Noname manuscript No.
(will be inserted by the editor)

A primal-dual modified log-barrier method for inequality constrained nonlinear optimization

Joshua Griffin · Riadh Omheni

Received: date / Accepted: date

Abstract We present a primal-dual modified log-barrier algorithm to solve inequality constrained nonlinear optimization problems. Basically, the algorithm is a Newton-like method applied to a perturbation of the optimality system that follows from a reformulation of the initial problem by introducing a modified log-barrier function to handle inequality constraints. The algorithm uses an outer/inner iteration scheme and the globalization is performed in the primal-dual space by means of a new primal-dual merit function. The robustness and efficiency of the algorithm is improved using quadratic extrapolation. The numerical performance of the new method is illustrated by comparing it with a primal-dual classical log-barrier method and two well-established interior-point solvers on two sets of problems from COPS and Hock-Schittkowski collections, including a set of problems that exhibits degeneracy.

Keywords Nonlinear programming · Inequality constrained optimization · Modified log-barrier method · Primal-dual method · Numerical tests

1 Introduction

We consider the following inequality constrained nonlinear programming problem

$$\text{minimize } f(x) \quad \text{subject to } \; c(x) \geq 0, \tag{1}$$

where $f(x)$ and $c(x) = (c_1(x), \ldots, c_m(x))^\top$ are twice continuously differentiable on $\mathbb{R}^n$. For the sake of simplicity, we omit the equality constraints that

Joshua Griffin
SAS Institute, Cary, NC, USA
E-mail: joshua.griffin@sas.com

Riadh Omheni
SAS Institute, Cary, NC, USA
E-mail: riadh.omheni@sas.com

can efficiently be handled using the sequential quadratic programming techniques [22,27] or primal-dual augmented Lagrangian methods [2].

The most straightforward way to handle the inequality constraints consists of transforming them into equality constraints by subtracting a vector of slack variables. This approach has been successfully used in building several solvers, e.g. KNITRO [8], LOQO [25], and IPOPT [27]. The main benefit of this transformation is finding an easy feasible starting point for interior-point based approaches. Another simple technique to convert inequality constraint to equality constraint is adding squared slack variables. Besides complicating the problems by adding equality constraints, this reformulation can cause difficulties and numerical instabilities [4,16].

The second approach for handling inequality constraints consists of placing them directly into a barrier function. The most common two examples of barrier functions are the log-barrier function and the inverse barrier function. The log-barrier is usually preferred because the logarithm increases so slowly as the boundary is approached. The classical log-barrier (CLB) method, originally introduced by Frish [14], transforms problem (1) to an unconstrained subproblem. Fiacco and McCormick [12] showed that the sequence of local minimizers of this subproblem converge to the local minimizer of problem (1) whenever the barrier parameter converges to zero. The main difficulty associated with the CLB method is finding a strictly feasible starting point. This task could have the same complexity as solving the original problem. Furthermore, Murray [19] showed that the Hessian matrix of the log-barrier function becomes increasingly ill-conditioned as the barrier parameter converges to zero. That is why this class of methods was quickly abandoned by practitioners. As an alternative to the CLB method, Polyak [23] introduced a new method based on a modified log-barrier (MLB) function. This method eliminates the most essential deficiencies associated to the CLB method while retaining its best features. Conn *et al.* [9] proposed a generalization of the MLB method of Polyak. Their method exhibits strong convergence properties. They proved that the barrier parameter is updated only a finite number of times and the algorithm requires only one inner iteration per outer iteration. The main drawback of this approach is that it requires shifting the starting point via an auxiliary problem after updating the barrier parameter. The authors have not provided any detailed numerical evidence that their approach is effective on general problems.

The advantages of the MLB method over CLB method are threefold. First, provided that the barrier parameter is sufficiently small, the sequence generated by the MLB method converges to a Karush-Kuhn Tucker point of (1), while keeping the barrier parameter constant and adjusting the Lagrange multiplier (see [23]). This nice feature allows to limit the effect of ill-conditioning associated to CLB method. Secondly, the MLB method naturally handles infeasibility while the CLB method requires a strictly feasible starting point in order for the logarithmic term to be well defined. To determine such a point, one can be led to solve a problem as difficult as (1). Thirdly, an interesting feature of the MLB is the ability of handling the equality constraints by treat-

ing them as range constraints. However, this transformation cannot be used by the CLB method since no strictly feasible point satisfying both inequalities could be found. The superiority of modified log-barrier function over classical one was illustrated in [6,17,21]. Note that these approaches are primal in the sense that the globalization in done only in the primal space.

In his seminal paper [23], Polyak noted that the modified barrier method is to the classical barrier method as the augmented Lagrangian to the classical quadratic penalty method. Motivated by this observation and the nice performance of the primal-dual augmented Lagrangian proposed in [2,3], the purpose of this paper is to propose a new primal-dual MLB algorithm for solving (1) and test whether the MLB method is superior to the CLB method in the primal-dual space. The main features of our algorithm compared to primal MLB methods are:

- The barrier parameter and the Lagrange multiplier estimate are updated depending on the reduction of the complementarity slackness. Our update rule for these parameters is motivated by the fact that our proposed algorithm could be reduced to the regularized Newton applied to the KKT conditions of problem (1). Hence, rapid convergence could be observed.
- Our method makes use of the outer/inner scheme, successfully used in [1, 2,3]. This allows to avoid performing inner iterations every iteration, and hence reducing the number of function evaluations.
- The control of iterates is performed in both primal and dual spaces, as long as "large" infeasibility is not detected. In particular, the globalization is done using a new primal-dual merit function.

In Section 2, we give a detailed description of our new primal-dual MLB method. This is followed by a brief description of the quadratic extrapolation in Section 3. The implementation details are given in Section 4. Numerical results showing the performance of our algorithm are presented in Section 5.

Notation For two real vectors x and z of the same length, the Euclidean scalar product is denoted by $x^\top z$. The associated norm is $\|x\| = (x^\top x)^{1/2}$. The positive part of a real number t is the function defined by $t^+ = \max\{t, 0\}$. We denote by $\nabla f(x) \in \mathbb{R}^n$ the gradient of f at x. The Jacobian matrix of c at x is denoted by $\nabla c(x)^\top$. It is an $m \times n$ matrix, whose i-th row is the vector $\nabla c_i(x)^\top$. We will use the standard notations of interior point methods: $C(x)$ and Z are diagonal matrices whose diagonal entries are given by the components of $c(x)$ and z, respectively. e is a vector of all ones. The Lagrangian function for problem (1) is defined by

$$\mathcal{L}(x, z) = f(x) - c(x)^\top z.$$

The gradient and the Hessian of $\mathcal{L}$ with respect to the primal variables are given by

$$\nabla_x \mathcal{L}(x, z) = \nabla f(x) - \nabla c(x) z \quad \text{and} \quad \nabla_x^2 \mathcal{L}(x, z) = \nabla^2 f(x) - \sum_{i=1}^{m} z_i \nabla^2 c_i(x).$$

The first order optimality conditions, also called Karush-Kuhn-Tucker (KKT) conditions, for problem (1) are given by

$$r_0(w) := \begin{pmatrix} \nabla f(x) - \nabla c(x)z \\ C(x)Ze \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad c(x) \geq 0 \quad \text{and} \quad z \geq 0. \tag{2}$$

2 The primal-dual modified log-barrier method

Polyak [23] proposed the following MLB function associated to problem (1)

$$\mathcal{M}_{\lambda,\mu}(x) := f(x) - \mu \sum_{i \in \mathcal{I}} \lambda_i \log\left(\frac{c_i(x)}{\mu} + 1\right), \tag{3}$$

where $\mathcal{I} = \{1, \ldots, m\}$, λ is a vector of positive estimates of the Lagrange multiplier and $\mu > 0$ is a barrier parameter. Polyak [23] motivated the MLB function by noting the equivalence between the constraints of Problem (1) and

$$\mu \log\left(\frac{c_i(x)}{\mu} + 1\right) \geq 0 \quad \text{for} \quad i \in \mathcal{I}.$$

The MLB method consists of finding a solution of (1) by solving a sequence of unconstrained minimization problems

$$\min_{x \in \mathbb{R}^n} \quad \mathcal{M}_{\lambda,\mu}(x). \tag{4}$$

The KKT conditions for problem (4) are

$$\nabla f(x) - \sum_{i \in \mathcal{I}} \frac{\mu \lambda_i}{c_i(x) + \mu} \nabla c_i(x) = 0.$$

By introducing $z_i = \dfrac{\mu \lambda_i}{c_i(x) + \mu}$, we get

$$r(w, \lambda, \mu) := \begin{pmatrix} \nabla f(x) - \nabla c(x)z \\ C(x)Ze + \mu(z - \lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{5}$$

These conditions appear as perturbation of the KKT conditions of the initial problem (1). The algorithm proposed in this paper is basically a Newton-like method applied to (5), while updating λ and μ to guarantee the global convergence. Our approach is more flexible than the CLB method and offers two possibilities to ensure the convergence to a point for which the KKT conditions (2) are satisfied: reduce the barrier parameter or update the Lagrange multiplier estimate. However, for the CLB method, only the barrier parameter should be driven to zero. We adopt the primal-dual scheme successfully proposed and used in [1,2,3].

2.1 Outer algorithm

Let k be the outer iteration index. Let $w_k = (x_k, z_k)$ be the current iterate, λ_k be an estimate of the Lagrange multiplier and μ_k be the barrier parameter. The minimization process starts by updating λ_k and μ_k, depending on the progress towards the complementarity slackness. More precisely, if the current primal-dual iterate satisfies

$$\|C(x_k)z_k\| \leq a\|C(x_{k-1})z_{k-1}\|, \tag{6}$$

where $a \in (0,1)$, then the Lagrange multiplier estimate is updated by setting $\lambda_{k+1} = z_k$ and the barrier parameter may be kept constant. However, if (6) is not satisfied then the barrier parameter is reduced and the Lagrange multiplier is kept unchanged to ensure that it does not behave too badly far from a solution. Inequality (6), used to measure the progress achieved by the current iterate, may be too demanding especially for nonlinear optimization. That is why we use a condition based on the non-monotonic decrease of the complementarity slackness. This leads us to consider the following inequality instead of (6)

$$\|C(x_k)z_k\| \leq a \max\left\{\|C(x_{i_j})z_{i_j}\| : (k-\ell)^+ \leq j \leq k\right\} + \zeta_k, \tag{7}$$

where $l \geq 1$, $\{i_k\}$ is a sequence whose elements are the iterations index at which $\lambda_{k+1} = z_k$ and $\{\zeta_k\}$ is a sequence converging to zero. Condition (7) is inspired from our recent work on primal-dual augmented Lagrangian methods for equality constrained problems [2].

After updating μ_k and λ_k, we compute a search direction by solving the following primal-dual linear system

$$\begin{pmatrix} \nabla^2_x \mathcal{L}(w_k) & -\nabla c(x_k) \\ Z_k \nabla c(x_k)^\top & C(x_k) + \mu_{k+1} I \end{pmatrix} \begin{pmatrix} d_x \\ d_z \end{pmatrix} = - \begin{pmatrix} \nabla_x \mathcal{L}(w_k) \\ Z_k C(x_k) e + \mu_{k+1}(z_k - \lambda_{k+1}) \end{pmatrix}. \tag{8}$$

This corresponds to the result of applying a Newton-like method to (5). Setting $\lambda_{k+1} = z_k$, the linear system (8) appears as a regularized Newton method applied to the KKT conditions of the initial problem (1). Hence a high rate of convergence could be expected, provided that the parameters are correctly updated and the Jacobian constraints is of full rank. Note that condition (7) was designed to make sure that the update $\lambda_{k+1} = z_k$ occurs as often as possible. For future reference, we define

$$J(w_k, \mu_{k+1}) := \begin{pmatrix} \nabla^2_x \mathcal{L}(w_k) & -\nabla c(x_k) \\ Z_k \nabla c(x_k)^\top & C(x_k) + \mu_{k+1} I \end{pmatrix}.$$

The linear system (8) can then be written as follows

$$J(w_k, \mu_{k+1})\ d = -r(w_k, \lambda_{k+1}, \mu_{k+1}).$$

After the step computation, the new trial iterate is given by

$$\hat{x}_k = x_k + \alpha_k^x d_k^x \quad \text{and} \quad \hat{z}_k = z_k + \alpha_k^z d_k^z,$$

where the step-lengths α_k^x and α_k^z are computed such that

$$\frac{c(x_k + \alpha_k^x d_k^x)}{\mu_{k+1}} + 1 > 0 \tag{9}$$

and

$$z_k + \alpha_k^z d_k^z > 0. \tag{10}$$

To decide whether the candidate iterate $\widehat{w}_k$ is selected, the following criterion is checked

$$\|r(\widehat{w}_k, \lambda_{k+1}, \mu_{k+1})\| \leq \varepsilon_k,$$

where $\{\varepsilon_k\}$ is a sequence that converges to zero. If this is not the case, we apply a sequence of inner iterations to find w_{k+1} satisfying

$$\|r(w_{k+1}, \lambda_{k+1}, \mu_{k+1})\| \leq \varepsilon_k.$$

We are now in a position to give the complete outer algorithm to solve (5). It is described as Algorithm 1.

Algorithm 1 Outer algorithm

1: (Initialization): Choose a starting point $w_0 = (x_0, z_0) \in \mathbb{R}^{n+p}$, an initial barrier parameter $\mu_0 > 0$, an initial estimate of the Lagrange multiplier $\lambda_0 > 0$ and three constants $a,\ \rho \in (0,1)$ and $\ell \in \mathbb{N}$. Set $k = 0$.
2: (Optimality check): If (2) is satisfied, then terminate; w_k is a first-order optimal point for (1).
3: (Measure progress towards complementarity slackness): Choose $\zeta_k \geq 0$ such that $\{\zeta_k\} \to 0$. If

$$\|C(x_k)z_k\| \leq a \max\left\{\|C(x_{i_j})z_{i_j}\| : (k-\ell)^+ \leq j \leq k\right\} + \zeta_k,$$

then go to Step 4. Otherwise, go to Step 5.
4: (Update Lagrange multiplier estimates): Choose $\mu_{k+1} \leq \mu_k$. Set $\lambda_{k+1} = z_k$, $i_{k+1} = k$ and go to Step 6.
5: (Reduce the barrier parameter): Choose $\mu_{k+1} \leq \rho\mu_k$. Set $\lambda_{k+1} = \lambda_k$ and $i_{k+1} = i_k$.
6: (Search direction computation): Compute a search direction $d_k = (d_k^x, d_k^z)$ by solving

$$J(w_k, \mu_{k+1})\, d_k = -r(w_k, \lambda_{k+1}, \mu_{k+1}). \tag{11}$$

7: (Step-length computation): Compute α_k^x and α_k^z such that (9) and (10) hold. Set $\hat{x}_k = x_k + \alpha_k^x d_k^x$, $\hat{z}_k = z_k + \alpha_k^z d_k^z$ and $\widehat{w}_k = (\hat{x}_x, \hat{z}_k)$.
8: (Inner iteration): Choose $\varepsilon_k > 0$ such that $\{\varepsilon_k\} \to 0$. If

$$\|r(\widehat{w}_k, \lambda_{k+1}, \mu_{k+1})\| \leq \varepsilon_k,$$

then set $w_{k+1} = \widehat{w}_k$. Otherwise, starting from $\widehat{w}_k$, apply a sequence of inner iterations to find w_{k+1} such that

$$\|r(w_{k+1}, \lambda_{k+1}, \mu_{k+1})\| \leq \varepsilon_k.$$

9: (k increment): Set $k \leftarrow k + 1$ and go to Step 2.

2.2 Inner algorithm

Our inner algorithm corresponds to the minimization of the new primal-dual merit function (12) with fixed values of the Lagrange multiplier estimate and the barrier parameter.

$$\Phi_{\lambda,\mu}(w) = \mathcal{M}_{\lambda,\mu}(x) + \nu\Psi_{\lambda,\mu}(w), \tag{12}$$

where $\nu \geq 0$ is a scaling parameter and

$$\Psi_{\lambda,\mu}(w) = \mu \sum_{i\in\mathcal{I}} \left(\frac{(c_i(x)+\mu)z_i}{\mu\lambda_i} - \log\left(\frac{(c_i(x)+\mu)z_i}{\mu\lambda_i} \right) \right).$$

Note that this merit function is inspired by the merit function proposed by Forsgren and Gill [13]. It is easy to check that w is a stationary point for $\Phi_{\lambda,\mu}$ if and only if $r(w,\lambda,\mu) = 0$. The inner algorithm is stated as Algorithm 2.

Algorithm 2 Inner algorithm

1: (Initialization): Choose $\eta \in (0,1)$ and set $i = 0$, $w_{k,i} = \widehat{w}_k$, $\lambda = \lambda_{k+1}$ and $\mu = \mu_{k+1}$.
2: (Search direction computation): Solve the linear system

$$J(w_{k,i},\mu)\, d = -r(w_{k,i},\lambda,\mu). \tag{13}$$

3: (Backtracking): Compute the step-length $\alpha_{k,i}$.
   (a) Set $\bar{\alpha} = 1$.
   (b) (Feasibility check): Find $\tilde{\alpha} \in (0,\bar{\alpha}]$ such that (9) and (10) hold.
   (c) (Armijo condition): Find $\alpha \in (0,\tilde{\alpha}]$ such that the Armijo condition is satisfied

$$\Phi_{\lambda,\mu}(w_{k,i}+\alpha d) \leq \Phi_{\lambda,\mu}(w_{k,i}) + \alpha\eta\nabla\Phi_{\lambda,\mu}(w_{k,i})^\top d_{k,i}, \tag{14}$$

   (d) If (9) and (10) hold at $w_{k,i}+\alpha d_{k,i}$ then set $\alpha_{k,i} = \alpha$ and go to step 4. Otherwise, set $\bar{\alpha} = \alpha$ and go to Step 3b.

4: (New inner iterate): Set $w_{k,i+1} = w_{k,i} + \alpha_{k,i} d_{k,i}$.
5: (Stopping condition): If $\|r(w_{k,i+1},\lambda,\mu)\| \leq \varepsilon_k$, then set $w_{k+1} = w_{k,i+1}$ and return to Algorithm 1. Else set $i \leftarrow i+1$ and go to Step 2.

Because the outer algorithm is not using a merit function, we allow a different step-length in the primal variable, x, from the dual variable, z. While this flexibility to choose the step-length should not impact the convergence of the algorithm, the numerical results show that this strategy has a positive impact on the performance of our algorithm. However, given the Armijo condition (14), Algorithm 2 uses only one step-length to compute the new iterate.

The following lemma states that the search direction computed at Step 2 of Algorithm 2 is a descent direction for the merit function.

Lemma 1 *Let $d = (d_x, d_z)$ denote the solution of (13). The directional derivative of $\Phi_{\lambda,\mu}$ at w is given by*

$$\nabla\Phi_{\lambda,\mu}(w)^\top d = -d_x^\top K(w,\mu)d_x - \| \left(C(x)+\mu I\right)^{-\frac{1}{2}} Z^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} \left(\mu\lambda - \left(C(x)+\mu I\right) z\right) \|^2,$$

where $K(w,\mu) := \nabla_x^2\mathcal{L}(w) + \nabla c(x)\left(C(x)+\mu I\right)^{-1}\nabla c(x)^\top$. In addition if $K(w,\mu)$ is positive definite and r, defined by (5), is nonzero, then d is a descent direction of $\Phi_{\lambda,\mu}$ at w.

Proof To simplify the notation, the functions $\Phi_{\lambda,\mu}$, $\mathcal{M}_{\lambda,\mu}$, and $\Psi_{\lambda,\mu}$ will be denoted by Φ, $\mathcal{M}$, and Ψ, respectively. The directional derivative of Φ is given by

$$\nabla\Phi(w)^\top d = \nabla_x\mathcal{M}(x)^\top d_x + \nu\nabla\Psi(w)^\top d.$$

The linear system (13) gives

$$\nabla_x^2\mathcal{L}(w)d_x - \nabla c(x)d_z = -\nabla f(x) + \nabla c(x)z, \tag{15}$$
$$Z\nabla c(x)^\top d_x + \left(C(x)+\mu I\right) d_z = -ZC(x)e - \mu(z-\lambda). \tag{16}$$

We have

$$\nabla_x\mathcal{M}(x)^\top d_x = \nabla f(x)^\top d_x - \mu e^\top \left(C(x)+\mu I\right)^{-1}\Lambda\nabla c(x)^\top d_x. \tag{17}$$

Or using (15), we get

$$\nabla f(x)^\top d_x = -d_x^\top\nabla_x^2\mathcal{L}(w)d_x + (z+d_z)^\top\nabla c(x)^\top d_x.$$

Using this, equation (17) gives

$$\nabla_x\mathcal{M}(x)^\top d_x = -d_x^\top\nabla_x^2\mathcal{L}(w)d_x + \left(z + d_z - \mu\Lambda\left(C(x)+\mu I\right)^{-1}e\right)^\top \nabla c(x)^\top d_x.$$

Using this and (16), we find

$$\nabla_x\mathcal{M}(x)^\top d_x = -d_x^\top\left(\nabla_x^2\mathcal{L}(w) + \nabla c(x)Z\big(C(x)+\mu I\big)^{-1}\nabla c(x)^\top\right)d_x.$$

We now compute the directional derivative of $\Psi_{\lambda,\mu}$. Simple calculations give

$$\begin{aligned}\nabla\Psi(w)^\top d &= \left(\nabla c(x)Z\Lambda^{-1}e - \mu\nabla c(x)\left(C(x)+\mu I\right)^{-1}e\right)^\top d_x + \left(\left(C(x)+\mu I\right)\Lambda^{-1}e - \mu Z^{-1}e\right)^\top d_z \\ &= -\left(Z^{-1}\left(C(x)+\mu I\right)^{-1}\Lambda^{-1}\left(\mu\lambda - \left(C(x)+\mu\right)z\right)\right)^\top \left(Z\nabla c(x)^\top d_x + \left(C(x)+\mu I\right) d_z\right).\end{aligned}$$

Recalling (16) we obtain

$$\nabla\Psi(w)^\top d = -\| Z^{-\frac{1}{2}}\left(C(x)+\mu I\right)^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}\left(Z\nabla c(x)^\top d_x + \left(C(x)+\mu I\right) d_z\right) \|^2.$$

Our assumption on $K(w,\mu)$ implies that $\nabla\Phi(w)^\top d \leq 0$. The equality holds if and only if $d_x = 0$ and $Z\nabla c(x)^\top d_x + \left(C(x)+\mu I\right) d_z = 0$, and thus $d = 0$. Using (15) and (16), this implies $r(w,\lambda,\mu) = 0$. This concludes the proof.

3 Handling infeasibility

In order for the logarithmic term of the MLB function (3) to be well defined, the following inequality should be satisfied

$$\frac{c_i(x)}{\mu} + 1 > 0, \quad \forall\, i \in \mathcal{I}. \tag{18}$$

Due to the non-convexity, inequality (18) can restrict how quickly the barrier parameter can be reduced. This situation mainly occurs when condition (7) is not satisfied and the current values of the constraints do not allow the barrier parameter to be reduced by a large factor. Hence a little progress could be expected and the algorithm may get stuck. This is a common issue associated with the MLB and is reported in [7,9,21]. To cope with this situation, Conn *et al.* [9] considered solving an auxiliary problem to determine a feasible starting point after the penalty parameter update. However, the authors did not provide any numerical evidence that their approach is efficient. On the other hand, Breitfeld and Shanno [7] and Nash *et al* [21] chose to use the quadratic extrapolation, originally introduced by Ben-Tal and Zibulevsky [5]. This idea can also be applied in our context. We modify the definition of the function $\mathcal{M}$ by distinguishing two cases based on the infeasibility of the current iterate. More precisely, we use

$$\mathcal{M}(x) = f(x) - \mu \sum_{i \in \mathcal{I}} \lambda_i \begin{cases} \log\left(\dfrac{c_i(x)}{\mu} + 1\right) & \text{if } c_i(x) \geq -\beta\mu, \quad \text{(19a)} \\ \dfrac{1}{2} q^a (c_i(x))^2 + q^b c_i(x) + q^c & \text{if } c_i(x) < -\beta\mu, \quad \text{(19b)} \end{cases}$$

where $0 \leq \beta < 1$ and the coefficients q^a, q^b, and q^c are given by

$$q^a = \frac{-1}{(\mu(1-\beta))^2}, \quad q^b = \frac{1-\beta}{\mu(1-\beta)^2}, \quad \text{and} \quad q^c = \frac{\beta(2-3\beta)}{2(1-\beta)^2} + \log(1-\beta).$$

The primal quadratic extrapolation can be seen as follows. Whenever a "large" infeasibility is detected (using (19b)), the algorithm switches to use the quadratic extrapolation term for the corresponding constraint. In this manner, the quadratic approximations could be seen as a penalty rather than a barrier. In other words, if the i-th constraint becomes increasingly infeasible, we penalize it using a quadratic term. Let $\mathcal{B}$ and $\mathcal{Q}$ be two non-intersecting index sets denoting the set of constraints satisfying (19a) and (19b), respectively. The linear system (13) will have the following form

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(w) & -\nabla c_{\mathcal{B}}(x) & -\nabla c_{\mathcal{Q}}(x) \\ Z_{\mathcal{B}} \nabla c_{\mathcal{B}}(x)^\top & C_{\mathcal{B}}(x) + \mu I & 0 \\ \mu q^a \Lambda_{\mathcal{Q}} \nabla c_{\mathcal{Q}}(x)^\top & 0 & -I \end{pmatrix} \begin{pmatrix} d_x \\ d_{z_{\mathcal{B}}} \\ d_{z_{\mathcal{Q}}} \end{pmatrix} = - \begin{pmatrix} \nabla_x \mathcal{L}(w) \\ C_{\mathcal{B}}(x) Z_{\mathcal{B}} e + \mu(z_{\mathcal{B}} - \lambda_{\mathcal{B}}) \\ \mu \Lambda_{\mathcal{Q}}(q^a c_{\mathcal{Q}}(x) + q^b e) - z_{\mathcal{Q}} \end{pmatrix},$$

where $\Lambda_{\mathcal{B}} = \operatorname{diag}(\lambda_{\mathcal{B}})$. During inner iterations, we use the following merit function

$$\Phi(w) = \mathcal{M}(x) + \nu\mu \sum_{i \in \mathcal{B}} \left(\frac{(c_i(x) + \mu) z_i}{\mu \lambda_i} - \log\left(\frac{(c_i(x) + \mu) z_i}{\mu \lambda_i} \right) \right).$$

Note the use of $\mathcal{B}$ instead of $\mathcal{I}$ in the second part of this definition. This means that when the inequality (19b) is satisfied, the algorithm switches to use the primal quadratic term associated with the corresponding i-th constraint. Let us assume that we solve an optimization problem with only one constraint, c_1, and that the inequality (19b) is satisfied ($c_1(x) < -\beta\mu$). In this case, we have $\mathcal{B} = \emptyset$ and $\mathcal{Q} = \{1\}$. The merit function is then defined by

$$\Phi(w) = \mathcal{M}(x) = f(x) - \mu\lambda_1 \left(\frac{1}{2} q^a (c_1(x))^2 + q^b c_1(x) + q^c \right),$$

which is a primal merit function.

4 Implementation

In this section we give the implementation details of the algorithm proposed in this paper. Our implementation is written in C and called PDMLBM, which stands for primal-dual Modified Log-Barrier Method.

Starting values An important advantage of the MLB method over classical one is that it does not require the starting point, x_0, to be feasible for problem (1). That is why the initial point provided by the user is always used. The initial value for the dual variable is $z_0 = 1$. The initial barrier parameter μ_0 is chosen as

$$\mu_0 = \max_{j \in \mathcal{I}} (-c_j(x_0) + 0.1,\ 0.1). \tag{20}$$

This guarantees that $\dfrac{c(x_0)}{\mu_0} + e \geq 0$, although sometimes at the cost of a large initial barrier parameter. At Step 1 of Algorithm 1, we set $a = 0.9$ and $l = 2$.

Update of the barrier parameter At Step 5 of Algorithm 1, the barrier parameter is updated by setting $\rho = 0.2$. When Step 4 is executed, we also choose to reduce the barrier parameter at a slower rate by setting $\mu_{k+1} = 0.5\mu_k$. This gives better performance than keeping it constant.

Quadratic extrapolation We use $\beta = 0.9$. In our implementation, the quadratic extrapolation is enabled only if the primal step-length gets very small preventing the algorithm from making progress to the local solution. The version of the code using the quadratic extrapolation will be referred as PDMLBM-QE.

Solution of the linear system The solutions of the linear systems (11) and (13) are computed by factorizing the following symmetric matrix

$$J_S(w, \mu) := \begin{pmatrix} \nabla_x^2 \mathcal{L}(w) + \delta I & \nabla c_{\mathcal{B}}(x) & \nabla c_{\mathcal{Q}}(x) \\ \nabla c_{\mathcal{B}}(x)^\top & -Z_{\mathcal{B}}{}^{-1}(C_{\mathcal{B}}(x) + \mu I) & 0 \\ \nabla c_{\mathcal{Q}}(x)^\top & 0 & \frac{1}{\mu q^a} \Lambda_{\mathcal{Q}}{}^{-1} \end{pmatrix}.$$

The regularization term, δ, is used to ensure that $J_S(w, \mu)$ has the correct inertia. A mechanism for updating δ can be found in [27, Algorithm IC]. Note that the inertia control is done using the LDL factorization.

Step-length selection At step 7 of Algorithm 1, we apply the fraction to the boundary rule to compute α_k^x and α_k^z that satisfy (9) and (10). More precisely, we compute

$$\alpha_k^x := \max\{\alpha \in (0,1] : \ \frac{c(x_k + \alpha d_k^x)}{\mu_k} + 1 \geq (1-\gamma)(\frac{c(x_k)}{\mu_k} + 1)\}, \quad \text{and}$$
$$\alpha_k^z := \max\{\alpha \in (0,1] : \ z_k + \alpha d_k^z \geq (1-\gamma) z_k\}$$

with $\gamma = 0.995$. At step 3 of Algorithm 2, a step-length satisfying (9), (10), and (14) is computed. We start by linearizing the constraints to get an initial estimate of the step-length and then proceed with a backtracking line search. Murray and Wright [20] proposed a specialized line search that is beneficial for the CLB method. It uses a logarithmic search direction function to interpolate for α. However, Breitfeld and Shanno [6] observed that the quadratic or cubic interpolation work better for primal MLB method and outperforms Murray and Wright's line search. That is why we chose to use quadratic and cubic interpolations.

Stopping conditions The algorithm terminates, declaring convergence to an optimal solution of (1) whenever $\|r_0(w)\| \leq 10^{-6}$. This condition is tested at the beginning of each inner and outer iteration. For the sequence $\{\varepsilon_k\}$, we chose

$$\varepsilon_k = 0.9 \max\left\{\|r(w_j, \lambda_j, \mu_j)\| : (k-4)^+ \leq j \leq k\right\} + 10\mu_{k+1}.$$

The convergence of $\{\varepsilon_k\}$ to zero is shown by simply applying [2, Lemma 3.1].

Primal-dual CLB method There is no available implementation of a primal-dual CLB method. That is why we also implemented a CLB method based on various works (see e.g. [22,26]). This algorithm will be referred as PDCLBM. Note finally that this implementation takes advantage from all the enhancements of PDMLBM (outer/inner scheme, relaxed stopping condition for inner iterations, inertia control, ...). Here, we give a brief description of the main components of the PDCLBM implementation. The linear system solved at outer and inner iterations has the following form

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(w) & \nabla c(x) \\ \nabla c(x)^\top & -Z^{-1}C(x) \end{pmatrix} \begin{pmatrix} d_x \\ -d_z \end{pmatrix} = - \begin{pmatrix} \nabla_x \mathcal{L}(w) \\ c(x) - \mu Z^{-1} e \end{pmatrix}. \tag{21}$$

As a merit function, we use the primal-dual function

$$\Phi(w) = f(x) - \mu \sum_{i \in \mathcal{I}} \log(c_i(x)) + c(x)^\top z - \mu \sum_{i \in \mathcal{I}} \log(c_i(x) z_i).$$

Finally, for the barrier parameter update, we tried different rules [15]. The best results, in terms of efficiency and robustness, were obtained by decreasing the barrier parameter every iteration by a factor of 0.5.

5 Numerical experiments

In this section, we evaluate the performance of the proposed primal-dual algorithm on a set of 105 problems from COPS [11] and Hock-Schittkowski [18] collections. We consider all problems that can be written under the form (1). The models are written in the OPTMODEL modeling language [24]. This first collection will be referred as *standard*. We create a second collection by manipulating the OPTMODEL models and adding the following constraint

$$-c_1(x)^2 \leq 0. \tag{22}$$

Note that this transformation does not have any impact on the feasible region but makes every problem degenerate since the Mangasarian-Fromovitz constraint qualification (MFCQ) is not satisfied at all solution points. This collection will be referred as *degenerate*. All the comparisons presented in this section make use of the Dolan-Moré performance profiles [10]. For $\tau \geq 0$, $\rho_s(\tau)$ is the fraction of problems for which the performance of a given solver is within 2τ of the best one. Note finally that the numerical results have been obtained on a 64-bit 2.50Ghz Intel® Xeon® CPU E7-4880 machine with 8 processors and 128 GB RAM.

5.1 Standard problems

We start our numerical study by showing the impact of the quadratic extrapolation on the performance of our primal-dual MLB method. We compare both versions of Algorithm 1 on 105 problems from the standard collection. Figure 1

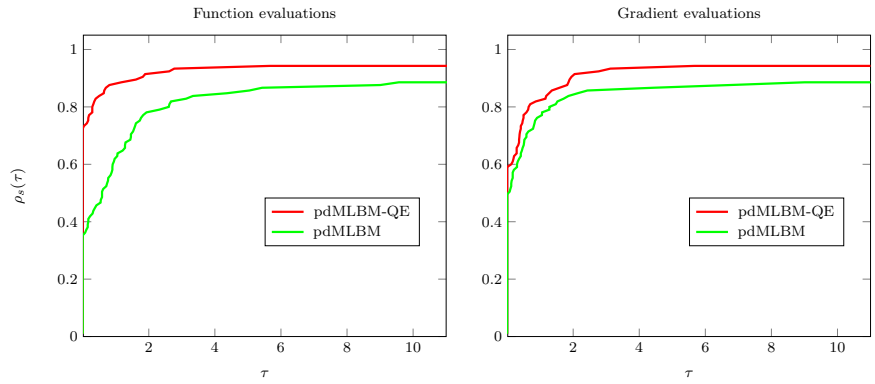

Fig. 1 Comparing PDMLBM-QE and PDMLBM in terms of number of function and gradient evaluations on 105 problems from the standard collection.

summarizes the performance of PDMLBM and PDMLBM-QE in terms of function and gradient evaluations. The result shows that the quadratic extrapolation

improves both the efficiency and robustness of the method. This improvement is more important in terms of function evaluations. In fact, on 73% of the problems, the number of function evaluations required by PDMLBM-QE is less than that required by PDMLBM. This gain is due to the fact that PDMLBM has to perform a backtracking to satisfy inequality (9) for the well-definedness of the MLB term. However, for PDMLBM-QE, whenever inequality (9) is not satisfied, the algorithm switches to use the quadratic term. Hence, no backtracking is required and the unit step-length could be chosen. In addition, as explained in Section 3, the quadratic extrapolation allows to avoid that the algorithm gets stuck, leading to an improvement in terms of robustness. PDMLBM-QE solves 7 more problems than PDMLBM. In terms of gradient evaluations, the right plot in Figure 1 shows that the gap between both methods is tighter but again PDMLBM-QE seems to require fewer gradient evaluations. It should be noted that the quadratic extrapolation was invoked by PDMLBM-QE to solve 36% of test problems.

As mentioned earlier, one of the advantages of the MLB method is that it naturally handles infeasibility by choosing a "large" initial barrier parameter. However, since the quadratic extrapolation is added to our MLB method, any choice of the initial barrier parameter (typically $\mu_0 = 0.1$) could be used. However, from our numerical results, we found that this strategy is beneficial only in some cases but overall PDMLBM-QE has better performance when using the initialization (20). This implies that it is beneficial that the control of iterates is, whenever possible, done in the primal-dual space before switching to the primal space and use the quadratic extrapolation whenever "large" infeasibility is encountered. Note that Nash *et al.* [21] found that better results were obtained by setting the initial barrier parameter to relatively small value, contrary to our observation.

Next, we compare in Figure 2 the performance of PDMLBM-QE and the primal-dual CLB method, PDCLBM. Note that the latter assumes a strictly feasible initial point with respect to the inequality constraint. That is why we only consider 62 problems from the standard collection for which the starting point is strictly feasible. As for the primal case presented in [21], Figure 2 confirms the superiority of the MLB method over the classical one on the primal-dual space. The gain is more important in terms of function evaluations. In fact, efficiency of PDMLBM-QE is 63% while that of PDCLBM does not exceed 40%. In terms of robustness, both algorithms solve 59 problems.

We also present a comparison between our method, PDMLBM-QE, and two well-established software, SPDOPT [2,3] and IPOPT [27]. Both solvers convert the inequality constraints to equality constraints using slack variables. SPDOPT applies a primal-dual augmented Lagrangian interior-point algorithm while IPOPT implements a line-search filter method. Both solvers were run using their default options, except for the scaling procedure. All runs are done without scaling to have a similar stopping test.

Figure 3 summarizes the performance of the three algorithms on the standard collection. The results indicate that PDMLBM-QE seems to perform equivalently to IPOPT and SPDOPT. In terms of robustness, IPOPT is the most robust

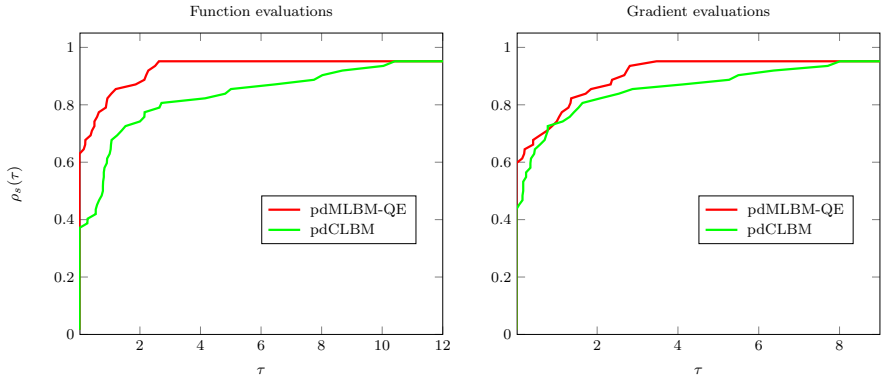

Fig. 2 Comparing PDMLBM-QE and PDCLBM in terms of number of function and gradient evaluations on 62 problems from the standard collection.

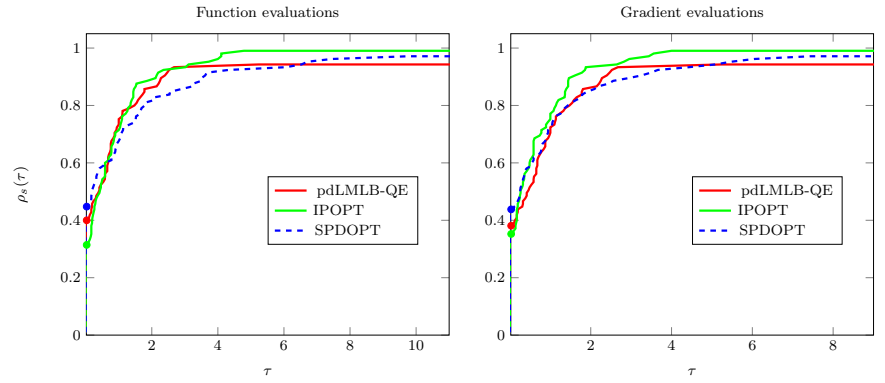

Fig. 3 Comparing PDMLBM-QE, IPOPT and SPDOPT in terms of number of function and gradient evaluations on 105 problems from the standard collection.

since it solves all problems but one. While PDMLBM-QE and SPDOPT solve 100 and 102 problems, respectively.

5.2 Degenerate problems

We conclude this numerical study by showing the performance of our new method on a class of degenerate problems. While the transformation (22) creates a particular type of degeneracy, the main purpose of the comparison presented in this subsection is to get an initial idea about the robustness and efficiency of our algorithm when the MFCQ is not satisfied. This is motivated by the fact that PDMLBM-QE introduces a natural regularization when solving the linear system. The performances of PDMLBM-QE, IPOPT, and SPDOPT are shown in Figure 4. The plots indicate that PDMLBM-QE clearly outperforms

IPOPT and SPDOPT when comparing function and gradient evaluations. In fact, efficiency of PDMLBM-QE is about 58% while that of IPOPT and SPDOPT does not exceed 30%. This is mostly likely due to the natural regularization introduced by the MLB term and the quadratic extrapolation. After adding the constraint (22) to Problem (1), the Jacobian of the constraints of the new problem becomes rank deficient. If the (2,2)-block of the KKT matrix is not correctly regularized, numerical difficulties could be observed while solving the linear system. When using our approach to solve this degenerate problem, the search direction is computed by solving a linear system whose coefficient matrix is of the form

$$\left(\begin{array}{c|ccccc} \nabla^2_x \mathcal{L}(w) & \nabla c_1(x) & \nabla c_2(x) & \dots & \nabla c_m(x) & 2c_1(x)\nabla c_1(x) \\ \hline \nabla c_1(x)^\top & -\dfrac{c_1(x)+\mu}{z_1} & 0 & \dots & 0 & 0 \\ \nabla c_2(x)^\top & 0 & -\dfrac{\mu(1-\beta)^2}{\lambda_2} & \vdots & 0 & 0 \\ \vdots & \vdots & \dots & \ddots & 0 & 0 \\ \nabla c_m(x)^\top & 0 & 0 & 0 & -\dfrac{\mu(1-\beta)^2}{\lambda_m} & 0 \\ 2c_1(x)\nabla c_1(x)^\top & 0 & 0 & 0 & 0 & -\dfrac{c_1(x)^2+\mu}{z_{m+1}} \end{array}\right).$$

Without loss of generality, we assume that constraints c_1 and (22) satisfy (19a) while constraints c_2, c_3, …, c_m satisfy (19b). When debugging some examples, we observe that our algorithm generates bounded Lagrange multipliers vector, z. Recalling the update rule used at Steps 4 and 5 of Algorithm 1, the estimate of the Lagrange multiplier, λ, is also bounded. This implies that the regularization terms in the (2,2)-block of the KKT matrix do not vanish. This has a positive impact on the KKT matrix and hence on the solution of the linear system. However, when using the CLB method, we observe that the Lagrange multipliers have tendency to grow very fast. This implies that the regularization term associated to the ith constraint, $\frac{c_i(x)}{z_i}$, gets closer to zero very quickly. This leads to some numerical difficulties while solving the linear system (21). Note that SPDOPT also solves a linear system with a natural regularization. But from Figure 4, it seems that the regularization used by PDMLBM-QE is more suitable for solving the kind of degeneracy created by adding the constraint (22).

Note finally that PDMLBM-QE has almost the same rate of robustness when solving problems from the standard or degenerate collections, while IPOPT and SPDOPT seems to be affected by the transformation (22).

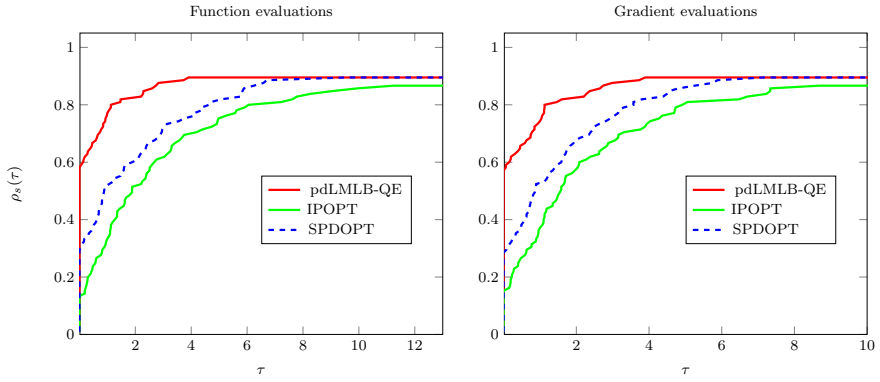

Fig. 4 Comparing PDMLBM-QE, IPOPT and SPDOPT in terms of number of function and gradient evaluations on 105 problems from the degenerate collection.

6 Conclusion

We have presented a new primal-dual MLB method for solving inequality constrained nonlinear optimization problems. The behavior of this new method has been improved by extrapolating the modified logarithmic term with quadratic approximations. To our knowledge, this is the first MLB method developed in the primal-dual space. While the theoretical analysis of this new algorithm has to be completed, the numerical study presented in this paper shows that our new method has favorable performance. In fact, our algorithm clearly outperforms the primal-dual CLB method and has a competitive behavior when compared to two well-established solvers on a collection of 105 problems. On a class of degenerate problems, our primal-dual MLB algorithm exhibits superior behavior.

We may be able to improve the performance of our method by using a better alternative in the primal-dual space to the quadratic extrapolation. Additionally, in some cases, we observe that the barrier parameter has a tendency to converge quickly to zero which creates difficulties to our algorithm to converge. One possible idea would be to increase the barrier parameter during inner iterations. This kind of strategy has been successfully used in the context of primal-dual augmented Lagrangian methods [3]. We will look into these improvements and the theoretical analysis as part of our future work.

References

1. Armand, P., Benoist, J., Omheni, R., Pateloup, V.: Study of a primal-dual algorithm for equality constrained minimization. Computational Optimization and Applications **59**(3), 405–433 (2014)
2. Armand, P., Omheni, R.: A globally and quadratically convergent primal-dual augmented lagrangian algorithm for equality constrained optimization. Optimization Methods and Software **32**(1), 1–21 (2017)

3. Armand, P., Omheni, R.: A mixed logarithmic barrier-augmented lagrangian method for nonlinear optimization. J. Optim. Theory Appl. **173**(2), 523–547 (2017)
4. Armand, P., Orban, D.: The squared slacks transformation in nonlinear programming. Sultan Qaboos University Journal for Science [SQUJS] **17**(1) (2012)
5. Ben-Tal, A., Zibulevsky, M.: Penalty/barrier multiplier methods for convex programming problems. SIAM Journal on Optimization **7**(2), 347–366 (1997)
6. Breitfeld, M.G., Shanno, D.F.: Preliminary computational experience with modified log-barrier functions for large-scale nonlinear programming. In: W.W. Hager, D.W. Hearn, P.M. Pardalos (eds.) Large Scale Optimization: State of the Art, pp. 45–67. Springer US, Boston, MA (1994)
7. Breitfeld, M.G., Shanno, D.F.: Computational experience with penalty-barrier methods for nonlinear programming. Ann. Oper. Res. **62**, 439–463 (1996). Interior point methods in mathematical programming
8. Byrd, R.H., Gilbert, J.C., Nocedal, J.: A trust region method based on interior point techniques for nonlinear programming. Math. Program. **89**(1, Ser. A), 149–185 (2000)
9. Conn, A.R., Gould, N., Toint, P.L.: A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. MATH. OF COMPUTATION **66**, 261–288 (1992)
10. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2, Ser. A), 201–213 (2002)
11. Dolan, E.D., Moré, J.J., Munson, T.S.: Benchmarking optimization software with COPS 3.0. Tech. rep., Argonne National Laboratory (2004)
12. Fiacco, A.V., McCormick, G.P.: Nonlinear programming: Sequential unconstrained minimization techniques. John Wiley and Sons, Inc., New York-London-Sydney (1968)
13. Forsgren, A., Gill, P.E.: Primal-dual interior methods for nonconvex nonlinear programming. SIAM J. Optim. **8**(4), 1132–1152 (electronic) (1998)
14. Frisch, K.R.: The logarithmic potential method of convex programming. Memorandum, University Institute of Economics, Oslo (1955)
15. Gay, D.M., Overton, M.L., Wright, M.A.H.: A primal-dual interior method for nonconvex nonlinear programming. In: Advances in nonlinear programming (Beijing, 1996), *Appl. Optim.*, vol. 14, pp. 31–56. Kluwer Acad. Publ., Dordrecht (1998)
16. Gill, P.E., Murray, W., Wright, M.H.: Practical optimization. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London (1981)
17. Goldfarb, D., Polyak, R., Scheinberg, K., Yuzefovich, I.: A modified barrier-augmented Lagrangian method for constrained minimization. Comput. Optim. Appl. **14**(1), 55–74 (1999)
18. Hock, W., Schittkowski, K.: Test examples for nonlinear programming codes, *Lecture Notes in Economics and Mathematical Systems*, vol. 187. Springer-Verlag, Berlin-New York (1981)
19. Murray, W.: Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. J. Optimization Theory Appl. **7**, 189–196 (1971)
20. Murray, W., Wright, M.H.: Line search procedures for the logarithmic barrier function. SIAM Journal on Optimization **4**(2), 229–246 (1994)
21. Nash, S.G., Polyak, R., Sofer, A.: A numerical comparison of barrier and modified barrier methods for large-scale bound-constrained optimization. In: Large scale optimization (Gainesville, FL, 1993), pp. 319–338. Kluwer Acad. Publ., Dordrecht (1994)
22. Nocedal, J., Wright, S.J.: Numerical optimization, second edn. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006)
23. Polyak, R.: Modified barrier functions (theory and methods). Math. Programming **54**(2, Ser. A), 177–222 (1992)
24. SAS Institute: The optmodel procedure. In: SAS/OR 14.3 Users Guide: Mathematical Programming, pp. 23–189. (2017)
25. Shanno, D.F., Vanderbei, R.J.: Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods. Math. Program. **87**(2, Ser. B), 303–316 (2000). Studies in algorithmic optimization
26. Silva, R., Soares, J., Vicente, L.N.: Local analysis of the feasible primal-dual interior-point method. Comput. Optim. Appl. **40**(1), 41–57 (2008)

27. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1, Ser. A), 25–57 (2006)