{k+1} = \mu_k$ or $\mu_{k+1} = \gamma_0\phi_{(\mu_k,\rho^*)}(w_k)$, $x_{k+1} = x_k + d_{xk}$, $s_{k+1} = s_k + d_{sk}$, and $\lambda_{k+1} = \lambda_k + d_{\lambda k}$ for all sufficiently large k. Moreover, $\|w_{k+1} - w^*\| = O(\|w_k - w^*\|^2)$.*

Proof. We need to prove that, for all sufficiently large k, $\alpha_k = 1$ will be accepted by the line search procedure (3.11). By Theorem 5.4,

$$\begin{aligned}
&\phi_{(\mu_k+\Delta\mu_k,\rho^*)}(w_k + d_k) \\
&= \phi_{(\mu_k+\Delta\mu_k,\rho^*)}(w_k + d_k) - \phi_{(0,\rho^*)}(w^*) \\
&= (\nabla_{(\mu,w)}\phi_{(\mu_k+\Delta\mu_k,\rho^*)}(w_k + d_k))^T((\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)) \\
&\quad + O(\|(\mu_k, w_k) + (\Delta\mu_k, d_k) - (0, w^*)\|^2) \\
&= O(\|(\mu_k, w_k) - (0, w^*)\|^2).
\end{aligned}$$

Note that $\phi_{(\mu_k,\rho^*)}(w_k) = \phi_{(\mu_k,\rho^*)}(w_k) - \phi_{(0,\rho^*)}(w^*) = O(\|(\mu_k, w_k) - (0, w^*)\|)$ 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 (3.11).

By Theorem 5.4 (1),

$$\|(\mu_{k+1}, w_{k+1}) - (0, w^*)\| = O(\|(\mu_k, w_k) - (0, w^*)\|^2). \tag{5.11}$$

Due to $\mu_k \leq \eta\phi_{(\mu_k,\rho^*)}(w_k)$, $\mu_k = O(\|w_k - w^*\|)$. The desired result follows from (5.11) immediately. □

6. Numerical experiments

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

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

by substituting (2.1) and (2.2), respectively, with

$$\begin{aligned} z_j(x, s; \mu, \rho) &\equiv \frac{1}{2\rho}(\sqrt{(s_j - \rho g_j(x))^2 + 4\rho\mu} - (s_j - \rho g_j(x))), \\ y_j(x, s; \mu, \rho) &\equiv \frac{1}{2\rho}(\sqrt{(s_j - \rho g_j(x))^2 + 4\rho\mu} + (s_j - \rho g_j(x))), \end{aligned}$$

where $g : \Re^n \to \Re^{m_{\mathcal{I}}}$ is a twice continuously differentiable real-valued function on $\Re^n$, $j = 1, \ldots, m_{\mathcal{I}}$. 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 [34] and further discussed by Byrd, Marazzi and Nocedal [10]:

$$\min \quad x \tag{6.1}$$

$$\text{s.t.} \quad x^2 - 1 \geq 0, \quad x - 2 \geq 0. \tag{6.2}$$

This problem is well-posed since it has a unique global minimizer $x^* = 2$, at which both LICQ and MFCQ hold. However, starting from $x_0 = -4$, [34] 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 CUTE collection [4]. Since the code is very elementary, we restricted our test problems to the 126 HS problems. 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 [23].

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. 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 [35] (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:

$$E(x_k, \lambda_k, s_k) \leq \epsilon, \tag{6.3}$$

where $E(x_k, \lambda_k, s_k) = \max\{\|\nabla f(x_k) - \nabla h(x_k)\lambda_k - s_k\|_\infty, \|h(x_k)\|_\infty, \|\max\{-(x_k+s_k), 0\}\|_\infty, \|x_k \circ s_k\|_\infty\}$, $x_k \circ s_k$ 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 [35], then the accuracy differences between Algorithm 1 and IPOPT should be in the range of the tolerance.

For test problem (6.1)–(6.2), 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 $s_1^* = -1.1972\text{e}-16$, $s_2^* = 1.0000$ 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 μ_k, $\phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$ and $E(x_k, \lambda_k, s_k)$, where μ_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 ℓ_∞ norm of violations of constraints, $\phi_k = \phi_{(\mu_k,\rho_k)}(x_k, \lambda_k, s_k)$, $E_k = E(x_k, \lambda_k, s_k)$. 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 [16] and [24] in totally 16 and 19 iterations, respectively.

Table 1: Output of Algorithm 1 for test problem (6.1)–(6.2)

k	μ_k	x_k	s_k	f_k	v_k	ϕ_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

When solving the HS test problems of the CUTE 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 (3.10) is degenerate. The latter three cases of termination can be resulted from that the Hessian does not satisfy Assumption 4.1 (3) and the condition (4) of Assumption 4.1 does not hold.

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 [35]. In our implementation, Algorithm 1 successfully solved 79 problems and terminated with (6.3), while IPOPT found the approximate solutions of 125 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 Figures 1–4 to show log scaling performance profiles (see Dolan and Moŕe [17]) 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. Figures 1–3 show that, under the measures on the former three items, IPRM performs approximate but inferior to IPOPT obviously. However, Figure 4 shows that IPRM needs less CPU time than IPOPT, which may be partially resulted from that the system (3.10) 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, it is not surprising that our implementation of Algorithm 1 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 (3.10) more efficiently.

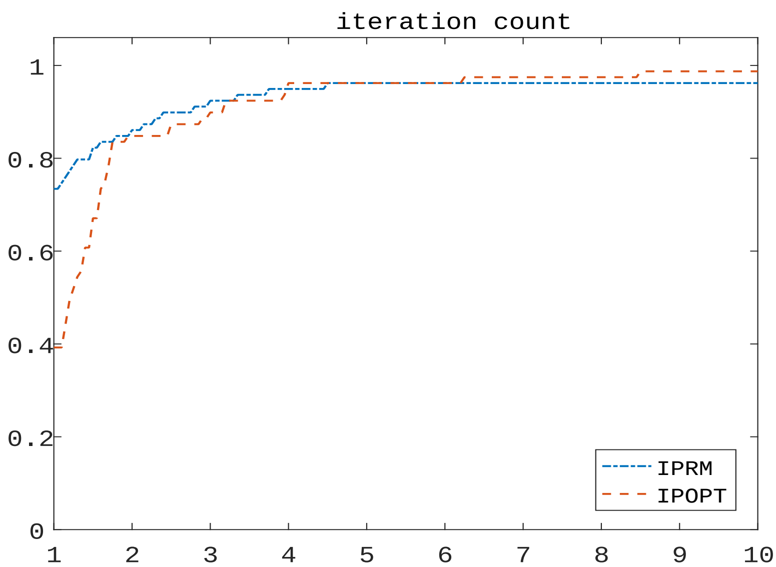

Figure 1: Performance plot for iteration count

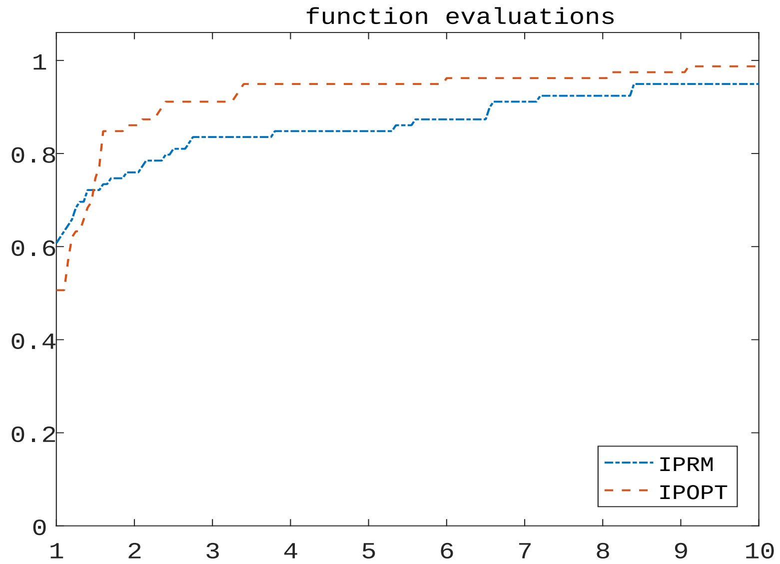

Figure 2: Performance plot for function evaluations

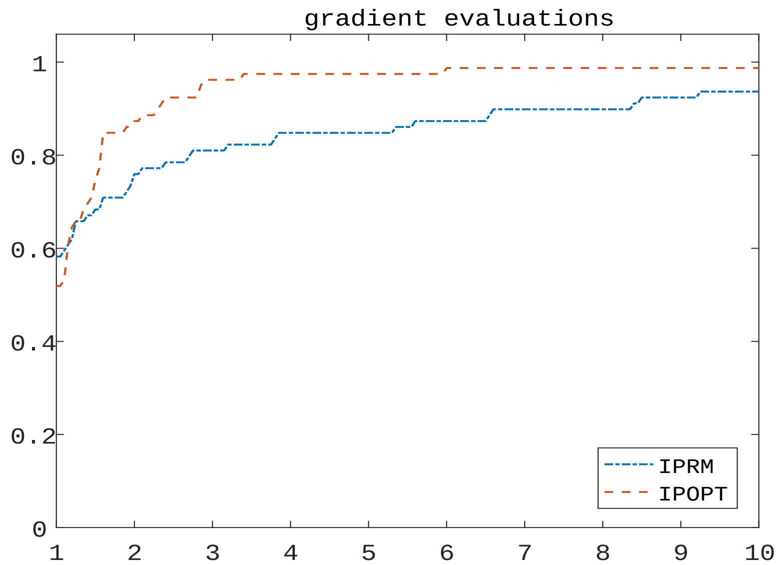

Figure 3: Performance plot for gradient evaluations

7. Conclusion

We present a novel primal-dual interior-point relaxation method with adaptively updating barrier for nonlinear programs in this paper. The method is based on solving a parametric equality constrained mini-max subproblem. It is of the interior-point variety, but does not require any primal or dual iterates to be interior, thus can circumvent the jamming difficulty resulted from the interior-point limitations. Since the update for barrier parameter is autonomous and adaptive, our method is capable of improving the ill conditioning of the classic primal-dual interior-point methods and averting the affect of the unappropriate initial selection of the barrier

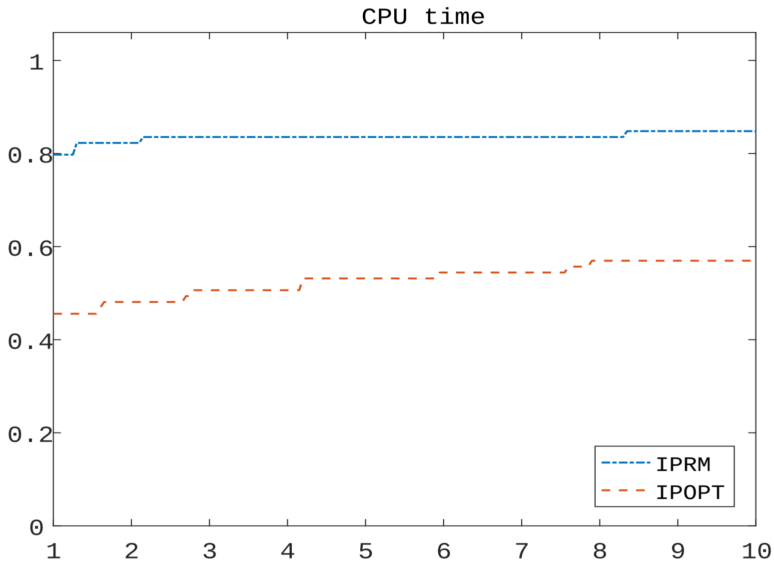

Figure 4: Performance plot for the CPU time

parameter. 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 CUTE collection show that our method is efficient.

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), 371–399.

[2] H.Y. Benson and D.F. Shanno, *Interior-point methods for nonconvex nonlinear programming: regularization and warmstarts,* Comput. Optim. Appl. 40 (2008), 143–189.

[3] H.Y. Benson, D.F. Shanno, R.J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: jamming and comparative numerical testing,* Math. Program., 99 (1), 35–48, 2004.

[4] 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.

[5] J.V. Burke, F.E. Curtis and H. Wang, *A sequential quadratic optimization algorithm with rapid infeasibility detection,* SIAM J. Optim., 24 (2014), 839–872.

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

[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), 149–185.

[8] R.H. Byrd, M. E. Hribar and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming,* SIAM J. Optim., 9 (1999), 877–900.

[9] R.H. Byrd, G. Liu and J. Nocedal, *On the local behaviour of an interior point method for nonlinear programming,* In D.F. Griffiths and D.J. Higham, editors, Numerical Analysis 1997, pp. 37-56, Addison-Wesley Longman, Reading, MA, 1997.

[10] R.H. Byrd, M. Marazzi and J. Nocedal, *On the convergence of Newton iterations to non-stationary points,* Math. Program., 99 (2004), 127–148.

[11] L.F. Chen and D. Goldfarb, *Interior-point ℓ_2-penalty methods for nonlinear programming with strong global convergence properties,* Math. Program., 108 (2006), 1–36.

[12] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *Testing a class of algorithms for solving minimization problems with simple bounds on the variables,* Math. Comp., 50 (1988), 399–430.

[13] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A),* Springer-Verlag, 1992.

[14] F.E. Curtis, *A penalty-interior-point algorithm for nonlinear constrained optimization,* Math. Program. Comput., 4 (2012), 181–209.

[15] F.E. Curtis, N.I.M. Gould, D.P. Robinson, *An interior-point trust-funnel algorithm for nonlinear optimization,* Math. Program., 161 (2017), 73–134.

[16] 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., 16 (2020), 1009–1035.

[17] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles,* Math. Program., 91 (2002), 201–213.

[18] 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.

[19] A. Forsgren and P.E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming,* SIAM J. Optim., 8 (1998), 1132–1152.

[20] E.M. Gertz and Ph. E. Gill, A primal-dual trust region algorithm for nonlinear optimization, *Math. Program.,* 100 (2004), 49–94.

[21] N.I.M. Gould, D. Orban and Ph.L. Toint, *An interior-point l_1-penalty method for nonlinear optimization,* Recent Developments in Numerical Analysis and Optimization, Proceedings of NAOIII 2014, Springer, Verlag, 134 (2015), 117–150.

[22] N.I.M. Gould and Ph.L. Toint, *Nonlinear programming without a penalty function or a filter,* Math. Program., 122 (2009), 155–196.

[23] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes,* Lecture Notes in Eco. and Math. Systems 187, Springer-Verlag, Berlin, New York, 1981.

[24] X.-W. Liu and Y.-H. Dai, *A globally convergent primal-dual interior-point relaxation method for nonlinear programs,* Math. Comput., 89 (2020), 1301–1329.

[25] X.-W. Liu and J. Sun, *A robust primal-dual interior point algorithm for nonlinear programs,* SIAM J. Optim., 14 (2004), 1163–1186.

[26] X.-W. Liu and Y.-X. Yuan, *A null-space primal-dual interior-point algorithm for nonlinear optimization with nice convergence properties,* Math. Program., 125 (2010), 163–193.

[27] X.-W. Liu and Y.-X. Yuan, *A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization,* SIAM J. Optim., 21 (2011), 545–571.

[28] J. Nocedal, F. Öztoprak and R.A. Waltz, *An interior point method for nonlinear programming with infeasibility detection capabilities,* Optim. Methods Softw. **4** (2014), 837–854.

[29] J. Nocedal and S. Wright, *Numerical Optimization,* Springer-Verlag New York, Inc., 1999.

[30] D.F. Shanno and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods,* Math. Program., 87 (2000), 303–316.

[31] W.Y. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming,* Springer, New York, 2006.

[32] M. Ulbrich, S. Ulbrich and L.N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming,* Math. Program., 100 (2004), 379–410.

[33] R.J. Vanderbei and D.F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming,* Comput. Optim. Appl., 13 (1999), 231–252.

[34] A. Wächter and L. T. Biegler, *Failure of global convergence for a class of interior point methods for nonlinear programming,* Math. Program., 88 (2000), 565–574.

[35] 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), 25–57.

[36] S.J. Wright, *Primal-Dual Interior-Point Methods,* SIAM Publications, Philadelphia, Pa, 1997.

[37] Y.Y. Ye, *Interior Point Algorithms: Theory and Analysis,* John Wiley & Sons, Inc., 1997.