

An augmented Lagrangian interior-point method using directions of negative curvature

Abstract. We describe an efficient implementation of an interior-point algorithm for non-convex problems that uses directions of negative curvature. These directions should ensure convergence to second-order KKT points and improve the computational efficiency of the procedure. Some relevant aspects of the implementation are the strategy to combine a direction of negative curvature and a modified Newton direction, and the conditions to ensure feasibility of the iterates with respect to the simple bounds. The use of multivariate barrier and penalty parameters is also discussed, as well as the update rules for these parameters. We analyze the convergence of the procedure; both the linesearch and the update rule for the barrier parameter behave appropriately. As the main goal of the paper is the practical usage of negative curvature, a set of numerical results on small test problems is presented. Based on these results, the relevance of using directions of negative curvature is discussed.

Key words. Primal-dual methods – Nonconvex optimization – Linesearches

1. Introduction

We are interested in developing an algorithm to compute local solutions for nonlinear, and possibly non-convex, problems of the form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & x \geq 0, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $c : \mathbb{R}^n \mapsto \mathbb{R}^m$. More specifically, we wish to compute second-order KKT points for problem (1), that is, points that can be assured to satisfy both the first-order conditions and the second-order necessary conditions [20].

The use of directions of negative curvature plays a crucial role in this context; only by considering this second-order information it is possible to ensure convergence to such points. Trust-region methods can take negative curvature into account provided that a sufficiently accurate approximation to an eigenvector corresponding to a (sufficiently) negative eigenvalue of the Hessian matrix can be computed [9]. Linesearch procedures,

J.M. Moguerza: School of Engineering, Univ. Rey Juan Carlos, Madrid, Spain, e-mail: j.moguerza@escet.urjc.es

Research supported by Spanish MEC grant TIC2000-1750-C06-04

F.J. Prieto: Dept. of Statistics and Econometrics, Univ. Carlos III de Madrid, Spain, e-mail: fjp@est-econ.uc3m.es

Research supported by Spanish MEC grant BEC2000-0167

Mathematics Subject Classification (1991): 49M37, 65K05, 90C30

while presenting interesting properties from a practical point of view, must consider second-order information by explicitly computing some direction of negative curvature, if it is available. The idea of using directions of negative curvature was proposed by Fiacco and McCormick [12]. Later, Moré and Sorensen [25] described how to modify Newton's method to incorporate this second-order information.

If exact second derivatives are used, the explicit computation of these directions can be carried out with limited cost, from an appropriate factorization of the coefficient matrix in the system of Newton equations, see [18] for example. Nevertheless, the requirement to obtain both descent and negative curvature directions from the Newton system of equations limits the choice of numerical procedures that can be used to compute the search direction.

In this paper we will be concerned with deriving a linesearch algorithm that uses negative curvature, extending the ideas of Moré and Sorensen [25] to nonlinearly constrained problems. The directions required to update the iterates will be generated using an interior-point approach. In this setting, problem (1) is transformed into a sequence of equality constrained problems of the form (see [12])

$$\begin{aligned} \min_x \quad & f(x) - \sum_{i=1}^n \mu_i \log x_i \\ \text{s.t.} \quad & c(x) = 0. \end{aligned} \tag{2}$$

The search directions are computed to approximate the solutions of these barrier problems. We have chosen to use a vector of barrier parameters $\mu \in \mathbb{R}^n$, one for each simple bound $x \geq 0$.

Interior-point methods have proved to be very successful for the solution of linear and general convex problems. More recently, a significant amount of effort has been devoted to extending these procedures to non-convex problems, see for example El-Bakry et al. [10], Gajulapalli [14], Gay et al. [16], Vanderbei and Shanno [28], Yamashita [32], among others. Nevertheless, very few of these proposals have taken into consideration the use of negative curvature directions.

Once directions of negative curvature have been obtained, it is still necessary to combine them with traditional descent directions. In general, iterates will fail to satisfy both first-order and second-order conditions; it is important to make use of both types of information simultaneously to ensure the efficiency of the procedure. The combination of descent and negative curvature directions has been considered by McCormick [23] and Moré and Sorensen [25] for the unconstrained case using linesearch approaches. Trust-region methods for the unconstrained case have been proposed by Byrd, Schnabel and Schultz [8]. Forsgren and Murray [13] have proposed a method for the linear equality-constrained case. Other approaches to ensure convergence to second-order KKT points for the constrained case are described in [1, 2, 5, 15]. Designing an efficient procedure to obtain a satisfactory combination of these directions poses considerable difficulties, as the Newton direction is well-scaled in general, and specifically near a stationary point, while directions of negative curvature have no inherent scale. Note that from a theoretical point of view this problem is not particularly relevant, but from a practical point of view it may have a large impact on the efficiency of the algorithm. Rather than attempting to provide solutions to this complication, our aim in this paper

has been to offer some suggestions on how to use these directions to reduce the impact of the scaling problem, and at the same time to improve the performance of methods that use negative curvature.

As we already mentioned, the approach implemented in the proposed algorithm will be similar to that in Moré and Sorensen [25]. In this approach we generate iterates from combinations of descent directions and negative curvature directions. These iterates should have reasonable properties, such as global convergence. A linesearch has been introduced in our algorithm as a mechanism to enforce this property. We will compute the iterates in such a manner that the value of an augmented Lagrangian merit function is decreased in each iteration. For problem (2) this merit function takes the form

$$L_A(x, \lambda; \rho, \mu) = f(x) - \sum_{i=1}^n \mu_i \log x_i - \lambda^T c(x) + \frac{1}{2} \sum_{j=1}^m \rho_j c_j(x)^2. \quad (3)$$

The penalty term in the merit function is defined using a vector of penalty parameters, one for each constraint, $\rho \in \mathbb{R}^m$. This function has been extensively studied by Bertsekas [3] among others. It has the advantage of being differentiable at all points where it is defined (the interior of the positive orthant). Also, under suitable assumptions the local minimizers for problem (2) are minimizers for this merit function, if all components of ρ are large enough.

This merit function introduces variables and parameters, λ and ρ , that have to be updated through the successive iterations in the algorithm to ensure convergence. Analogously, the barrier problems (2) also include parameters μ that must be updated. The choice of updating strategies may affect significantly the efficiency of the overall procedure and will be considered in some detail in the following sections.

Our proposal will address the three issues discussed in the preceding paragraphs: the definition of the search directions, their combination and the updating of the parameters, in a manner that produces an efficient and robust procedure. We intend to do this from a practical point of view, that is, we wish to derive a procedure that is efficient in practice, and we check that this is the case by conducting numerical tests on a set of small problems. We are also interested in studying the impact that using negative curvature information may have in practice on the efficiency of the algorithm. Although the paper is mostly numerically oriented, we will also present some theoretical results on the convergence of the algorithm. These results should help to justify some of our implementation choices.

The paper is organized as follows: In Section 2 we introduce the general results and the notation used to motivate and describe the algorithm, as well as its general scheme. Section 3 describes the procedure to compute the search directions. In Section 4 we indicate how to combine the directions to obtain the next iterate. Section 5 gives the rules for updating the algorithm parameters. In Section 6 we derive some convergence results. Section 7 discusses implementation issues and gives the structure of the algorithm. Finally, in Section 8 we present and comment some computational results on a problem test set.

2. Notation and general comments

The first-order Karush-Kuhn-Tucker (KKT) conditions for problem (1) are:

$$\begin{aligned} \nabla f(x) - \nabla c^T(x)\lambda - z &= 0, \\ c(x) &= 0, \\ Zx &= 0, \\ x, z &\geq 0, \end{aligned} \tag{4}$$

where λ and z are the multipliers for the equality and bound constraints respectively, and Z denotes a diagonal matrix having as entries the elements of z , $Z = \text{diag}(z)$.

In the proposed algorithm, instead of considering directly the preceding conditions, we solve a sequence of problems (2) such that $\mu_i \rightarrow 0$ for all i , following [12]. The first-order KKT conditions for (2) are:

$$\begin{aligned} \nabla f(x) - \nabla c^T(x)\lambda - X^{-1}\mu &= 0, \\ c(x) &= 0, \end{aligned} \tag{5}$$

where $X = \text{diag}(x)$. Replacing

$$z = X^{-1}\mu, \tag{6}$$

in the first equation of (5), the first-order KKT conditions for the barrier problem can be rewritten as:

$$\begin{aligned} \nabla f(x) - \nabla c^T(x)\lambda - z &= 0, \\ c(x) &= 0, \\ Zx &= \mu. \end{aligned} \tag{7}$$

The set of equations (7) is known as the primal-dual equations for problem (2). Initially implemented by Mehrotra [24], the search directions obtained from them have better theoretical and practical properties than those computed from (5). The algorithm will compute search directions based on these primal-dual KKT equations. In addition to trying to satisfy these conditions, to ensure that the logarithmic terms in the objective function of (2) are well defined, the algorithm should force the variables x_i to remain strictly positive. From the comparison of these conditions and (4), it will be of interest to have $z \geq 0$ in all iterations of the algorithm.

We also want to satisfy the necessary second-order condition. For problem (2) this condition requires that

$$W_A^T \left(\nabla_{xx} L(x, \lambda) + M X^{-2} \right) W_A \quad \text{p.s.d.}, \tag{8}$$

where L is defined as $L(x, \lambda) \equiv f(x) - \lambda^T c(x)$, M is a diagonal matrix with entries those of μ , $M = \text{diag}(\mu)$ and W_A has columns that form a basis for the null-space of $\nabla c(x)$.

We will use the notation $u = O(v)$ for two functions u and v of a common variable x whenever $\lim_{x \rightarrow 0} u(x)/v(x) \leq K$ for some positive constant K . Analogously, $u = o(v)$ will be used to indicate two functions satisfying $\lim_{x \rightarrow 0} u(x)/v(x) = 0$.

The algorithm we propose must carry out the following tasks in each iteration:

- The computation of search directions, both to improve the satisfaction of the first-order KKT conditions (7) based on Newton's method, and the satisfaction of the second-order condition (8) using directions of negative curvature for the merit function L_A .
- The combination of these directions to compute the next iterate, ensuring sufficient decrease for the augmented Lagrangian merit function (3).
- Finally, the updating of the multipliers λ and penalty parameters ρ in the merit function, and the barrier parameters μ associated with problem (2).

In the following sections we indicate how to conduct these tasks in an efficient manner. A schematic version of the algorithm, indicating the Sections of the paper where the different steps are described, is presented below.

<i>Initialize variables (x^0, λ^0, z^0), barrier (μ^0) and penalty (ρ^0) parameters</i>	<i>Section 7.3</i>
<i>repeat</i>	
<i>From the Newton primal-dual equations:</i>	<i>Sections 3.1, 3.2</i>
<i>Compute a descent direction, d_x^k, for the primal variables x</i>	
<i>Compute search directions, d_λ^k and d_z^k, for the multipliers λ and z</i>	
<i>Compute, if it exists, d_n^k, a direction of negative curvature</i>	<i>Section 3.3</i>
<i>Adjust the penalty parameter ρ^k</i>	<i>Section 5.2</i>
<i>Compute α_p using a curvilinear search</i>	<i>Section 4</i>
<i>Update the primal variables using</i> $x^{k+1} = x^k + \alpha_p^2 d_x^k + \alpha_p d_n^k$	
<i>Update the multipliers</i>	<i>Sections 4, 5.1</i>
<i>Decrease the barrier parameter vector μ^k</i>	<i>Section 5.3</i>
<i>until convergence</i>	<i>Section 7.2</i>

It can be shown that this algorithm has good global and local convergence properties, under reasonable assumptions on the problem. In Section 6 we show that the method converges to second-order KKT points for problem (1), under some assumptions on the penalty parameter. A more general analysis of the behavior of this penalty parameter can be derived by adapting the results in [26]. Also, the local convergence properties (super-linear convergence) could be derived in a straightforward manner from the arguments in [33].

3. Computation of the search directions

We will start by considering the computation of a descent direction for the variables x , d_x , based on a modified Newton method applied to the primal-dual equations (7).

3.1. The descent direction

Newton's method provides search directions d_x , d_λ and d_z , corresponding to update directions for the variables x , λ and z respectively. From the first-order Taylor series expansion for the primal-dual KKT conditions (7) about the values x , λ and z , the resulting system of linear equations defining the search directions is:

$$\begin{pmatrix} H & -\nabla c^T & -I \\ \nabla c & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_z \end{pmatrix} = \begin{pmatrix} -\nabla f + \nabla c^T \lambda + z \\ -c \\ \mu - Zx \end{pmatrix}, \quad (9)$$

where $H(x, \lambda) = \nabla_{xx} L(x, \lambda) = \nabla^2 f(x) - \sum_j \lambda_j \nabla^2 c_j(x)$, $Z = \text{diag}(z)$, I denotes the identity matrix and $X = \text{diag}(x)$.

From the last set of equations in (9), we have

$$d_z = X^{-1} \mu - z - X^{-1} Z d_x. \quad (10)$$

Replacing (10) in the first two sets of equations in (9), the movement direction d_x can be computed as the solution of the symmetric system

$$K \begin{pmatrix} d_x \\ -d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla f + \nabla c^T \lambda + X^{-1} \mu \\ -c \end{pmatrix}, \quad (11)$$

where K is defined as

$$K = \begin{pmatrix} G & \nabla c^T \\ \nabla c & 0 \end{pmatrix}, \quad (12)$$

for $G = H + X^{-1} Z$.

3.2. Solving the system of equations

The direction obtained from (11) may fail to provide descent for any reasonable merit function; for example, if the iterates are close to a stationary point that is not a minimizer. To ensure good global convergence properties for the algorithm it is important to adapt system (11) so that the direction d_x provides sufficient decrease for the merit function (3).

The gradient of the merit function is given by

$$\nabla_x L_A = \nabla f - X^{-1} \mu - \nabla c^T (\lambda - Rc), \quad (13)$$

where $R = \text{diag}(\rho)$. The Hessian of L_A with respect to x will be given by

$$\begin{aligned} \nabla_{xx} L_A &= \nabla^2 f + M X^{-2} - \sum_j (\lambda_j - \rho_j c_j) \nabla^2 c_j + \nabla c^T R \nabla c \\ &= \nabla_{xx} L(x, \lambda - Rc) + M X^{-2} + \nabla c^T R \nabla c. \end{aligned} \quad (14)$$

The Newton direction for the minimization of the merit function (3) will be given by

$$(\nabla_{xx} L(x, \lambda - Rc) + M X^{-2} + \nabla c^T R \nabla c) d_x = -(\nabla f - X^{-1} \mu - \nabla c^T (\lambda - Rc)). \quad (15)$$

As the minimization of the merit function is not a very efficient procedure to ensure the satisfaction of the constraints, the additional condition that $\nabla c(x)d_x = -c(x)$, already included in (11), is imposed. Using this condition, (15) now becomes

$$(\nabla_{xx}L(x, \lambda - Rc) + MX^{-2})d_x = -(\nabla f - X^{-1}\mu - \nabla c^T\lambda). \quad (16)$$

To ensure that d_x is a direction of descent for (3), we may replace the coefficient matrix $\nabla_{xx}L(x, \lambda - Rc) + MX^{-2}$ in (16) with a matrix $\bar{G}_\rho = \bar{H} + X^{-1}Z$ that is positive definite on the null space of the constraints $\nabla c(x)$, that is, such that $W_A^T \bar{G}_\rho W_A$ is positive definite. \bar{H} is a bounded approximation for $H(x, \lambda - Rc(x))$, obtained as described below.

The modified system used to define the search directions is then

$$\begin{pmatrix} \bar{G}_\rho & \nabla c^T \\ \nabla c & 0 \end{pmatrix} \begin{pmatrix} d_x \\ -d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla f + \nabla c^T(\lambda - Rc) + X^{-1}\mu \\ -c \end{pmatrix}, \quad (17)$$

where its coefficient matrix (and \bar{G}_ρ in particular) is computed from a modification of

$$K_\rho = \begin{pmatrix} G_\rho & \nabla c^T \\ \nabla c & 0 \end{pmatrix}, \quad (18)$$

for $G_\rho = \nabla_{xx}L(x, \hat{\lambda} - Rc) + X^{-1}Z$ and $\hat{\lambda}$ an approximation to λ .

The matrix \bar{G}_ρ in (17) can be generated in the process of factorizing K_ρ . Instead of a modified Cholesky factorization of the reduced Hessian $W_A^T G_\rho W_A$, see [16], we have chosen to use a version of the symmetric indefinite factorization on the full matrix K_ρ , see [7] for example, incorporating the modifications proposed in [13]. This alternative is able to obtain the desired modification for the reduced Hessian directly from system (17), it allows the computation of appropriate directions of negative curvature, as we will describe in Section 4, and it can be applied to medium-sized and large problems.

The modified factorization selects the pivots using rules that ensure that the rows of the Jacobian ∇c are considered first. In this manner, the modifications in the pivots will not affect the part of the system corresponding to this Jacobian matrix. For additional details see [13]. We now state the basic properties of this factorization. Assume that the LDL^T factorization of K_ρ , defined in (18), has been computed using the factorization algorithm described in [13], and that the matrix D in the factorization is partitioned into

$$D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}, \quad (19)$$

where D_1 and D_2 are block-diagonal matrices with 1×1 and 2×2 blocks and D_1 includes all the pivots chosen from elements of $\nabla c(x)$. The precise rules to choose these pivots can be found in [13]. For D_2 the Schur decomposition is computed, that is, $D_2 = U\Lambda U^T$, where U is a matrix whose columns are the eigenvectors of D_2 and Λ is a diagonal matrix of eigenvalues, π_i . This computation is straightforward since the blocks in D_2 are at most of size two. For a given $\varepsilon > 0$, define a diagonal matrix $\bar{\Lambda}$ having as its i -th diagonal element the value $\bar{\pi}_i = \max(|\pi_i|, \varepsilon)$. Construct \bar{D}_2 as $\bar{D}_2 = U\bar{\Lambda}U^T$ where $\bar{\Lambda} = \text{diag}(\bar{\pi}_i)$ and define \bar{D} as:

$$\bar{D} = \begin{pmatrix} D_1 & 0 \\ 0 & \bar{D}_2 \end{pmatrix}.$$

Let \tilde{d} be the solution of

$$L\tilde{D}L^T\tilde{d} = P^T b, \quad (20)$$

for P an appropriate permutation matrix, and b the right-hand side of (17). Finally let $d = P\tilde{d}$, the vector of unknowns in system (17),

$$d = \begin{pmatrix} d_x \\ -d_\lambda \end{pmatrix}.$$

It holds that $\nabla c d_x = -c$ and $W_A^T \bar{G}_\rho W_A$ is positive definite, where \bar{G}_ρ is the submatrix corresponding to the appropriate rows and columns of $L\tilde{D}L^T$, see [13].

The factorization in [13] requires that the matrix ∇c has full row rank. This cannot be guaranteed in practice, but our algorithm detects this rank deficiency within the factorization procedure and takes into account those rows of ∇c that numerically are linearly dependent, that is, those associated with very small pivots. This modification introduces errors in the solution of the system, but the computed directions seem to behave reasonably well in practice.

3.3. Second-order directions

If we wish to avoid convergence to points that do not satisfy the second-order necessary condition (8), we must make use of directions of negative curvature. For an unconstrained problem $\min_x f(x)$, we would look for directions satisfying the classical definition, see [25], that is, we will require a direction of negative curvature d to satisfy at an iterate x

$$\nabla f(x)^T d \leq 0 \quad \text{and} \quad d^T \nabla^2 f(x) d < 0. \quad (21)$$

To ensure convergence to second-order KKT points, these directions should also satisfy

$$d_k^T \nabla^2 f(x_k) d_k \rightarrow 0 \Rightarrow \limsup_{k \rightarrow \infty} \lambda_{\min}(\nabla^2 f(x_k)) \geq 0 \text{ and } d_k \rightarrow 0,$$

where $\lambda_{\min}(\nabla^2 f(x_k))$ denotes the smallest eigenvalue of $\nabla^2 f(x_k)$.

For equality constrained problems these conditions can be easily generalized: from (8) negative curvature information is only relevant on the subspace spanned by W_A ; we consider only negative curvature directions that lie in this subspace, $d = W_A v$. However, for (nonlinear) inequality constrained problems it is not clear how to define these directions, or how to use them. Now negative curvature information will depend on the current estimate of the active set and the procedure to update it. This information will be relevant in a given iteration both when this estimate does not change, and when the estimate is modified by moving away from some of the active bounds. We will build directions of negative curvature for the merit function using the preceding unconstrained conditions, under certain additional restrictions on the infeasibility of the current iterate.

We now give a more precise statement for these conditions. The gradient of the merit function (3) and its Hessian matrix with respect to x are given by (13) and (14), respectively. Analogously to the case of the descent Newton direction, and due to the limitations of the merit function regarding the satisfaction of the constraints, we will

impose the additional condition that the direction of negative curvature should lie in the null-space of the matrix ∇c . This is reasonable, as the constraints must hold with equality at the solution, and the descent direction is computed to satisfy these constraints. From the definition (21) and this additional condition, a direction of negative curvature d_n for our algorithm should satisfy

$$d_n = W_A w, \quad d_n^T (\nabla f - X^{-1} \mu) \leq 0, \quad d_n^T (\nabla_{xx} L(x, \hat{\lambda} - Rc) + MX^{-2}) d_n < 0. \quad (22)$$

At each iterate we need to determine if negative curvature is present, and if that is the case we also need to compute a direction satisfying the preceding conditions. We will conduct this analysis and compute this direction from matrix (18), used to define the descent direction, to reduce as much as possible the computational cost within the algorithm. However, the matrix used in definition (22), $\nabla_{xx} L(x, \hat{\lambda} - Rc) + MX^{-2}$, may differ from $\nabla_{xx} L(x, \hat{\lambda} - Rc) + X^{-1}Z$, the matrix that appears in (18). Both matrices will be close if (6) is approximately satisfied. A direction of negative curvature will be computed as an approximate eigenvector corresponding to a negative eigenvalue of the coefficient matrix of (18), and we will check if conditions (22) are satisfied before using it in the search. Note that close to a stationary point for the barrier problem (2), condition (6) will be approximately satisfied and directions of negative curvature, if they exist, will eventually be accepted.

Another theoretical issue must be considered. The definition (22) has been made for the barrier problem (2), but the problem of interest is (1). It would be important to ensure that by computing second-order KKT points for problem (2), we are in fact solving problem (1). This issue has been treated in detail in [16] and [29]; we now present the basic result that justifies the validity of our approach.

A second-order KKT point (x^*, λ^*) for problem (1) will satisfy the first-order conditions (4), and the matrix $W_J^T \nabla_{xx} L(x^*, \lambda^*) W_J$ will be positive semidefinite. Here, W_J denotes a matrix whose columns form a basis for the null-space of the Jacobian of the active constraints at the solution of (1). If $\hat{I} \in \mathbb{R}^{p \times n}$ denotes the p rows of the identity associated with the components of x^* that are equal to zero (the active bounds in $x \geq 0$), then W_J corresponds to a basis for the null-space of the matrix

$$J^* \equiv \begin{pmatrix} \nabla c(x^*) \\ \hat{I} \end{pmatrix}.$$

Without loss of generality, we can construct a basis W_A for the null-space of $\nabla c(x^*)$, having the form $W_A = (Y \ W_J)$, where Y is a set of p columns in W_A that do not belong to the null-space of J^* (p is the number of active bounds at x^*). Let $v^* = (x^*, \lambda^*, z^*)$ denote the corresponding values at a second-order KKT point of (1). We will use the notation $H_\rho = \nabla_{xx} L(x, \hat{\lambda} - Rc)$ and $G_\rho = H_\rho + X^{-1}Z$.

Theorem 1. *Given $v = (x, \lambda, z)$ such that $\|v - v^*\| < \epsilon$ for some small enough ϵ , for the matrix $W_A^T G_\rho W_A$ evaluated at v it holds that*

- *The largest (in magnitude) p eigenvalues of $W_A^T G_\rho W_A$ are positive and unbounded as $v \rightarrow v^*$.*
- *The remaining $n - m - p$ eigenvalues are within $O(\epsilon)$ of the eigenvalues of $W_J^T H_\rho W_J$.*

– G_ρ has invariant subspaces for which there exist bases \tilde{Y} and \tilde{W} such that

$$\|Y - \tilde{Y}\| = O(\epsilon) \quad \|W_J - \tilde{W}\| = O(\epsilon),$$

and in particular

$$\|\tilde{W}^T G_\rho \tilde{W} - W_J^T H_\rho W_J\| = O(\epsilon).$$

Proof. See Theorem 3.3 of [30] and Theorem V.2.7 of [27]. \square

As a consequence of this result, close to a stationary point, the observation of the finite eigenvalues of $W_A^T G_\rho W_A$ from (18) provides enough information on the eigenvalues of $W_J^T H_\rho W_J$, the ones entering the definition of second-order KKT points for problem (1). Moreover, if we are close enough to a second-order KKT point of (1), the (finite) negative eigenvalues of $W_A^T G_\rho W_A$ and their associated eigenvectors will provide good approximations to the corresponding ones in $W_J^T H_\rho W_J$. As a consequence, we will be able to compute in an efficient manner directions of negative curvature as approximate eigenvectors corresponding to negative eigenvalues of the matrix in (18), while ensuring convergence to second-order KKT points of (1).

3.4. Computation of directions of negative curvature

We compute a direction of negative curvature d_n (assuming that it exists) from the same symmetric indefinite factorization used to obtain the descent direction d_x in (20). If no negative curvature is available at the current iterate, we set $d_n = 0$. Let K_ρ be the matrix defined in (18), and assume that its symmetric indefinite factorization $K_\rho = LDL^T$ has been computed using the algorithm in [13]. Assume that from the factorization it has been determined that this matrix has more than m negative eigenvalues, implying that $W_A^T G_\rho W_A$ has at least one negative eigenvalue, see [19].

We obtain d_n in the following manner: let w be defined as $w = P\tilde{w}$, where P is the permutation matrix in (20) and $\tilde{w} = (\tilde{w}_1^T \tilde{w}_2^T)^T$ satisfies

$$\begin{pmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{pmatrix} \begin{pmatrix} \tilde{w}_1 \\ \tilde{w}_2 \end{pmatrix} = \pm \sqrt{-\lambda_{\min}(D_2)} \begin{pmatrix} 0 \\ u_\lambda \end{pmatrix}, \quad (23)$$

where $\lambda_{\min}(D_2)$ denotes the most negative eigenvalue of D_2 , defined in (19), and u_λ is a unit eigenvector corresponding to this smallest eigenvalue. The negative curvature direction d_n is defined as the first n components of w . In [13] it is shown that

$$\nabla c(x)d_n = 0, \quad (24)$$

so that d_n lies in the correct subspace, and there exist positive constants c_1 and c_2 such that

$$d_n^T G_\rho d_n \leq -c_1 \lambda_{\min}^2(W_A^T G_\rho W_A) \quad \text{and} \quad d_n^T d_n \leq -c_2 \lambda_{\min}(W_A^T G_\rho W_A). \quad (25)$$

Nevertheless, this scaling may not be adequate for the search procedure. We rescale this negative curvature direction using the norm of the descent direction d_x , to ensure that both directions are comparable in size.

If condition (6) is approximately satisfied, the direction d_n computed using the preceding procedure will be a direction of negative curvature for the merit function $L_A(x, \lambda; \rho, \mu)$. In this case, as $\nabla c d_n = 0$, in order to satisfy (22) it will be enough to choose the sign of d_n so that

$$d_n^T (\nabla f - X^{-1} \mu) \leq 0. \quad (26)$$

As in general (6) may not be satisfied, each time a direction of negative curvature is computed we will also check if

$$d_n^T (H_\rho + M X^{-2}) d_n < -\beta_{nc} c_1 \lambda_{\min}^2 (W_A^T G_\rho W_A) \quad (27)$$

is satisfied for some prespecified constant $\beta_{nc} \in (0, 1)$. If this is not the case, the negative curvature direction d_n will not be used.

An additional condition on the use of negative curvature is related to the violation of the constraints. If this violation is too large, that is, if $\|c(x)\| > \beta_v$ for some positive constant β_v , we set $d_n = 0$. We only use negative curvature when it has been computed close enough to feasible points, where it is most relevant.

The primal iterates x^k must remain positive throughout the algorithm to ensure that the barrier function is well-defined. This condition is imposed in the search procedure, described in the following Section, by limiting the largest acceptable step. This limitation may impact the convergence of the procedure, if the size of some component of the negative curvature direction becomes too large compared with the corresponding primal iterate. To ensure that this does not happen we impose the additional condition

$$(d_n^k)_i \geq -\beta_{dn} x_i^k, \quad (28)$$

where $\beta_{dn} \gg 1$ is a prespecified constant. This condition is automatically satisfied for large enough β_{dn} if the matrix L in $K_\rho = L D L^T$ is computed so that it remains bounded as $x^k \rightarrow 0$. The factorization described above satisfies this condition, see [6]. Note that from (23)

$$\begin{pmatrix} H + X^{-1} Z \nabla c^T \\ \nabla c & 0 \end{pmatrix} \begin{pmatrix} d_n \\ w_2 \end{pmatrix} = \pm (-\lambda_{\min}(D_2))^{3/2} P^T L \begin{pmatrix} 0 \\ u_\lambda \end{pmatrix},$$

and $X^{-1} d_n$ remains bounded if the other matrices are bounded. For other factorizations, condition (28) can be enforced as part of the algorithm, without affecting its convergence. Let \bar{d}_n^k denote the direction obtained from the computation in (23); if $\beta_{dn} x_i^k < z_i^k$ and $(\bar{d}_n^k)_i < -\beta_{dn} x_i^k$, set $(d_n^k)_i = -\beta_{dn} x_i^k$, otherwise $(d_n^k)_i = (\bar{d}_n^k)_i$. The values of those components of d_n^k satisfying $\beta_{dn} x_i^k \geq z_i^k$ are readjusted to ensure (24) is satisfied. If this is not possible or the readjusted direction does not satisfy (25), set $d_n^k = 0$.

4. The curvilinear search

For a given iterate (x, λ) , classical linesearch methods applied to problem (2) compute a direction of movement $d = (d_x^T d_\lambda^T)^T$, and then determine a scalar α such that the next iterate $(x + \alpha d_x, \lambda + \alpha d_\lambda)$ provides sufficient decrease for an appropriate merit

function. The role of the merit function is to ensure, through the proper choice of α , the convergence to a minimizer of this merit function, that should correspond to a minimizer of (2). The computation of α is usually referred to as a linesearch (see [18], for example).

This approach works quite well in practice whenever there is a single search direction d_x . In our case we may have a pair of search directions at a given iteration, d_x and d_n . In this case, the preceding procedure must be modified to take into account that the next iterate must be found by searching on a subspace having dimension two, instead of dimension one as was the case for the classical approach. We have chosen to use a curvilinear search, defined on the subspace generated by both directions, and applied to the augmented Lagrangian merit function (3). This curvilinear search will be based on the directions d_x computed from (17) and d_n obtained as described in the preceding section from (23).

A search will also be carried out on the multipliers. Their search direction will be defined from d_λ , obtained from (17), but it will be modified to take into account the right-hand side used in (17) and the Newton direction for the merit function given in (16). The actual search direction is defined as

$$d_{\lambda\rho} = d_\lambda - Rc(x). \quad (29)$$

To combine the preceding directions we will follow the proposal in [25]. Given an iterate (x, λ) and directions d_x , d_n and d_λ , the next iterate will be obtained as a point on the curves

$$x(\alpha) = x + \alpha^2 d_x + \alpha d_n, \quad (30)$$

$$\lambda(\alpha) = \lambda + \alpha^2 d_{\lambda\rho}. \quad (31)$$

This combination is the simplest one that guarantees sufficient decrease for the merit function whenever either we have a direction of descent or we have a direction of negative curvature. See Lemma 2.2 in [25].

The value of α is determined to ensure that $(x(\alpha), \lambda(\alpha))$ provides sufficient decrease for the augmented Lagrangian merit function (3). Let

$$\phi(\alpha) = L_A(x(\alpha), \lambda(\alpha); \rho, \mu),$$

where ρ is the penalty parameter vector. For an initial value α_{\max} , defined later on, we will check if

$$\phi(\alpha_{\max}) \leq \phi(0) + \gamma \frac{1}{2} \alpha_{\max}^2 \phi''(0). \quad (32)$$

If this condition is satisfied, we choose $\tilde{\alpha} = \alpha_{\max}$. Otherwise, we apply a backtracking procedure from α_{\max} to find a value $\tilde{\alpha} \in (0, \alpha_{\max})$ satisfying

$$\phi(\tilde{\alpha}) \leq \phi(0) + \gamma \frac{\tilde{\alpha}^2}{2} \phi''(0), \quad (33)$$

$$\phi'(\tilde{\alpha}) \geq \eta(\phi'(0) + \tilde{\alpha}\phi''(0)), \quad (34)$$

where γ and η are scalar parameters satisfying $0 < \gamma < \frac{1}{2}$ and $\frac{1}{2} < \eta < 1$.

In addition to the sufficient decrease conditions (33)–(34), we also force the iterates to remain on the domain of the merit function (the positive orthant). Thus, α is chosen so that $x(\alpha) > 0$ by defining α_{\max} appropriately, that is, we choose α_{\max} so that all $\alpha \in [0, \alpha_{\max}]$ satisfy

$$P_i(\alpha) = \alpha^2(d_x)_i + \alpha(d_n)_i + x_i > 0 \quad \forall i. \quad (35)$$

If $\tilde{\alpha}_i$ denotes the smallest positive root of $P_i(\alpha)$, if it exists, or ∞ otherwise, the preceding condition is satisfied for a given i and all $\alpha \in [0, \tilde{\alpha}_i)$, as $x_i > 0$. As a consequence, we must choose $\alpha_{\max} < \min_i \tilde{\alpha}_i$.

On the other hand, to take advantage of the good local convergence properties of the Newton direction we consider the step to this Newton direction ($\alpha = 1$) whenever it is reasonable, that is, whenever the Newton step lies within the positive orthant and there is no negative curvature available. Consequently, the initial step α_{\max} is defined as

$$\alpha_{\max} = \min(\delta \min_i(\tilde{\alpha}_i), 1), \quad (36)$$

where the parameter δ , introduced to ensure the strict positivity of the iterates, has been defined as

$$\delta = \max(0.995, 1 - \|\mu\|). \quad (37)$$

Near the solution (when $\mu \simeq 0$), the term $1 - \|\mu\|$ guarantees that a step of one will not be prevented by the positivity requirement. This is relevant to ensure adequate local convergence properties.

Finally, we impose an additional condition that helps to ensure that the iterates remain in a compact set throughout the algorithm. The next iterate is computed as $x(\alpha_p)$, where $\alpha_p = \tilde{\alpha}$ if $\tilde{\alpha}$, satisfying (33)–(34), also satisfies

$$\|c(x(\alpha))\| \leq \beta_c, \quad (38)$$

for some positive constant β_c satisfying $\beta_c > 2\beta_v$, where β_v was the constant introduced at the end of Section 3.4. Otherwise, a value α_p is determined by applying a backtracking procedure from $\tilde{\alpha}$ until conditions (33) and (38) are both satisfied. From (30) and (31), the next iterates are defined as

$$\begin{aligned} x^{k+1} &= x^k + (\alpha_p^k)^2 d_x^k + \alpha_p^k d_n^k, & \lambda^{k+1} &= \lambda^k + (\alpha_p^k)^2 d_{\lambda\rho}^k \\ \hat{\lambda}^{k+1} &= \lambda^k + \max\left(\beta_\lambda, (\alpha_p^k)^2\right) d_{\lambda\rho}^k, \end{aligned} \quad (39)$$

where $\hat{\lambda}$ is the value used in the definition of G_ρ , (18), and H_ρ , and $\beta_\lambda \in [0, 1]$ is a prespecified constant.

5. Parameter updates

A complete specification of the algorithm should indicate how to update the different parameters that appear in the computation of the search directions and the curvilinear search. In the following paragraphs we describe the procedures used to update the multiplier estimates, the penalty parameters and the barrier parameters.

5.1. The multipliers

Two sets of dual variables are generated by the algorithm, the equality constraint multipliers λ and the approximations to the multipliers for the bound constraints z . The multipliers λ are updated within the curvilinear search using (31) and the value α_p chosen for the variables x according to the procedure described in the preceding section.

The solution of Newton's system of equations (9) provides a search direction for the multipliers z , d_z , defined in (10). These dual variables will be updated from

$$z(\alpha_d) = z + \alpha_d d_z, \quad (40)$$

using an adequate value of α_d . The only condition on the values of the dual variables is their non-negativity. The scalar α_d is chosen as the largest reasonable value that satisfies this condition, as follows. Let

$$\bar{\alpha}_d = \min \left(\delta \min \left(\frac{-z_i}{(d_z)_i} \mid (d_z)_i < 0 \right), 1 \right), \quad (41)$$

where δ is defined as in (37). The step length for the dual variables, α_d , is defined as

$$\alpha_d = \max \left(\beta_z, (\alpha_p / \alpha_{\max})^2 \right) \bar{\alpha}_d, \quad (42)$$

where $\beta_z \in [0, 1]$ is another prespecified parameter. The correction to $\bar{\alpha}_d$ is introduced to scale α_d so that its value is related to the value of α_p obtained from the linesearch.

5.2. The penalty parameters

The penalty parameters ρ_j are used in the algorithm to ensure the convergence to points satisfying the constraints $c(x) = 0$. The Newton direction should generate iterates that satisfy this condition in the limit, but if the penalty parameters are not sufficiently large, this Newton direction may not be a descent direction for the merit function and will not be accepted. As a consequence, the penalty parameters are chosen so that the sufficient decrease condition given by inequality (33) can be satisfied. The updating of these parameters is also very relevant for the computational efficiency of the procedure. A very large value of these parameters may cause numerical problems in the computation of the search directions from (17). Also, these parameters have an impact on the updating of the λ multiplier estimates.

The update formula will be derived in terms of condition (33). This condition includes the initial derivatives of the merit function along the curve $x(\alpha)$ defined in (30),

$$\phi'(0) = d_n^T (\nabla f - X^{-1} \mu), \quad (43)$$

$$\begin{aligned} \phi''(0) = & d_n^T (\nabla_{xx} L(x, \hat{\lambda} - Rc) + M X^{-2}) d_n + 2d_x^T (\nabla f - X^{-1} \mu - \nabla c^T (\lambda - Rc)) \\ & - 2d_{\lambda\rho}^T c. \end{aligned} \quad (44)$$

As (24) holds, the terms $d_n^T \nabla c^T (\lambda - Rc)$ and $d_n^T \nabla c^T R \nabla c d_n$ have been removed from the preceding expressions. To simplify the following arguments, define θ_ρ as

$$\theta_\rho = d_x^T (\nabla f - X^{-1} \mu - \nabla c^T \lambda) - d_\lambda^T c. \quad (45)$$

Also, note that (17) implies $\theta_\rho = -d_x^T \bar{G}_\rho d_x$.

If negative curvature is available in the current iteration, that is, if $d_n \neq 0$, from (26) and $\nabla c d_n = 0$ it follows that $\phi'(0) \leq 0$. We still need to have a sufficient descent condition

$$\phi''(0) \leq \hat{\delta}_\rho (\min(\theta_\rho, 0) - \|c\|^2), \quad (46)$$

for some constant $0 < \hat{\delta}_\rho \ll 1$. If the current values of d_x , d_n , $d_{\lambda\rho}$ and ρ are such that (46) is not satisfied, we set $d_n = 0$ for the search.

If no negative curvature has been detected or the preceding condition has resulted in having $d_n = 0$, then $\phi'(0) = 0$ and condition (33) becomes equivalent to

$$\phi(\tilde{\alpha}) \leq \phi(0) + \gamma \tilde{\alpha}^2 (d_x^T (\nabla f - X^{-1} \mu - \nabla c^T (\lambda - Rc)) - d_{\lambda\rho}^T c).$$

For the curvilinear search to be well defined we need to have again a sufficient descent condition equivalent to (46) (see [25]):

$$\phi''(0) = d_x^T (\nabla f - X^{-1} \mu - \nabla c^T (\lambda - Rc)) - d_{\lambda\rho}^T c \leq \hat{\delta}_\rho (\min(\theta_\rho, 0) - \|c\|^2).$$

This condition is a slightly stronger version of the classical descent requirement $d_x^T \nabla_x L_A \leq 0$, but (46) must take into account that the multiplier update $\lambda(\alpha)$ (31) is also included in the curvilinear search.

Condition (46) can always be satisfied for an adequate choice of the penalty parameter vector ρ . If at the current iterate the equality constraints are satisfied, $c(x) = 0$, (46) can be rewritten as

$$d_x^T (\nabla f - X^{-1} \mu) \leq \hat{\delta}_\rho \min(\theta_\rho, 0) \Leftrightarrow \theta_\rho \leq \hat{\delta}_\rho \min(\theta_\rho, 0),$$

using (45), $\nabla c d_x = -c$ from (11), and $c = 0$. From (17), $\nabla c d_x = 0$ and the positive definiteness of \bar{G}_ρ in the appropriate subspaces,

$$-d_x^T \bar{G}_\rho d_x = d_x^T (\nabla f + X^{-1} \mu) = \theta_\rho \leq 0, \quad (47)$$

and the preceding condition, and (46), are satisfied.

If $c(x) \neq 0$, (46) may not hold for the current value of the penalty parameter ρ , and it must be modified. We can rewrite (46) as

$$\phi''(0) = \theta_\rho + d_x^T \nabla c^T Rc \leq \hat{\delta}_\rho (\min(\theta_\rho, 0) - \|c\|^2). \quad (48)$$

From (17), implying $\nabla c d_x = -c$, (48) is equivalent to

$$\theta_\rho - c^T Rc \leq \hat{\delta}_\rho (\min(\theta_\rho, 0) - \|c\|^2). \quad (49)$$

Define $\zeta \in \mathbb{R}^m$ as $\zeta_j = c_j^2$, and note that $c^T R c = \zeta^T \rho$. Following the procedure used in code NPSOL [17], the update of the penalty parameter vector ρ is obtained from the following problem:

$$\begin{aligned} \min \quad & \frac{1}{2} \rho^T \rho \\ \text{s.t.} \quad & \zeta^T \rho \geq \theta_\rho - \hat{\delta}_\rho (\min(\theta_\rho, 0) - \|c\|^2) \\ & \rho \geq 0. \end{aligned} \quad (50)$$

The solution of this problem is given by

$$\rho^* = \max \left(0, \theta_\rho - \hat{\delta}_\rho (\min(\theta_\rho, 0) - \|c\|^2) \right) \frac{\zeta}{\zeta^T \zeta}$$

This solution satisfies (49) and (46).

If (46) does not hold and $\rho_j < \delta_\rho \rho_j^*$, the j -th component of ρ will be updated to $\delta_\rho \rho_j^*$ for some $\delta_\rho > 1$. In practice, it will also be necessary to ensure that ρ does not become too large (see [18]) to avoid ill-conditioning. If ρ_j is much larger than ρ_j^* , we will reduce its value while ensuring that (46) is still satisfied. The strategy we will follow to update ρ is similar to the one described in [11]. We will compute a trial value $\hat{\rho}_j$:

$$\hat{\rho}_j = \sqrt{\rho_j (\bar{\delta}_\rho + \rho_j^*)},$$

where $\bar{\delta}_\rho \geq 1$, and the new value of ρ at iteration k will be defined as

$$\rho_j^{k+1} = \begin{cases} \delta_\rho \rho_j^* & \text{if } \delta_\rho \rho_j^* > \rho_j^k, \\ \hat{\rho}_j & \text{if } \hat{\rho}_j \leq \frac{1}{2} \rho_j^k, \\ \rho_j^k & \text{otherwise.} \end{cases} \quad (51)$$

To avoid having to modify ρ too often, the parameter $\bar{\delta}_\rho$ is increased at each iteration where ρ is modified.

5.3. The barrier parameters

The vector of barrier parameters in (2) is also updated in each iteration. The updating rule is based on the relationship between the satisfaction of the first-order conditions, the complementarity conditions and the values of the barrier parameters. The definition of μ_i , given below, can be shown to be $O(\|F(x, \lambda, z)\|_2^2)$ near a KKT point, that is one of the conditions required to attain superlinear convergence (see [10] or [33]), where $F(x, \lambda, z)$ is related to the satisfaction of the first-order KKT conditions for problem (1) at the current iterate.

Define

$$F(x, \lambda, z) = \begin{pmatrix} \nabla f(x) - \nabla c(x)^T \hat{\lambda} - z \\ c(x) \\ (1 - \beta_m)Xz + \beta_m(z - X^{-1}\mu + \mu) \end{pmatrix}, \quad (52)$$

for some prespecified constant $\beta_m \in [0, 1]$, and set

$$\theta_\mu = \begin{cases} \|F(x, \lambda, z)\| & \text{if } \|F(x, \lambda, z)\| \geq 1, \\ \|F(x, \lambda, z)\|^2 & \text{otherwise,} \end{cases} \quad (53)$$

and let $y = Xz$. Vector μ is updated in a manner similar to the penalty parameter ρ . The problem

$$\begin{aligned} \min \quad & \frac{1}{2} \mu^T \mu \\ \text{s.t.} \quad & y^T \mu = \theta_\mu \\ & \mu \geq 0 \end{aligned} \quad (54)$$

has its solution for $\mu = \theta_\mu y / \|y\|^2$. A small perturbation is introduced on this solution to define μ^* as

$$\mu^* = \theta_\mu \frac{y + \beta_y \|y\| e}{y^T y}, \quad (55)$$

where $e = (1 \dots 1)^T$ and $\beta_y \in [0, 1/\sqrt{n}]$ is a prespecified constant.

Definition (53) has been introduced to prevent μ_i^* from becoming too large when far from a KKT point. On the other hand, if y_i is small then μ_i^* may become too small. To avoid this situation we compute a reference value $\hat{\mu}$, similar to the one used in [10],

$$\hat{\mu} = \frac{x^T z}{n}, \quad (56)$$

and define the new value of μ at iteration k as

$$\mu_i^{k+1} = \begin{cases} \beta_\mu \delta^k \max((\mu_i^*)^k, \hat{\mu}^k) & \text{if } \delta^k \max((\mu_i^*)^k, \hat{\mu}^k) < \mu_i^k \\ \mu_i^k & \text{otherwise,} \end{cases} \quad (57)$$

where $\beta_\mu < 1$ is a prespecified constant,

$$\delta^k = \min(0.25, \exp(-(1/\theta_\mu^k))), \quad (58)$$

and θ_μ^k is the value of θ_μ at iteration k . Note that μ_i^k will not be decreased in every iteration, but only when a sufficient reduction in the satisfaction of the KKT conditions has been achieved. This definition of μ ensures that $\mu \rightarrow 0$ if problem (2) has a solution.

6. Theoretical results

Although the convergence analysis of the proposed procedure is not the main goal of this paper, in this Section we study some of its properties to justify the validity of some of the choices made in the algorithm. In particular, we wish to justify that the choice of linesearch and barrier parameter updates (the more unusual parts of the algorithm) are reasonable from a theoretical point of view. Nevertheless, from a numerical point of view the contents of this section could be omitted, as they have no impact on the

remainder of the paper. For the sake of a clearer presentation, the detailed proofs for these results have been collected in an Appendix at the end of the paper.

We will study the properties of the limit points for the sequence generated by the algorithm (if they exist). We will see that they are second-order KKT points for the problem of interest (1), under some regularity assumptions on the problem. We will also introduce some conditions on the sequence of iterates, to simplify the arguments in the proofs. The assumptions on problem (1) are:

- A.1 The functions f and c_j have Lipschitz-continuous third derivatives on the region defined by the bounds.
- A.2 Strict complementarity holds at all first-order KKT points.
- A.3 The iterates generated by the algorithm remain in a compact set.
- A.4 The penalty parameter remains bounded in the algorithm.
- A.5 The Jacobian matrix of the equality constraints, $\nabla c(x)$, has full row rank at all feasible points of problem (1).

Note that A.1 and A.3 imply that the objective function remains bounded below in the algorithm.

Assumptions A.1, A.2 and A.5 are related to the problem, while the other two assumptions concern the algorithm. Excluding these assumptions would imply a significant increase in the technical complication of the proofs, not our main concern for this paper. Assumption A.3 is mostly relevant for the primal variables, as for the dual ones the boundedness condition can be imposed within the algorithm by limiting the size of the corresponding search directions. Assumption A.4 is quite strong, but it would be very difficult to show the boundedness of the parameter (without making any assumption on it), given the complex update rule we use. For slightly simpler update rules, see [26] for example, it is possible to remove this condition, under some alternative assumptions on the behavior of the multiplier estimates in the algorithm. Assumption A.5 is also very strong, but it simplifies significantly the arguments in the proofs. It ensures that $\nabla c(x^k)d_x^k = -c(x^k)$ close to the solution, and as a consequence that both (47) and condition (46) hold in the algorithm.

As an additional simplification, we will ignore condition (34). It ensures that the steplength is not unnecessarily small. While this is a reasonable practical requirement, it does not impact the theoretical properties of the algorithm. From the definition of ρ , that ensures descent for the merit function, and the existence of a step α , see Moré and Sorensen [25], the iterates generated by the algorithm are well-defined. Also, from Assumption A.4 and (51), as the value $\hat{\rho}_j$ is increased by a finite amount every time that ρ_j is reduced, the number of reductions is finite and the value of ρ eventually remains constant in the algorithm. Several constants were introduced in the definition of the algorithm; to simplify the proofs we only analyze the case where $\beta_\lambda = 1$ in (39), $\beta_z \in (0, 1]$ in (42), $\beta_m \in (0, 1]$ in (52) and $\beta_y \in (0, 1/\sqrt{n}]$ in (55).

We now summarize the structure of the arguments in this section:

- Our main result shows that the update rule we use for the barrier vector parameter enforces its convergence to zero. This same rule guarantees convergence to first-order KKT points for problem (1). In addition to these results, we also show that the limit points must be second-order KKT points. We prove that the descent conditions

in the linesearch force the algorithm to move away from points that do not satisfy the required second-order conditions. All these properties are proved in Theorem 2.

- To establish this main result, we first show that if the algorithm updates the barrier parameter only a finite number of times, then the primal variables converge to first-order KKT points for the barrier problem (2). This result, presented in Lemma 3, will be used in Theorem 2 to prove a contradiction, as the barrier parameter must eventually be updated when sufficiently close to a first-order KKT point for problem (2). An auxiliary result in Lemma 4 shows that dual variables also converge to the correct values.
- A previous intermediate result, proved in Lemma 2, shows that whenever one component of the barrier vector parameter converges to zero, all of them must converge to zero. As a consequence, we only need to consider the cases when all components are bounded away from zero, Lemma 3, and when all of them converge to zero, Theorem 2.
- Our first auxiliary result, Lemma 1, establishes the boundedness of the primal iterates whenever the barrier parameter is bounded away from zero.

From (57), the components of the barrier parameter μ are nonincreasing. It might happen that some or all of the components in μ^k would not converge to zero. The following Lemmas show that this cannot happen. To prove this result, we start by introducing a bound on the values of the primal variables in terms of the barrier parameter value.

Lemma 1. *Let \mathcal{I} denote the set of components μ_i^k ($i \in \mathcal{I}$) of the barrier parameter that change only a finite number of times in the course of the algorithm, and assume \mathcal{I} is nonempty. Then there exists $\bar{\gamma}_x > 0$ such that $x_i^k \geq \bar{\gamma}_x$ for all sufficiently large k and all $i \in \mathcal{I}$.*

We can now present a first result related to the convergence of the barrier parameter μ^k . In it we show that the case in which some of the components of the parameter converge to zero and other components remain bounded away from zero cannot happen in the algorithm. We will continue using the notation introduced in Lemma 1.

Lemma 2. *Let \mathcal{J} denote the set of components μ_i^k ($i \in \mathcal{J}$) of the barrier parameter that change an infinite number of times in the course of the algorithm, and assume \mathcal{J} is nonempty. Then $\mathcal{J} = \{1, \dots, n\}$ and $\liminf_{k \rightarrow \infty} \theta_\mu^k = 0$.*

From this result only two situations are possible in the algorithm: i) the components of the barrier parameter change only a finite number of times in the algorithm, and the barrier parameter eventually remains constant, and ii) all components of the barrier parameter change an infinite number of times. The following Lemma considers the first case.

Lemma 3. *Let \mathcal{J} denote the set of components μ_i^k ($i \in \mathcal{J}$) of the barrier parameter μ that change an infinite number of times in the course of the algorithm, and assume \mathcal{J} is empty. Then any limit point x^* of the sequence $\{x^k\}$ generated by the algorithm is a first-order KKT point of the barrier problem (2).*

We need an additional auxiliary lemma to show that the sequences of dual variables $\{\hat{\lambda}^k\}$, $\{z^k\}$, also converge to the desired values.

Lemma 4. *Let \mathcal{J} denote the set of components μ_i^k ($i \in \mathcal{J}$) of the barrier parameter μ that change an infinite number of times in the course of the algorithm, and assume \mathcal{J} is empty. The sequences $\{x^k\}$, $\{\hat{\lambda}^k\}$ and $\{z^k\}$ generated by the algorithm satisfy*

$$\lim_{k \rightarrow \infty} \|\nabla f(x^k) - (X^k)^{-1} \mu^k - \nabla c(x^k)^T \hat{\lambda}^k\| = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \|z^k - (X^k)^{-1} \mu^k\| = 0.$$

We now show that the barrier parameter converges to zero, that is, that the situation analyzed in Lemma 3 cannot happen in practice, and also that the limit points of the sequence generated by the algorithm must have the correct properties.

Proving that $\mu^k \rightarrow 0$ and the limit points must be first-order KKT points follows easily from the definition of the barrier parameter in the algorithm. The proof that any limit point must also be a second-order KKT point is less straightforward, as we have to consider now that μ^k can be arbitrarily small and the arguments in the proofs of Lemmas 1 and 3 can no longer be applied. We show the desired result by proving that in this case the step-length α is also bounded away from zero, and as a consequence that the algorithm cannot converge unless $(\phi^k)''(0) \rightarrow 0$, implying that any negative curvature must also vanish.

The algorithm obtains both the descent and the negative curvature information from the barrier problem (2). The convergence of the algorithm to second-order KKT points for problem (1) requires the existence of a relationship between the negative curvature in both problems, as shown in Theorem 1. One of the requirements for that theorem is the correct identification of the active set at the solution, that follows from the strict complementarity Assumption A.2.

Theorem 2. *The barrier parameter is updated an infinite number of times in the algorithm. Furthermore, there exists a subsequence of iterates converging to second-order KKT points of problem (1).*

Although our aim is not the study of the local convergence of the algorithm, we provide a few comments on its local properties. The following condition would also be required:

A.6 The sufficient optimality conditions hold at all second-order KKT points of problem (1).

Under this assumption and Theorem 2 there can only be a finite number of iterations where negative curvature is used, as in a ball around the solution point the relevant Hessian matrices are strictly positive definite. Asymptotically, only Newton directions are used to obtain the new iterates. The arguments in [33] can be adapted to the proposed algorithm to prove superlinear convergence, by showing that the steps taken close to the solution coincide with the feasible steps $\tilde{\alpha}_p^k$ and $\tilde{\alpha}_d^k$, that these steps converge to one at a rate related to the size of μ^k and that having a vector barrier parameter does not affect the arguments in the proofs. Different approaches, such as the one in [21], can also be adapted but may require modifying the barrier parameter update rule.

7. Implementation issues

The algorithm described in the preceding sections includes certain parameters and conditions that have not been completely specified yet. In the following paragraphs we indicate how to carry out some of these computations.

7.1. Use of directions of negative curvature

After the factorization process has detected the presence of negative curvature in a given iteration, some additional conditions are checked before using this negative curvature direction in the curvilinear search. Appropriately restricting the use of negative curvature has a significant impact on the efficiency of the algorithm. In particular, the three following conditions should hold:

$$d_n^T \nabla_{xx} L_A d_n \leq -\varepsilon_1, \quad (59)$$

$$d_n^T \nabla_{xx} L_A d_n \leq d_n^T G_\rho d_n + \varepsilon_2, \quad (60)$$

$$\|c(x)\| \leq \varepsilon_3, \quad (61)$$

where ε_1 , ε_2 and ε_3 are positive constants. Condition (59) guarantees that d_n is a direction of negative curvature for the augmented Lagrangian merit function. Condition (60) takes into account those cases where (6) is far from being satisfied. Finally, (61) guarantees that we only use negative curvature when we are close enough to feasibility. If any of these conditions is not satisfied, the algorithm sets $d_n = 0$. In practice, if these conditions are not taken into account the algorithm may get stuck away from a solution.

In our implementation we have defined $\varepsilon_1 = 10^{-7}$ and $\varepsilon_2 = 10^{-3}$. The parameter ε_3 is defined in each iteration as:

$$\varepsilon_3^k = \min(10^{-2}(0.1 + \|x^k\| + |f^k|), 3),$$

where f^k is the objective function of (1) at iteration k .

7.2. Convergence criterion

The stopping criterion for the algorithm will be related to the satisfaction of the first and second-order KKT conditions for problem (1). The algorithm will stop if no negative curvature has been detected at the current iteration and the condition

$$\|F(x, \lambda, z)\|_2 \leq \varepsilon(1 + \|\nabla f(x)\|)$$

is satisfied at the current iterate. In this condition $F(x, \lambda, z)$ denotes the measure of optimality defined in (52) and $f(x)$ is the objective function for problem (1). We have taken $\varepsilon = 10^{-8}$.

7.3. Initial values of parameters and variables

Let \bar{x}^0 denote the starting point for the algorithm, assumed to be specified by the user. Sometimes these initial points may not satisfy the bound constraints. The algorithm transforms the given initial point following an automatic strategy similar to that described in [28]. Basically, for problems with zero lower bounds, such as (1), if the initial value of one of the variables is outside the bounds it has been redefined by

$$x_i^0 = \max\left(1, \sum_i |\bar{x}_i^0|/n\right).$$

For problems with upper and lower bounds, $l \leq x \leq u$,

$$x_i^0 = \max \left(\min \left(\bar{x}_i^0, u_i - 0.1(u_i - l_i) \right), l_i + 0.1(u_i - l_i) \right),$$

that is, the initial value has been redefined by moving it inside the bounds to within 10% of the closest bound.

The remaining initial values for the variables and parameters are defined from x^0 . The initial value of the dual variables z^0 is defined as $z^0 = (X^0)^{-1}e$. The initial Lagrange multiplier estimate λ^0 is chosen as an approximation to the least-squares solution of the linear system

$$\nabla c(x^0)^T \lambda^0 = \nabla f(x^0) - z^0.$$

The penalty parameter vector ρ^0 is initially taken to be zero. The initial barrier vector μ^0 is defined using (57) evaluated at the preceding values.

7.4. Other parameters

The constant β_c in (38) is updated recursively. In iteration $k + 1$ it is defined to be

$$\beta_c^{k+1} = \max \left(\sqrt{K \|c(x^k)\| \beta_c^k}, 1 \right),$$

where $K > 1$ is a constant (in our implementation we use $K = 7.5$). The initial value of β_c is defined as

$$\beta_c^0 = \max \left(\sqrt{K \|c(x^0)\|}, 1 \right).$$

The parameter $\hat{\delta}_\rho$ in (46), controlling the sufficient descent in the linesearch, has been set to zero in the algorithm. Also, β_μ , introduced in (57), the update of the barrier parameter, has been set to one. Other constants have been taken in the implementation as $\beta_\lambda = 0$ in (39), $\beta_z = 0$ in (42), $\beta_m = 0$ in (52) and $\beta_y = 0$ in (55). These values are in the limit of their theoretically acceptable ranges, but this does not seem to have an impact in practice.

7.5. Numerical difficulties

In the preceding description of the algorithm we have considered only simple bounds of the form $x \geq 0$, to simplify the resulting expressions. The implementation of the algorithm used for the tests is able to handle simple bounds of the form $l \leq x \leq u$, where some of the entries in l could be equal to $-\infty$, and some of those in u could be ∞ . The finite bounds are included in the objective function via logarithmic barrier terms.

If a variable gets very close to its corresponding bound at a given iteration, it is possible that due to roundoff errors its value may be considered to be equal to the bound and the logarithmic function will not be defined in subsequent iterations. A possible solution is given in [14], where the variables are forced to be removed from their bounds by a

fixed distance, chosen as 10^{-9} . However, this strategy presents a clear disadvantage: the solution of a particular problem, for a sufficiently small value of μ , might be closer to the bound than the previous tolerance. Also, from our numerical experience this option may delay convergence by a significant number of iterations.

In our strategy, the information on the distance to the bounds will be kept in a vector r , that will be updated separately from the values of the variables, using the same information. The independent term in inequality (35) will also be defined in terms of this vector.

Another numerical problem that might arise is the ill-conditioning of the symmetric system (17) to solve in each iteration, due to the terms $X^{-1}Z$. Under reasonable conditions, it can be shown that this ill-conditioning is benign (see [29] and [31]).

7.6. The algorithm

We present below a more detailed scheme of the proposed interior point algorithm (Curvilinear Search Interior Point Method - CSIPM).

Algorithm CSIPM

Select x^0 by modifying the user-specified initial value \bar{x}^0 , if necessary

Choose initial values for λ^0 and z^0 .

Choose initial values for vectors ρ^0 and μ^0

Set $k = 0$

repeat

Compute d_x^k , d_λ^k and d_z^k from (17) using the factorization described in [13]

Compute $d_{\lambda\rho}^k$ from (29)

Compute, if it exists, d_n^k , a direction of negative curvature from (23)

Compute ρ^{k+1} from (51)

Set $d_n^k = 0$ if any one of the conditions (59), (60) or (61) is not satisfied

Compute α_p using the curvilinear search procedure satisfying (33), (34) and (38)

Compute α_d from (42)

$x^{k+1} = x^k + \alpha_p^2 d_x^k + \alpha_p d_n^k$

$\lambda^{k+1} = \lambda^k + \alpha_p^2 d_{\lambda\rho}^k$

$z^{k+1} = z^k + \alpha_d d_z^k$

Compute the updated barrier vector μ^{k+1} from (57)

$k = k + 1$

until convergence

8. Numerical results

We have conducted a set of numerical experiments on a collection of test problems using algorithm CSIPM. The algorithm has been implemented, and the tests have been carried out, on MATLAB.

8.1. Test problems

The test set we have considered is composed of 145 small problems from the CUTE collection [4], selected from those nonlinear constrained problems having less than 100 variables and continuous derivatives (note that exact first and second derivatives have been used). The initial points given in CUTE have been used.

Table 1 shows the results obtained by CSIPM for these problems. The columns in the table correspond to:

- Prob.: problem name.
- Obj.: value of the objective function $f(x)$ at the solution.
- Const.: norm of the constraint vector, $\|c(x)\|$, at the solution, including slacks.
- KKT: norm of the first-order KKT conditions at the solution, $\|F(x, \lambda, z)\|$.
- Iter.: iteration count (number of factorizations of the primal-dual system).
- Eval.: number of evaluations of the objective function and the constraints.
- NC: number of iterations in which directions of negative curvature were used.

In those cases where negative curvature was detected the problem was solved a second time, setting the negative curvature direction to zero. Table 1 includes two lines for those problems, one for the results from each of the two versions of the algorithm.

8.2. Analysis of the results

The algorithm was able to solve all problems but two, problems HS13 (with a rank-deficient Jacobian at the solution) and HS109 (after exceeding 250 iterations no solution was reached). For some of the problems the code finds better local minimizers than those given in [22] (this happens for problems HS106, HS107, HS112 and HS116, for example), while for other problems the local minimizers found are worse (HS97 and HS98). Problem HS99 is an example of a badly scaled problem. The termination tolerance is satisfied when the norm of the first-order KKT conditions is 0.4994. Introducing a more demanding stopping criterion (a tolerance of 10^{-14}), the norm of the KKT conditions goes down to 10^{-6} after 3 additional iterations, but the value of the merit function remains basically unaltered.

In general, the number of iterations required to solve the problems is fairly small. The number of function evaluations is higher, but no particular care was taken with the strategy to choose the value of α (a standard backtracking search was implemented). It is also interesting to note the large number of cases in which a unit step was directly accepted.

Table 1. Results for small-size problems

Prob.	Obj.	Const.	KKT	Iter.	Eval.	NC
AIRPORT	47952.7018	1.5e-07	1.4e-04	15	15	0
ALJAZZAF	75.005	5.6e-10	2.0e-07	20	27	0
ALSOTAME	0.08208499	2.2e-16	6.8e-11	8	8	0
BIGGSC4	-24.499999	8.9e-16	7.1e-08	19	19	3
	-24.499999	0	6.9e-08	20	20	0
BT13	1.29e-121	5.1e-09	5.1e-09	21	25	0
CANTILVR	1.33995636	5.2e-11	7.9e-11	15	18	0
CB2	1.95222449	3.3e-14	5.1e-14	11	12	0
CB3	2.0	4.1e-11	5.4e-11	10	11	0
CHACONN1	1.95	3.2e-09	5.0e-09	9	11	0
CHACONN2	2.0	3.8e-09	5.1e-09	10	12	0
CONGIGMZ	28.0	1.1e-07	2.2e-07	33	37	0
CSFI1	-49.0752	1.6e-11	1.2e-09	33	46	2
	-49.0752	1.1e-09	1.0e-08	35	55	0
CSFI2	55.0176056	9.2e-12	1.6e-09	58	62	0
DEMYMALO	-2.9999999	1.1e-09	3.5e-09	13	14	0
DIPIGRI	680.63006	9.0e-10	2.3e-09	9	13	1
	680.63006	1.9e-09	5.1e-09	10	15	0
DISC2	1.5625	1.6e-14	1.6e-14	45	53	0
DUAL1	0.035012968	2.2e-16	1.7e-09	18	18	0
DUAL2	0.033733671	5.1e-16	1.7e-11	14	14	0
DUAL4	0.746090649	1.0e-16	1.1e-08	14	14	0
EXPFITA	0.0011366117	2.9e-14	1.3e-11	33	33	1
	0.0011366117	3.7e-14	1.2e-11	33	33	0
FCCU	11.14910914	4.4e-15	4.2e-15	8	8	0
GIGOMEZ1	-3.0	2.4e-16	1.4e-15	17	19	0
HATFLDH	-24.499999	2.9e-15	2.4e-07	12	12	1
	-24.499999	2.3e-15	2.3e-07	12	12	0
HIMMELBI	-1735.569545	1.1e-06	1.5e-05	26	26	0
HIMMELBK	0.0518143	3.9e-12	1.7e-10	25	28	0
HIMMELP2	-62.0538698	1.4e-08	2.1e-08	16	34	3
	-62.0538698	1.4e-11	1.9e-11	19	32	0
HIMMELP3	-59.01312394	1.1e-09	3.3e-09	15	33	5
	-59.01312394	1.8e-12	3.4e-10	15	25	0
HIMMELP4	-59.01312394	1.6e-10	1.6e-10	24	50	4
	-59.01312391	1.4e-07	1.4e-07	21	42	0
HIMMELP5	-59.01312395	4.5e-12	4.5e-12	68	112	1
	-59.01312395	4.4e-11	4.4e-11	57	80	0
HIMMELP6	-59.01312395	1.2e-12	1.2e-12	42	64	1
	-59.01312395	3.7e-12	4.2e-12	35	44	0
HONG	22.57108736	5.5e-17	6.4e-13	7	7	0
HS10	-1.0	4.2e-14	6.6e-14	15	22	0
HS11	-8.49846422	1.3e-10	6.5e-10	7	8	0
HS12	-30.0	7.6e-14	7.7e-14	10	19	0
HS13	--	--	--	--	--	--
HS14	1.39346498	7.1e-15	1.7e-13	7	10	0
HS15	306.50	6.7e-14	2.2e-11	17	17	0
HS16	0.25	1.1e-16	1.1e-16	15	17	0

Table 1. (cont.)

Prob.	Obj.	Const.	KKT	Iter.	Eval.	NC
HS17	1.0	5.4e-11	1.1e-08	16	16	0
HS18	4.999999	8.3e-13	8.4e-13	13	15	0
HS19	-6961.81388	7.8e-08	4.9e-05	17	17	0
HS20	40.19872981	1.2e-16	5.8e-14	8	9	0
HS21	-99.9599999	0	2.7e-13	5	5	0
HS21MOD	-99.9599998	0	3.1e-07	12	12	0
HS22	1.0	3.3e-16	3.2e-15	6	6	0
HS23	2.0	3.5e-12	1.5e-10	9	10	0
HS24	-1.0	4.4e-16	4.6e-12	8	10	3
	-1.0	9.1e-16	1.2e-12	6	6	0
HS29	-22.62741699	6.0e-10	6.1e-10	10	17	1
	-22.62741699	6.0e-10	6.0e-10	10	16	0
HS30	1.0	5.1e-10	9.6e-10	6	7	0
HS31	5.999999	3.1e-12	1.5e-09	5	5	0
HS32	1.00000001	7.9e-13	7.0e-08	14	14	0
HS33	-4.58578643	3.0e-14	5.2e-14	9	10	1
	-4.58578643	8.0e-13	2.2e-12	13	21	0
HS34	-0.83403244	4.6e-14	1.2e-13	9	9	0
HS35	0.11111111	1.1e-17	1.9e-10	7	7	0
HS36	-3299.9999	3.5e-15	9.2e-12	8	8	1
	-3299.9999	4.4e-26	1.2e-11	8	8	0
HS37	-3455.9999	3.5e-15	2.0e-06	7	8	0
HS41	1.92592592	0	1.3e-12	7	7	0
HS43	-44.0	7.7e-08	2.3e-07	14	23	0
HS44	-14.999999	1.8e-15	3.4e-12	9	9	3
	-12.999999	1.4e-15	6.6e-15	9	9	0
HS44NEW	-14.999999	1.8e-15	3.4e-15	9	9	3
	-12.999999	1.4e-15	6.6e-15	9	9	0
HS53	4.0930232	1.8e-15	6.2e-14	4	4	0
HS57	0.030647619	6.0e-11	3.7e-10	22	49	12
	0.030647619	1.7e-09	3.1e-09	27	50	0
HS59	-7.80278947	7.0e-11	7.2e-11	26	31	1
	-7.80278947	4.0e-10	8.6e-10	25	30	0
HS60	0.03256682	6.5e-12	1.8e-11	7	7	0
HS63	961.7151721	4.1e-11	1.2e-09	6	9	0
HS64	6299.84243	1.2e-16	2.5e-14	26	30	0
HS65	0.95352886	9.8e-32	1.4e-15	15	26	2
	0.95352886	2.1e-08	2.8e-11	17	31	0
HS66	0.518163274	6.2e-14	8.1e-14	8	8	0
HS67	-1162.119226	1.0e-09	7.9e-08	11	19	0
HS68	-0.920425004	1.4e-17	1.1e-14	24	35	0
HS69	-956.7128869	4.4e-09	9.4e-06	9	9	0
HS70	0.1870436431	0	1.0e-08	24	27	0
HS71	17.0140173	2.2e-14	3.7e-14	8	8	0
HS72	727.67936	1.7e-18	7.3e-12	24	24	0
HS73	29.894378	8.0e-15	1.2e-13	16	16	0
HS74	5126.4981	1.4e-11	6.5e-09	8	8	0
HS75	5174.4127	2.5e-13	4.3e-10	9	9	0

Table 1. (cont.)

Prob.	Obj.	Const.	KKT	Iter.	Eval.	NC
HS76	-4.681818181	1.3e-16	2.3e-10	7	7	0
HS80	0.0539498	9.3e-09	9.9e-09	9	9	0
HS81	0.0539498	1.5e-10	1.6e-10	9	9	0
HS83	-30665.539	3.4e-09	1.2e-12	19	19	0
HS84	-5280335.13	1.0e-08	1.8e-07	36	43	0
HS86	-32.348679	1.0e-14	1.0e-08	14	14	0
HS88	1.362656815	1.7e-17	1.1e-12	25	33	0
HS91	1.36265681	1.4e-13	4.9e-09	14	18	6
	1.36265681	3.2e-17	7.8e-13	19	34	0
HS92	1.36265681	4.6e-15	2.9e-09	21	27	6
	1.36265681	1.3e-14	8.3e-09	28	42	0
HS93	135.075963	3.2e-15	1.5e-07	9	9	0
HS95	0.0156195	6.9e-12	2.7e-10	11	11	0
HS96	0.0156195	2.6e-11	8.3e-10	11	11	0
HS97	4.0712463	6.2e-12	2.0e-09	13	13	0
HS98	4.0712463	9.7e-15	1.2e-12	16	19	1
	4.0712463	1.0e-10	2.0e-08	16	17	0
HS99	-8.3108e+08	4.4e-11	0.49945913	6	6	0
HS100	680.630057	9.0e-10	2.3e-09	9	13	1
	680.630057	1.9e-09	5.1e-09	10	15	0
HS104	3.9511634	1.5e-13	1.0e-12	12	12	0
HS105	1044.611692	2.6e-18	4.0e-09	17	21	2
	1044.611692	4.3e-18	9.4e-12	17	19	0
HS106	7049.24802	4.1e-11	4.1e-11	13	14	0
HS107	4797.98185	5.2e-14	7.5e-09	9	13	0
HS108	-0.8660254	3.2e-10	5.3e-10	19	24	0
HS109	--	--	--	--	--	--
HS110	-45.7784697	--	4.8e-13	5	5	0
HS111	-47.7610917	5.0e-08	9.0e-08	12	18	0
HS112	-47.7610908	2.5e-06	1.8e-08	12	18	0
HS113	24.306209	1.4e-11	2.3e-11	18	25	3
	24.306209	5.1e-09	5.8e-08	41	54	0
HS114	-1768.80696	3.1e-11	1.0e-10	17	18	0
HS116	97.5875096	2.9e-13	8.4e-11	32	36	0
HS117	32.3486790	1.2e-10	5.7e-10	23	27	0
HS118	664.820450	2.0e-14	3.1e-10	14	14	0
HS119	244.899697	2.2e-16	2.9e-08	12	12	0
HS268	2.5e-9	5.6e-15	5.0e-09	18	19	0
HUBFIT	0.016893495	2.9e-17	2.8e-09	7	7	0
KIWCRESC	-3.0e-15	9.0e-15	1.2e-14	10	15	0
LAUNCH	9.004903149	8.5e-11	1.9e-08	36	70	3
	9.004903149	5.9e-11	1.5e-08	37	69	0
LIN	-0.0175775	3.9e-17	4.1e-16	8	9	1
	-0.0175775	6.2e-17	5.6e-10	8	9	0
LOADBAL	0.4528510391	3.1e-12	4.8e-10	13	13	0
MADSEN	0.6164324355	1.1e-11	1.5e-11	11	12	0
MAKELA1	-1.414213564	2.0e-09	3.1e-09	12	12	0
MAKELA2	7.19999999	2.3e-11	8.8e-11	7	7	0

Table 1. (cont.)

MAKELA3	-1.2e-25	3.1e-14	3.2e-14	16	21	0
MATRIX2	1.1e-18	1.2e-17	3.8e-13	33	33	2
	8.5e-23	2.6e-24	6.1e-15	56	56	0
MIFFLIN1	-1.0	2.2e-15	5.9e-15	6	6	0
MIFFLIN2	-1.0	3.5e-15	4.0e-15	23	27	0
MINMAXBD	115.7064397	4.2e-09	4.3e-09	27	50	0
MINMAXRB	3.49e-16	2.5e-10	2.5e-10	8	10	0
MISTAKE	-1.0	2.7e-14	2.8e-14	11	11	3
	-1.0	4.7e-09	1.2e-08	10	10	0
ODFITS	-2380.026775	4.0e-14	1.2e-05	7	7	0
POLAK1	2.718281833	2.3e-08	3.1e-08	7	7	0
POLAK2	54.59815003	1.7e-09	4.0e-09	14	30	0
POLAK3	5.933003353	2.5e-08	3.0e-08	22	29	0
POLAK4	-3.8e-18	2.2e-18	3.4e-13	69	95	2
	-1.7e-18	3.0e-16	6.9e-13	115	153	0
POLAK6	-44.0	3.1e-10	3.3e-10	20	34	0
PRODPL0	58.79010	2.4e-13	3.5e-10	13	13	0
PRODPL1	35.73896744	2.3e-14	2.1e-11	16	16	1
	35.73896744	2.0e-13	5.2e-11	19	20	0
RK23	0.083333335	6.8e-10	2.6e-09	7	7	0
ROSENNMX	-44.0	6.2e-09	1.9e-08	33	53	1
	-44.0	8.2e-10	5.3e-09	147	205	0
S268	2.5e-09	5.6e-15	5.0e-09	18	19	0
TAME	0	0	1.9e-15	5	37	0
TENBARS4	368.4931619	2.8e-10	3.9e-10	34	43	0
TRUSPYR1	11.22874087	1.7e-12	1.5e-11	9	9	0
TRUSPYR2	11.22874090	1.9e-09	4.3e-08	10	11	0
TRY-B	1.8e-27	3.2e-14	8.1e-14	10	10	0
TWOBARs	1.508652417	3.8e-10	2.3e-09	6	6	0
WOMFLET	1.6e-14	2.6e-12	2.7e-12	15	159	1
	3.0e-14	4.1e-12	4.1e-12	16	135	0
ZECEVIC2	-4.125	4.6e-16	1.4e-11	8	8	0
ZECEVIC3	97.30945002	1.8e-08	2.5e-07	9	12	1
	97.30945014	5.9e-11	1.0e-09	11	15	0
ZECEVIC4	7.557507769	8.3e-13	9.6e-10	10	11	0
ZY2	2.0	3.2e-09	3.6e-09	6	6	0

Table 2 compares the results from the proposed algorithm (with and without negative curvature) to those of other codes reported in the literature, in particular those from [28], [16] and [32], on a set of 22 HS problems (all the problems that were reported in all of the references). The columns in the table correspond to the number of iterations (matrix factorizations) required by:

- CS: the proposed algorithm, using negative curvature.
- CS-nc: the proposed algorithm, when negative curvature was disabled.
- VS: iteration counts for LoQo, as reported in [28].
- Y: iteration counts reported in [32].
- GOW: iteration counts reported in [16].

From these results the proposed algorithm works better on the average, particularly when negative curvature is used, than any of the other three codes. Note that none of the three algorithms uses negative curvature explicitly. All initial points for the algorithms are

Table 2. Iteration counts for different nonlinear interior point codes

Prob.	CS	CS-nc	VS	Y	GOW
HS64	26	26	28	29	--
HS65	15	17	14	15	10
HS71	8	8	12	8	15
HS72	24	24	21	43	--
HS73	16	16	20	12	11
HS83	19	19	15	16	16
HS84	36	36	18	21	25
HS93	9	9	10	29	17
HS95	11	11	18	13	26
HS96	11	11	22	12	27
HS97	13	13	18	22	31
HS98	16	16	19	20	27
HS100	9	10	11	16	10
HS104	12	12	14	19	12
HS106	13	13	33	39	45
HS108	19	19	23	62	13
HS109	--	--	49	21	32
HS113	18	41	16	25	13
HS114	17	17	31	47	15
HS116	32	32	33	82	--
HS117	23	23	22	36	33
HS118	14	14	17	34	17
Average	17.19	17.95	21.09	28.23	20.79

Table 3. Iterations and function evaluations with and without negative curvature

Prob.	Iterations		Function evals.	
	CS	CS-nc	CS	CS-nc
Average	20.94	27.38	34.29	40.91

those indicated in [22]. For the GOW algorithm, only those results corresponding to these starting points are shown.

Regarding the impact of the use of negative curvature on the whole set of 145 test problems, Table 3 shows the average iteration counts and function evaluations required by the proposed algorithm for the 34 problems where negative curvature was detected, both when using negative curvature directions (CS) and when these directions were disabled (CS-nc). The average reductions in the number of iterations and function evaluations, whenever negative curvature was detected, is approximately equal to 24% for the iteration numbers and 16% for the function evaluations.

Table 4 presents a brief summary of the results. The table includes the total number of problems in the test set, the number of problems in which negative curvature was used, and the number of problems in which the use of negative curvature implied a decrease/no change/increase in the number of iterations and the number of function evaluations.

From the results in Table 4, negative curvature was used in 23% of the cases. Although in some cases (18%) there was an increase in the iteration count, in more than half of the cases (53%) there was a reduction in the number of iterations. Regarding the

Table 4. Impact of negative curvature

	# of problems	145
	# with neg. curv.	34
# Its.	Better	18
	Same	10
	Worse	6
# Eval	Better	15
	Same	6
	Worse	13

number of function evaluations the results were less satisfactory, although an overall decrease was still observed. In fact, the reductions in iterations and function evaluations were far more marked than the increases. The largest deterioration in the number of iterations amounted to 9 iterations (20%) for problem HIMMELP5, while the largest improvement was 114 iterations in problem ROSENMMX (80%). For those problems where negative curvature was used, a (geometric) average improvement of 15% in the number of iterations and 12% in the number of function evaluations was observed. As a consequence, using negative curvature seems to provide significant advantages, although from the observation of the different behavior in the numbers of iterations and function evaluations, special care should be taken in the procedure to compute the step length in the search to reduce the number of function evaluations when negative curvature is used.

9. Conclusions

In this paper we have described an efficient procedure that makes use of negative curvature directions to compute local solutions for nonconvex problems. The procedure is based on a primal-dual interior point method to define the search directions, and a curvilinear search to combine them. Particular care has been taken to introduce conditions that restrict in an appropriate manner the use of negative curvature. For example, whenever the constraints or the complementarity conditions are far from being satisfied any negative curvature information that may be available is ignored, and the effort is devoted to attaining feasibility or to satisfying the complementarity conditions. From a practical point of view, these restrictions on the use of negative curvature play a very significant role to ensure the efficiency of the procedure.

The implemented version of the algorithm has been run on a set of test problems. The results show that the procedure works quite well on these problems. The use of vector penalty and barrier parameters is in part responsible for this good behavior. The impact of the negative curvature is not very significant on these small problems (it is used in only 23% of them), but it can be quite important in some of the cases when it is used. Given the limited cost of computing a direction of negative curvature whenever an appropriate factorization is used to obtain the movement directions, it would seem that reasonable algorithms should incorporate the use of this second-order information.

A. Appendix

In this Appendix we include the detailed proofs for the theoretical results described in Section 6.

A.1. Proof of Lemma 1

The mechanism of the algorithm and the properties of the linesearch guarantee that for all k ,

$$L_A(x^{k+1}, \lambda^{k+1}; \rho^{k+1}, \mu^{k+1}) \leq L_A(x^k, \lambda^k; \rho^{k+1}, \mu^{k+1}). \quad (62)$$

Define the set $\mathcal{J} = \{1, \dots, n\} \setminus \mathcal{I}$. From Assumption A.3 there exists $\bar{x}_i > 1$ such that for all i and all k , $x_i^k \leq \bar{x}_i$. From Assumption A.4 and the definition of \mathcal{I} , there exists an index $r \geq 0$ such that $\rho^k = \rho^r$ and $\mu_i^k = \mu_i^r$ for all $k \geq r$ and all $i \in \mathcal{I}$. For all such k , the definition of the merit function (3) implies

$$\begin{aligned} L_A(x^k, \lambda^k; \rho^{k+1}, \mu^{k+1}) &= L_A(x^k, \lambda^k; \rho^k, \mu^k) - \sum_{i \in \mathcal{J}} (\mu_i^{k+1} - \mu_i^k) \log x_i^k \\ &\leq L_A(x^k, \lambda^k; \rho^k, \mu^k) - \sum_{i \in \mathcal{J}} (\mu_i^{k+1} - \mu_i^k) \log \bar{x}_i. \end{aligned} \quad (63)$$

Define $\sum_{i \in \mathcal{J}} \mu_i^r \log \bar{x}_i \equiv K$. Iterating (63) and using (62) it holds for all $k \geq r$,

$$\begin{aligned} L_A(x^k, \lambda^k; \rho^k, \mu^k) &= L(x^k, \lambda^k) - \sum_{i \in \mathcal{I}} \mu_i^r \log x_i^r - \sum_{j \in \mathcal{J}} \mu_j^k \log x_j^k \\ &\quad + \frac{1}{2} \sum_{l=1}^m \rho_l^r c_l^2(x^k) \\ &\leq L_A(x^r, \lambda^r; \rho^r, \mu^r) - \sum_{i \in \mathcal{J}} (\mu_i^k - \mu_i^r) \log \bar{x}_i \\ &\leq L_A(x^r, \lambda^r; \rho^r, \mu^r) + K. \end{aligned} \quad (64)$$

From Assumptions A.1 and A.3, there exists a constant $\delta \in \mathbb{R}$ such that for all $k \geq r$, $L(x^k, \lambda^k) \geq \delta$. The nonnegativity of the penalty parameter ρ implies $\sum_l \rho_l^r c_l^2(x^k) \geq 0$. From (57) we have $\mu_j^k \rightarrow 0$ for all $j \in \mathcal{J}$ and from Assumption A.3 there exists $\epsilon > 0$ such that $-\sum_{j \in \mathcal{J}} \mu_j^k \log x_j^k \geq -\epsilon$ for all $k \geq r$.

Inequality (64) and the above thus combine to yield

$$-\sum_{i \in \mathcal{I}} \mu_i^r \log x_i^r \leq L_A(x^r, \lambda^r; \rho^r, \mu^r) + K - \delta + \epsilon, \quad (65)$$

for all $k \geq r$. Let $\mu_+ = \max\{\mu_i^r, i \in \mathcal{I}\}$ and $\mu_- = \min\{\mu_i^r, i \in \mathcal{I}\}$. Also, for a given iteration k , let \mathcal{I}_+^k be the set of indices in \mathcal{I} such that $\log x_i^k$ is positive and $\mathcal{I}_-^k = \mathcal{I} \setminus \mathcal{I}_+^k$. From Assumption A.3 we have $-\sum_{i \in \mathcal{I}_+^k} \log x_i^k \geq -\sum_{i \in \mathcal{I}_+^k} \log \bar{x}_i \equiv -K^+$. Using these definitions we can write

$$\begin{aligned} -\sum_{i \in \mathcal{I}} \mu_i^r \log x_i^k &= -\sum_{i \in \mathcal{I}_+^k} \mu_i^r \log x_i^k - \sum_{i \in \mathcal{I}_-^k} \mu_i^r \log x_i^k \\ &\geq -\mu_+ \sum_{i \in \mathcal{I}_+^k} \log x_i^k - \mu_- \sum_{i \in \mathcal{I}_-^k} \log x_i^k \\ &= -\mu_- \sum_{i \in \mathcal{I}} \log x_i^k - (\mu_+ - \mu_-) \sum_{i \in \mathcal{I}_+^k} \log x_i^k \\ &\geq -\mu_- \sum_{i \in \mathcal{I}} \log x_i^k - (\mu_+ - \mu_-) K^+. \end{aligned} \quad (66)$$

Combining (64), (65) and (66) we obtain

$$\begin{aligned}
-\sum_{i \in \mathcal{I}} \log x_i^k &\leq \mu_-^{-1}(L_A(x^r, \lambda^r; \rho^r, \mu^r) + K - \delta + \epsilon + (\mu_+ - \mu_-)K^+) \\
&\Rightarrow \prod_{i \in \mathcal{I}} x_i^k \geq \exp\left(-\mu_-^{-1}(L_A(x^r, \lambda^r; \rho^r, \mu^r) + K - \delta + \epsilon + (\mu_+ - \mu_-)K^+)\right) \\
&\Rightarrow x_i^k \geq \exp\left(-\mu_-^{-1}(L_A(x^r, \lambda^r; \rho^r, \mu^r) + \hat{K})\right) \bar{x}_i / \prod_{j \in \mathcal{I}} \bar{x}_j,
\end{aligned}$$

for all $i \in \mathcal{I}$ and all $k \geq r$, where \bar{x}_i denotes an upper bound on x_i^k implied by Assumption A.3 and $\hat{K} \equiv K - \delta + \epsilon + (\mu_+ - \mu_-)K^+$. \square

A.2. Proof of Lemma 2

Consider a component μ_i^k , $i \in \mathcal{J}$, of the barrier parameter μ and let \mathcal{K}_i denote the (infinite) subsequence of iterations where it is updated. From (57), $\mu_i^{k+1} < \beta_\mu \mu_i^k$ for $k \in \mathcal{K}_i$, implying $\mu_i^k \rightarrow 0$. Also from (57), $\mu_i^k > \delta^k \hat{\mu}^k$ for $k \in \mathcal{K}_i$, implying either $\hat{\mu}^k \rightarrow 0$ or $\delta^k \rightarrow 0$ along the subsequence \mathcal{K}_i . Consider both cases:

- If $\delta^k \rightarrow 0$ for $k \in \mathcal{K}_i$, from (58) it follows $\min(0.25, \exp(-1/\theta_\mu^k)) \rightarrow 0 \Rightarrow \theta_\mu^k \rightarrow 0$ for $k \in \mathcal{K}_i$. From (52) and (53) it follows that

$$x_j^k z_j^k \rightarrow 0, \quad (67)$$

for $k \in \mathcal{K}_i$ and all j .

- If $\hat{\mu}^k \rightarrow 0$ for $k \in \mathcal{K}_i$, from (56) and $x^k \geq 0$, $z^k \geq 0$ the result in (67) follows also for this case.

Thus, (67) holds for $k \in \mathcal{K} \equiv \cup_{i \in \mathcal{J}} \mathcal{K}_i$ and all j .

Assume now that $\mathcal{I} = \{1, \dots, n\} \setminus \mathcal{J}$ is nonempty; from the definition of \mathcal{I} there exists $\bar{\mu} > 0$ such that $\mu_l^k \geq \bar{\mu}$ for all k and all $l \in \mathcal{I}$. Let $\bar{\gamma} \equiv \min(0.005, \bar{\mu})$; from the update of the dual variables (40)–(42), the definition of δ^k , (37), and $\bar{\alpha}_d^k \geq \alpha_d^k$, for any iteration k and any $l \in \mathcal{I}$,

$$\begin{aligned}
z_l^{k+1} &= z_l^k + \alpha_d^k (d_z^k)_l \geq z_l^k + \alpha_d^k \min(0, (d_z^k)_l) \\
&\geq z_l^k + \bar{\alpha}_d^k \min(0, (d_z^k)_l) \geq (1 - \delta^k) z_l^k \geq \bar{\gamma} z_l^k.
\end{aligned} \quad (68)$$

The definition of the search direction d_x^k from (17), the modifications in the coefficient matrix (18) and Assumption A.3 imply the existence of a constant \bar{x} such that $x_i^k \leq \bar{x}$ and $|(d_x^k)_i| \leq \bar{x}$ for all i and all k . Define $\gamma_z \equiv \min(\bar{\mu}/(4\bar{x}), \min_l z_l^0/2)$. For any iteration k and any $l \in \mathcal{I}$ such that $z_l^k < \gamma_z$, the definition of the dual search direction d_z^k in (10) and the preceding bounds imply

$$\begin{aligned}
x_l^k (d_z^k)_l &= \mu_l^k - (x_l^k + (d_x^k)_l) z_l^k \geq \bar{\mu} - 2\bar{x} z_l^k \geq \frac{1}{2} \bar{\mu} > 0 \Rightarrow (d_z^k)_l > 0 \\
&\Rightarrow z_l^{k+1} > z_l^k.
\end{aligned} \quad (69)$$

We now show that $x_l^k z_l^k$ is bounded away from zero for $l \in \mathcal{I}$. Consider the first iteration index m (if it exists) such that $z_l^m < \bar{\gamma} \gamma_z$, and $z_l^{m-1} \geq \bar{\gamma} \gamma_z$. From (68), $z_l^m \geq \bar{\gamma} z_l^{m-1} \Rightarrow z_l^{m-1} < \gamma_z$ but then (69) implies $z_l^m > z_l^{m-1} \geq \bar{\gamma} \gamma_z$, a contradiction implying no such iteration index m exists and $z_l^k \geq \bar{\gamma} \gamma_z$ for all k and all $l \in \mathcal{I}$. This bound and Lemma 1 yield

$$x_i^k z_i^k \geq \bar{\gamma}_x \bar{\gamma} \gamma_z > 0 \quad (70)$$

for all iterations k and all $i \in \mathcal{I}$. But (70) contradicts (67), as a consequence $\mathcal{I} = \emptyset$ whenever \mathcal{J} is nonempty.

Assume \mathcal{J} is nonempty, then $\mathcal{J} = \{1, \dots, n\}$ and $\lim_{k \rightarrow \infty} \mu_l^k = 0$ for all l . From (67) it holds that $y^k = X^k z^k \rightarrow 0$ for $k \in \mathcal{K}$. Also, (57), (58) and (55) imply

$$\begin{aligned} \mu_i^k &> \delta^k (\mu_i^*)^k = \theta_\mu^k \min(0.25, \exp(-(1/\theta_\mu^k))) \frac{y_i^k + \beta_y \|y^k\|}{\|y^k\|^2} \\ &\geq \beta_y \theta_\mu^k \min(0.25, \exp(-(1/\theta_\mu^k))) / \|y^k\|, \end{aligned} \quad (71)$$

for all $k \in \mathcal{K}_i$ and any i . From Assumption A.3 there exists a value K such that $\|y^k\| = \|X^k z^k\| \leq K$ for all k . Replacing this bound in (71) we have

$$\mu_i^k > \frac{\beta_y}{K} \theta_\mu^k \min(0.25, \exp(-(1/\theta_\mu^k))),$$

for all i and all iterations $k \in \mathcal{K}_i$. From this bound, $\theta_\mu^k \geq 0$ and $\lim_{k \rightarrow \infty} \mu_i^k = 0$ it follows that $\lim_{k \in \mathcal{K}} \theta_\mu^k = 0$. □

A.3. Proof of Lemma 3

If $\mathcal{J} = \emptyset$, from Assumption A.4 there exists an iteration index r such that $\rho^k = \rho^r$ and $\mu_i^k = \mu_i^r$ for all $k \geq r$ and all i . Adding conditions (32) and (33) over all iterations from r to any $k > r$ it holds that

$$L_A(x^{k+1}, \lambda^{k+1}; \rho^{k+1}, \mu^{k+1}) - L_A(x^r, \lambda^r; \rho^r, \mu^r) \leq \frac{\gamma}{2} \sum_{l=r}^k (\alpha_p^l)^2 (\phi^l)''(0). \quad (72)$$

From Assumptions A.1 and A.3, $L_A(x^{k+1}, \lambda^{k+1}; \rho^{k+1}, \mu^{k+1}) \geq \hat{\gamma}$ for all k and some constant $\hat{\gamma}$. Replacing this bound in (72), it follows that

$$-\sum_{l=r}^k (\alpha_p^l)^2 (\phi^l)''(0) \leq \frac{2}{\gamma} (L_A(x^r, \lambda^r; \rho^r, \mu^r) - \hat{\gamma}). \quad (73)$$

But for all k from (46) it holds that $(\phi^k)''(0) < 0$ and $\alpha_p^k > 0$. Then, (73) implies

$$\lim_{k \rightarrow \infty} (\alpha_p^k)^2 (\phi^k)''(0) = 0. \quad (74)$$

We now show that $(\alpha_p^k)^2$ is bounded away from zero. From the definition of d_x^k , (17), the conditions satisfied by d_n^k , (22), the definition of $d_{\lambda\rho}^k$ (29) and Assumptions A.1 and

A.4, there exists a constant γ_d such that $\|d_x^k\| \leq \gamma_d$, $\|d_n^k\| \leq \gamma_d$ and $\|d_{\lambda\rho}^k\| \leq \gamma_d$. From the definition of $\bar{\alpha}_i$ in (35) and Lemma 1,

$$\begin{aligned} x_i^k + (d_n^k)_i \bar{\alpha}_i^k + (d_x^k)_i (\bar{\alpha}_i^k)^2 &= 0 \Rightarrow x_i^k - |(d_n^k)_i| \bar{\alpha}_i^k - |(d_x^k)_i| (\bar{\alpha}_i^k)^2 \leq 0 \\ \Rightarrow x_i^k - \left(|(d_n^k)_i| + |(d_x^k)_i| \right) \min(1, \bar{\alpha}_i^k) &\leq 0 \Rightarrow \bar{\alpha}_i^k \geq \frac{x_i^k}{|(d_n^k)_i| + |(d_x^k)_i|} \geq \frac{\bar{\gamma}_x}{2\gamma_d}. \end{aligned}$$

This bound, together with (36) and (37), implies for all k ,

$$\alpha_{\max}^k \geq \frac{\delta^k \bar{\gamma}_x}{2\gamma_d} \geq \frac{0.995 \bar{\gamma}_x}{2\gamma_d} > 0. \quad (75)$$

The search procedure either defines $\tilde{\alpha}^k$ to be equal to α_{\max}^k or from (33) $\tilde{\alpha}^k$ satisfies

$$\phi^k(2\tilde{\alpha}^k) > \phi^k(0) + 2\gamma(\tilde{\alpha}^k)^2(\phi^k)''(0).$$

Replacing a Taylor series expansion for $\phi^k(2\tilde{\alpha}^k)$ around 0 in the preceding condition and using $(\phi^k)'(0) \leq 0$ from (43) and (26) and $(\phi^k)''(0) \leq 0$ for all k ,

$$\begin{aligned} &\phi^k(0) + 2\tilde{\alpha}^k(\phi^k)'(0) + 2(\tilde{\alpha}^k)^2(\phi^k)''(0) + \frac{4}{3}(\tilde{\alpha}^k)^3(\phi^k)'''(\xi^k) \\ &> \phi^k(0) + 2\gamma(\tilde{\alpha}^k)^2(\phi^k)''(0) \\ \Rightarrow &-(1-\gamma)(\tilde{\alpha}^k)^2(\phi^k)''(0) > \frac{2}{3}(\tilde{\alpha}^k)^3(\phi^k)'''(\xi^k) \\ \Rightarrow &\tilde{\alpha}^k > -\frac{3(1-\gamma)}{2} \frac{(\phi^k)''(0)}{(\phi^k)'''(\xi^k)}, \end{aligned} \quad (76)$$

where $\xi^k \in [0, 2\tilde{\alpha}^k]$. Consider the third derivative term $(\phi^k)'''(\xi^k)$; from Assumptions A.1, A.3 and A.4, the bounds on the search directions, the expressions for the first, second and third derivatives of the merit function in terms of the functions f and c , and Lemma 1, implying that the matrices $(X^k)^{-s}$ for $s = 1, 2, 3$ remain bounded, there exists a constant γ_1 such that $(\phi^k)'''(\xi^k) \leq \gamma_1$. Replacing this bound in (76) and using (75) it follows that

$$\tilde{\alpha}^k \geq \min \left(\frac{0.995 \bar{\gamma}_x}{2\gamma_d}, -(\phi^k)''(0) \frac{3(1-\gamma)}{2\gamma_1} \right). \quad (77)$$

The last step in the computation of α_p^k is based on the satisfaction of condition (38). Define $\psi^k(\alpha) \equiv \|c(x^k(\alpha))\|$. If $\psi^k(\tilde{\alpha}^k) \leq \beta_c$, then $\alpha_p^k = \tilde{\alpha}^k$ and (77) holds for α_p^k . Otherwise, consider two cases:

- If $\psi^k(0) = \|c(x^k)\| < \frac{1}{2}\beta_c$ holds. Define α_c^k as the largest value such that $\psi^k(\alpha) \leq \beta_c$ for all $\alpha \in [0, \alpha_c^k]$. From the Taylor series expansion of ψ^k around zero,

$$\beta_c = \psi^k(\alpha_c^k) = \psi^k(0) + \alpha(\psi^k)'(\zeta^k),$$

for $\zeta^k \in [0, \alpha_c^k]$. Assumptions A.1 and A.3 and the boundedness of the search directions imply the existence of a constant γ_{c1} such that $(\psi^k)'(\zeta^k) = (2\zeta^k d_x^k + d_n^k)^T \nabla c(x^k(\zeta^k)) c(x^k(\zeta^k)) / \|c(x^k(\zeta^k))\| \leq \gamma_{c1}$, and it holds that

$$\alpha_c^k = \frac{\psi^k(\alpha_c^k) - \psi^k(0)}{(\psi^k)'(\zeta^k)} \geq \frac{\beta_c}{2\gamma_{c1}}. \quad (78)$$

- If $\psi^k(0) \geq \frac{1}{2}\beta_c$, from $\beta_c > 2\beta_v$ it holds that $d_n^k = 0$ (see Section 3.4). It follows that $(\psi^k)'(0) = c(x^k)^T \nabla c(x^k) d_n^k / \|c(x^k)\| = 0$ and from $\nabla c(x^k) d_x^k = -c(x^k)$, a consequence of (17), and $d_n^k = 0$,

$$\begin{aligned} (\psi^k)''(0) &= (\|\nabla c(x^k) d_n^k\|^2 + \sum_j c_j(x^k) (d_n^k)^T \nabla^2 c_j(x^k) d_n^k \\ &\quad + 2c(x^k)^T \nabla c(x^k) d_x^k \|c(x^k)\| - (c(x^k)^T \nabla c(x^k) d_n^k)^2) / \|c(x^k)\|^2 \\ &= -2\|c(x^k)\| = -2\psi^k(0). \end{aligned}$$

For the Taylor series expansion of $\psi(\alpha_c^k)$ around zero, using the preceding expressions for the derivatives,

$$\begin{aligned} \psi^k(\alpha_c^k) &= \psi^k(0) + \alpha_c^k (\psi^k)'(0) + \frac{1}{2} (\alpha_c^k)^2 (\psi^k)''(0) + \frac{1}{6} (\alpha_c^k)^3 (\psi^k)'''(\zeta^k) \\ &= \psi^k(0) - (\alpha_c^k)^2 \psi^k(0) + \frac{1}{6} (\alpha_c^k)^3 (\psi^k)'''(\zeta^k), \end{aligned} \quad (79)$$

where $\zeta^k \in [0, \alpha]$. From Assumptions A.1 and A.3, there exists a constant γ_{c3} such that $(\psi^k)'''(\zeta^k) \leq \gamma_{c3}$. From (38) it holds that $\psi^k(0) \leq \beta_c$ for all k , and $\psi^k(\alpha_c^k) - \psi^k(0) \geq 0$. Replacing these bounds in (79) we obtain,

$$\frac{1}{2} (\alpha_c^k)^2 \left(-\psi^k(0) + \frac{1}{3} \alpha_c^k (\psi^k)'''(\zeta^k) \right) \geq 0 \Rightarrow \alpha_c^k \geq \frac{3\psi^k(0)}{(\psi^k)'''(\zeta^k)} \geq \frac{3\beta_c}{2\gamma_{c3}}. \quad (80)$$

From the backtracking procedure used to compute α_p^k from $\tilde{\alpha}^k$, (77), (78) and (80),

$$\alpha_p^k \geq \min \left(\frac{1}{2} \alpha_c^k, \tilde{\alpha}^k \right) \geq \min \left(\frac{\beta_c}{4\gamma_c}, \frac{0.995\bar{\gamma}_x}{2\gamma_d}, -(\phi^k)''(0) \frac{3(1-\gamma)}{2\gamma_1} \right), \quad (81)$$

for all k , where $\gamma_c = \max(\gamma_{c1}, \gamma_{c3})$. Replacing this bound in (74) we obtain

$$\lim_{k \rightarrow \infty} (\phi^k)''(0) = 0. \quad (82)$$

From Assumption A.5 and (46) we have $(\phi^k)''(0) \leq -\|c(x^k)\|^2$ and (82) implies $\lim_{k \rightarrow \infty} c(x^k) = 0$. The definition (45) and the equalities in (17) yield

$$\theta_\rho^k = -(d_x^k)^T \bar{G}_\rho^k d_x^k - 2c(x^k)^T d_\lambda^k + c(x^k)^T R^k c(x^k). \quad (83)$$

To proceed with the analysis, we write d_x^k as the direct sum of a component in the null-space of $\nabla c(x^k)$, d_1^k , and another component in the orthogonal subspace, d_2^k , as $d_x^k = W_A^k d_1^k + \nabla c(x^k)^T d_2^k$. From (17) and Assumption A.5, $\nabla c(x^k) \nabla c(x^k)^T d_2^k = -c(x^k)$; this equality and $c(x^k) \rightarrow 0$ imply $\lim_{k \rightarrow \infty} \|d_2^k\| = 0$. Then, (83) can be written as

$$\begin{aligned} \theta_\rho^k + (d_1^k)^T (W_A^k)^T \bar{G}_\rho^k W_A^k d_1^k &= -2(d_1^k)^T (W_A^k)^T \bar{G}_\rho^k \nabla c(x^k)^T d_2^k \\ &\quad - (d_2^k)^T \nabla c(x^k) \bar{G}_\rho^k \nabla c(x^k)^T d_2^k - 2c(x^k)^T d_\lambda^k + c(x^k)^T R^k c(x^k). \end{aligned}$$

From Assumptions A.1, A.3, $\lim_{k \rightarrow \infty} \|c(x^k)\| = 0$ and $\lim_{k \rightarrow \infty} \|d_2^k\| = 0$, the right-hand side of the preceding expression converges to zero, and

$$\lim_{k \rightarrow \infty} \left(\theta_\rho^k + (d_1^k)^T (W_A^k)^T \bar{G}_\rho^k W_A^k d_1^k \right) = 0. \quad (84)$$

As (46) and (82) imply $\lim_{k \rightarrow \infty} \min(\theta_\rho^k, 0) = 0$, using this result in (84) it follows

$$\begin{aligned} \lim_{k \rightarrow \infty} \left(\max(\theta_\rho^k, 0) + (d_1^k)^T (W_A^k)^T \bar{G}_\rho^k W_A^k d_1^k \right) &= 0 \\ \Rightarrow \lim_{k \rightarrow \infty} (d_1^k)^T (W_A^k)^T \bar{G}_\rho^k W_A^k d_1^k &= 0 \Rightarrow \lim_{k \rightarrow \infty} \|d_1^k\| = 0, \end{aligned}$$

where the last implication follows from the positive definiteness of $(W_A^k)^T \bar{G}_\rho^k W_A^k$. Thus, for the sequence of search directions, $\lim_{k \rightarrow \infty} \|d_x^k\| = 0$, and from (17),

$$\begin{aligned} \bar{G}_\rho^k d_x^k &= -\nabla f(x^k) + (X^k)^{-1} \mu^k + \nabla c(x^k)^T (\lambda^k + d_\lambda^k) \\ \Rightarrow \lim_{k \rightarrow \infty} \|\nabla f(x^k) - (X^k)^{-1} \mu^k - \nabla c(x^k)^T (\lambda^k + d_\lambda^k)\| &= 0. \end{aligned}$$

This result, together with $\lim_{k \rightarrow \infty} \|c(x^k)\| = 0$, implies that the limit points of the sequence generated by the algorithm are first-order KKT points for problem (2). \square

A.4. Proof of Lemma 4

From (82) in Lemma 3, (44) and (27), it follows that if \mathcal{J} is empty then $\lim_{k \rightarrow \infty} \|d_n^k\| = 0$ and $\lim_{k \rightarrow \infty} \|x^{k+1} - x^k\| = 0$. Also, there exists a value $\bar{\mu} > 0$ such that $\mu^k \geq \bar{\mu}$ for all k .

Consider the sequence $\{\hat{\lambda}^k\}$, defined from $\hat{\lambda}^{k+1} = \lambda^k + d_{\lambda,\rho}^k = \lambda^k + d_\lambda^k - R^k c(x^k)$. From Taylor series expansions and Assumptions A.1 and A.3

$$\begin{aligned} \nabla f(x^k) &= \nabla f(x^{k-1}) + O(\|x^k - x^{k-1}\|) \\ \nabla c(x^k) &= \nabla c(x^{k-1}) + O(\|x^k - x^{k-1}\|), \end{aligned}$$

and using Lemma 1, implying the boundedness of $(X^k)^{-1}$, and the convergence of $\{\mu^k\}$,

$$\begin{aligned} (X^k)^{-1} \mu^k &= (X^{k-1})^{-1} \mu^{k-1} + (X^k)^{-1} (\mu^k - \mu^{k-1}) \\ &\quad - (X^k - X^{k-1})(X^k X^{k-1})^{-1} \mu^{k-1} = (X^{k-1})^{-1} \mu^{k-1} + o(1). \end{aligned}$$

Combining these results, it holds that

$$\begin{aligned} \nabla f(x^k) - (X^k)^{-1} \mu^k - \nabla c(x^k)^T \hat{\lambda}^k &= \nabla f(x^{k-1}) - (X^{k-1})^{-1} \mu^{k-1} \\ &\quad - \nabla c(x^{k-1})^T (\lambda^{k-1} + d_\lambda^{k-1} - R^{k-1} c(x^{k-1})) + o(1). \end{aligned}$$

From Assumption A.4 and Lemma 3 we have that $R^k c(x^k) \rightarrow 0$ and

$$\lim_{k \rightarrow \infty} \|\nabla f(x^k) - (X^k)^{-1} \mu^k - \nabla c(x^k)^T \hat{\lambda}^k\| = 0.$$

Consider now the sequence $\{z^k\}$. From (40) we have

$$\begin{aligned} \|\mu^{k+1} - X^{k+1} z^{k+1}\| &\leq \|\mu^{k+1} - \mu^k\| + \|X^{k+1} - X^k\| \|z^{k+1}\| + \|\mu^k - X^k z^{k+1}\| \\ &\leq \|\mu^{k+1} - \mu^k\| + \|X^{k+1} - X^k\| \|z^{k+1}\| \\ &\quad + (1 - \alpha_d^k) \|\mu^k - X^k z^k\| + \alpha_d^k \|\mu^k - X^k (z^k + d_z^k)\| \\ &\leq \tau^k + (1 - \alpha_d^k) \|\mu^k - X^k z^k\|, \end{aligned} \tag{85}$$

where $t^k \equiv \|\mu^{k+1} - \mu^k\| + \|X^{k+1} - X^k\| \|z^{k+1}\| + \alpha_d^k \|\mu^k - X^k(z^k + d_z^k)\|$. From (10) it holds that $\mu^k - X^k(z^k + d_z^k) = Z^k d_x^k$ and from Lemma 3, $\|d_x^k\| \rightarrow 0$; then Assumption A.3 implies $\lim_{k \rightarrow \infty} t^k = 0$. To simplify the notation, let $u^k \equiv \|\mu^k + X^k z^k\|$; from Assumption A.3 there exists a constant γ_u such that $u^k \leq \gamma_u$ and $t^k \leq \gamma_u$ for all iterations k . Also, let $\omega^k \equiv \prod_{i=j}^{k-1} (1 - \alpha_d^i)$.

From (10), Lemma 1 and $\|d_x^k\| \rightarrow 0$ in Lemma 3, if $(d_z^k)_i < 0$ it holds that $z_i^k > \mu_i^k / (x_i^k + (d_x^k)_i) \geq \bar{\mu}_i / (2\bar{x}_i)$ for some iteration s and all $k \geq s$, where \bar{x}_i is an upper bound on x_i^k from Assumption A.3. Again from (10), as \bar{z}_i satisfies from Assumption A.3 $z_i^k \leq \bar{z}_i$ for some constant \bar{z}_i ,

$$|(d_z^k)_i| = \frac{|\mu_i^k - z_i^k(x_i^k + (d_x^k)_i)|}{x_i^k} \leq \frac{\bar{\mu}_i + 2\bar{z}_i\bar{x}_i}{\bar{\gamma}_x},$$

and from (41) and (37) it follows that $\bar{\alpha}_d^k \geq 0.995\bar{\mu}_i\bar{\gamma}_x / (2\bar{x}_i(\bar{\mu}_i + 2\bar{z}_i\bar{x}_i)) \equiv \bar{\gamma}_z$. From (42) it then follows that $\alpha_d^k \geq \beta_z\bar{\gamma}_z > 0$ for all $k \geq s$. As $z_i^k > 0$ implies $\alpha_d^k > 0$ for all k and all i , define $\tilde{\gamma}_z \equiv \min(\min_{0 \leq k \leq s} \alpha_d^k, \beta_z\bar{\gamma}_z) > 0$ and it holds that $\alpha_d^k \geq \tilde{\gamma}_z$ for all k .

Iterating on (85) and using the bound on α_d^k we obtain the inequality

$$u^k \leq \sum_{j=0}^{k-1} \omega^{j+1,k} t^j + \omega^{0,k} u^0 \leq \sum_{j=0}^{k-1} (1 - \tilde{\gamma}_z)^{k-j-1} t^j + (1 - \tilde{\gamma}_z)^k u^0. \quad (86)$$

For any $\epsilon > 0$, from $\lim_{k \rightarrow \infty} t^k = 0$ there exists an iteration index K such that $t^j \leq \epsilon\tilde{\gamma}_z/2$. Consider iterations k such that $k \geq K + \lceil \log(\epsilon\tilde{\gamma}_z/(2\gamma_u)) / \log(1 - \tilde{\gamma}_z) \rceil$. From $1 - \beta_z < 1$ and (86),

$$\begin{aligned} u^k &\leq \sum_{j=K}^{k-1} (1 - \tilde{\gamma}_z)^{k-j-1} t^j + \sum_{j=0}^{K-1} (1 - \tilde{\gamma}_z)^{k-j-1} t^j + (1 - \tilde{\gamma}_z)^k u^0 \\ &\leq \frac{1}{2}\epsilon\tilde{\gamma}_z \sum_{j=K}^{k-1} (1 - \tilde{\gamma}_z)^{k-j-1} + \gamma_u \sum_{j=0}^K (1 - \tilde{\gamma}_z)^{k-j} \\ &\leq \frac{1}{2}\epsilon\tilde{\gamma}_z \sum_{j=0}^{\infty} (1 - \tilde{\gamma}_z)^j + \gamma_u (1 - \tilde{\gamma}_z)^{k-K} \frac{1 - (1 - \tilde{\gamma}_z)^{K+1}}{\tilde{\gamma}_z} \\ &\leq \frac{1}{2}\epsilon + \frac{\gamma_u (1 - \tilde{\gamma}_z)^{k-K}}{\tilde{\gamma}_z} \leq \frac{1}{2}\epsilon + \frac{1}{2}\epsilon = \epsilon, \end{aligned}$$

implying $\lim_{k \rightarrow \infty} u^k = 0$, or equivalently $\lim_{k \rightarrow \infty} \|X^k z^k - \mu^k\| = 0$. □

A.5. Proof of Theorem 2

Assume that the barrier parameter is updated a finite number of times. Then, there exists an iteration index r such that $\mu^k = \mu^r$ for all $k \geq r$. From Lemmas 3 and 4 it holds in

this case

$$\begin{aligned}\lim_{k \rightarrow \infty} \|\nabla f(x^k) - z^k - \nabla c(x^k)^T \hat{\lambda}^k\| &= 0 \\ \lim_{k \rightarrow \infty} \|c(x^k)\| &= 0, \quad \lim_{k \rightarrow \infty} X^k z^k = \mu^r.\end{aligned}\tag{87}$$

These results imply $\lim_{k \rightarrow \infty} y^k = \lim_{k \rightarrow \infty} X^k z^k = \mu^r$ and $\lim_{k \rightarrow \infty} \|(X^k)^{-1} \mu^r - z^k\| = 0$. From (52), (53) and (87),

$$\lim_{k \rightarrow \infty} \theta_\mu^k = \begin{cases} \|\mu^r\|^2 & \text{if } \|\mu^r\| < 1 \\ \|\mu^r\| & \text{otherwise.} \end{cases}$$

From (55),

$$\lim_{k \rightarrow \infty} (\mu^*)^k = \begin{cases} \mu^r + \beta_y \|\mu^r\| e & \text{if } \|\mu^r\| < 1 \\ \mu^r / \|\mu^r\| + \beta_y e & \text{otherwise.} \end{cases}$$

Finally, from (56) we have $\lim_{k \rightarrow \infty} \hat{\mu}^k = \sum_i \mu_i^r / n$.

Let $l \in \arg \max_i \mu_i^r$ and note that $\mu_l^r \geq \|\mu^r\| / \sqrt{n}$ and $\mu_l^r + \beta_y \|\mu^r\| \leq 2\mu_l^r$. From the preceding limits there exists an iteration index $K > r$ such that $(\mu_l^*)^k \leq 3\mu_l^r$ and $\hat{\mu}^k \leq 3 \sum_i \mu_i^r / n \leq 3\mu_l^r$ for all $k \geq K$. Also, from (58) it holds that $\delta^k \leq 0.25$ implying

$$\delta^k \max \left((\mu_l^*)^k, \hat{\mu}^k \right) \leq \frac{3}{4} \mu_l^r < \mu_l^r = \mu_l^k,$$

for all $k \geq K$. From (57) the update condition is satisfied for μ^{k+1} and $\mu_l^{k+1} = \beta_\mu \delta^k \max \left((\mu_l^*)^k, \hat{\mu}^k \right) \leq \frac{3}{4} \beta_\mu \mu_l^r < \mu_l^r = \mu_l^{k+1}$, a contradiction. As a consequence, μ^k must be updated an infinite number of times and Lemma 2 applies, implying $\lim_{k \rightarrow \infty} \mu^k = 0$ and the existence of an infinite sequence \mathcal{K} such that $\theta_\mu^k \rightarrow 0$ for $k \in \mathcal{K}$. From the definition of θ_μ , (53), it holds that

$$\lim_{k \in \mathcal{K}} \|\nabla f(x^k) - \nabla c(x^k)^T \hat{\lambda}^k - z^k\| = 0, \quad \lim_{k \in \mathcal{K}} \|c(x^k)\| = 0.\tag{88}$$

For the third set of components in θ_μ it holds that $\lim_{k \in \mathcal{K}} (1 - \beta_m) x_i^k z_i^k + \beta_m (z_i^k + \mu_i^k - \mu_i^k / x_i^k) = 0$ for all i . For any $\epsilon > 0$ and any i there exists an iteration index \bar{K} such that for all $k \in \mathcal{K}$, $k \geq \bar{K}$,

$$\left| \frac{1 - \beta_m}{\beta_m} x_i^k z_i^k + z_i^k - \frac{\mu_i^k}{x_i^k} \right| \leq \epsilon.\tag{89}$$

Define $\gamma_2 \equiv \epsilon \beta_m / ((1 - \beta_m) \bar{\gamma}_z)$, where from Assumption A.3 $\bar{\gamma}_z > 0$ is a constant such that $z_i^k \leq \bar{\gamma}_z$ for all k and all i . From $\lim_{k \rightarrow \infty} \mu^k = 0$ there exists another iteration index \hat{K} such that $\mu_i^k \leq \epsilon \gamma_2 (\beta_m + 2\gamma_2(1 - \beta_m)) / (2\beta_m + \gamma_2(1 - \beta_m))$ for all $k \geq \hat{K}$ and all i ; let $K = \max(\bar{K}, \hat{K})$.

For any $k \in \mathcal{K}$, $k \geq K$, if $x_i^k \leq \gamma_1$ from (89) it holds that

$$\left| z_i^k - \frac{\mu_i^k}{x_i^k} \right| \leq \epsilon + \frac{1 - \beta_m}{\beta_m} \bar{\gamma}_z \gamma_2 = 2\epsilon,\tag{90}$$

while if $x_i^k > \gamma_1$, using again (89),

$$\begin{aligned}
& \left| \frac{1-\beta_m}{\beta_m} x_i^k z_i^k + z_i^k - \frac{\mu_i^k}{x_i^k} \right| \geq \left(\frac{1-\beta_m}{\beta_m} x_i^k + 1 \right) z_i^k - \frac{\mu_i^k}{\gamma_2} \\
& \Rightarrow \left(\frac{1-\beta_m}{\beta_m} x_i^k + 1 \right) z_i^k \leq \epsilon + \frac{\mu_i^k}{\gamma_2} \Rightarrow z_i^k \leq \frac{\epsilon + \mu_i^k / \gamma_2}{1 + \gamma_2(1-\beta_m)/\beta_m} \\
& \Rightarrow \left| z_i^k - \frac{\mu_i^k}{x_i^k} \right| \leq \frac{\epsilon + \mu_i^k / \gamma_2}{1 + \gamma_2(1-\beta_m)/\beta_m} + \frac{\mu_i^k}{\gamma_2} \leq 2\epsilon,
\end{aligned} \tag{91}$$

and conditions (90) and (91), together with (89), imply

$$\lim_{k \in \mathcal{K}} \|(X^k)^{-1} \mu^k - z^k\| = 0 \quad , \quad \lim_{k \in \mathcal{K}} \|X^k z^k\| = 0. \tag{92}$$

As the algorithm ensures that $x^k \geq 0$ and $z^k \geq 0$ for all k , (88) and (92) establish that along the sequence \mathcal{K} all limit points are first-order KKT points of problem (1).

We now show that these limit points are also second-order KKT points of (1). In particular, we show that if this is not the case, then the merit function (3) must be unbounded below, against Assumptions A.1 and A.3. Consider an infinite subsequence $\hat{\mathcal{K}} \subset \mathcal{K}$ such that $\lim_{k \in \hat{\mathcal{K}}} x^k = x^*$, $\lim_{k \in \hat{\mathcal{K}}} \hat{\lambda}^k = \lambda^*$ and $\lim_{k \in \hat{\mathcal{K}}} z^k = z^*$. The existence of this subsequence is a consequence of Assumption A.3. From (88) and (92) these values satisfy

$$\begin{aligned}
& \|\nabla f(x^*) - z^* - \nabla c(x^*)^T \lambda^*\| = 0 \quad , \quad \|c(x^*)\| = 0 \quad , \\
& \|X^* z^*\| = 0 \quad , \quad x^* \geq 0, z^* \geq 0.
\end{aligned} \tag{93}$$

Assume that $\lambda_{\min}((W_j^*)^T \nabla_{xx} L(x^*, \lambda^*) W_j^*) \equiv \gamma_\lambda < 0$ for W_j^* defined as in Theorem 1, implying that x^* is not a second-order KKT point of problem (1).

Under this condition, from Assumption A.2, Theorem 1, (92) and (25) there exists an iteration index r_1 such that for all $k \geq r_1$, $k \in \hat{\mathcal{K}}$, it holds that

$$(d_n^k)^T G_\rho^k d_n^k \leq -\frac{1}{2} c_1 \gamma_\lambda^2. \tag{94}$$

From (93), implying $\lim_{k \in \hat{\mathcal{K}}} \|d_x^k\| = 0$, Assumptions A.1 and A.3, there exists another iteration index r_2 such that

$$\begin{aligned}
& |2(d_x^k)^T (\nabla f(x^k) - (X^k)^{-1} \mu^k - \nabla c(x^k)^T (\lambda^k - R^k c(x^k))) - 2(d_{\lambda\rho}^k)^T c(x^k)| \\
& \leq \frac{1}{8} c_1 \gamma_\lambda^2,
\end{aligned} \tag{95}$$

for all $k \geq r_2$, $k \in \hat{\mathcal{K}}$. Again from (93),

$$|(d_n^k)^T (M^k (X^k)^{-2} - (X^k)^{-1} Z^k) d_n^k| \leq \frac{1}{8} c_1 \gamma_\lambda^2, \tag{96}$$

for some iteration index r_3 and $k \geq r_3$, $k \in \hat{\mathcal{K}}$. Select r_4 large enough that from Assumption A.4 it holds $\rho^k = \rho^r$ for all $k \geq r_4$, and let $r \equiv \max(r_1, r_2, r_3, r_4)$. Replacing (94), (95) and (96) in (44) yields

$$(\phi^k)''(0) \leq -\frac{1}{4} c_1 \gamma_\lambda^2, \tag{97}$$

for all $k \geq r$ and $k \in \hat{\mathcal{K}}$.

The change in the merit function for iterations larger than r is due to the modification of the variables in the search and the update of the barrier parameters. From (33), for $k \geq r$ it holds that

$$L_A(x^{k+1}, \lambda^{k+1}; \rho^{k+1}, \mu^{k+1}) \leq L_A(x^k, \lambda^k; \rho^k, \mu^k) + \frac{1}{2}\gamma(\alpha_p^k)^2(\phi^k)''(0) \\ + \sum_{i=1}^n(\mu_i^k - \mu_i^{k+1}) \log x_i^{k+1}.$$

From Assumption A.3 there exists $\bar{x}_i > 1$ such that $x_i^k \leq \bar{x}_i$ for all k and all i . Replacing this bound in the preceding inequality, we have

$$L_A(x^{k+1}, \lambda^{k+1}; \rho^{k+1}, \mu^{k+1}) \leq L_A(x^k, \lambda^k; \rho^k, \mu^k) + \frac{1}{2}\gamma(\alpha_p^k)^2(\phi^k)''(0) \\ + \sum_{i=1}^n(\mu_i^k - \mu_i^{k+1}) \log \bar{x}_i.$$

Adding these bounds from r to k , taking into account that $\log \bar{x}_i > 0$ and $(\phi^j)''(0) < 0$, and using (97),

$$L_A(x^{k+1}, \lambda^{k+1}; \rho^{k+1}, \mu^{k+1}) \leq L_A(x^r, \lambda^r; \rho^r, \mu^r) + \frac{1}{2}\gamma \sum_{j=r}^k (\alpha_p^j)^2 (\phi^j)''(0) \\ + \sum_{i=1}^n (\mu_i^r - \mu_i^{k+1}) \log \bar{x}_i \\ \leq L_A(x^r, \lambda^r; \rho^r, \mu^r) + \frac{1}{2}\gamma \sum_{j \in \hat{\mathcal{K}}: j \leq k} (\alpha_p^j)^2 (\phi^j)''(0) \\ + \sum_{i=1}^n \mu_i^r \log \bar{x}_i \\ \leq L_A(x^r, \lambda^r; \rho^r, \mu^r) - \frac{1}{8}\gamma c_1 \gamma_\lambda^2 \sum_{j \in \hat{\mathcal{K}}: j \leq k} (\alpha_p^j)^2 \\ + \sum_{i=1}^n \mu_i^r \log \bar{x}_i,$$

for any $k \geq r$. From Assumptions A.1, A.3 and $-\sum_i \mu_i^k \log x_i^k \geq -\sum_i \mu_i^0 \log \bar{x}_i$, where \bar{x}_i is an upper bound on x_i^k from Assumption A.3, there exists a constant γ_L such that $L_A(x^k, \lambda^k; \rho^k, \mu^k) \geq \gamma_L$ for all k . As a consequence, if (97) holds then

$$\frac{1}{8}\gamma c_1 \gamma_\lambda^2 \sum_{j \in \hat{\mathcal{K}}: j \leq k} (\alpha_p^j)^2 \leq -\gamma_L + L_A(x^r, \lambda^r; \rho^r, \mu^r) + \sum_{i=1}^n \mu_i^r \log \bar{x}_i, \quad (98)$$

for any $k \geq r$. From (98) it should hold that $\lim_{k \in \hat{\mathcal{K}}} \alpha_p^k = 0$.

To complete the proof of convergence to second-order KKT points we now show that α_p^k is bounded away from zero along the sequence $\hat{\mathcal{K}}$, which contradicts (98) and as a consequence implies that (97) cannot hold. The computation of α_p^k is carried out in three stages: it must satisfy the condition (38) on the constraints, but the bound is in this case identical to the ones obtained in Lemma 3, (78) and (80), as the arguments were independent of μ . It must also preserve the positivity of the primal variables and produce sufficient descent on the merit function; in both of these cases, the arguments in the proof of Lemma 3 are no longer valid, as they depended on the boundedness of the barrier parameter, and the corresponding bound away from zero on the primal variables. Now $\mu^k \rightarrow 0$ and the term $-\sum_i \mu_i \log x_i$ has unbounded higher derivatives as $\mu^k \rightarrow 0$.

We derive a bound for the step to ensure sufficient descent on the merit function, condition (33). Let $\phi^k(\alpha) = \zeta^k(\alpha) - \sum_i \mu_i^k \log x_i^k(\alpha)$, where $\zeta^k(\alpha) \equiv f(x^k(\alpha)) - \lambda^k(\alpha)^T c(x^k(\alpha)) + 0.5 \sum_j \rho_j^k c_j^2(x^k(\alpha))$ has continuous and uniformly bounded derivatives from Assumption A.1. We show that sufficient descent on ζ implies sufficient

descent on ϕ , and as a consequence that the corresponding step is bounded away from zero. The definition of ζ and (44) imply

$$(\phi^k)''(0) = (\zeta^k)''(0) + (d_n^k)^T M^k (X^k)^{-2} d_n^k - 2d_x^k (X^k)^{-1} \mu^k. \quad (99)$$

From (93), Assumption A.1 and (17) it follows that $\lim_{k \in \hat{\mathcal{K}}} \|d_x^k\| = 0$ and there exists an iteration index $\hat{r} \geq \bar{r}$ such that, from (92),

$$\begin{aligned} (d_n^k)^T M^k (X^k)^{-2} d_n^k - 2d_x^k (X^k)^{-1} \mu^k &= \sum_{i=1}^n \frac{\mu_i^k}{x_i^k} \left(\frac{(d_n^k)_i^2}{x_i^k} - (d_x^k)_i \right) \geq -\frac{1}{16} c_1 \gamma_\lambda^2 \\ \Rightarrow (\zeta^k)''(0) &\leq (\phi^k)''(0) + \frac{1}{16} c_1 \gamma_\lambda^2 \leq \frac{3}{4} (\phi^k)''(0) \leq -\frac{3}{16} c_1 \gamma_\lambda^2, \end{aligned} \quad (100)$$

for all $k \geq \hat{r}$ and $k \in \hat{\mathcal{K}}$, where the last inequality follows from (97) and (99). The arguments used in the proof of Lemma 3 to derive the bound in (81) can be applied to the function ζ^k . It holds that

$$\zeta^k(\alpha^k) \leq \zeta^k(0) + \alpha^k (\zeta^k)'(0) + \gamma (\alpha^k)^2 (\zeta^k)''(0), \quad (101)$$

for all $\alpha^k \in (0, \hat{\alpha}^k]$, where $\hat{\alpha}^k$ satisfies

$$\begin{aligned} \zeta^k(\hat{\alpha}^k) &= \zeta^k(0) + \hat{\alpha}^k (\zeta^k)'(0) + \gamma (\hat{\alpha}^k)^2 (\zeta^k)''(0) \\ \Rightarrow \gamma (\hat{\alpha}^k)^2 (\zeta^k)''(0) &= \frac{1}{2} (\hat{\alpha}^k)^2 (\zeta^k)''(0) + \frac{1}{6} (\hat{\alpha}^k)^3 (\zeta^k)'''(\xi^k) \\ \Rightarrow \hat{\alpha}^k &= -(\zeta^k)''(0) \frac{3 - 6\gamma}{(\zeta^k)'''(\xi^k)} \geq c_1 \gamma_\lambda^2 \frac{9 - 18\gamma}{16\gamma_{1a}}, \end{aligned}$$

where the bound follows from the use of Taylor series expansions for ζ^k around zero, and γ_{1a} denotes a bound on $(\zeta^k)'''(\xi^k)$ for $\xi^k \in [0, \hat{\alpha}^k]$.

We show that the preceding value $\hat{\alpha}$ also provides a bound for the steplength in the search on the full merit function ϕ^k . Let $\Delta^k \equiv -\sum_i \mu_i^k (\log x_i^k(\alpha) - \log x_i^k)$; from the definition of ζ^k and (101), for $\alpha \in [0, \hat{\alpha}^k]$ and $k \in \hat{\mathcal{K}}$ we have

$$\begin{aligned} \phi^k(\alpha) &= \zeta^k(\alpha) - \sum_i \mu_i^k \log x_i^k(\alpha) \leq \zeta^k(0) + \alpha (\zeta^k)'(0) + \gamma \alpha^2 (\zeta^k)''(0) \\ &\quad - \sum_i \mu_i^k \log x_i^k(\alpha) \\ &\leq \zeta^k(0) - \sum_i \mu_i^k \log x_i^k + \gamma (\alpha^k)^2 (\zeta^k)''(0) + \Delta^k + \alpha (\zeta^k)'(0) \\ &\leq \phi^k(0) + \gamma \alpha^2 (\zeta^k)''(0) + \Delta^k + \alpha (\zeta^k)'(0) \\ &\leq \phi^k(0) + \frac{3}{4} \gamma \alpha^2 (\phi^k)''(0) + \Delta^k + \alpha (\zeta^k)'(0), \end{aligned} \quad (102)$$

where the last inequality follows from (100).

We now prove that (102) implies the satisfaction of (33). Consider the term Δ^k ; from the definition of $\bar{\alpha}_i$ in Section 4 it holds that $x_i^k + (\bar{\alpha}_i^k)^2 (d_x^k)_i + \bar{\alpha}_i^k (d_n^k)_i = 0$ and $2\bar{\alpha}_i^k (d_x^k)_i + (d_n^k)_i \leq 0$, as $\bar{\alpha}_i$ is the smallest positive root of $P_i(\alpha)$ and $P_i(0) > 0$. For i

such that $\bar{\alpha}_i < \infty$, from (36), (37) and the preceding conditions we also have

$$\begin{aligned}
x_i^k(\alpha) &\geq x_i^k + (\alpha_p^k)^2 (d_x^k)_i + \alpha_p^k (d_n^k)_i \geq x_i^k + (\alpha_{\max}^k)^2 (d_x^k)_i + \alpha_{\max}^k (d_n^k)_i \\
&\geq x_i^k + (\delta^k)^2 (\bar{\alpha}_i^k)^2 (d_x^k)_i + \delta^k \bar{\alpha}_i^k (d_n^k)_i \\
&\geq (1 + (\delta^k)^2 - 2\delta^k) x_i^k + 2(\bar{\alpha}_i^k)^2 ((\delta^k)^2 - \delta^k) (d_x^k)_i + \bar{\alpha}_i^k ((\delta^k)^2 - \delta^k) (d_n^k)_i \\
&\geq (1 + (\delta^k)^2 - 2\delta^k) x_i^k - \bar{\alpha}_i^k \delta^k (1 - \delta^k) \left(2\bar{\alpha}_i^k (d_x^k)_i + (d_n^k)_i \right) \\
&\geq (1 + (\delta^k)^2 - 2\delta^k) x_i^k \\
&\geq \left(1 + (1 - \|\mu^k\|)^2 - 2(1 - \|\mu^k\|) \right) x_i^k = \|\mu^k\|^2 x_i^k.
\end{aligned}$$

As a consequence,

$$\Delta^k = - \sum_i \mu_i^k (\log x_i^k(\alpha) - \log x_i^k) \leq 2 \sum_i \mu_i^k \log \|\mu^k\| \Rightarrow \lim_{k \rightarrow \infty} \Delta^k = 0. \quad (103)$$

Consider now $(\zeta^k)'(0) = (d_n^k)^T (X^k)^{-1} \mu^k$. From (97) and the definition of G_ρ it follows that

$$\begin{aligned}
(d_n^k)^T G_\rho^k d_n^k &\leq -\frac{1}{2} c_1 \gamma_\lambda^2 \\
\Rightarrow (d_n^k)^T (X^k)^{-1} Z^k d_n^k &\leq -\frac{1}{2} c_1 \gamma_\lambda^2 - (d_n^k)^T \nabla_{xx} L(x^k, \hat{\lambda}^k - R^k c(x^k)) d_n^k \leq K,
\end{aligned}$$

where the last bound follows from Assumptions A.1, A.3 and A.4 and the boundedness of d_n^k from condition (25). These bounds, Assumption A.2 and (93) imply that, if $x_i^* > 0$ then $x_i^* = 0$ and $(d_n^k)_i \rightarrow 0$ for $k \in \hat{\mathcal{K}}$. Then, (92) yields

$$\lim_{k \in \hat{\mathcal{K}}} (d_n^k)^T (X^k)^{-1} \mu^k = \lim_{k \in \hat{\mathcal{K}}} (d_n^k)^T z^k = 0. \quad (104)$$

From (103) and (104) there exists an iteration index $\tilde{r} \geq \hat{r}$ such that for all $k \geq \tilde{r}$, $k \in \hat{\mathcal{K}}$

$$\begin{aligned}
\Delta^k &\leq \frac{1}{128} \gamma c_1^3 \gamma_\lambda^6 ((9 - 18\gamma)/(16\gamma_{1a}))^2 \leq -\frac{1}{8} \gamma (\hat{\alpha}^k/2)^2 (\phi^k)''(0) \\
(\zeta^k)'(0) &\leq \frac{1}{128} \gamma c_1^2 \gamma_\lambda^4 (9 - 18\gamma)/(16\gamma_{1a}) \leq -\frac{1}{8} \gamma (\hat{\alpha}^k/4) (\phi^k)''(0)
\end{aligned}$$

implying from (102) for all $\alpha \in [\hat{\alpha}^k/2, \hat{\alpha}^k]$,

$$\phi^k(\alpha) \leq \phi^k(0) + \frac{1}{2} \gamma (\alpha^k)^2 (\phi^k)''(0),$$

and the satisfaction of (33) for all $k \geq \tilde{r}$ and $k \in \hat{\mathcal{K}}$ for a bounded value

$$\hat{\alpha}^k/2 \geq c_1 \gamma_\lambda^2 \frac{9 - 18\gamma}{32\gamma_{1a}} > 0. \quad (105)$$

Consider now the steplength to ensure that the primal variables remain positive, α_{\max}^k . From the definition of $\bar{\alpha}_i^k$ as the smallest positive root of (35), it holds that $0 = 1 + (\bar{\alpha}_i^k)^2 ((d_x^k)_i / x_i^k) + \bar{\alpha}_i^k ((d_n^k)_i / x_i^k)$ if $\bar{\alpha}_i^k$ is finite. From (17) and $\bar{G}_\rho^k = \bar{H}^k + (X^k)^{-1} Z^k$,

$$(\bar{H}^k + (X^k)^{-1} Z^k) d_x^k = -\nabla f(x^k) + \nabla c(x^k)^T (\lambda^k + d_\lambda^k) + (X^k)^{-1} \mu^k,$$

and the boundedness by construction of \bar{H}^k , Assumptions A.1, A.2 and A.3, (92) and (93) imply that there exists a constant γ_{dx} such that $|(d_x^k)_i|/x_i^k \leq \gamma_{dx}$ for all k and all i . Regarding the direction of negative curvature d_n^k , condition (28) implies $|(d_n^k)_i|/x_i^k \leq \beta_{dn}$. From these bounds it holds that if $\bar{\alpha}_i^k$ is finite, either $\bar{\alpha}_i^k \geq 1$ or

$$\begin{aligned} 0 &= 1 + (\bar{\alpha}_i^k)^2((d_x^k)_i/x_i^k) + \bar{\alpha}_i^k((d_n^k)_i/x_i^k) \geq 1 - (\bar{\alpha}_i^k)^2\gamma_{dx} - \bar{\alpha}_i^k\beta_{dn} \\ &\geq 1 - \bar{\alpha}_i^k\gamma_{dx} - \bar{\alpha}_i^k\beta_{dn} \Rightarrow \bar{\alpha}_i^k \geq \frac{1}{\gamma_{dx} + \beta_{dn}}, \end{aligned}$$

and as a consequence of (36), $\alpha_{\max}^k \geq 1/(\gamma_{dx} + \beta_{dn})$ and from this bound, (78), (80) and (105), α_p^k is bounded away from zero along the sequence \hat{K} .

But if α_p^k is bounded, (98) cannot hold, and the algorithm must converge to second-order KKT points of problem (1) along the sequence \hat{K} .

□

Acknowledgements. We wish to thank A. Forsgren, M.H. Wright and two anonymous referees for their helpful comments and suggestions.

References

1. Auslender, A.: Penalty methods for computing points that satisfy second-order necessary conditions. *Math. Program.* **17**(2), 229–238 (1979)
2. Bannert, T.: A trust-region algorithm for nonsmooth optimization. *Math. Program.* **67**, 247–264 (1994)
3. Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York, 1982
4. Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, Ph.L.: CUTE: Constrained and unconstrained testing environment. *Trans. ACM Math. Software* **21**(1), 123–160 (1995)
5. Bonnans, J.F., Launay, G.: Sequential quadratic programming with penalization of the displacement. *SIAM J. Optim.* **5**(4), 782–812 (1995)
6. Bunch, J.R., Parlett, B.N.: Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numer. Anal.* **8**, 639–655 (1971)
7. Bunch, J.R., Kaufman, L., Parlett, B.N.: Decomposition of a symmetric matrix. *Numer. Math.* **27**, 95–109 (1976)
8. Byrd, R.H., Schanbel, R.B., Schultz, G.A.: Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Math. Program.* **40**, 247–263 (1988)
9. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: *Trust-region Methods*. SIAM, Philadelphia, 2000
10. El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., Zhang, Y.: On the formulation and theory of the Newton interior-point method for nonlinear programming. *J. Optim. Theory Appl.* **89**, 507–541 (1996)
11. Eldersveld, S.K.: *Large-scale Sequential Quadratic Programming Algorithms*. Technical Report SOL 92-4. Stanford University, 1992
12. Fiacco, A.V., McCormick, G.P.: *Nonlinear programming: Sequential unconstrained minimization techniques*. Society for Industrial and Applied Mathematics, Philadelphia, 1990 (Originally published by Research Analysis Corporation, McLean, Virginia)
13. Forsgren, A., Murray, W.: Newton methods for large-scale linear equality-constrained minimization. *SIAM J. Matrix Anal. Appl.* **14**, 560–587 (1993)
14. Gajulapalli, R.S.: *INTOPT: An interior point algorithm for large scale nonlinear optimization*. Ph. D. Thesis. The University of Texas at Austin, 1995
15. Gay, D.M.: A trust-region approach to linearly constrained optimization. In: D.F. Griffiths (ed.), *Numerical Analysis Proceedings, Dundee 1983*, Lecture Notes in Mathematics, 1066, pp. 72–105, Springer-Verlag, Heidelberg, 1984
16. Gay, D.M., Overton, M.L., Wright, M.H.: A primal-dual interior method for nonconvex nonlinear programming. In: Y. Yuan (ed.), *Advances in Nonlinear Programming*, pp. 31–56. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998

17. Gill, P.E., Murray, W., Saunders, M.A., Wright, M.H.: *User's guide for NPSOL (version 4.0): A FORTRAN package for nonlinear programming*. Technical Report SOL 86-2. Stanford University, 1986
18. Gill, P.E., Murray, W., Wright, M.H.: *Practical Optimization*. Academic Press, London/New York, 1981
19. Gould, N.I.M.: On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem. *Math. Program.* **32**, 90–99 (1985)
20. Gould, N.I.M., Toint, Ph.L.: A note on the second-order convergence of optimization algorithms using barrier functions. *Math. Program.* **85**, 433–438 (1999)
21. Gould, N.I.M., Orban, D., Sartenaer, A., Toint, Ph.L.: Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. *SIAM J. Optim.* **11**(4), 974–1002 (2001)
22. Hock, W., Schittkowski, K.: *Test Examples for Nonlinear Programming Codes*. Springer Verlag, Berlin, 1981
23. McCormick, G.: A modification of Armijo's step-size rule for negative curvature. *Math. Program.* **13**, 111–115 (1977)
24. Mehrotra, S.: On the implementation of a primal-dual interior point method. *SIAM J. Optim.* **2**, 575–601 (1992)
25. Moré, J.J., Sorensen, D.C.: On the use of directions of negative curvature in a modified Newton method. *Math. Program.* **16**, 1–20 (1979)
26. Murray, W., Prieto, F.J.: A Sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM J. Optim.* **5**, 590–640 (1995)
27. Stewart, G.W., Sun, J.: *Matrix Perturbation Theory*. Academic Press, Boston, 1990
28. Vanderbei, R.J., Shanno, D.F.: An interior-point algorithm for nonconvex nonlinear programming. *Comput. Optim. Appl.* **13**, 231–252 (1999)
29. Wright, M.H.: Ill-conditioning and computational error in primal-dual methods for nonlinear programming. *SIAM J. Optim.* **9**, 84–111 (1998)
30. Wright, M.H.: Some properties of the Hessian of the logarithmic barrier function. *Math. Program.* **67**, 265–295 (1994)
31. Wright, S.J.: *Effects of finite-precision arithmetic on interior-point methods for nonlinear programming*. Preprint ANL/MCS-P705-0198. Mathematics and Computer Science Division, Argonne National Laboratory, 1998
32. Yamashita, H.: A globally convergent primal-dual interior point method for constrained optimization. *Optim. Meth. Software* **10**, 443–469 (1998)
33. Yamashita, H., Yabe, H.: Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization. *Math. Program.* **75**, 377–397 (1996)