On the solution of linearly constrained optimization problems by means of barrier algorithms

E. G. Birgin[1] · J. L. Gardenghi[2] · J. M. Martínez[3] · S. A. Santos[3]

Abstract

Many practical problems require the solution of large-scale constrained optimization problems for which preserving feasibility is a key issue, and the evaluation of the objective function is very expensive. In these cases it is mandatory to start with a feasible approximation of the solution, the obtention of which should not require objective function evaluations. The necessity of solving this type of problems motivated us to revisit the classical barrier approach for nonlinear optimization, providing a careful implementation of a modern version of this method. This is the main objective of the present paper. For completeness, we provide global convergence results and comparative numerical experiments with one of the state-of-the-art interior-point solvers for continuous optimization.

Keywords Linearly constrained optimization · Feasible barrier methods · Convergence · Numerical experiments

Mathematics Subject Classification 49M37 · 65K05 · 90C30 · 90C51

✉ E. G. Birgin
egbirgin@ime.usp.br

J. L. Gardenghi
john.gardenghi@unb.br

J. M. Martínez
martinez@ime.unicamp.br

S. A. Santos
sandra@ime.unicamp.br

[1] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, SP, Brazil

[2] Faculty UnB Gama, University of Brasília, Brasília, DF, Brazil

[3] Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, University of Campinas, Campinas, SP, Brazil

1 Introduction

Algorithms for solving continuous constrained optimization problems are iterative. Very frequently a feasible initial point is not available so that we must start with an approximation x^0 that is neither feasible nor optimal. Most algorithms compute successive iterations trying to achieve feasibility and optimality more or less simultaneously or, at least, without giving priority for one feature over another. Moreover, optimization algorithms usually compute both the objective function and the constraints (almost) at every iteration. When the objective function is very expensive and the evaluation of constraints is cheap, this may be a poor strategy. On the one hand, many times, what users really need is a feasible point with a reasonable objective function value. On the other hand, finding a feasible point may demand a non-negligible number of iterations that could become very expensive if we are forced to evaluate the objective function simultaneously with the constraints.

These observations lead us to prefer algorithms that start at a feasible point and preserve feasibility at every iteration. Classical barrier methods, introduced by Frisch (1955) and Fiacco and McCormick (1968), are among the most attractive alternatives for solving constrained optimization problems with those characteristics.

In this paper, we address the case in which the constraints are linear. Namely, the problem considered here will be:

$$\text{Minimize} f(x) \text{ subject to } Ax = b \text{ and } \ell \leq x \leq u, \tag{1}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, $A \in \mathbb{R}^{m \times n}$ $\ell, u \in (\mathbb{R} \cup \{-\infty, +\infty\})^n$, and $m < n$. Let $\mathcal{F} = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } \ell \leq x \leq u\}$ be the feasible set for problem (1) and

$$\mathcal{F}_0 = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } \ell < x < u\} \tag{2}$$

be the set of points that strictly satisfy the box constraints of (1), called the relative interior of $\mathcal{F}$. If an initial feasible approximation is not available, we assume that such approximation may be obtained at low cost by a suitable Linear Programming solver or an specific method for solving affine feasibility problems. Moreover, we assume that, by means of an appropriate pre-processing scheme, we purify the matrix A in order to have linear independent rows.

Pioneered by Frisch (1955) for convex problems, and further analyzed by Fiacco and McCormick (1968) for general nonlinear programming problems, interior-point methods had a revival with the work of Karmarkar (1984) within the linear programming context, specially after the equivalence with logarithmic barrier methods has been established by Gill et al. (1986). The renewed interest in logarithm barrier methods for the general nonlinear programming problem led to the development of the Ipopt algorithm, proposed by Wächter and Biegler (2006). This is one of the state-of-the-art solvers used nowadays in theoretical and practical problems.

The barrier method described in this work starts with an interior approximation such that $\ell < x^0 < u$ and $Ax^0 = b$. Given an interior feasible x^k, the new iterate x^{k+1}

is computed using matrix decompositions and manipulations by means of which instability due to extreme barrier parameters is avoided as much as possible and, additionally, possible sparsity is preserved. At each iteration, a linear system of equations is solved by means of which one computes new primal and dual approximate solutions. This system approximates KKT conditions for the minimization of a quadratic approximation of the function subject to the linear constraints, except for the fact that the primal $n \times n$ matrix, which represents the Hessian of the Lagrangian, might need to be modified in order to preserve positive definiteness onto the null-space of A. In other implementations of primal–dual methods (see, for example Wächter and Biegler 2006), an additional (primal) modification may be demanded to ensure that a solution of the KKT system exists. This modification is not needed at all in our approach due to the pre-processing scheme that guarantees full-rank of the matrix A. In linear constrained optimization, the primal modification may cause a temporary loss of feasibility of the next iterate, a feature that we want to avoid completely due to the characteristics of the problems that we have in mind, as stated above. We will show that the resulting method is globally convergent under mild assumptions, preserving feasibility of constraints along the iterations, therefore guaranteeing a feasible solution at the end of the optimization process. The proposed method has been implemented, and a comparative numerical study with Ipopt is provided.

Ipopt is a method for solving optimization problems with constraints of the form $h(x) = 0$, $\ell \leq x \leq u$, where the function h is, in general, nonlinear. In the case in which $h(x) \equiv Ax - b$, assuming that the iterate x^k is feasible and interior (that is, $Ax^k = b$ and $\ell < x^k < u$), the Ipopt basic iteration may be described as follows. Assume that $\mu > 0$ is a barrier parameter and that $B(x, \mu)$ is the corresponding logarithmic barrier function, which is well defined whenever $\ell < x < u$ and goes to infinity when x tends to the boundary. Consider the Newton iteration corresponding to the nonlinear system defined by the optimality conditions for the minimization of $f(x) + B(x, \mu)$ subject to $Ax = b$. This leads to a linear system of equations whose matrix may not satisfy desired inertia conditions. In order to correct this inertia, the whole diagonal of the matrix may be modified. The solution of the corrected linear system leads to a trial point that may violate both the (interiority with respect to the) bound constraints $\ell \leq x \leq u$ and the linear constraints $Ax = b$. This is because, after correction, the search direction may not belong to the null-space of A. The lack of interiority with respect to the bound constraints is fixed by means of a restriction of the step size. This is not the case of the infeasibility with respect to $Ax = b$, which is fixed using the same restoration procedure that is used in the nonlinear case. The acceptance of the trial point obeys to a filter criterion that combines a measure of infeasibility and the objective function value. The main differences with respect to the method presented in this paper are: (i) the inertia correction does not involve the modification of whole matrix's diagonal, as we assume that A is full-row rank, a feature that is guaranteed by a pre-processing scheme. As a consequence, feasibility is always preserved and restoration is never necessary; (ii) the acceptance of the trial point is related to sufficient descent for the merit function $f(x) + B(x, \mu)$.

The rest of this work is organized as follows. Section 2 introduces the proposed barrier method. Section 3 presents its theoretical convergence results. Section 4 shows implementation details and exhibits numerical experiments. Conclusions are given in the last section.

Notation If $v = (v_1, \ldots, v_n)^{\mathrm{T}} \in \mathbb{R}^n$, diag($v$) denotes the $n \times n$ diagonal matrix whose diagonal entries are given by $v_1, \ldots, v_n$. If $\mathcal{K} = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$ with $k_j < k_{j+1}$ for all j then we denote $\mathcal{K} \underset{\infty}{\subset} \mathbb{N}$.

2 Proposed barrier method

Let us consider the problem

$$\text{Minimize } \varphi_\mu(x) \text{ subject to } Ax = b, \tag{3}$$

where

$$\varphi_\mu(x) \overset{\text{def}}{=} f(x) - \mu \sum_{i \in \mathcal{I}_\ell} \log(x_i - \ell_i) - \mu \sum_{i \in \mathcal{I}_u} \log(u_i - x_i), \tag{4}$$

μ is a positive parameter, $\mathcal{I}_\ell \overset{\text{def}}{=} \{i : \ell_i \neq -\infty\}$, and $\mathcal{I}_u \overset{\text{def}}{=} \{i : u_i \neq +\infty\}$. The Lagrange conditions for (3) are given by

$$\begin{aligned} \nabla f(x) + A^{\mathrm{T}}\lambda - \mu L^\dagger e + \mu U^\dagger e &= 0 \\ Ax - b &= 0, \end{aligned} \tag{5}$$

e being the n-dimensional vector of all ones, the diagonal matrices L and U defined componentwise by

$$L_{i,i} = \begin{cases} x_i - \ell_i, & \text{if } i \in \mathcal{I}_\ell \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad U_{i,i} = \begin{cases} u_i - x_i, & \text{if } i \in \mathcal{I}_u \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

with pseudo-inverses $L^\dagger$ and $U^\dagger$ given by

$$L_{i,i}^\dagger = \begin{cases} \dfrac{1}{x_i - \ell_i}, & \text{if } i \in \mathcal{I}_\ell \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad U_{i,i}^\dagger = \begin{cases} \dfrac{1}{u_i - x_i}, & \text{if } i \in \mathcal{I}_u \\ 0, & \text{otherwise.} \end{cases}$$

Defining

$$[z_\ell]_i = \mu L_{i,i}^\dagger \quad \text{and} \quad [z_u]_i = \mu U_{i,i}^\dagger, \tag{7}$$

we have

$$\begin{aligned} LZ^\ell e - \mu e &= 0 \\ UZ^u e - \mu e &= 0, \end{aligned}$$

where $Z^\ell = \text{diag}(z_\ell)$ and $Z^u = \text{diag}(z_u)$; and, putting all together into (5), we obtain

$$\nabla f(x) + A^{\mathrm{T}}\lambda - z_\ell + z_u = 0, \tag{8a}$$

$$Ax - b = 0, \tag{8b}$$

$$Lz_\ell - \mu e = 0, \tag{8c}$$

$$Uz_u - \mu e = 0. \tag{8d}$$

A natural approach for finding x^* that approximately solves (3) consists in obtaining $(x^*, \lambda^*, z_\ell^*, z_u^*)$ approximately satisfying (8). In practice, we do not use the definition (7) of z_ℓ and z_u, but we indeed find an approximate solution to (8). Since φ in (3) is not defined for $x \notin (\ell, u)$ and goes to infinity as any component of x gets closer to ℓ or u, necessarily $\ell < x^* < u$. This fact, Eqs. (8c) and (8d), and the positivity of μ imply that $[z_\ell^*]_i \geq 0$, for all $i \in \mathcal{I}_\ell$ and $[z_u^*]_i \geq 0$, for all $i \in \mathcal{I}_u$. Therefore, z_ℓ^* and z_u^* are μ-approximations to the Lagrange multipliers in the KKT conditions of problem (1).

These ideas give support to a method for solving problem (1) that consists in solving a sequence of subproblems of the form (3) by driving to zero the so-called *barrier parameter* μ_k, positive for all k. Denoting the *outer iterations* by the index k, an approximate solution to (8) is computed by an iterative process indexed by j, the *inner iterations*.

To describe an inner iteration of the method, suppose we have $(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$, with $j \geq 0$, $x^{k,j} \in \mathcal{F}_0$, and positive vectors $z_\ell^{k,j}$ and $z_u^{k,j}$. We consider a line search method to compute $(x^{k,j+1}, \lambda^{k,j+1}, z_\ell^{k,j+1}, z_u^{k,j+1})$ such that $x^{k,j+1} \in \mathcal{F}_0$, $[z_\ell^{k,j+1}]_i > 0$ for each $i \in \mathcal{I}_\ell$, and $[z_u^{k,j+1}]_i > 0$ for each $i \in \mathcal{I}_u$. For defining the $(j+1)$th inner approximation, a search direction $(d_x^{k,j}, d_\lambda^{k,j}, d_{z_\ell}^{k,j}, d_{z_u}^{k,j})$ must be computed, with $d_x^{k,j}$ a descent direction for $\varphi_{\mu_k}(\cdot)$ from $x^{k,j}$, and a step size $\alpha_{k,j}$ satisfying a sufficient decrease condition.

The Newton's direction to system (8) from $(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$ is the solution to the $(3n + m)$-dimensional linear system given by

$$\begin{pmatrix} \nabla^2 f(x^{k,j}) & A^{\mathrm{T}} & -I & I \\ A & 0 & 0 & 0 \\ Z_{k,j}^\ell & 0 & L_{k,j} & 0 \\ -Z_{k,j}^u & 0 & 0 & U_{k,j} \end{pmatrix} \begin{pmatrix} d_x^{k,j} \\ d_\lambda^{k,j} \\ d_{z_\ell}^{k,j} \\ d_{z_u}^{k,j} \end{pmatrix} = - \begin{pmatrix} \nabla f(x^{k,j}) + A^{\mathrm{T}}\lambda^{k,j} - z_\ell^{k,j} + z_u^{k,j} \\ 0 \\ L_{k,j} z_\ell^{k,j} - \mu_k e \\ U_{k,j} z_u^{k,j} - \mu_k e \end{pmatrix}, \tag{9}$$

the well-known *primal–dual* system (Nocedal and Wright 2006). From the last two blocks of (9), we get

$$d_{z_\ell}^{k,j} = -z_\ell^{k,j} + \mu_k L_{k,j}^\dagger e - L_{k,j}^\dagger Z_{k,j}^\ell d_x^{k,j} \text{ and} \tag{10a}$$

$$d_{z_u}^{k,j} = -z_u^{k,j} + \mu_k U_{k,j}^\dagger e + U_{k,j}^\dagger Z_{k,j}^u d_x^{k,j}. \tag{10b}$$

Using (10) into the first two blocks of (9), we obtain the $(n + m)$-dimensional linear system

$$\begin{pmatrix} H_{k,j} & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} \begin{pmatrix} d_x^{k,j} \\ d_\lambda^{k,j} \end{pmatrix} = - \begin{pmatrix} \nabla f(x^{k,j}) + A^{\mathrm{T}} \lambda^{k,j} - \mu_k L_{k,j}^\dagger e + \mu_k U_{k,j}^\dagger e \\ 0 \end{pmatrix}, \qquad (11)$$

where $H_{k,j} = \nabla^2 f(x^{k,j}) + L_{k,j}^\dagger Z_{k,j}^\ell + U_{k,j}^\dagger Z_{k,j}^u$. From Nocedal and Wright (2006, Thm.16.3), we have that

$$\text{inertia} \begin{pmatrix} H_{k,j} & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} = \text{inertia}(W^T H_{k,j} W) + (m, m, 0), \qquad (12)$$

where $W \in \mathbb{R}^{n \times (n-m)}$ is a matrix whose columns form a basis to the kernel of A and inertia(M) is a triple that denotes the number of positive, negative, and null eigenvalues of the square symmetric matrix M, respectively. Therefore, the matrix of the linear system (11) is nonsingular if $H_{k,j}$ is nonsingular in the kernel of A. Furthermore, we have the following result.

Lemma 1 *Consider the linear system* (11). *If the matrix* $H_{k,j}$ *is positive definite in the kernel of A and the component* $d_x^{k,j}$ *of the solution is nonzero, then* $d_x^{k,j}$ *is a descent direction for* $\varphi_{\mu_k}(\cdot)$ *from* $x^{k,j}$.

Proof The first block of equations in system (11) implies that

$$H_{k,j} d_x^{k,j} + A^{\mathrm{T}} d_\lambda^{k,j} = -\nabla f(x^{k,j}) - A^{\mathrm{T}} \lambda^{k,j} + \mu_k L_{k,j}^\dagger e - \mu_k U_{k,j}^\dagger e. \qquad (13)$$

Since $\nabla \varphi_{\mu_k}(x^{k,j}) = \nabla f(x^{k,j}) - \mu_k L_{k,j}^\dagger e + \mu_k U_{k,j}^\dagger e$, from (13),

$$H_{k,j} d_x^{k,j} + A^{\mathrm{T}} d_\lambda = -\nabla \varphi_{\mu_k}(x^{k,j}) - A^{\mathrm{T}} \lambda^{k,j}. \qquad (14)$$

The second block of equations in (11) implies that $d_x^{k,j}$ is in the kernel of A. So, pre-multiplying (14) by $-(d_x^{k,j})^{\mathrm{T}}$,

$$\nabla \varphi_{\mu_k}(x^{k,j})^{\mathrm{T}} d_x^{k,j} = -(d_x^{k,j})^{\mathrm{T}} H_{k,j} d_x^{k,j}. \qquad (15)$$

Thus, if $H_{k,j}$ is positive definite in the kernel of A, then $(d_x^{k,j})^{\mathrm{T}} H_{k,j} d_x^{k,j} > 0$ whenever $d_x^{k,j} \neq 0$, which, together with (15), imply that $d_x^{k,j}$ is a descent direction for $\varphi_{\mu_k}(\cdot)$ from $x^{k,j}$. $\qquad \square$

To ensure that (i) the solution of (11) exists and (ii) $d_x^{k,j}$ is a descent direction for $\varphi_{\mu_k}(\cdot)$ from $x^{k,j}$, relation (12) and Lemma 1 indicate that we need

$$\text{inertia} \begin{pmatrix} H_{k,j} & A^{\mathrm{T}} \\ A & 0 \end{pmatrix} = (n, m, 0).$$

According with (12), this can be accomplished by guaranteeing that $H_{k,j}$ is positive definite in the kernel of A. For this reason, if the inertia of the matrix of the system (11) is not equal to $(n, m, 0)$, we search for a scalar $\xi_{k,j} > 0$ such that the inertia of the matrix

$$\begin{pmatrix} H_{k,j} + \xi_{k,j}I & A^{\mathrm{T}} \\ A & 0 \end{pmatrix}$$

is $(n, m, 0)$ and then solve the following perturbed version of the system (11)

$$\begin{pmatrix} H_{k,j} + \xi_{k,j}I & A^{T} \\ A & 0 \end{pmatrix} \begin{pmatrix} d_x^{k,j} \\ d_\lambda^{k,j} \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_{\mu_k}(x^{k,j}) + A^{\mathrm{T}}\lambda_{k,j} \\ 0 \end{pmatrix}. \tag{16}$$

The subproblem iterates are stated as

$$x^{k,j+1} = x^{k,j} + \alpha_{k,j}d_x^{k,j}, \tag{17a}$$

$$\lambda^{k,j+1} = \lambda^{k,j} + d_\lambda^{k,j}, \tag{17b}$$

$$\bar{z}_\ell^{k,j+1} = z_\ell^{k,j} + \alpha_{k,j}^{z_\ell}d_{z_\ell}^{k,j}, \tag{17c}$$

$$\bar{z}_u^{k,j+1} = z_u^{k,j} + \alpha_{k,j}^{z_u}d_{z_u}^{k,j}, \tag{17d}$$

in which $\alpha_{k,j}, \alpha_{k,j}^{z_\ell}, \alpha_{k,j}^{z_u} \in (0, 1]$ determine the step sizes. Since $\ell < x^{k,j} < u$, $[z_\ell^{k,j}]_i > 0$ for each $i \in \mathcal{I}_\ell$, and $[z_u^{k,j}]_i > 0$ for each $i \in \mathcal{I}_u$, we need to preserve these properties in the new iterate. Following Wächter and Biegler (2006), we use a *fraction-to-the-boundary* parameter $\tau_k = \max\{\tau_{\min}, 1 - \mu_k\}$, with $\tau_{\min} \in (0, 1)$. Moreover, we define the sets $\mathcal{D}_-^{k,j} \overset{\text{def}}{=} \{i : [d_x^{k,j}]_i < 0\}$ and $\mathcal{D}_+^{k,j} \overset{\text{def}}{=} \{i : [d_x^{k,j}]_i > 0\}$, and compute

$$\alpha_{k,j}^\ell = \max_{i \in \mathcal{I}_\ell \cap \mathcal{D}_-^{k,j}} \left\{ \alpha \in (0, 1] : (x_i^{k,j} + \alpha[d_x^{k,j}]_i) - \ell_i \geq (1 - \tau_k)(x_i^{k,j} - \ell_i) \right\},$$

$$\alpha_{k,j}^u = \max_{i \in \mathcal{I}_u \cap \mathcal{D}_+^{k,j}} \left\{ \alpha \in (0, 1] : u_i - (x_i^{k,j} + \alpha[d_x^{k,j}]_i) \geq (1 - \tau_k)(u_i - x_i^{k,j}) \right\},$$

$$\alpha_{k,j}^{z_\ell} = \max_{i \in \mathcal{I}_\ell \cap \mathcal{D}_+^{k,j}} \left\{ \alpha \in (0, 1] : [z_\ell^{k,j}]_i + \alpha[d_{z_\ell}^{k,j}]_i \geq (1 - \tau_k)[z_\ell^{k,j}]_i \right\},$$

$$\alpha_{k,j}^{z_u} = \max_{i \in \mathcal{I}_u \cap \mathcal{D}_-^{k,j}} \left\{ \alpha \in (0, 1] : [z_u^{k,j}]_i + \alpha[d_{z_u}^{k,j}]_i \geq (1 - \tau_k)[z_u^{k,j}]_i \right\}.$$

Notwithstanding, whenever $d_x^{k,j}$ is small enough, we take $\alpha_{k,j}^{z_\ell} = \alpha_{k,j}^{z_u} = 1$, in order to not impair the global convergence of the method. A backtracking must be done to obtain $\alpha_{k,j} \in (0, \alpha_{k,j}^{\max}]$, with $\alpha_{k,j}^{\max} = \min\{\alpha_{k,j}^\ell, \alpha_{k,j}^u\}$, such that the sufficient decrease condition

$$\varphi_{\mu_k}\left(x^{k,j} + \alpha_{k,j} d_x^{k,j}\right) \leq \varphi_{\mu_k}(x^{k,j}) + \gamma \alpha_{k,j} \nabla \varphi_{\mu_k}(x^{k,j})^T d_x^{k,j}$$

is satisfied, for some $\gamma \in (0, 1)$.

Last, but not the least, we have to ensure that $z_\ell^{k,j+1}$ and $z_u^{k,j+1}$ approximately maintain the relationship with $x^{k,j+1}$ established in (7). These equations guarantee that the north-west matrix in system (11) is an approximation to the Hessian of the log-barrier function (4). Therefore, we consider

$$[z_\ell^{k,j+1}]_i = \begin{cases} \max\left\{\min\left\{[\bar{z}_\ell^{k,j+1}]_i, \kappa_z\left(\dfrac{\mu_k}{x_i^{k,j+1} - \ell_i}\right)\right\}, \dfrac{1}{\kappa_z}\left(\dfrac{\mu_k}{x_i^{k,j+1} - \ell_i}\right)\right\}, & \text{if } i \in \mathcal{I}_\ell \\ 0, & \text{otherwise} \end{cases}$$

and

$$[z_u^{k,j+1}]_i = \begin{cases} \max\left\{\min\left\{[\bar{z}_u^{k,j+1}]_i, \kappa_z\left(\dfrac{\mu_k}{u_i - x_i^{k,j+1}}\right)\right\}, \dfrac{1}{\kappa_z}\left(\dfrac{\mu_k}{u_i - x_i^{k,j+1}}\right)\right\}, & \text{if } i \in \mathcal{I}_u \\ 0, & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, n$ and a constant $\kappa_z \geq 1$. Hence,

$$\begin{aligned} [z_\ell^{k,j+1}]_i &\in \left[\dfrac{1}{\kappa_z}\left(\dfrac{\mu_k}{x_i^{k,j+1} - \ell_i}\right), \kappa_z\left(\dfrac{\mu_k}{x_i^{k,j+1} - \ell_i}\right)\right], \quad \text{for } i \in \mathcal{I}_\ell \text{ and} \\ [z_u^{k,j+1}]_i &\in \left[\dfrac{1}{\kappa_z}\left(\dfrac{\mu_k}{u_i - x_i^{k,j+1}}\right), \kappa_z\left(\dfrac{\mu_k}{u_i - x_i^{k,j+1}}\right)\right], \quad \text{for } i \in \mathcal{I}_u, \end{aligned}$$

which means that (7) will be satisfied with precision κ_z (see Wächter and Biegler 2006).

We consider that a point $(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$ is an approximate solution to subproblem (3) whenever

$$E_{\mu_k}(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j}) \leq \kappa_\varepsilon \mu_k, \tag{18}$$

where

$$E_\mu(x, \lambda, z_\ell, z_u) \overset{\text{def}}{=} \left\| \begin{pmatrix} \nabla f(x) + A^T \lambda - z_\ell + z_u \\ L z_\ell - \mu e \\ U z_u - \mu e \end{pmatrix} \right\|_\infty, \qquad (19)$$

and $\kappa_\varepsilon > 0$. Thus, (18) implies that Eqs. (8a), (8c), and (8d) are approximately satisfied, and by the definition of the method, $Ax^{k,j} = b$, so that (8b) holds. Therefore, $(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$ approximately satisfies the optimality conditions (8) of subproblem (3).

After a subproblem is approximately solved, we compute a new barrier parameter

$$\mu_{k+1} = \min\left\{ \kappa_\mu \mu_k, \mu_k^{\theta_\mu} \right\},$$

where $\kappa_\mu \in (0, 1)$ and $\theta_\mu \in (1, 2)$. With such an update, the barrier parameter may converge superlinearly to zero.

Algorithms 1, 2, and 3 below summarize the proposed method. Algorithm 1 corresponds to the *outer algorithm*, while Algorithm 2 corresponds to the *inner algorithm*, that is used by Algorithm 1 for solving the subproblems. Algorithm 3, taken from Wächter and Biegler (2006, p 36) and reproduced here for completeness, corresponds to the inertia correction procedure.

Algorithm 1: Feasible Interior-Point Method - Outer Iterations

---

Input. Let $x^0 \in \mathcal{F}_0$, $\lambda^0 \in \mathbb{R}^m$, $z_\ell^0 \in \mathbb{R}^n$, and $z_u^0 \in \mathbb{R}^n$ such that $[z_\ell^0]_i > 0$ for $i \in \mathcal{I}_\ell$ and $[z_\ell^0]_i = 0$ otherwise, and $[z_u^0]_i > 0$ for $i \in \mathcal{I}_u$ and $[z_u^0]_i = 0$ otherwise, $A \in \mathbb{R}^{m \times n}$ a full row-rank matrix, and $\mu_0 > 0$; constants $\varepsilon_{\text{tol}} > 0$, $\kappa_\varepsilon > 0$, $\kappa_\mu \in (0, 1)$, $\theta_\mu \in (1, 2)$, $\gamma \in (0, 1)$, $\tau_{\min} > 0$, and $\kappa_z \geq 1$. Initialize $k \leftarrow 0$.

Step 1.1. If

$$E_0(x^k, \lambda^k, z_\ell^k, z_u^k) \leq \varepsilon_{\text{tol}} \qquad (20)$$

then **stop** and **return** $(x^k, \lambda^k, z_\ell^k, z_u^k)$.

Step 1.2. Starting from x^k, λ^k, z_ℓ^k, and z_u^k and using Algorithm 2, compute x^{k+1}, λ^{k+1}, z_ℓ^{k+1}, and z_u^{k+1} such that

$$E_{\mu_k}(x^{k+1}, \lambda^{k+1}, z_\ell^{k+1}, z_u^{k+1}) \leq \kappa_\varepsilon \mu_k.$$

Step 1.3. Compute $\mu_{k+1} = \min\{\kappa_\mu \mu_k, \mu_k^{\theta_\mu}\}$.

Step 1.4. Let $k \leftarrow k + 1$ and go to Step 1.1.

Algorithm 2: Feasible Interior-Point Method - Inner Iterations

Input. Let $x^{k,0} \in \mathcal{F}_0$, $\lambda^{k,0} \in \mathbb{R}^m$, $z_\ell^{k,0} \in \mathbb{R}^n$, and $z_u^{k,0} \in \mathbb{R}^n$ such that $[z_\ell^{k,0}]_i > 0$ for $i \in \mathcal{I}_\ell$ and $[z_\ell^{k,0}]_i = 0$ otherwise, and $[z_u^{k,0}]_i > 0$ for $i \in \mathcal{I}_u$ and $[z_u^{k,0}]_i = 0$ otherwise, $A \in \mathbb{R}^{m \times n}$ a full row-rank matrix, and $\mu_k > 0$; constants $\kappa_\varepsilon > 0$, $\gamma \in (0,1)$, $\tau_{\min} > 0$, and $\kappa_z \geq 1$. Initialize $j \leftarrow 0$.

Step 2.1. If
$$E_{\mu_k}(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j}) \leq \kappa_\varepsilon \mu_k,$$
then **stop** and **return** $(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$.

Step 2.2. Using Algorithm 3, compute $\xi_{k,j} \geq 0$ such that inertia$(M_{k,j}) = (n, m, 0)$, where
$$M_{k,j} = \begin{pmatrix} \nabla^2 f(x^{k,j}) + L_{k,j}^\dagger Z_{k,j}^L + U_{k,j}^\dagger Z_{k,j}^U + \xi_{k,j} I & A^T \\ A & 0 \end{pmatrix}. \quad (21)$$

Step 2.3. Compute $d_x^{k,j}$ and $d_\lambda^{k,j}$ as the solution to the linear system
$$M_{k,j} \begin{pmatrix} d_x^{k,j} \\ d_\lambda^{k,j} \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_{\mu_k}(x^{k,j}) + A^T \lambda^{k,j} \\ 0 \end{pmatrix}. \quad (22)$$

Step 2.4. Compute
$$d_{z_\ell}^{k,j} = -z_\ell^{k,j} + \mu_k L_{k,j}^\dagger e - L_{k,j}^\dagger Z_{k,j}^L d_x^{k,j} \text{ and}$$
$$d_{z_u}^{k,j} = -z_u^{k,j} + \mu_k U_{k,j}^\dagger e + U_{k,j}^\dagger Z_{k,j}^U d_x^{k,j}.$$

Step 2.5. Compute $\tau_k = \max\{\tau_{\min}, 1 - \mu_k\}$,
$$\alpha_{k,j}^\ell = \max_{i \in \mathcal{I}_\ell \cap \mathcal{D}_-^{k,j}} \{\alpha \in (0,1] : ([x^{k,j}]_i + \alpha[d_x^{k,j}]_i) - \ell_i \geq (1 - \tau_k)([x^{k,j}]_i - \ell_i)\},$$
$$\alpha_{k,j}^u = \max_{i \in \mathcal{I}_u \cap \mathcal{D}_+^{k,j}} \{\alpha \in (0,1] : u_i - ([x^{k,j}]_i + \alpha[d_x^{k,j}]_i) \geq (1 - \tau_k)(u_i - [x^{k,j}]_i)\},$$

and $\alpha_{k,j}^{\max} = \min\{\alpha_{k,j}^\ell, \alpha_{k,j}^u\}$. If both
$$\begin{aligned} [d_x^{k,j}]_i &\leq \frac{\mu_k}{[z_\ell^{k,j}]_i}, \text{ for all } i \in \mathcal{I}_\ell \cap \mathcal{D}_+^{k,j} \text{ and} \\ [d_x^{k,j}]_i &\geq -\frac{\mu_k}{[z_u^{k,j}]_i}, \text{ for all } i \in \mathcal{I}_u \cap \mathcal{D}_-^{k,j}, \end{aligned} \quad (23)$$

then $\alpha_{k,j}^{z_\ell} = \alpha_{k,j}^{z_u} = 1$, otherwise,
$$\alpha_{k,j}^{z_\ell} = \max_{i \in \mathcal{I}_\ell \cap \mathcal{D}_+^{k,j}} \{\alpha \in (0,1] : [z_\ell^{k,j}]_i + \alpha[d_{z_\ell}^{k,j}]_i \geq (1 - \tau_k)[z_\ell^{k,j}]_i\} \text{ and}$$
$$\alpha_{k,j}^{z_u} = \max_{i \in \mathcal{I}_u \cap \mathcal{D}_-^{k,j}} \{\alpha \in (0,1] : [z_u^{k,j}]_i + \alpha[d_{z_u}^{k,j}]_i \geq (1 - \tau_k)[z_u^{k,j}]_i\}.$$

Step 2.6. Let $\alpha \leftarrow \alpha_{k,j}^{\max}$. While
$$\varphi_{\mu_k}(x^{k,j} + \alpha d_x^{k,j}) > \varphi_{\mu_k}(x^{k,j}) + \gamma \alpha \nabla \varphi_{\mu_k}(x^{k,j})^T d_x^{k,j} \quad (24)$$
set $\alpha \leftarrow \frac{1}{2}\alpha$. Then, set $\alpha_{k,j} \leftarrow \alpha$.

Step 2.7. Compute $x^{k,j+1} = x^{k,j} + \alpha_{k,j} d_x^{k,j}$ and $\lambda^{k,j+1} = \lambda^{k,j} + d_\lambda^{k,j}$.

Step 2.8. Compute $\bar{z}_\ell^{k,j+1} = z_\ell^{k,j} + \alpha_{k,j}^{z_\ell} d_{z_\ell}^{k,j}$, $\bar{z}_u^{k,j+1} = z_u^{k,j} + \alpha_{k,j}^{z_u} d_{z_u}^{k,j}$,
$$[z_\ell^{k,j+1}]_i = \max\left\{\min\left\{[\bar{z}_\ell^{k,j+1}]_i, \kappa_z \left(\frac{\mu_k}{[x^{k,j+1}]_i - \ell_i}\right)\right\}, \frac{1}{\kappa_z}\left(\frac{\mu_k}{[x^{k,j+1}]_i - \ell_i}\right)\right\},$$
for $i \in \mathcal{I}_\ell$, and
$$[z_u^{k,j+1}]_i = \max\left\{\min\left\{[\bar{z}_u^{k,j+1}]_i, \kappa_z \left(\frac{\mu_k}{u_i - [x^{k,j+1}]_i}\right)\right\}, \frac{1}{\kappa_z}\left(\frac{\mu_k}{u_i - [x^{k,j+1}]_i}\right)\right\},$$
for $i \in \mathcal{I}_u$.

Step 2.9. Set $j \leftarrow j + 1$ and go to Step 2.1.

Algorithm 3: Inertia correction

Input. Constants $0 < \xi_{\min} < \xi_{\max}$; $0 < \kappa_\xi^- < 1 < \kappa_\xi^+ < \bar{\kappa}_\xi^+$; $\xi_{\mathrm{ini}} > 0$, and the prior correction $\xi_{k,j-1}$ ($\xi_{k,j-1} = 0$ if $k = j = 0$).

Step 3.1. $\xi_{k,j} \leftarrow 0$.

Step 3.2. If the inertia of the linear system (16) is $(n, m, 0)$, set $\xi_{k+1,-1} = \xi_{k,j}$ and **stop**.

Step 3.3. If $\xi_{k,j-1} = 0$, then $\xi_{k,j} \leftarrow \xi_{\mathrm{ini}}$, else $\xi_{k,j} \leftarrow \max\{\xi_{\min}, \kappa_\xi^- \xi_{k,j-1}\}$.

Step 3.4. If the inertia of the linear system (16) is $(n, m, 0)$, set $\xi_{k+1,-1} = \xi_{k,j}$ and **stop**.

Step 3.5. If $\xi_{k,j-1} = 0$, then $\xi_{k,j} \leftarrow \bar{\kappa}_\xi^+ \xi_{k,j}$, else $\xi_{k,j} \leftarrow \kappa_\xi^+ \xi_{k,j}$.

Step 3.6. If $\xi_{k,j} > \xi_{\max}$, then **stop**, because it was not possible to compute a value for $\xi_{k,j}$. Else, go to Step 3.4.

---

3 Global convergence

The convergence theory of the proposed method is given in this section. We first present well definiteness results for Algorithm 2.

Lemma 2 *In Step 2.2 of Algorithm* 2, *there exists* $\xi_{k,j} \geq 0$ *large enough such that* $\mathrm{inertia}(M_{k,j}) = (n, m, 0)$, *with* $M_{k,j}$ *defined in* (21).

Proof Let $\tilde{H}_{\xi_{k,j}} = \nabla^2 f(x^{k,j}) + L_{k,j}^\dagger Z_{k,j}^L + U_{k,j}^\dagger Z_{k,j}^U + \xi_{k,j} I$ and $W \in \mathbb{R}^{n \times n-m}$ a matrix whose columns form a basis to the kernel of A. Notice that, if $\tilde{H}_{\xi_{k,j}}$ is positive definite, then $W^\mathrm{T} \tilde{H}_{\xi_{k,j}} W$ is also positive definite. Thus, on the one hand, if $\tilde{H}_0$ is positive definite, then $W^\mathrm{T} \tilde{H}_0 W$ is positive definite as well, which together with (12) implies that $\mathrm{inertia}(M_{k,j}) = (n, m, 0)$ for $\xi_{k,j} = 0$. On the other hand, if $\tilde{H}_0$ is not positive definite, let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of the matrix $\tilde{H}_0$, with $\lambda_1 \leq 0$. Therefore, the eigenvalues of $\tilde{H}_{|\lambda_1|+\epsilon}$, for $\epsilon > 0$, are

$$\epsilon \leq \lambda_2 + |\lambda_1| + \epsilon \leq \cdots \leq \lambda_n + |\lambda_1| + \epsilon,$$

which implies that $\tilde{H}_{|\lambda_1|+\epsilon}$ is positive definite and, consequently, so it is $W^\mathrm{T} \tilde{H}_{|\lambda_1|+\epsilon} W$. Hence, (12) yields $\mathrm{inertia}(M_{k,j}) = (n, m, 0)$ for $\xi_{k,j} \geq |\lambda_1| + \epsilon$. □

Next we show that, as long as $x^{k,j}$ is not stationary for problem (3), it is always possible to compute an adequate search direction in Algorithm 2.

Lemma 3 *In Step 2.3 of Algorithm* 2, *it is possible to compute the search directions* $d_x^{k,j}$ *and* $d_\lambda^{k,j}$. *Moreover, if* $d_x^{k,j} \neq 0$, *then* $d_x^{k,j}$ *is a descent direction for* $\varphi_{\mu_k}(\cdot)$ *from* $x^{k,j}$. *So*, Step 2.6 *finishes in a finite number of iterations.*

Proof From Lemma 2, it is possible to find $\xi_{k,j}$ such that inertia$(M_{k,j}) = (n, m, 0)$. Therefore, $M_{k,j}$ is a nonsingular matrix, and the linear system in Step 2.3 has a unique solution. Notice that, if $d_x^{k,j} = 0$ and $d_\lambda^{k,j} = 0$, the right-hand side of the linear system (22) yields the pair $(x^{k,j}, \lambda^{k,j})$ to be primal–dual stationary for problem (3). In case $d_x^{k,j} = 0$ but $d_\lambda^{k,j} \neq 0$, also from (22) we obtain the primal–dual stationary pair $(x^{k,j}, \lambda^{k,j} + d_\lambda^{k,j}) = (x^{k,j}, \lambda^{k+1,j})$ for problem (3). Now, from Lemma 1, since inertia$(M_{k,j}) = (n, m, 0)$, we have that, whenever a direction $d_x^{k,j} \neq 0$ is computed, it will be of descent for $\varphi_{\mu_k}(\cdot)$ from $x^{k,j}$. So, there exists $\alpha_{k,j}$ such that the sufficient decrease condition at Step 2.6 is verified. $\square$

Lemmas 2 and 3 show that Algorithm 2 is well defined. The well definiteness of Algorithm 1 is subject to the global convergence of Algorithm 2, and this is established in the sequel. By global convergence, we mean that we analyze the properties of the infinite sequence generated by the method that emerges when the stopping criterion at Step 2.1 is removed from Algorithm 2. With some abuse of notation, we refer to this sequence as the infinite sequence generated by Algorithm 2. Analyzing the properties of this infinite sequence, we prove that the stopping criterion at Step 2.1 of Algorithm 2 is satisfied in finite time. Based on ideas from Chen and Goldfarb (2006), we present a global convergence analysis for the proposed algorithm, but in a more detailed fashion, and strongly connected with the structure of problem (1). The global convergence results rest upon the following assumptions.

Assumption 1 The set $\mathcal{F}_0$, defined in (2), is nonempty.

Assumption 2 The objective function f of problem (1) is continuous and at least twice continuously differentiable.

Assumption 3 The sequence $\{x^{k,j}\}$ generated by Algorithm 2 is bounded, for all k.

Assumption 4 The matrices $\bar{H}_{k,j} = H_{k,j} + \xi_{k,j}I$ satisfy $d^{\mathrm{T}}\bar{H}_{k,j}d \geq \sigma\|d\|^2$, for all $d \in \mathbb{R}^n$, $d \neq 0$, such that $Ad = 0$, for some $\sigma > 0$ and for all k and j.

Although Assumption 3 is a conjecture about the sequence generated by the method, which we have no control at a first glance, this is implicitly assured when (i) box constraints exist for every variable, i.e., when $-\infty < \ell_i \leq u_i < +\infty$, for $i = 1, \ldots, n$, or (ii) when the level set $\{x \in \mathcal{F} \mid f(x) \leq f(x^0)\}$ is bounded, where $x^0 \in \mathcal{F}_0$. Assumption 4 establishes that the matrices $\{\bar{H}_{k,j}\}$ must be uniformly positive definite in the kernel of matrix A. Compared with other requirements, as in the work by Chen and Goldfarb (2006, Condition (C-5)), our hypothesis is slightly weaker, since Lemma 2 guarantees the attainment of a scalar $\xi_{k,j}$ in Step 2.2 of Algorithm 2 which properly adjusts the inertia of the matrix of the system (22) . Thus, Assumption 4 may be accomplished by the numerical nature of the inertia correction method used.

Lemma 4 *Suppose that Assumptions 1, 2 and 3 hold and consider that the sequence generated by Algorithm 2 is $\{x^{k,j+1}, \lambda^{k,j+1}, z_{\ell}^{k,j+1}, z_u^{k,j+1}\}$. Then, there exists $\delta > 0$ such that*

I. *for all $j \in \{0, 1, 2, \ldots\}$, it holds*

   (a) $\ell_i + \delta \leq x_i^{k,j+1}$, *for all $i \in \mathcal{I}_{\ell}$ and*

   (b) $x_i^{k,j+1} \leq u_i - \delta$, *for all $i \in \mathcal{I}_u$;*

II. *for all $j \in \{0, 1, 2, \ldots\}$, it holds*

   (a) $[z_{\ell}^{k,j+1}]_i \in \dfrac{\mu_k}{\delta}\left[\dfrac{1}{\kappa_z}, \kappa_z\right]$, *for all $i \in \mathcal{I}_{\ell}$ and*

   (b) $[z_u^{k,j+1}]_i \in \dfrac{\mu_k}{\delta}\left[\dfrac{1}{\kappa_z}, \kappa_z\right]$, *for all $i \in \mathcal{I}_u$.*

Proof To show (I), suppose, by contradiction, that there exist an infinite set $\mathcal{J} \subseteq \mathbb{N}_{\infty}$, and $\hat{\imath} \in \mathcal{I}_{\ell}$ such that

$$\lim_{j \in \mathcal{J}} x_{\hat{\imath}}^{k,j+1} = \ell_{\hat{\imath}}. \tag{25}$$

By Step 2.1 of Algorithm 2, $E_{\mu_k}(x^{k,j+1}, \lambda^{k,j+1}, z_{\ell}^{k,j+1}, z_u^{k,j+1}) > \kappa_{\varepsilon}\mu_k$, and by the line search of Step 2.6 of Algorithm 2,

$$\varphi_{\mu_k}(x^{k,j+1}) \leq \varphi_{\mu_k}(x^{k,j}) + \gamma \alpha_{k,j-1} \nabla \varphi_{\mu_k}(x^{k,j})^{\mathrm{T}} d_x^{k,j}. \tag{26}$$

Assumptions 2 and 3 imply that the sequences $\{f(x^{k,j+1})\}$, $\{x_i^{k,j+1} - \ell_i\}$, for all $i \in \mathcal{I}_{\ell}$, and $\{u_i - x_i^{k,j+1}\}$, for all $i \in \mathcal{I}_u$, are bounded. Thus, by (25) and by the definition of the function $\varphi_{\mu}(\cdot)$ in (4),

$$\lim_{j \in \mathcal{J}} \varphi_{\mu_k}(x^{k,j+1}) = +\infty,$$

which contradicts (26). An analogous reasoning applies in case there exist an infinite set $\mathcal{J} \subseteq \mathbb{N}_{\infty}$, and $\hat{\imath} \in \mathcal{I}_u$ such that $\{x_{\hat{\imath}}^{k,j+1}\}_{j \in \mathcal{J}} \to u_{\hat{\imath}}$.

Part (II) follows directly from the computations within Step 2.8 of Algorithm 2. $\square$

In the following, auxiliary results ascertain the boundedness of the sequences generated by Algorithm 2, in preparation to the global convergence theorem.

Lemma 5 *Suppose Assumptions 2 and 3 hold. Then, the sequence $\{\bar{H}_{k,j}\}$ generated by Algorithm 2 is bounded.*

Proof Assumptions 2 and 3 imply that the sequence $\{\nabla^2 f(x^{k,j})\}$ generated by Algorithm 2 is bounded. Furthermore, Assumption 3 and Lemma 4(I) imply that the

sequence $\{L_{k,j}^{\dagger}, U_{k,j}^{\dagger}\}$ generated by Algorithm 2 is bounded. In addition, Lemma 4(II) implies that the sequence $\{Z_{k,j}^{L}, Z_{k,j}^{U}\}$ generated by Algorithm 2 is bounded. Finally, according with Lemma 2, $\xi_{k,j} \geq |\lambda_1| + \epsilon$ is large enough to correct the inertia of matrix $M_{k,j}$, which gives an implicit upper bound in $\xi_{k,j}$. Putting all these facts together, the result follows. □

Lemma 6 *Suppose that Assumptions* 1, 2, *and* 3 *hold. Then, the sequence* $\{d_x^{k,j}, \lambda^{k,j+1}, z_{\ell}^{k,j+1}, z_u^{k,j+1}\}$ *generated by Algorithm 2 is bounded.*

Proof According with Lemma 4(II), the sequence $\{z_{\ell}^{k,j+1}, z_u^{k,j+1}\}$ is bounded. In order to get a contradiction, suppose there exists an infinite set $\mathcal{J} \underset{\infty}{\subset} \mathbb{N}$ such that

$$\lim_{j \in \mathcal{J}} \|(d_x^{k,j}, \lambda^{k,j} + d_{\lambda}^{k,j})\| = +\infty. \tag{27}$$

Assumptions 2 and 3 and Lemma 4(I) imply that the sequence

$$\left\{ \nabla \varphi_{\mu_k}(x^{k,j}) \right\}_{j \in \mathcal{J}} = \left\{ \nabla f(x^{k,j}) - \mu_k L_{k,j}^{\dagger} e + \mu_k U_{k,j}^{\dagger} e \right\}_{j \in \mathcal{J}}$$

is bounded. This fact, together with Lemmas 4(II) and 5, assure the existence of an infinite set $\hat{\mathcal{J}} \underset{\infty}{\subset} \mathcal{J}$ such that

$$\lim_{j \in \hat{\mathcal{J}}} \nabla \varphi_{\mu_k}(x^{k,j}) = \nabla \varphi_{\mu_k}(x^{k,*}), \tag{28}$$

$$\lim_{j \in \hat{\mathcal{J}}}(z_{\ell}^{k,j}, z_u^{k,j}) = (z_{\ell}^{k,*}, z_u^{k,*}), \text{ and} \\ \lim_{j \in \hat{\mathcal{J}}} \bar{H}_{k,j} = \bar{H}_{k,*}. \tag{29}$$

Using (21), we can rewrite the system (16) in order to get

$$M_{k,j} \begin{pmatrix} d_x^{k,j} \\ \lambda^{k,j} + d_{\lambda}^{k,j} \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_{\mu_k}(x^{k,j}) \\ 0 \end{pmatrix}, \tag{30}$$

recalling that (29) implies that $\lim_{j \in \hat{\mathcal{J}}} M_{k,j} = M_{k,*}$.

By (28), we have that the right-hand side of the system (30) is bounded for $j \in \hat{\mathcal{J}}$. Besides that, Step 2.2 of Algorithm 2 guarantees that $M_{k,j}$ will be nonsingular for all j. Thus, from (30), it follows that

$$\lim_{j \in \hat{\mathcal{J}}} \begin{pmatrix} d^{k,j} \\ \lambda^{k,j} + d_{\lambda}^{k,j} \end{pmatrix} = \lim_{j \in \hat{\mathcal{J}}} -M_{k,j}^{-1} \begin{pmatrix} \nabla \varphi_{\mu_k}(x^{k,j}) \\ 0 \end{pmatrix} = -M_{k,*}^{-1} \begin{pmatrix} \nabla \varphi_{\mu_k}(x^{k,*}) \\ 0 \end{pmatrix},$$

contradicting (27). □

Lemma 7 *Consider that Assumptions* 1, 2, 3, *and* 4 *hold. Then, the sequence* $\{d_x^{k,j}\}$ *generated by Algorithm 2 goes to zero as j tends to infinity.*

Proof Lemma 6 implies that the sequence $\{d_x^{k,j}\}$ is bounded, therefore it admits some convergent subsequence. Let us consider, by contradiction, that there exists an infinite subset $\mathcal{J} \underset{\infty}{\subseteq} \mathbb{N}$ such that

$$\lim_{j \in \mathcal{J}} d_x^{k,j} = d_x^{k,*} \neq 0. \tag{31}$$

Lemma 5, Assumption 3, and Lemma 6 imply that there exists an infinite set $\hat{\mathcal{J}} \underset{\infty}{\subseteq} \mathcal{J}$ such that

$$\lim_{j \in \hat{\mathcal{J}}} \bar{H}_{k,j} = \bar{H}_{k,*} \quad \text{and} \quad \lim_{j \in \hat{\mathcal{J}}} \left(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j} \right) = \left(x^{k,*}, \lambda^{k,*}, z_\ell^{k,*}, z_u^{k,*} \right).$$

Pre-multiplying the first block of equations from the system (16) by $d_x^{k,j}$, which, by the second block of equations from the system (16), belongs to the kernel of A, we have that

$$\begin{aligned}
(d_x^{k,j})^{\mathrm{T}} \bar{H}_{k,j} d_x^{k,j} &= -(d_x^{k,j})^{\mathrm{T}} \nabla \varphi_{\mu_k}(x^{k,j}) \\
&\geq \sigma \|d_x^{k,j}\|^2, \text{ by Assumption 4.}
\end{aligned}$$

Thus,

$$\nabla \varphi_{\mu_k}(x^{k,j})^{\mathrm{T}} d_x^{k,j} \leq -\sigma \|d_x^{k,j}\|^2. \tag{32}$$

Taking the limit in (32) for $j \in \hat{\mathcal{J}}$, it follows that

$$\nabla \varphi_{\mu_k}(x^{k,*})^{\mathrm{T}} d_x^{k,*} \leq -\sigma \|d_x^{k,*}\|^2. \tag{33}$$

By Lemma 4(I), we have that $\ell_i + \delta \leq x_i^{k,j}$, for all $i \in \mathcal{I}_\ell$ and $x_i^{k,j} \leq u_i - \delta$, for all $i \in \mathcal{I}_u$, with $\delta > 0$. Taking the limit for $j \in \hat{\mathcal{J}}$, it follows that $\ell_i < x_i^{k,*}$, for all $i \in \mathcal{I}_\ell$ and $x_i^{k,*} < u_i$, for all $i \in \mathcal{I}_u$. Therefore, there exists $\hat{\alpha} \in (0,1]$ such that, for all $\alpha \in (0, \hat{\alpha}]$,

$$\ell_i < x_i^{k,*} + \alpha [d_x^{k,*}]_i, \text{ for all } i \in \mathcal{I}_\ell \text{ and } x_i^{k,*} + \alpha [d_x^{k,*}]_i < u_i, \text{ for all } i \in \mathcal{I}_u. \tag{34}$$

Since $d_x^{k,*} \neq 0$ and (33) implies that $\nabla \varphi_{\mu_k}(x^{k,*})^T d_x^{k,*} < 0$, then there exists $\tilde{\alpha} \in (0, \hat{\alpha}]$ such that, for all $\alpha \in (0, \tilde{\alpha}]$, (34) holds and

$$\varphi_{\mu_k}(x^{k,*} + \alpha d_x^{k,*}) \leq \varphi_{\mu_k}(x^{k,*}) + \bar{\gamma} \alpha \nabla \varphi_{\mu_k}(x^{k,*})^T d_x^{k,*}$$

is verified, with $\bar{\gamma} \in (\gamma, 1)$. Notice that this is a sufficient decrease condition, which ensures the existence of $\tilde{\alpha}$. Nonetheless, this is a more rigorous condition than the one required in Step 2.6 of Algorithm 2, given that $\bar{\gamma} > \gamma$. Since $\varphi_\mu(\cdot)$ is a continuously differentiable function, from the strict fulfillment of the bound constraints (34), and the fact that $d_x^{k,*}$ is a descent direction from $x^{k,*}$, according with (33), we can define

$$\rho^* = \min \{\rho \in \{0, 1, 2, \dots\}\}$$

such that

$$\alpha_{k,*} \overset{\text{def}}{=} \tilde{\alpha}\left(\frac{1}{2}\right)^{\rho^*} \le \alpha_{k,j}$$

and

$$\varphi_{\mu_k}(x^{k,j} + \alpha_{k,*}d_x^{k,j}) \le \varphi_{\mu_k}(x^{k,j}) + \gamma\alpha_{k,*}\nabla\varphi_{\mu_k}(x^{k,j})^{\mathrm{T}}d_x^{k,j}, \tag{35}$$

for all $j \in \hat{\mathcal{J}}$, j large enough. Then,

$$
\begin{aligned}
\varphi_{\mu_k}(x^{k,j+1}) &\le \varphi_{\mu_k}(x^{k,j}) + \gamma\alpha_{k,j}\nabla\varphi_{\mu_k}(x^{k,j})^{\mathrm{T}}d_x^{k,j} \\
&\le \varphi_{\mu_k}(x^{k,j}) - \gamma\alpha_{k,j}\sigma\|d_x^{k,j}\|^2, \text{ by } (32) \\
&\le \varphi_{\mu_k}(x^{k,j}) - \tfrac{1}{2}\gamma\alpha_{k,j}\sigma\|d_x^{k,*}\|^2, \text{ by } (31) \\
&\le \varphi_{\mu_k}(x^{k,j}) - \tfrac{1}{2}\gamma\alpha_{k,*}\sigma\|d_x^{k,*}\|^2, \text{ for } j \text{ large enough,}
\end{aligned}
$$

which implies that

$$\lim_{j\in\mathcal{J}} \varphi_{\mu_k}(x^{k,j}) = -\infty,$$

contradicting Assumptions 2 and 3 .   □

Theorem 1 establishes the global convergence result for Algorithm 2.

Theorem 1 *Suppose that Assumptions* 1, 2, 3, *and* 4 *hold. Then, any limit point of the sequence* $\{x^{k,j} + \alpha_{k,j}d_x^{k,j}, \lambda^{k,j} + d_\lambda^{k,j}, z_\ell^{k,j} + \alpha_{k,j}^{z_\ell}d_{z_\ell}^{k,j}, z_u^{k,j} + \alpha_{k,j}^{z_u}d_{z_u}^{k,j}\}$ *generated by Algorithm* 2 *satisfies the first-order optimality conditions* (8) *for problem* (3).

Proof Let $(x^{k,*}, \lambda^{k,*}, z_\ell^{k,*}, z_u^{k,*})$ be any limit point of the sequence

$$\left\{ x^{k,j} + \alpha_{k,j}d_x^{k,j}, \lambda^{k,j} + d_\lambda^{k,j}, z_\ell^{k,j} + \alpha_{k,j}^{z_\ell}d_{z_\ell}^{k,j}, z_u^{k,j} + \alpha_{k,j}^{z_u}d_{z_u}^{k,j} \right\},$$

namely the subsequence whose indexes belong to the infinite set $\mathcal{J} \underset{\infty}{\subset} \mathbb{N}$. Taking the limit in (16) for $j \in \mathcal{J}$, by Lemma 7 we have that

$$A^{\mathrm{T}}\lambda^{k,*} = -\nabla\varphi_{\mu_k}(x^{k,*}) = -\nabla f(x^{k,*}) + \mu_k L_{k,*}^{\dagger}e - \mu_k U_{k,*}^{\dagger}e.$$

In other words,

$$\nabla f(x^{k,*}) + A^{\mathrm{T}}\lambda^{k,*} - \mu_k L_{k,*}^{\dagger}e + \mu_k U_{k,*}^{\dagger}e = 0. \tag{36}$$

Lemma 7 implies that, for $j \in \mathcal{J}$ large enough, $d_x^{k,j}$ will be small enough and, for this reason, conditions (23) will be satisfied and $\alpha_{k,j}^{z_\ell} = \alpha_{k,j}^{z_u} = 1$. Therefore, taking limits in (10) for $j \in \mathcal{J}$ and considering (17c) and (17d), we have that

$$z_\ell^{k,*} = \mu_k L_{k,*}^{\dagger}e \text{ e } z_u^{k,*} = \mu_k U_{k,*}^{\dagger}e. \tag{37}$$

Thus, (36) and (37) together imply that

$$\begin{aligned}
\nabla f(x^{k,*}) + A^{\mathrm{T}}\lambda^{k,*} - z_\ell^{k,*} + z_u^{k,*} &= 0, \\
L_{k,*}z_\ell^{k,*} - \mu_k e &= 0, \\
U_{k,*}z_u^{k,*} - \mu_k e &= 0.
\end{aligned} \tag{38}$$

Therefore, since $Ax^{k,*} = b$, (38) gives us that (8) holds in $(x^{k,*}, \lambda^{k,*}, z_\ell^{k,*}, z_u^{k,*})$. $\square$

Now, we are ready to show that Algorithm 1 is well defined. First, by Assumption 1, it is always possible to find an initial point $x^0 \in \mathcal{F}_0$. Thus, to prove the well definiteness of Algorithm 1, it is enough to check that Step 1.2 is well defined, which is closely related to the well definiteness of Algorithm 2. The next result completes the analysis.

Lemma 8 *Consider that Assumptions 1, 2, 3, and 4 hold. Then, for all k, it is possible to find $(x^{k+1}, \lambda^{k+1}, z_\ell^{k+1}, z_u^{k+1})$ in finite time using Algorithm 2 such that*

$$E_{\mu_k}(x^{k+1}, \lambda^{k+1}, z_\ell^{k+1}, z_u^{k+1}) \leq \kappa_\varepsilon \mu_k.$$

Proof First, notice that $\mu_0 > 0$ and, from Step 1.3, $\mu_k > 0$ for all k. Thus, we have that $\kappa_\varepsilon \mu_k > 0$ for all k. On the other hand, for each k, Theorem 1 implies that, as j tends to infinity, the sequence generated by Algorithm 2 converges to a point $(x^{k,*}, \lambda^{k,*}, z_\ell^{k,*}, z_u^{k,*})$ such that $E_{\mu_k}(x^{k,*}, \lambda^{k,*}, z_\ell^{k,*}, z_u^{k,*}) = 0$ (according with (38)). Therefore, for j large enough, Algorithm 2 can find a point $(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$ such that $E_{\mu_k}(x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j}) \leq \kappa_\varepsilon \mu_k$. Consequently, it is possible to find $(x^{k+1}, \lambda^{k+1}, z_\ell^{k+1}, z_u^{k+1}) = (x^{k,j}, \lambda^{k,j}, z_\ell^{k,j}, z_u^{k,j})$ in finite time with Algorithm 2. $\square$

With the previous results, we can establish the global convergence of Algorithm 1. Once again, with some abuse of notation, we refer to the infinite sequence generated by the method that emerges when the stopping criterion (Step 1.1) is removed from Algorithm 1 as "the infinite sequence generated by Algorithm 1".

Theorem 2 *Consider that Algorithm 1 generates an infinite sequence of iterates and that Assumptions 1, 2, 3, and 4 hold. If the sequence generated by Algorithm 1 admits a limit point $(x^*, \lambda^*, z_\ell^*, z_u^*)$, then*

$$E_0(x^*, \lambda^*, z_\ell^*, z_u^*) = 0, \tag{39}$$

with $E_\mu(x, \lambda, z_\ell, z_u)$ defined in (19).

Proof Let $\mathcal{K} \underset{\infty}{\subset} \mathbb{N}$ be an infinite set such that

$$\lim_{k \in \mathcal{K}} (x^{k+1}, \lambda^{k+1}, z_\ell^{k+1}, z_u^{k+1}) = (x^*, \lambda^*, z_\ell^*, z_u^*). \tag{40}$$

Suppose that (39) does not hold. Then,

$$\|\nabla f(x^*) + A^{\mathrm{T}}\lambda^* - z_\ell^* + z_u^*\|_\infty > 0 \text{ or} \tag{41a}$$

$$\|L_* Z_*^\ell e\|_\infty > 0 \text{ or} \tag{41b}$$

$$\|U_* Z_*^u e\|_\infty > 0. \tag{41c}$$

By Step 1.2 of Algorithm 1, we have that

$$E_{\mu_k}\left(x^{k+1}, \lambda^{k+1}, z_\ell^{k+1}, z_u^{k+1}\right) \le \kappa_\varepsilon \mu_k,$$

for all k, which means that

$$\|\nabla f(x^{k+1}) + A^{\mathrm{T}} \lambda^{k+1} - z_\ell^{k+1} + z_u^{k+1}\|_\infty \le \kappa_\varepsilon \mu_k \text{ and} \tag{42a}$$

$$\|L_{k+1} Z_{k+1}^\ell e - \mu_k e\|_\infty \le \kappa_\varepsilon \mu_k \text{ and} \tag{42b}$$

$$\|U_{k+1} Z_{k+1}^u e - \mu_k e\|_\infty \le \kappa_\varepsilon \mu_k, \tag{42c}$$

for all k. Since $\mathcal{K} \underset{\infty}{\subseteq} \mathbb{N}$, we have, by Step 1.3 of Algorithm 1, that

$$\lim_{k \in \mathcal{K}} \mu_k = 0. \tag{43}$$

From (40), (41b), and (42b), for all k large enough, we have that

$$0 < \|L_{k+1} Z_{k+1}^\ell e\|_\infty - \mu_k \le \kappa_\varepsilon \mu_k,$$

which yields $0 < \mu_k(\kappa_\varepsilon + 1)$. It is a contradiction, since (43) holds. Analogously, (40), (41c), and (42c) produce the same contradiction.

On the other hand, from (40), (41a), and (42a), for all k large enough,

$$0 < \|\nabla f(x^{k+1}) + A^T \lambda^{k+1} - z_\ell^{k+1} + z_u^{k+1}\|_\infty \le \kappa_\varepsilon \mu_k,$$

which implies that $0 < \kappa_\varepsilon \mu_k$, being also in contradiction with (43). Therefore, (41) cannot occur, implying that (39) holds. □

Theorem 2 assures that, given any sequence generated by Algorithm 1, if such a sequence admits a limit point, then this point satisfies the set of equations in (8) with $\mu = 0$. Consequently, this limit point also satisfies the KKT conditions for the original problem (1), since z_ℓ^* and z_u^* are nonnegative and, by the definition of the method, they satisfy the complementarity relations in (8c) and (8d) with $\mu = 0$. Therefore, the next result is obtained.

Corollary 1 *Suppose Algorithm 1 generates an infinite sequence of iterates and that Assumptions* 1, 2, 3, *and* 4 *hold. If the sequence generated by Algorithm 1 admits any limit point* $(x^*, \lambda^*, z_\ell^*, z_u^*)$, *then this point satisfies the KKT conditions for problem* (1).

Table 1 Distribution of the number of variables n and the number of constraints m in the considered 200 problems from the CUTEst collection

n_{max}	m_{max}	# Problems with $n \geq \omega n_{max}$			# Problems with $m \geq \omega m_{max}$		
		$\omega = 0.1$	$\omega = 0.01$	$\omega = 0.001$	$\omega = 0.1$	$\omega = 0.01$	$\omega = 0.001$
$251,001$	$250,498$	55	123	147	6	100	132

Additionally, the convex case yields the following result.

Corollary 2 *Suppose Algorithm 1 generates an infinite sequence of iterates, that Assumptions 1, 2, 3, and 4 hold, and that, in addition, the objective function f is convex. If the sequence generated by Algorithm 1 admits any limit point $(x^*, \lambda^*, z_\ell^*, z_u^*)$, then this point is a global minimizer for problem (1).*

Proof The proof follows from Corollary 1 and the fact that, in the convex case, every KKT point is a global minimizer (see Nocedal and Wright 2006). ☐

4 Implementation and numerical experiments

We now present numerical experiments to evaluate the performance of Algorithms 1 and 2. We consider all the 200 problems from CUTEst collection (Gould et al. 2015) with linear equality and box constraints. Table 1 displays the distribution of the number of variables n and the number of constraints m in the considered set of problems. It should be noted that, in all the problems, a constraint of the form, $\ell_i \leq x_i \leq u_i$ with $-10^{20} \leq \ell_i \leq u_i \leq 10^{20}$ for $i = 1, \dots, n$ is present; this being a sufficient condition for the satisfaction of Assumption 3.

We implemented Algorithms 1, 2, and 3, referred as Lcmin from now on, in Fortran 2008. The codes are freely available in the web.[1] Tests were conducted in an Intel Core i7-8700 3.20GHz processor with 32 GB RAM, running Ubuntu 18.04.3 LTS operating system. Codes were compiled using the GNU Compiler Collection version 7.4.0 with -O3 flag enabled.

In practice, we consider a scaled version of the stopping criterion (20) at Step 1.1 of Algorithm 1 given by

$$E_0^s(x^k, \lambda^k, z_\ell^k, z_u^k) \leq \varepsilon_{tol}, \tag{44}$$

where $s = (s_d, s_\ell, s_u)$,

---

[1] https://github.com/johngardenghi/lcmin.

$$s_d \stackrel{\text{def}}{=} \max \left\{ s_{\max}, \frac{\|\lambda\|_1 + \|z_\ell\|_1 + \|z_u\|_1}{m + 2n} \right\} / s_{\max},$$

$$s_\ell \stackrel{\text{def}}{=} \max \left\{ s_{\max}, \frac{\|z_\ell\|_1}{n} \right\} / s_{\max},$$

$$s_u \stackrel{\text{def}}{=} \max \left\{ s_{\max}, \frac{\|z_u\|_1}{n} \right\} / s_{\max},$$

$s_{\max} \geq 1$ is a given constant, and

$$E_\mu^s(x, \lambda, z_\ell, z_u) \stackrel{\text{def}}{=} \max \left\{ \frac{\|\nabla f(x) + A^T \lambda - z_\ell + z_u\|_\infty}{s_d}, \frac{\|LZ^\ell e - \mu e\|_\infty}{s_\ell}, \frac{\|UZ^u e - \mu e\|_\infty}{s_u} \right\}.$$

In theory, all iterates of Lcmin are feasible. However, in practice, numerical errors may lead to some loss of feasibility. For this reason, once the stopping criterion (44) has been satisfied, we check the value of $\|Ax^k - b\|_\infty$. We consider "the problem has been solved" (stopping criterion SC1) if

$$\|Ax - b\|_\infty \leq \varepsilon_{\text{feas}},$$

where $\varepsilon_{\text{feas}} > 0$ is a given constant. If

$$\|Ax - b\|_\infty \leq \sqrt{\varepsilon_{\text{feas}}},$$

we say that "an acceptable feasible point was obtained" (stopping criterion SC2). Otherwise, we declare that "convergence to an infeasible point was obtained" (stopping criterion SC3). In addition to (44), Lcmin also stops whenever

SC4: $\|x^{k,j}\|_\infty \geq \kappa_x$, where κ_x is a large positive given value;
SC5: $k \geq k_{\max}$, where $k_{\max} > 0$ is given; or
SC6: $\mu_k \leq \varepsilon_{\text{tol}}/10$ and $j \geq j_{\max}$, where $j_{\max} > 0$ is given.

In the experiments, following Wächter and Biegler (2006), we set $\mu_0 = 0.1$, $\varepsilon_{\text{tol}} = 10^{-8}$, $\kappa_\varepsilon = 10$, $\kappa_\mu = 0.2$, $\theta_\mu = 1.5$, $\gamma = 10^{-4}$, $\tau_{\min} = 0.99$, $\kappa_z = 10^{10}$, $\kappa_\xi^- = 1/3$, $\kappa_\xi^+ = 8$ e $\bar{\kappa}_\xi^+ = 100$, $\xi_{\text{ini}} = 10^{-4}$, $\xi_{\min} = 10^{-20}$, $\xi_{\max} = 10^{20}$, $s_{\max} = 100$, $\varepsilon_{\text{feas}} = 10^{-8}$, $\kappa_x = 10^{20}$, $k_{\max} = 50$, and $j_{\max} = 200$. Three implementation features are in order. Routine HSL MA57[2] was used to solve the linear systems. Matrix A of the constraints of problem (1) may not have full row rank as required, and may even be such that $m > n$. Thus, routine HSL MC58[3] was used to check whether (i) $\text{rank}(A) = m$; (ii); $\text{rank}(A) < m$ and $\text{rank}(A) = \text{rank}(A|b)$; or (iii) $\text{rank}(A) < m$ and $\text{rank}(A) \neq \text{rank}(A|b)$. In the first case, A satisfies the full row-rank assumption and there is nothing to be done. In the second case, constraints $Ax = b$ are replaced by an equivalent set of constraints $\bar{A}x = \bar{b}$ in which $\bar{A}$ satisfies the full row-rank assumption ($\bar{A}$ is given by routine MC58 and $\bar{b}$ can be easily computed). In the third case, the problem is infeasible and there is nothing to be done.

---

[2] http://www.hsl.rl.ac.uk/catalogue/hsl_ma57.html.
[3] Available at http://www.hsl.rl.ac.uk/catalogue/mc58.html.

Table 2 Number of problems in which Ipopt and Lcmin found a point satisfying (47) with $\varepsilon_{\text{feas}} = 10^{-8}$ and (48) with $\varepsilon_{\text{bnd}} \in \{10^{-1}, 10^{-2}, \ldots, 10^{-16}, 0\}$

$\log_{10}(\varepsilon_{\text{bnd}})$																ε_{bnd}
−1	−2	−3	−4	−5	−6	−7	−8	−9	−10	−11	−12	−13	−14	−15	−16	0
Ipopt 154	153	151	150	143	141	137	126	65	65	65	65	65	64	64	64	64
Lcmin 174	174	174	174	174	174	174	174	174	174	174	174	174	174	174	174	174

(Infeasibility was detected in 6 out of the 200 problems at this pre-processing stage.) Finally, an interior point $x^0 \in \mathcal{F}_0$ is required to start Algorithm 1. For this reason, we tried to find such a point using a *phase 1* procedure, that consists in approximately solving the feasibility problem

$$Ax = b \text{ plus } \ell_i + [\delta_\ell]_i \leq x_i \leq u_i - [\delta_u]_i \text{ for } i = 1, \ldots, n, \tag{45}$$

with

$$\begin{aligned} \left[\delta_\ell\right]_i &= \min\{\kappa_1 \max\{1, |\ell_i|\}, \kappa_2(u_i - \ell_i)\}, \\ \left[\delta_u\right]_i &= \min\{\kappa_1 \max\{1, |u_i|\}, \kappa_2(u_i - \ell_i)\}, \end{aligned}$$

for $i = 1, \ldots, n$, $\kappa_1 > 0$, and $\kappa_2 \in (0, \frac{1}{2})$. To approximately solve (45), we apply ALGENCAN (Andreani et al. 2008; Birgin and Martínez (2014)) with the option IGNORE-OBJECTIVE-FUNCTION enabled. The phase 1 procedure starts from the given initial point, making it somehow useful in the computation of the initial interior point. (Infeasibility was detected in phase 1 for only 1 problem out of the remaining $194 = 200 - 6$ problems.)

We have applied Ipopt (Wächter and Biegler 2006), version 3.12.13, within the same computational environment, also using the HSL MA57 routine for solving the linear systems, taking into account the same time budget for each problem, and considering all its default parameters, except for `honor_original_bounds no`. Such a parameter, which does not affect the overall performance of Ipopt, inhibits this solver to project[4] the final iterate onto the box defined by the bound constraints of problem (1), allowing us to measure the violation of the bounds at the final iterate. Additional experiments with Ipopt considering the default choice `honor_original_bounds yes` were also carried on; the comparison showed results qualitatively similar to those reported below.

Detailed output of both methods for each one of the 200 problems, as well as tables summarizing the results, with a CPU time budget of 10 minutes per problem, can be viewed at the same repository the code is located[5]. Since the methods under analysis have different stopping criteria, we consider that a problem $p \in \{1, 2, \ldots, 200\}$ is solved by a method $M \in \{\text{Ipopt}, \text{Lcmin}\}$ if

---

[4] The bound constraints might be dynamically relaxed by Ipopt during the optimization process (Wächter and Biegler (2006), §3.5), starting from a relative relaxation factor whose initial value is 10^{-8}.

[5] https://github.com/johngardenghi/lcmin/tree/master/paper.

Table 3 Number of problems in which Ipopt and Lcmin found a point satisfying (47) with $\varepsilon_{\text{feas}} = 10^{-8}$, (48) with $\varepsilon_{\text{bnd}} = 0$, and (46) with $f_{\text{tol}} \in \{10^{-1}, 10^{-2}, \ldots, 10^{-8}, 0\}$

	$\log_{10}(f_{\text{tol}})$								f_{tol}
	-1	-2	-3	-4	-5	-6	-7	-8	0
Ipopt	61	60	58	58	58	57	57	56	42
Lcmin	58	58	58	58	58	58	55	52	29

$$f_M^p \leq f_{\min}^p + f_{\text{tol}} \max\{1, |f_{\min}^p|\}, \tag{46}$$

where $f_{\min}^p = \min\{f_{\text{Ipopt}}^p, f_{\text{Lcmin}}^p\}$, and $f_{\text{tol}} \in [0, 1]$,

$$\|Ax - b\|_\infty \leq \varepsilon_{\text{feas}}, \tag{47}$$

with $\varepsilon_{\text{feas}} \geq 0$, and

$$\max\{\|(\ell - x)_+\|_\infty, \|(x - u)_+\|_\infty\} \leq \varepsilon_{\text{bnd}}, \tag{48}$$

with $\varepsilon_{\text{bnd}} \geq 0$ and $(\cdot)_+ = \max\{\cdot, 0\}$.

We first take a close look at the feasibility of the final iterate found by the methods. In Table 2, we show the number of problems in which each method found a point satisfying (47) with $\varepsilon_{\text{feas}} = 10^{-8}$ and (48) with $\varepsilon_{\text{bnd}} \in \{10^{-1}, 10^{-2}, \ldots, 10^{-16}, 0\}$, no matter the objective function value. Since Lcmin preserves feasibility during all the optimization process, the amount of problems whose bound constraints are satisfied does not depend on ε_{bnd}. On the other hand, the number of problems whose bound constraints hold for Ipopt varies according to the tolerance ε_{bnd}. The $26 = 200 - 174$ failures in Lcmin correspond to (i) 7 problems detected as being infeasible, 6 in the pre-processing of the coefficients' matrix A and 1 during phase 1; (ii) 7 problems in which Lcmin generated a final iterate whose feasibility does not satisfy (47) with $\varepsilon_{\text{feas}} = 10^{-8}$; and (iii) 12 problems in which Lcmin exceeded the 10 min established as CPU time budget. When $\varepsilon_{\text{bnd}} = 0.1$, the $46 = 200 - 154$ failures in Ipopt correspond to (i) 10 problems in which Ipopt generated a final iterate that does not satisfy (47) with $\varepsilon_{\text{feas}} = 10^{-8}$; (ii) 13 problems in which Ipopt exceeded the 10 min established as CPU time budget; and (iii) 23 problems to which Ipopt is not applicable because of the degree of freedom of A in the constraints of the problem[6]. For other values of ε_{bnd}, the increasing number of failures is due to the bound constraints violation at the final iterate.

Lcmin detected the problem is infeasible at phase 1 in 7 problems; and it exceeded the CPU time limit of 10 min in 13 problems. In the remaining 180 problems, it stopped satisfying the stopping criteria SC1, SC2, ..., SC6 in 168, 6, 0, 0, 3, and 3 problems, respectively. As a consequence, it found a feasible point (satisfying (47) with $\varepsilon_{\text{feas}} = 10^{-8}$ and (48) with $\varepsilon_{\text{bnd}} = 0$) in 174 out of the 200 considered

---

[6] It means that, in problem (1), A has more rows than columns. Lcmin eliminates redundant constraints, which makes A to have full row rank in most cases, except in those in which the feasible set is empty. Ipopt does not start optimization in these cases, stopping with the output `Problem has too few degrees of freedom`.
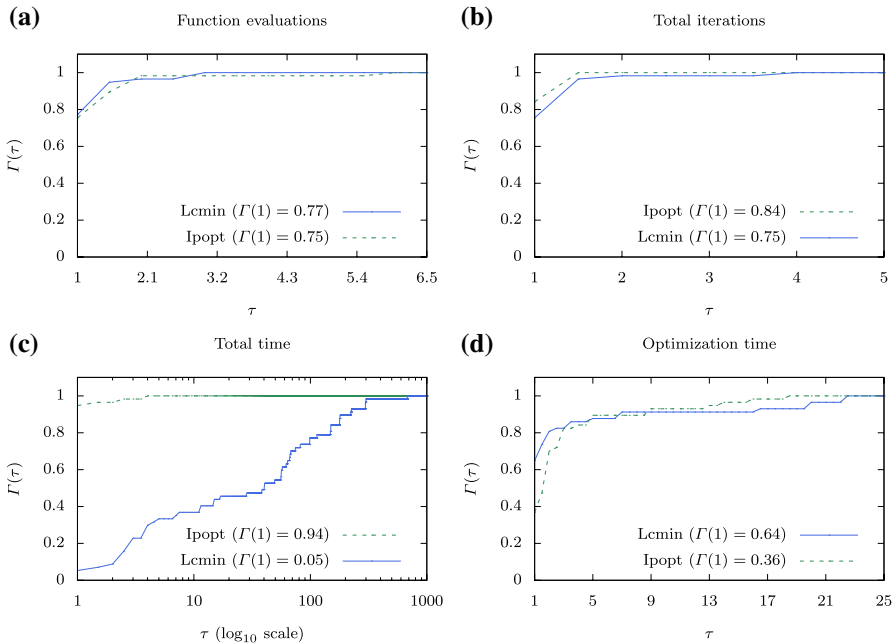
(a)

(b)

(c)

(d)

Fig. 1 Performance profiles comparing **a** the number of functional evaluations, **b** the number of iterations, and c–d the CPU time of Lcmin and Ipopt in the subset of 57 problems in which both solvers found iterates satisfying (46) with $f_{tol} = 0.1$, (47) with $\varepsilon_{feas} = 10^{-8}$, and (48) with $\varepsilon_{bnd} = 0$. In **d** the CPU time spent by Lcmin to find a feasible initial point has been ignored

problems. Considering these 174 problems, Lcmin performed, in average, 6.35 outer iterations (being 51 the maximum) and 30.54 inner iterations (being 610 the maximum) per problem. In 132 out of the 174, the inertia of the matrix of coefficients of the linear system (16) was never corrected, meaning that a single matrix factorization per iteration was performed. In the remaining 42 problems, the average number of matrix factorizations per iteration was 1.26.

Now, we are interested in those problems in which both Ipopt and Lcmin converged to a point satisfying (47) with $\varepsilon_{feas} = 10^{-8}$ and (48) with $\varepsilon_{bnd} = 0$. For this set, composed by 62 problems, Table 3 shows, for each solver, the number of problems in which (46) holds with $f_{tol} \in \{10^{-1}, 10^{-2}, \ldots, 10^{-8}, 0\}$.

We now consider, on the one hand, the set of 57 problems in which both Lcmin and Ipopt found a final iterate satisfying (46) with $f_{tol} = 0.1$, (47) with $\varepsilon_{feas} = 10^{-8}$, and (48) with $\varepsilon_{bnd} = 0$. Figure 1 depicts, for these problems, the performance profiles Dolan and Moré (2002) using as performance measure the number of functional evaluations, the number of iterations, and the CPU time consumed by each solver. Considering the remaining $143 = 200 - 57$ problems, we have that: *(i)* in 24 problems, none of the methods found a point satisfying (47) with $\varepsilon_{feas} = 10^{-8}$ and (48) with $\varepsilon_{bnd} = 0$; **(ii)** in 2 problems, Ipopt found a point satisfying (47) with $\varepsilon_{feas} = 10^{-8}$ and (48) with $\varepsilon_{bnd} = 0$, while Lcmin failed; **(iii)** in 112 problems, Lcmin found a point satisfying (47) with $\varepsilon_{feas} = 10^{-8}$

and (48) with $\varepsilon_{\text{bnd}} = 0$, while Ipopt failed; (iv) in 5 problems both found a point satisfying (47) with $\varepsilon_{\text{feas}} = 10^{-8}$ and (48) with $\varepsilon_{\text{bnd}} = 0$, but the objective functional value of one of them does not satisfy (46) with $f_{\text{tol}} = 0.1$; (v) regarding the 5 problems mentioned in (iv), the objective function found by Ipopt was smaller than the objective functional value find by Lcmin in 4 problems, while the opposite situation occurred in 1 problem.

5 Final remarks

In this work, a feasible line-search interior-point method for linearly constrained optimization has been described, implemented, and analyzed. The global convergence theory is accompanied with numerical experiments, encompassing a classical test set from the literature. The performance of the proposed algorithm was put into perspective with Ipopt, a current state-of-the-art solver.

No winner emerged from the comparative results, which was somehow expected, since both methods have the interior-point strategy as the main principle. Nevertheless, we point out that feasibility may be an issue: while the general purpose solver Ipopt may relax bounds, Lcmin always guarantees a feasible final iterate, except when numerical difficulties may occur, as in the 7 cases of failure of Lcmin in the numerical experiments, which evidences that the problem is numerically difficult or even numerically infeasible. Therefore, Lcmin is recommended for applications in which feasibility is a desired feature.

References

Andreani R, Birgin EG, Martínez JM, Schuverdt ML (2008) On augmented Lagrangian methods with general lower-level constraints. SIAM J Optim 18(4):1286–1309

Birgin EG, Martínez JM (2014) Practical augmented Lagrangian methods for constrained optimization. Society for Industrial and Applied Mathematics, Philadelphia

Chen L, Goldfarb D (2006) Interior-point ℓ_2-penalty methods for nonlinear programming with strong global convergence properties. Math Progr 108(1):1–36

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Math Progr 91(2):201–213

Fiacco AV, McCormick GP (1968) Nonlinear programming: sequential unconstrained minimization techniques. Wiley, New York (1968, Reprinted by SIAM Publications, 1990)

Frisch KR (1955) The logarithmic potential method of convex programming. Tech. rep., University Institute of Economics, Oslo, Norway

Gill PE, Murray W, Saunders MA, Tomlin JA, Wright MH (1986) On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. Math Progr 36(2):183–209

Gould NIM, Orban D, Toint PL (2015) CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput Optim Appl 60(3):545–557

Karmarkar N (1984) A new polynomial-time algorithm for linear programming. In: Proceedings of the Sixteenth Annual ACM Symposium on theory of computing, STOC '84, pp. 302–311. ACM, New York, NY, USA

Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer, New York

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.