A Projected-Search Interior-Point Method for Nonlinearly Constrained Optimization

Philip E. Gill and Minxin Zhang*

Department of Mathematics, University of California San Diego,
La Jolla, 92093-0112, CA, USA.

*Corresponding author(s). E-mail(s): miz151@ucsd.edu;
Contributing author(s): pgill@ucsd.edu;

Abstract

This paper concerns the formulation and analysis of a new interior-point method for constrained optimization that combines a shifted primal-dual interior-point method with a projected-search method for bound-constrained optimization. The method involves the computation of an approximate Newton direction for a primal-dual penalty-barrier function that incorporates shifts on both the primal and dual variables. Shifts on the dual variables allow the method to be safely "warm started" from a good approximate solution and avoids the possibility of very large solutions of the associated path-following equations. The approximate Newton direction is used in conjunction with a new projected-search line-search algorithm that employs a flexible non-monotone quasi-Armijo line search for the minimization of each penalty-barrier function. Numerical results are presented for a large set of constrained optimization problems. For comparison purposes, results are also given for two primal-dual interior-point methods that do not use projection. The first is a method that shifts both the primal and dual variables. The second is a method that involves shifts on the primal variables only. The results show that the use of both primal and dual shifts in conjunction with projection gives a method that is more robust and requires significantly fewer iterations. In particular, the number of times that the search direction must be computed is substantially reduced. Results from a set of quadratic programming test problems indicate that the method is particularly well-suited to solving the quadratic programming subproblem in a sequential quadratic programming method for nonlinear optimization.

Keywords: Nonlinearly constrained optimization, interior-point methods, primal-dual methods, shifted penalty and barrier methods, projected-search methods, Armijo line search, augmented Lagrangian methods.

1 Introduction

This paper concerns the formulation and analysis of a new primal-dual interior-point method for the solution of the nonlinear optimization problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \quad \begin{pmatrix} \ell^x \\ \ell^s \end{pmatrix} \le \begin{pmatrix} x \\ c(x) \end{pmatrix} \le \begin{pmatrix} u^x \\ u^s \end{pmatrix}, \tag{NIP}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$, $f : \mathbb{R}^n \mapsto \mathbb{R}$, and (ℓ^x, ℓ^s) and (u^x, u^s) are constant vectors of lower and upper bounds. Problem (NIP) may be reformulated as

$$\underset{x\in\mathbb{R}^n,\, s\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) - s = 0, \;\; \begin{pmatrix} \ell^x \\ \ell^s \end{pmatrix} \le \begin{pmatrix} x \\ s \end{pmatrix} \le \begin{pmatrix} u^x \\ u^s \end{pmatrix}, \tag{NPs}$$

where x and the "slack variables" s are treated as independent variables. For simplicity, in our discussion of the theoretical aspects of the method we consider the problem

$$\underset{x\in\mathbb{R}^n,\, s\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) - s = 0, \quad s \ge 0, \tag{NIPs}$$

which is equivalent to minimizing $f(x)$ subject to the inequality constraints $c(x) \ge 0$. The methods designed to solve (NIPs) may be easily applied to solve the more general problem (NPs) by treating the bound constraints on x in the same way as treating the bounds on s.

The proposed method is based on combining a new primal-dual interior-point method with a projected-search method for bound-constrained optimization that uses a flexible non-monotone *quasi-Armijo* line search. Unlike conventional interior-point methods, which impose an upper bound on the step size to prevent the variables from becoming infeasible, the proposed projected-search interior-point method projects the underlying search direction onto a superset of the feasible region defined by perturbing the constraint bounds. With this approach the direction of the search path may change multiple times along the boundary of the perturbed feasible region at the cost of computing a single direction. Projected-search interior-point methods have the potential of requiring fewer iterations than a conventional interior-point method, thereby reducing the number of times that a search direction must be computed. The direction for the projected search is an approximate Newton direction associated with minimizing a shifted primal-dual penalty-barrier function. This function involves a primal-dual shifted penalty term for the equality constraints $c(x) - s = 0$ and an analogous primal-dual shifted barrier term for enforcing the nonnegativity constraints on the variables s and their associated multipliers. For problems with a mixture of upper and lower bounds on x and s, the method may be regarded as shifting both the primal and dual variables, see Gill and Zhang [1]. This extends the shifted primal-dual penalty-barrier function of Gill, Kungurtsev and Robinson [2], which only involves shifts on the primal variables. It is shown that a specific approximate Newton method for the unconstrained minimization of the penalty-barrier function generates directions that are identical to those associated with a variant of the conventional path-following method. In this context the penalty-barrier function is used as a merit function for assessing points generated by Newton's method for a zero of the path-following equations.

The projected-search method is specifically designed for the all-shifted penalty-barrier function and generates a sequence of feasible iterates $\{v_k\}_{k=0}^{\infty}$ such that $v_{k+1} = \mathbf{proj}_{\Omega_k}(v_k + \alpha_k \Delta v_k)$, where $\mathbf{proj}_{\Omega_k}(v)$ is the projection of the vector v of primal-dual variables onto a set Ω_k that is a perturbation of the feasible region. The perturbation is chosen to ensure that every optimal solution of (NIPs) lies in Ω_k (see equation (15)). Under mild assumptions, it is shown that there exists a limit point of the computed iterates that is either an infeasible stationary point, or a complementary approximate Karush-Kuhn-Tucker (CAKKT) point, i.e., it satisfies reasonable stopping criteria and is a Karush-Kuhn-Tucker (KKT) point under a complementary approximate KKT regularity condition (see Andreani, Martínez and Svaiter [3]).

Constraint shifts provide a number of important benefits. First, analogous to the definition of the original shifted penalty method of Powell [4] (equivalent to the augmented Lagrangian method) the penalty-barrier terms need not go to infinity (see, e.g., Powell [4], Hestenes [5]). In addition, if the optimal dual variables are known, then the problem may be solved in a single unconstrained minimization. Second, it is not necessary for the initial values of the variables to lie in the strict interior of the feasible region, i.e., the initial point can lie on the boundary. Similarly, dual shifts allow a dual variable to be initialized with any non-negative value. This implies that if the method is started at a primal-dual solution, the method will terminate immediately with the optimal point. Finally, shifts introduce a regularization term in the linear equations that are solved at each iteration. This mitigates the ill-conditioning of the associated linear equations that may occur when strict complementarity does not hold or the active constraints are not linearly independent at a solution (see Section 3, equation (13)).

The focus of this paper is on the formulation and analysis of an interior-point method for the solution of problems with a nonlinear objective function and nonlinear constraints. However, a significant benefit of the proposed method is that it can be used as part of an efficient second-derivative sequential quadratic programming (SQP) method. In general, interior-point methods and sequential quadratic programming methods are two alternative approaches to handling inequality constraints. Both interior methods and SQP methods have an inner/outer iteration structure, with the work for an inner iteration being dominated by the cost of solving a large sparse system of symmetric indefinite linear equations. In the case of SQP methods, these equations involve a subset of the variables and constraints and are related to the equations that were solved in the preceding iteration. This implies that matrix factorization methods can be used to update the QP solution as the inner iterations proceed. In the case of interior-point methods, the equations involve all the constraints and variables, and the equations must be solved from scratch at each inner iteration. Broadly speaking, the advantages and disadvantages of SQP methods and interior methods complement each other. Interior-point methods are most efficient when implemented with exact second derivatives (see Gill, Saunders and Wong [6]). Moreover, they can converge in few inner iterations—even for very large problems. As the dimension and zero/nonzero structure of the Newton equations remains *fixed*, these Newton equations may be solved efficiently using either iterative or direct methods available in the form of advanced "off-the-shelf" linear algebra software. On the negative side, although interior methods are very effective for solving "one-off" problems, they are more difficult to adapt to solving a sequence of related problems. In contrast, SQP methods have the potential of being able to capitalize on a good initial starting point, but are difficult to implement when exact second derivatives are available, and require customized matrix updating techniques. Over the years, algorithm developers have avoided the difficulty of using second derivatives by solving a QP subproblem defined with a positive semidefinite quasi-Newton approximate Hessian (see, e.g., Gill, Murray and Saunders [7]). Many of the difficulties associated with using second derivatives in an SQP method would be resolved if an interior-point method could be used to solve the QP subproblem. However, QP solvers based on conventional interior methods have had limited success within SQP methods because they are difficult to "warm start" from a near-optimal point. This makes it difficult to capitalize on the property that, as the outer iterates converge, the solution of one QP subproblem is a very good estimate of the solution of the next. In addition, the need to solve many QP subproblems using a method that must solve equations involving all of the constraints from scratch can be prohibitively expensive.

The interior-point method proposed in this paper is particularly well-suited to solving the quadratic programming subproblem in an SQP method. The shifts on the primal and dual variables allow the method to be safely "warm started" from a good approximate solution. In addition, the numerical results of Section 6 show that the method requires significantly fewer iterations than an unprojected interior-point method when applied to large set of quadratic programming problems.

The paper is organized in six sections. In Section 2 we review the method of Gill, Kungurtsev and Robinson. Section 3 concerns the extension of this method to include shifts on the dual variables as well as the variables s. In Section 4 a projected-search algorithm is proposed for minimizing the all-shifted primal-dual penalty-barrier function for fixed penalty and barrier parameters. The convergence of this algorithm is established under certain assumptions. Section 5 presents an algorithm for solving problem (NIPs) that builds upon the work from Section 4. Global convergence results are also established. Some numerical results are presented in Section 6.

1.1 Notation and terminology

Given vectors x and y, the vector consisting of x augmented by y is denoted by (x, y). The subscript i is appended to vectors to denote the ith component of that vector, whereas the subscript k is appended to a vector to denote its value during the kth iteration of an algorithm, e.g., x_k represents the value for x during the kth iteration, whereas $[\,x_k\,]_i$ denotes the ith component of the vector x_k. Given vectors a and b with the same dimension, the vector with ith component $a_i b_i$ is denoted by $a \cdot b$. Similarly, $\min(a, b)$ denotes a vector with components $\min(a_i, b_i)$. The vector e denotes the column vector of ones, and I denotes the identity matrix. The dimensions of e and I are defined by the context. The vector two-norm or its induced matrix norm are denoted by $\|\cdot\|$. The inertia of a real symmetric matrix A, denoted by $\mathrm{In}(A)$, is the integer triple (a_+, a_-, a_0) giving the number of positive, negative and zero eigenvalues of A. The n-vector $\nabla f(x)$ denotes gradient of $f(x)$, and the $m \times n$ matrix $J(x)$ denotes the constraint Jacobian, which has ith row $\nabla c_i(x)^{\mathrm{T}}$. The Hessian with respect to x of the Lagrangian function associated with problem (NIPs) is denoted by $H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x)$, where y is the m-vector of dual variables associated with the constraints $c(x) - s = 0$.

2 Background

Given an appropriate constraint qualification, the first-order optimality conditions for problem (NIPs) are given by

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y &= 0, & y - w &= 0, \\ c(x) - s &= 0, & s &\geq 0, \\ s \cdot w &= 0, & w &\geq 0, \end{aligned}\right\} \tag{1}$$

where the vectors y and w constitute the Lagrange multipliers for the equality constraints $c(x) - s = 0$ and nonnegativity constraints $s \geq 0$ respectively. Following standard practice, any point satisfying the conditions (1) will be referred to as a first-order KKT point.

Primal-dual path-following methods generate a sequence of iterates that approximate a continuous primal-dual path that passes through a solution of (NIPs). Points on this path satisfy a system of nonlinear equations that represent the deviations from a perturbation of the first-order optimality conditions (1). In a conventional path-following approach, the perturbed optimality conditions correspond to replacing the equality constraints and complementarity conditions of (1) by $c(x) - s = \mu y$ and $s \cdot w = \mu e$, where μ is a small positive parameter such that $\mu \to 0$. This method is closely related to penalty-barrier methods for solving (NIPs). Penalty and barrier methods involve the minimization of a sequence of unconstrained functions parameterized by a sequence of penalty-barrier parameters $\{\,\mu_k\,\}$ such that $\mu_k \to 0$ (see, e.g., Fiacco and McCormick [8], Frisch [9] and Fiacco [10]). Under certain conditions on f and c the continuous trajectory of penalty-barrier minimizers associated with a continuous penalty-barrier parameter μ coincides with the primal-dual path.

In the neighborhood of a first-order KKT point, computing the search direction as the solution of the Newton equations for a zero of the perturbed optimality conditions provides the favorable local convergence rate associated with Newton's method. Given the close connection with penalty-barrier methods, solving for a zero of the perturbed optimality conditions provides an alternative to solving the ill-conditioned equations associated with a conventional barrier method. In this context, the penalty-barrier function may be regarded as a merit function for forcing convergence of the sequence of Newton iterates of the path-following method. For examples of this approach, see Byrd, Hribar and Nocedal [11], Wächter and Biegler [12], Forsgren and Gill [13], and Gertz and Gill [14].

In a conventional path-following interior-point method, it is necessary to force $\mu \to 0$ to ensure that points near the path eventually satisfy the optimality conditions (1). However, if an augmented Lagrangian method defined with multiplier estimate y^E and penalty parameter μ^P is used to minimize $f(x)$ subject to $c(x) = 0$, then perturbed conditions of the form $c(x) = \mu^P(y^E - y)$ hold at a minimizer. It follows that μ^P need not go to zero if y^E is chosen to converge to the optimal multipliers. Based on this observation, the method of Gill, Kungurtsev and Robinson [2] is based on the perturbed optimality conditions

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y = 0, \qquad & y - w = 0, \\ c(x) - s = \mu^P(y^E - y), \qquad & s \ge 0, \\ s \cdot w = \mu^B(w^E - w), \qquad & w \ge 0, \end{aligned}\right\} \tag{2}$$

where μ^P and μ^B are positive scalars and y^E and w^E denote estimates of the Lagrange multipliers for the constraints $c(x) - s = 0$ and $s \ge 0$, respectively. The perturbed complementarity condition in (2) may be written in the form $(s + \mu^B e) \cdot w = \mu^B w^E$, which implies that if $w^E > 0$ then $s + \mu^B e > 0$ and $w > 0$. Gill, Kungurtsev and Robinson show that an appropriate merit function for a path-following interior-point method based on the conditions (2) is the shifted primal-dual penalty-barrier function

$$\begin{aligned} M(x, s, y, w\,; y^E, w^E, \mu^P, \mu^B) &= f(x) - (c(x) - s)^{\mathrm{T}} y^E \\ &\quad + \frac{1}{2\mu^P}\|c(x) - s\|^2 + \frac{1}{2\mu^P}\|c(x) - s + \mu^P(y - y^E)\|^2 \\ &\quad - \sum_{i=1}^m \mu^B w_i^E \ln\big(s_i + \mu^B\big) - \sum_{i=1}^m \mu^B w_i^E \ln\big(w_i(s_i + \mu^B)\big) + \sum_{i=1}^m w_i(s_i + \mu^B). \end{aligned}$$

In the neighborhood of a minimizer of (NIPs) satisfying certain second-order optimality conditions, the Newton equations for a zero of the conditions (2) are equivalent to the Newton equations for a minimizer of M. Under certain assumptions, a limit point of the iterates generated by the algorithm may always be found that is either an infeasible stationary point or a complementary approximate KKT point (see Andreani, Martínez and Svaiter [3]). The reader is referred to Gill, Kungurtsev and Robinson [2] for more details. This reference provides some numerical examples that illustrate the performance of the method compared to the widely-used interior-point method `IPOPT`.

In the following section, the Gill-Kungurtsev-Robinson algorithm is extended to include shifts on the dual variables w in addition to the shifts on the slack variables s.

3 An All-Shifted Primal-Dual Penalty-Barrier Function

In order to use shifts for the dual variables, we consider the perturbed optimality conditions

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y = 0, &\qquad & y - w = 0, \\ c(x) - s = \mu^{P}(y^{E} - y), &\qquad & s \geq 0, \\ s \cdot w = \mu^{B}(w^{E} - w) + \mu^{B}(s^{E} - s), &\qquad & w \geq 0, \end{aligned}\right\} \tag{3}$$

where $y^{E} \in \mathbb{R}^{m}$ is an estimate of a Lagrange multiplier vector for the constraints $c(x) - s = 0$, $w^{E} \in \mathbb{R}^{m}$ is an estimate of a Lagrange multiplier for the constraints $s \geq 0$, $s^{E} \in \mathbb{R}^{m}$ is an estimate of the optimal slacks, and μ^{P} and μ^{B} are positive scalars. The last equation of (3) may be written in the form $(s + \mu^{B} e) \cdot (w + \mu^{B} e) = \mu^{B}(s^{E} + w^{E} + \mu^{B} e)$, which implies that if $s^{E} + w^{E} + \mu^{B} e > 0$ then $s + \mu^{B} e > 0$ and $w + \mu^{B} e > 0$. If $F(x, s, y, w; s^{E}, y^{E}, w^{E}, \mu^{P}, \mu^{B})$ denotes the vector-valued function

$$F(x, s, y, w\,; s^{E}, y^{E}, w^{E}, \mu^{P}, \mu^{B}) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ c(x) - s + \mu^{P}(y - y^{E}) \\ s \cdot w - \mu^{B}(w^{E} - w + s^{E} - s) \end{pmatrix}, \tag{4}$$

then any point (x, s, y, w) that satisfies the perturbed optimality conditions (3) must satisfy $F(x, s, y, w\,; s^{E}, y^{E}, w^{E}, \mu^{P}, \mu^{B}) = 0$. Let $F(v)$ denote the function at a given point $v = (x, s, y, w)$. The Newton equations for the change in variables Δv are given by $F'(v)\Delta v = -F(v)$, i.e.,

$$\begin{pmatrix} H(x,y) & 0 & -J(x)^{\mathrm{T}} & 0 \\ 0 & 0 & I_m & -I_m \\ J(x) & -I_m & \mu^{P} I_m & 0 \\ 0 & W + \mu^{B} I_m & 0 & S + \mu^{B} I_m \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ c(x) - s + \mu^{P}(y - y^{E}) \\ s \cdot w - \mu^{B}(w^{E} - w + s^{E} - s) \end{pmatrix}, \tag{5}$$

where S and W denote diagonal matrices with diagonal entries s_i and w_i such that $s_i + \mu^{B} > 0$ and $w_i + \mu^{B} > 0$.

The next step is to formulate a penalty-barrier function M such that in a neighborhood of a minimizer of M, the Newton equations for minimizing M approximate the Newton equations (5). Consider the shifted primal-dual penalty-barrier function

$$
\left.
\begin{aligned}
M(x,s,y,w\,;s^E,y^E,w^E,\mu^P,\mu^B) = {} & \underbrace{f(x)}_{(A)} - \underbrace{(c(x)-s)^{\mathrm{T}}y^E}_{(B)} + \underbrace{\frac{1}{2\mu^P}\|c(x)-s\|^2}_{(C)} \\
& + \underbrace{\frac{1}{2\mu^P}\|c(x)-s+\mu^P(y-y^E)\|^2}_{(D)} \\
& \underbrace{- 2\sum_{i=1}^m \mu^B(w_i^E+s_i^E+\mu^B)\ln(s_i+\mu^B)}_{(E)} \\
& \underbrace{- \sum_{i=1}^m \mu^B(w_i^E+s_i^E+\mu^B)\ln(w_i+\mu^B)}_{(F)} \\
& + \underbrace{\sum_{i=1}^m w_i(s_i+\mu^B)}_{(G)} + \underbrace{2\mu^B\sum_{i=1}^m s_i}_{(H)}\,.
\end{aligned}
\right\} \qquad (6)
$$

Let S^E denote the diagonal matrix with diagonal entries s_i^E and define

$$
S_B = S + \mu^B I_m, \quad S_B^E = S^E + \mu^B I_m \;\text{ and }\; W_B = W + \mu^B I_m.
$$

Given the positive-definite matrices

$$
D_P = \mu^P I_m \quad \text{and} \quad D_B = S_B W_B^{-1},
$$

and auxiliary vectors

$$
\pi^Y(x) = y^E - \frac{1}{\mu^P}(c(x)-s) \quad \text{and} \quad \pi^W(s) = \mu^B(S+\mu^B I)^{-1}(w^E - s + s^E),
$$

the gradient of M may be written as

$$
\nabla M = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}}\big(\pi^Y + (\pi^Y - y)\big) \\ (\pi^Y - y) + (\pi^Y - \pi^W) + (w - \pi^W) \\ -D_P(\pi^Y - y) \\ -D_B(\pi^W - w) \end{pmatrix}, \qquad (7)
$$

and the Hessian $\nabla^2 M$ may be written in the form

$$\begin{pmatrix} H + 2J(x)^T D_P^{-1} J(x) & -2J(x)^T D_P^{-1} & J(x)^T & 0 \\ -2D_P^{-1} J(x) & 2(D_P^{-1} + D_B^{-1} W_B^{-1} \Pi^W + \mu^B S_B^{-1}) & -I_m & I_m \\ J(x) & -I_m & D_P & 0 \\ 0 & I_m & 0 & D_B W_B^{-1} \Pi^W + \mu^B W_B^{-2} S_B \end{pmatrix}, \tag{8}$$

where $H = H\big(x, \pi^Y + (\pi^Y - y)\big)$ and $\Pi^W = \operatorname{diag}(\pi^W)$.

Given the kth primal-dual iterate $v_k = (x_k,\ s_k,\ y_k,\ w_k)$, the search direction $\Delta v_k = (\Delta x_k,\ \Delta s_k,\ \Delta y_k,\ \Delta w_k)$ is computed by solving the linear equations

$$H_k^M \Delta v_k = -\nabla M(v_k), \tag{9}$$

where H_k^M is a positive-definite approximation of $\nabla^2 M(v_k)$. For the remainder of this section we focus on the computation of the search direction for a single iteration and omit the subscript k. The matrix H^M in the equations $H^M \Delta v = -\nabla M(v)$ is defined by substituting y for π^Y, w for π^W, s for s^E and a symmetric matrix $\widehat{H}$ for H in (8). This gives

$$H^M = \begin{pmatrix} \widehat{H} + 2J(x)^T D_P^{-1} J(x) & -2J(x)^T D_P^{-1} & J(x)^T & 0 \\ -2D_P^{-1} J(x) & 2(D_P^{-1} + D_B^{-1}) & -I_m & I_m \\ J(x) & -I_m & D_P & 0 \\ 0 & I_m & 0 & D_B \end{pmatrix}, \tag{10}$$

where $\widehat{H}$ is chosen such that $\widehat{H} \approx H(x,y)$ and H^M is positive definite. A generalization of Theorem 5.1 of Gill, Kungurtsev and Robinson [2] may be used to show that the choice $\widehat{H} = H(x,y)$ is allowed in the neighborhood of a solution satisfying certain second-order optimality conditions.

The distinctive property of the approximate Newton equations (9) is that under certain conditions on H, their solution is also a solution of the perturbed path-following equations (5). Consider the upper-triangular matrix

$$U = \begin{pmatrix} I_m & 0 & -2J(x)^T D_P^{-1} & 0 \\ 0 & I_m & 2D_P^{-1} & -2D_B^{-1} \\ 0 & 0 & I_m & 0 \\ 0 & 0 & 0 & W + \mu^B I_m \end{pmatrix}. \tag{11}$$

The matrix U is nonsingular and it follows that the solution Δv of (9) must satisfy

$$U H^M \Delta v = -U \nabla M(v). \tag{12}$$

Upon multiplication by U and the application of the identity $W_B D_B = S_B$, the equations (12) may be rewritten as

$$\begin{pmatrix} \widehat{H} & 0 & -J(x)^T & 0 \\ 0 & 0 & I_m & -I_m \\ J(x) & -I_m & D_P & 0 \\ 0 & W+\mu^B I_m & 0 & S+\mu^B I_m \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^T y \\ y - w \\ c(x) - s + \mu^P (y - y^E) \\ s \cdot w - \mu^B (w^E - w + s^E - s) \end{pmatrix}. \tag{13}$$

These equations are identical to the shifted path-following equations (5) when $\widehat{H} = H(x,y)$. The solution of (13) is given by

$$\Delta w = y - w + \Delta y \quad \text{and} \quad \Delta s = -D_B(y + \Delta y) + \mu^B W_B^{-1}(w^E + s^E - s),$$

where Δx and Δy satisfy the equations

$$\begin{pmatrix} \widehat{H} & J(x)^T \\ J(x) & -(D_P + D_B) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^T y \\ D_P(y - \pi^Y) + D_B(y - \pi^W) \end{pmatrix}. \tag{14}$$

The matrix H^M in (10) is positive definite if $\widehat{H} + J(x)^T(D_P + D_B)^{-1}J(x)$ is positive definite or, equivalently, if the $(n+m)\times(n+m)$ matrix associated with (14) has inertia $(n, m, 0)$. If this condition does not hold for $\widehat{H} = H(x,y)$, a common choice of $\widehat{H}$ is the matrix $H(x,y) + \delta I_n$ for some positive scalar δ (see Section 6.1).

It should be noted that in the neighborhood of a solution, both the approximate Newton equations $H_k^M \Delta v_k = -\nabla M(v_k)$ and the KKT equations (14) are ill-conditioned for small values of μ^P and μ^B. However, the sensitivity of the solution of (14) is independent of the magnitudes of μ^P and μ^B (see Forsgren, Gill and Shinnerl [15], Ponceleón [16] and Wright [17, 18]).

4 Minimizing the Merit Function using Projected Search

In this section, we propose a projected-search algorithm that utilizes a *non-monotone flexible quasi-Armijo* line search for minimizing the merit function $M(x,\, s,\, y,\, w\,;s^E,\, y^E,\, w^E,\, \mu^P,\, \mu^B)$ of (6) with fixed parameters s^E, y^E, w^E, μ^P and μ^B. The flexible quasi-Armijo line search is a generalization of the quasi-Armijo search (see Ferry et al [19] and Zhang [20]) that allows the acceptance of a step under a wider range of conditions. The generalization uses the idea of a flexible line search proposed by Curtis and Nocedal [21], and also exploits the connection between minimizing the merit function and finding a zero of the shifted path-following function $F(x,\, s,\, y,\, w\,;s^E,\, y^E,\, w^E,\, \mu^P,\, \mu^B)$ of (4). In our description, we simplify the notation by writing $M(x,s,y,w\,;s^E,y^E,w^E,\mu^P,\mu^B)$ and $F(x,s,y,w\,;s^E,y^E,w^E,\mu^P,\mu^B)$ as $M(v\,;\mu^P)$ and $F(v\,;\mu^P)$, respectively.

4.1 The algorithm

For the merit function $M(v\,;\mu^P)$ to be well-defined, the variables must satisfy the implicit bounds $s > -\mu^B e$, and $w > -\mu^B e$. Thus, minimizing the merit function $M(v\,;\mu^P)$ is equivalent to solving the bound-constrained problem

$$\underset{v}{\text{minimize}}\;\; M(v\,;\mu^P) \quad \text{subject to} \quad v > \ell, \tag{IPBC}$$

with $\ell = \big(-\infty, -\mu^B e, -\infty, -\mu^B e\big)$, where an entry of "$-\infty$" is used to indicate that the associated variable has no lower bound. Let $\mathbf{proj}_{\Omega_k}(v)$ be the projection of v onto the perturbed feasible region

$$\Omega_k = \big\{\, v : v \ge \min\big\{\, v_k - \sigma(v_k - \ell), 0 \,\big\}\,\big\}, \tag{15}$$

with σ a fixed positive scalar such that $0 < \sigma < 1$. The quantity σ may be interpreted as the "fraction to the boundary" parameter used in many conventional interior-point methods. The proposed projected-search method for problem (IPBC) is given in Algorithm 1. It generates a sequence of feasible iterates $\{v_k\}_{k=0}^{\infty}$ such that $v_{k+1} = \mathbf{proj}_{\Omega_k}(v_k + \alpha_k \Delta v_k)$, where Δv_k is the search direction computed as in Section 3, and α_k is a step computed using a flexible quasi-Armijo search.

To perform the flexible quasi-Armijo search, we choose a line-search Armijo parameter μ^L such that $\mu^L \ge \mu^P$. At an iteration k, let $\psi_k(\alpha\,;\mu)$ and $\phi_k(\alpha\,;\mu)$ denote the functions $M\big(\mathbf{proj}_{\Omega_k}(v_k + \alpha\Delta v_k)\,;\mu\big)$ and $\big\|F\big(\mathbf{proj}_{\Omega_k}(v_k + \alpha\Delta v_k)\,;\mu\big)\big\|$. A step α_k is acceptable if all of the three conditions

$$\psi_k(\alpha_k\,;\mu^P) < \max\big\{\, \psi_k(0\,;\mu^P), M_{\max} \,\big\}, \tag{16a}$$
$$\psi_k(\alpha_k\,;\mu^L) < \max\big\{\, \psi_k(0\,;\mu^L), M_{\max} \,\big\}, \quad \text{and} \tag{16b}$$
$$\phi_k(\alpha_k\,;\mu^P) \le \eta_F \min\big\{\, \phi_k(0\,;\mu^P), \eta_F^{m_k} F_{\max} \,\big\} \tag{16c}$$

are satisfied, or

$$\psi_k(\alpha_k\,;\mu_k^F) \le \psi_k(0\,;\mu_k^F) + \alpha_k \eta_A \nabla M(v_k\,;\mu^P)^{\mathrm{T}} \Delta v_k, \tag{16d}$$

for some value $\mu_k^F \in [\mu^P, \mu^L]$ and some positive $\eta_F < 1$. In these conditions, $M_{\max}$ and $F_{\max}$ are large preassigned parameters and m_k is the number of iterations prior to iteration k at which (16a)–(16c) were satisfied. The use of the sufficient decrease parameter of the form μ_k^F is characteristic of a flexible line search (see Curtis and Nocedal [21]). In practice the step may be found by reducing α_k by a constant factor until (16a)–(16c) holds, or (16d) is satisfied with either $\mu_k^F = \mu^L$ or $\mu_k^F = \mu^P$. The approximate Newton direction is a descent direction for $\mu_k^F = \mu^P$, but the idea is to choose the larger value $\mu_k^F = \mu^L$ when possible because the associated penalty-barrier function is less nonlinear. It is shown in Lemma 5.2 that $\mu_k^F = \mu^L$ for all k sufficiently large. Any α_k satisfying the conditions (16a)–(16c) or the condition (16d) is classified as a flexible quasi-Armijo step. Alternatively, an α_k that satisfies (16d) for $\mu_k^F = \mu^P$ is simply known as a quasi-Armijo step (see Ferry et al. [19]). The conditions (16a)–(16d) allow a step to be accepted if either (16a)–(16c) holds, which implies that α_k gives a sufficient decrease in the norm of the shifted path-following function F (4), or (16d) holds, which implies that α_k satisfies a flexible variant of the quasi-Armijo condition for the minimization of M.

The convergence analysis in subsection 4.2 below establishes the convergence of Algorithm 1 under typical assumptions. However, the ultimate purpose is to develop a practical algorithm for the solution of problem (NIPs) that uses Algorithm 1 as a basis for minimizing the underlying merit function. The slack-variable reset in Step 18 of Algorithm 1 plays a crucial role in this more general algorithm for handling (locally) infeasible problems (see Lemma 5.5). Analogous slack-variable resets are used in Gill, Murray and Saunders [22], and Gill, Kungurtsev and Robinson [2]. As defined in Step 17 of Algorithm 1, $\widehat{s}_{k+1}$ is the unique minimizer, with respect to s, of the sum of the terms (B), (C), (D), (G) and (H) in the definition of the function M. In particular, it follows from Step 17 and Step 18 of Algorithm 1 that the value of s_{k+1} computed in Step 18 satisfies

$$s_{k+1} \geq \widehat{s}_{k+1} = c(x_{k+1}) - \mu_k^F\big(y^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B\big),$$

which implies, after rearrangement, that

$$c(x_{k+1}) - s_{k+1} \leq \mu_k^F\big(y^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B\big). \tag{17}$$

Algorithm 1 Minimizing M for fixed parameters s^E, y^E, w^E, μ^P, μ^B and μ^L.

1: **procedure merit-proj**$(x_0, s_0, y_0, w_0, s^E, w^E, \mu^P, \mu^B, \mu^L)$
2:     **Restrictions:** $s_0 + \mu^B e > 0$, $w_0 + \mu^B e > 0$, $s^E + w^E + \mu^B e > 0$, $\mu^L \geq \mu^P > 0$, $\mu^B > 0$;
3:     **Constants:** $\{\, \eta_A, \gamma_A, \eta_F \,\} \in (0,1)$;
4:     Set $v_0 \leftarrow (x_0, s_0, y_0, w_0)$;
5:     **while** $\|\nabla M(v_k)\| > 0$ **do**
6:         Choose $H_k^M \succ 0$, and then compute the search direction Δv_k from (9);
7:         Set $\alpha_k \leftarrow 1$;
8:         **loop**
9:             **if** (16a)–(16c) hold or (16d) holds for $\mu_k^F = \mu^L$ **then**
10:                 break;
11:             **else if** (16d) holds for $\mu_k^F = \mu^P$ **then**
12:                 break;
13:             **end if**
14:             Set $\alpha_k \leftarrow \gamma_A \alpha_k$;
15:         **end loop**
16:         Set $v_{k+1} \leftarrow \mathbf{proj}_{\Omega_k}(v_k + \alpha_k \Delta v_k)$;
17:         Set $\widehat{s}_{k+1} \leftarrow c(x_{k+1}) - \mu_k^F\big(y^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B\big)$;
18:         Perform a slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, \widehat{s}_{k+1}\}$;
19:         Set $v_{k+1} \leftarrow (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$;
20:     **end while**
21: **end procedure**

4.2 Convergence analysis

The following assumptions are made for the convergence analysis:

Assumption 4.1. *The functions f and c are twice differentiable.*

Assumption 4.2. *The sequence of matrices $\{H_k^M\}_{k\geq 0}$ used in (9) are chosen to be uniformly positive definite and bounded in norm.*

Assumption 4.3. *The sequence of iterates $\{x_k\}$ is contained in a bounded set.*

Assumption 4.3 is not a restrictive assumption. If the problem has the form (NIP), the iterates are bounded if the upper and lower bounds are finite.

It will be shown in Section 5 (proof of Lemma 5.2) that μ_k^F is fixed for all k sufficiently large if μ^L is chosen appropriately. In this section, without loss of generality, we assume that the parameter μ_k^F in Algorithm 1 is fixed at a value μ^F, with either $\mu^F = \mu^P$ or $\mu^F = \mu^L$. In order to simplify the notation, let $M(v\,;\mu^F)$ denote the function $M(x,\, s,\, y,\, w\,; s^E,\, y^E,\, w^E,\, z^E,\, \mu^F,\, \mu^B)$.

Lemma 4.1. *The sequence of iterates $\{v_k\}$ computed by* Algorithm 1 *is such that $\big\{M(v_k\,;\mu^F)\big\}$ is bounded. In particular, if α_k is a step that satisfies (16d), then $M(v_{k+1}\,;\mu^F) < M(v_k\,;\mu^F)$.*

Proof. As H_k^M is positive definite by Assumption 4.2 and $\nabla M(v_k\,;\mu^P)$ is assumed to be nonzero for all $k \geq 0$, the vector Δv_k is a descent direction for M at v_k. This property, together with equations (16a) and (16b), imply that the line search performed in Algorithm 1 produces an α_k such that the new point $v_{k+1} = \mathbf{proj}_{\Omega_k}(v_k + \alpha_k \Delta v_k)$ satisfies $M(v_{k+1}\,;\mu^F) < \max\{M(v_k\,;\mu^F), M_{\max}\}$. In particular, if (16d) holds, then $M(v_{k+1}\,;\mu^F) < M(v_k\,;\mu^F)$. It follows that the only way that the desired result cannot hold is if the slack-reset procedure of Step 18 of Algorithm 1 causes M to increase. The proof is complete if it can be shown that this cannot happen.

The vector $\widehat{s}_{k+1}$ used in the slack reset is the unique minimizer of the sum of the terms (B), (C), (D), (G) and (H) defining the function $M(v\,;\mu^F)$, so that the sum of these terms cannot increase. Also, the (A) term is independent of s, so that its value does not change. The slack-reset procedure has the effect of possibly increasing the value of some of the components of s_{k+1}, which means that the (E) and (F) terms in the definition of M can only decrease. In total, this implies that the slack reset can never increase the value of M, which completes the proof. ☐

Lemma 4.2. *The sequence of iterates $\{v_k\} = \{(x_k, s_k, y_k, w_k)\}$ computed by* Algorithm 1 *satisfies the following properties.*

- **(i)** *The sequences $\{s_k\}$, $\{c(x_k) - s_k\}$, $\{y_k\}$, and $\{w_k\}$ are bounded.*
- **(ii)** *For every i it holds that*

$$\liminf_{k\geq 0}\, [\, s_k + \mu^B e\,]_i > 0 \quad \textit{and} \quad \liminf_{k\geq 0}\, [\, w_k + \mu^B e\,]_i > 0.$$

- **(iii)** *The sequences $\big\{\pi^Y(x_k, s_k)\big\}$, $\big\{\pi^W(s_k)\big\}$, and $\big\{\nabla M(v_k\,;\mu^P)\big\}$ are bounded.*
- **(iv)** *There exists a scalar M_{low} such that $M(v_k\,;\mu^F) \geq M_{\text{low}} > -\infty$ for all k.*

Proof. First, we consider the case where (16c) holds only finitely many times. For a proof by contradiction, assume that $\{s_k\}$ is unbounded. As $s_k + \mu^B e > 0$ by construction, there exists a subsequence of iterations $\mathcal{S}$ and component i such that

$$\lim_{k\in\mathcal{S}} [\, s_k \,]_i = \infty \;\text{ and }\; [\, s_k \,]_i \geq [\, s_k \,]_j \;\text{ for every } j \text{ and all } k \in \mathcal{S}. \tag{18}$$

Next it will be shown that M must go to infinity on $\mathcal{S}$. It follows from Assumption 4.3 and the continuity of f that the term (A) in the definition of M is bounded below for all k. Similarly, Assumption 4.3, the continuity of c and (18) implies that (B) cannot go to $-\infty$ any faster than $\|s_k\|$ on $\mathcal{S}$, and that (C) converges to ∞ on $\mathcal{S}$ at the same rate as $\|s_k\|^2$. It is also clear that (D) is bounded below by zero. On the other hand, (E) goes to $-\infty$ on $\mathcal{S}$ at the rate $-\ln\big([\, s_k \,]_i + \mu^B\big)$. Next, note that (H) is bounded below. Now, if (F) is bounded below on $\mathcal{S}$, then the preceding arguments imply that M converges to infinity on $\mathcal{S}$, which contradicts boundedness of M established in Lemma 4.1. Otherwise, if (F) goes to $-\infty$ on $\mathcal{S}$, then (G) converges to ∞ faster than (F) converges to $-\infty$. Thus, M converges to ∞ on $\mathcal{S}$, which again contradicts Lemma 4.1. These arguments imply that $\{s_k\}$ is bounded, which establishes the first part of result (**i**). The second part of (**i**), i.e., the uniform boundedness of $\{c(x_k)-s_k\}$, follows from the first result, the continuity of c, and Assumption 4.3.

The next step is to establish the third bound in part (**i**), i.e., that $\{y_k\}$ is bounded. For a proof by contradiction, assume that there exists some subsequence $\mathcal{S}$ and component i such that

$$\lim_{k\in\mathcal{S}} \big|[\, y_k \,]_i\big| = \infty \quad \text{and} \quad \big|[\, y_k \,]_i\big| \geq \big|[\, y_k \,]_j\big| \;\text{ for every } j \text{ and all } k \in \mathcal{S}.$$

Using arguments similar to those of the preceding paragraph, together with the result established above that $\{s_k\}$ is bounded, it follows that (A), (B) and (C) are bounded below over all k, and that (D) converges to ∞ on $\mathcal{S}$ at the rate of $[\, y_k \,]_i^2$ because $\{s_k\}$ is bounded, as has been shown above. Using the uniform boundedness of $\{\, s_k \,\}$ and the assumption that $s^E + w^E + \mu^B > 0$, it may be deduced that (E) is bounded below. If (F) is bounded below on $\mathcal{S}$, then (G) is also bounded, and as (H) is bounded below by zero we would conclude, in totality, that $\lim_{k\in\mathcal{S}} M(v_k) = \infty$, which contradicts Lemma 4.1. Thus, (F) must converge to $-\infty$, which implies that (G) converges to ∞ faster than (F) converges to $-\infty$, so that $\lim_{k\in\mathcal{S}} M(v_k\,;\mu^F) = \infty$ on $\mathcal{S}$, which contradicts Lemma 4.1. Thus, $\{\, y_k \,\}$ is bounded.

We now establish the final bound in part (**i**), i.e., we show that $\{\, w_k \,\}$ is bounded. The boundedness of $\{\, x_k \,\}$, $\{\, s_k \,\}$ and $\{\, y_k \,\}$ imply that (A), (B), (C), (D) and (H) are bounded and that (E) is bounded below. For a proof by contradiction, assume that the set is unbounded, which implies the existence of a subsequence $\mathcal{S}$ and a component i such that

$$\lim_{k\in\mathcal{S}} [\, w_k \,]_i = \infty.$$

Then (F) converges to $-\infty$, while (G) converges to ∞ faster than (F) converges to $-\infty$, so that $\lim_{k\in\mathcal{S}_1} M(v_k\,;\mu^F) = \infty$ on $\mathcal{S}$, which contradicts Lemma 4.1. It follows that $\{\, w_k \,\}$ is bounded.

Part (**ii**) is also proved by contradiction. Suppose that $\{\, [\, s_k + \mu^B e \,]_i \,\} \to 0$ on some subsequence $\mathcal{S}$ and for some component i. As before, (A), (B), (C), (D), (G) and (H) are all bounded from below over all k. We may also use $w^E + s^E + \mu^B > 0$ and the fact that $\{ s_k \}$ and $\{ w_k \}$ were proved to be bounded in part (**i**) to conclude that (E) and (F) converge to ∞ on $\mathcal{S}$. It follows that $\lim_{k\in\mathcal{S}} M(v_k\,;\mu^F) = \infty$, which contradicts Lemma 4.1, and therefore establishes that $\liminf\,[\,s_k + \mu^B e\,]_i > 0$ for every $1 \le i \le m$. A similar argument may be used to prove that $\liminf\,[\,w_k + \mu^B e\,]_i > 0$ for every $1 \le i \le m$, which completes the proof.

Part (**iii**) and Part (**iv**) can be proved similarly as in the proof of Lemma 3.2(iii) and (iv) in [2]. □

Certain results hold when the gradient of $M(v\,;\mu^P)$ is bounded away from zero.

Lemma 4.3. *If there exists a positive scalar ϵ and a subsequence of iterates $\mathcal{S}$ satisfying*

$$\|\nabla M(v_k\,;\mu^P)\| \ge \epsilon \;\; \textit{for all} \;\; k \in \mathcal{S},$$

then the following results must hold.

(i) *The set $\{\, \|\Delta v_k\|\,\}_{k\in\mathcal{S}}$ is bounded above and bounded away from zero.*

(ii) *There exists a positive scalar δ such that $\nabla M(v_k\,;\mu^P)^{\mathrm{T}} \Delta v_k \le -\delta$ for all $k \in \mathcal{S}$.*

Proof. See the proof of Lemma 3.3 in [2]. □

Next we establish the main convergence result for Algorithm 1.

Theorem 4.1 (Flexible quasi-Armijo search)**.** *Under* Assumptions 4.1–4.3*, there exists an iteration subsequence $\mathcal{S}$ such that*

$$\lim_{k\in\mathcal{S}} \nabla M(v_k\,;\mu^P) = 0.$$

Proof. First, consider the case where there exists an infinite subsequence of iterates $\mathcal{S}$ such that the line-search conditions (16a)–(16c) hold for all $k \in \mathcal{S}$. Then the line-search condition (16c) implies that $\lim_{k\in\mathcal{S}} \|F(v_k\,;\mu^P)\| = 0$. By (12), $F(v_k\,;\mu^P) = U_k \nabla M(v_k\,;\mu^P)$, where U_k is a matrix of the form (11). Lemma 4.2(**ii**) implies that $\{\,\|U_k\|\,\}$ is uniformly bounded away from zero, which ensures that $\lim_{k\in\mathcal{S}} \nabla M(v_k\,;\mu^P) = 0$.

Now assume the complementary case where the subsequence of iterates such that the line-search conditions (16a)–(16c) hold is finite. This implies that there exists k_0 such that for all $k > k_0$, the line-search condition (16d) must hold. Thus, all the subsequent iterates $\{\, v_k \,\}_{k>k_0}$ lies within the level set

$$\mathcal{L}\big(M(v_{k_0}\,;\mu^F)\big) \triangleq \big\{\, v \in \mathit{\Omega} : M(v\,;\mu^F) \le M(v_{k_0}\,;\mu^F) \,\big\},$$

where $\mathit{\Omega}$ represents the open set in which the merit function $M(v\,;\mu^F)$ is well defined, i.e.,

$$\mathit{\Omega} = \big\{\, v = (x, s, y, w) : v > \ell \,\big\}, \;\; \text{with} \;\; \ell = \big(-\infty, -\mu^B e, -\infty, -\mu^B e\big).$$

Notice that the value of $M(v\,;\mu^F)$ is $+\infty$ on the boundary of $\mathit{\Omega}$. Then by the continuity of the function $M(v\,;\mu^F)$, the level set $\mathcal{L}\big(M(v_{k_0}\,;\mu^F)\big)$ is a closed subset of $\mathit{\Omega}$. Moreover, Assumption 4.3 and Lemma 4.2(**i**) imply that the set of iterates $\{\, v_k \,\}_{k>k_0}$

is a bounded subset of $\mathcal{L}\big(M(v_{k_0}\,;\mu^{F})\big)$. Hence, there exists a compact subset of Ω such that $\{\, v_k \,\}_{k>k_0}$ lies within the compact subset. It follows that

$$\kappa \triangleq \min_{k>k_0,1\le i\le n} \{\, [\, v_k\,]_i - [\,\ell\,]_i \,\} > 0.$$

We show by contradiction that $\lim_{k\to\infty} \nabla M(v_k\,;\mu^{P}) = 0$. Suppose there exists a constant $\epsilon > 0$ and a subsequence $\mathcal{G}$ such that $\|\nabla M(v_k\,;\mu^{P})\| \ge \epsilon$ for all $k \in \mathcal{G}$. It follows from Lemma 4.1 and Lemma 4.2(**iv**) that $\lim_{k\to\infty} M(v_k\,;\mu^{F}) = M_{\min} > -\infty$. Using this result and the assumption that the line-search condition (16d) is satisfied for all k sufficiently large, it must follow that

$$\lim_{k\to\infty} \alpha_k \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k = 0,$$

which, together with Lemma 4.3(**ii**), implies that $\lim_{k\in\mathcal{G}} \alpha_k = 0$. For each k, define $\beta_k \triangleq \alpha_k/\gamma_{A}$. Then $\lim_{k\in\mathcal{G}} \beta_k = 0$ and the backtracking procedure in Algorithm 1 implies that the condition (16d) does not hold for the step β_k for all k sufficiently large. This means that the more stringent quasi-Armijo condition does not hold, i.e.,

$$M\big(\mathbf{proj}_{\,\Omega_k}(v_k + \beta_k \Delta v_k)\,;\mu^{P}\big) > M\big(v_k\,;\mu^{P}\big) + \alpha_k \eta_{A} \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k \tag{19}$$

for all k sufficiently large. By Lemma 4.3(**i**), we also have $\lim_{k\in\mathcal{G}} \|\beta_k \Delta v_k\| = 0$. Thus, there exists $\bar{k}$ such that every component of $\beta_k \Delta v_k$ satisfies $|[\,\beta_k \Delta v_k\,]_i| < \sigma\gamma$ for all $k > \bar{k}$ in $\mathcal{G}$. It follows that $v_k + \beta_k \Delta v_k \in \Omega_k$, which implies $\mathbf{proj}_{\,\Omega_k}(v_k + \beta_k \Delta v_k) = v_k + \beta_k \Delta v_k$. Now let $\mathcal{G}'$ denote the indices $k > \max\{\, k_0, \bar{k} \,\}$ of iterations at which a reduction in the initial step length was necessary, i.e., $\mathcal{G}' = \{\, k : \alpha_k < 1,\ k \in \mathcal{G},\ k > \max\{\, k_0, \bar{k} \,\} \,\}$. As α_k converges to zero, $\mathcal{G}'$ must be an infinite set. The inequality (19) implies that

$$M(v_k + \beta_k \Delta v_k\,;\mu^{P}) > M(v_k\,;\mu^{P}) + \beta_k \eta_{A} \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k$$

for all k in $\mathcal{G}'$. Adding $-\beta_k \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k$ to both sides and rearranging gives

$$\begin{aligned} M(v_k + \beta_k \Delta v_k\,;\mu^{P}) - M(v_k\,;\mu^{P}) - \beta_k \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k &> -\beta_k (1-\eta_{A}) \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k \\ &> \beta_k (1-\eta_{A})\delta, \quad \text{for all } \ k \in \mathcal{G}'. \end{aligned} \tag{20}$$

The Taylor expansion of $M(v_k + \beta_k \Delta v_k\,;\mu^{P})$ gives

$$\begin{aligned} &M(v_k + \beta_k \Delta v_k\,;\mu^{P}) - M(v_k\,;\mu^{P}) - \beta_k \nabla M(v_k\,;\mu^{P})^{\mathrm{T}} \Delta v_k \\ &\qquad = \beta_k \int_0^1 \big(\nabla M(v_k + \tau\beta_k \Delta v_k\,;\mu^{P}) - \nabla M(v_k\,;\mu^{P})\big)^{\mathrm{T}} \Delta v_k \, d\tau. \end{aligned} \tag{21}$$

If $\|\cdot\|_{D}$ denotes the norm dual to $\|\cdot\|$, i.e., $\|v\|_{D} = \max_{u\ne 0} |v^{\mathrm{T}} u|/\|u\|$, then

$$\left|\big(\nabla M(v_k+\tau\beta_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\big)^{\rm T}\Delta v_k\right| \le \|\nabla M(v_k+\tau\beta_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\|_D\|\Delta v_k\|.$$

If this inequality is substituted in (21), it then follows from (20) that

$$\begin{aligned}(1-\eta_A)\delta &< \int_0^1 \big(\nabla M(v_k+\tau\beta_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\big)^{\rm T}\Delta v_k\,d\tau\\ &\le \max_{0\le\tau\le 1}\|\nabla M(v_k+\tau\beta_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\|_D\|\Delta v_k\|, \ \text{ for all } \ k\in\mathcal{G}'.\end{aligned}$$

The continuity of ∇M implies that there exists some $\tau_k\in[0,\beta_k]$ such that

$$\max_{0\le\tau\le 1}\|\nabla M(v_k+\tau\beta_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\|_D = \|\nabla M(v_k+\tau_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\|_D.$$

Then

$$(1-\eta_A)\delta < \|\nabla M(v_k+\tau_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\|_D\|\Delta v_k\|. \tag{22}$$

However, $\alpha_k\Delta v_k\to 0$ implies $\tau_k\Delta v_k\to 0$ for $k\in\mathcal{G}$, and the continuity of ∇M gives

$$\|\nabla M(v_k+\tau_k\Delta v_k\,;\mu^P)-\nabla M(v_k\,;\mu^P)\|_D\to 0.$$

Lemma 4.3(**i**) implies that the right-hand side of (22) converges to zero, which gives the required contradiction. □

5 Solving the Nonlinear Optimization Problem

In this section, a projected-search interior-point method for solving the nonlinear optimization problem (NIPs) is formulated and analyzed. The method incorporates the projected-search algorithm presented in Section 4 with strategies for adjusting the parameters in the definition of the merit function. These parameters were fixed in Algorithm 1.

5.1 The algorithm

The proposed method is given in Algorithm 2. The method uses the distinction among O-iterations, M-iterations and F-iterations, which are described below.

The definition of an O-iteration is based on the optimality conditions for problem (NIPs). Progress towards optimality of the iterate $v_{k+1}=(x_{k+1},\ s_{k+1},\ y_{k+1},\ w_{k+1})$ is defined in terms of the following feasibility, stationarity, and complementarity measures:

$$\begin{aligned}\chi_{\rm feas}(v_{k+1}) &= \|c(x_{k+1})-s_{k+1}\|,\\ \chi_{\rm stny}(v_{k+1}) &= \max\big(\|\nabla f(x_{k+1})-J(x_{k+1})^{\rm T}y_{k+1}\|,\|y_{k+1}-w_{k+1}\|\big), \ \text{ and}\\ \chi_{\rm comp}(v_{k+1},\mu_k^B) &= \big\|\min\big(q_1(v_{k+1}),q_2(v_{k+1},\mu_k^B)\big)\big\|,\end{aligned}$$

where

$$q_1(v_{k+1}) = \max\big(|\min(s_{k+1}, w_{k+1}, 0)|, |s_{k+1} \cdot w_{k+1}|\big), \quad \text{and}$$
$$q_2(v_{k+1}, \mu_k^B) = \max\big(\mu_k^B e, |\min(s_{k+1} + \mu_k^B e, w_{k+1} + \mu_k^B e, 0)|, |(s_{k+1} + \mu_k^B e) \cdot (w_{k+1} + \mu_k^B e)|\big).$$

A first-order KKT point v_{k+1} for problem (NIPs) satisfies $\chi(v_{k+1}, \mu_k^B) = 0$, where

$$\chi(v, \mu) = \chi_{\text{feas}}(v) + \chi_{\text{stny}}(v) + \chi_{\text{comp}}(v, \mu). \tag{23}$$

Given these definitions, the kth iteration is designated as an O-iteration if $\chi(v_{k+1}, \mu_k^B) \le \chi_k^{\max}$, where $\{\chi_k^{\max}\}$ is a monotonically decreasing positive sequence. At an O-iteration the parameters are updated as $y_{k+1}^E = y_{k+1}$, $w_{k+1}^E = w_{k+1}$ and $\chi_{k+1}^{\max} = \frac{1}{2}\chi_k^{\max}$ (see Step 11 of Algorithm 2). These updates ensure that $\{\chi_k^{\max}\}$ converges to zero if infinitely many O-iterations occur. The point v_{k+1} is called an O-iterate.

If the condition for an O-iteration does not hold, a test is made to determine if $v_{k+1} = (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$ is an approximate first-order solution of the problem

$$\underset{v=(x,s,y,w)}{\text{minimize}} \;\; M(v\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B). \tag{24}$$

In particular, the kth iteration is called an M-iteration if v_{k+1} satisfies

$$\|\nabla_x M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k, \tag{25a}$$
$$\|\nabla_s M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k, \tag{25b}$$
$$\|\nabla_y M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k \|D_{k+1}^P\|_\infty, \quad \text{and} \tag{25c}$$
$$\|\nabla_w M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k \|D_{k+1}^B\|_\infty, \tag{25d}$$

where τ_k is a positive tolerance, $D_{k+1}^P = \mu_k^P I$, and $D_{k+1}^B = (S_{k+1} + \mu_k^B I)(W_{k+1} + \mu_k^B I)^{-1}$. In this case v_{k+1} is called an M-iterate because it is an approximate first-order solution of (24). The estimates s_{k+1}^E, y_{k+1}^E and w_{k+1}^E are defined by the safeguarded values

$$\left.\begin{aligned} s_{k+1}^E &= \min\big(\max(0, s_{k+1}), s_{\max} e\big), \\ y_{k+1}^E &= \max\big(-y_{\max} e, \min(y_{k+1}, y_{\max} e)\big), \\ w_{k+1}^E &= \min\big(w_{k+1}, w_{\max} e\big) \end{aligned}\right\} \tag{26}$$

for some large positive constants $s_{\max}$, $y_{\max}$ and $w_{\max}$. Next, Step 15 checks if the condition

$$\chi_{\text{feas}}(v_{k+1}) \le \tau_k \tag{27}$$

holds. If the condition holds, then $\mu_{k+1}^P \leftarrow \mu_k^P$; otherwise, $\mu_{k+1}^P \leftarrow \frac{1}{2}\mu_k^P$ to place more emphasis on satisfying the constraints $c(x) - s = 0$ in subsequent iterations. Similarly, Step 16 checks the inequalities

$$\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \le \tau_k, \quad s_{k+1} \ge -\tau_k e, \quad \text{and} \quad w_{k+1} \ge -\tau_k e. \tag{28}$$

If these conditions hold, then $\mu^{B}_{k+1} \leftarrow \mu^{B}_{k}$; otherwise, $\mu^{B}_{k+1} \leftarrow \frac{1}{2}\mu^{B}_{k}$ to place more emphasis on achieving complementarity in subsequent iterations.

An iteration that is not an O- or M-iteration is called an F-iteration. In an F-iteration none of the parameters in the merit function are changed, so that progress is measured solely in terms of the reduction in the merit function.

Algorithm 2 An all-shifted projected-search interior-point method.

1: **procedure pdProj**(x_0, s_0, y_0, w_0)
2:     **Restrictions:** $s_0 \geq 0$ and $w_0 \geq 0$;
3:     **Constants:** $\{\, \eta_A, \gamma_A \,\} \subset (0,1)$ and $\{\, y_{\max}, w_{\max}, s_{\max} \,\} \subset (0,\infty)$;
4:     Choose y^{E}_{0}; $\chi^{\max}_{0} > 0$; $\{\, \mu^{P}_{0}, \mu^{B}_{0} \,\} \subset (0,\infty)$; and $\mu^{L}_{0} \geq \mu^{P}_{0}$;
5:     Choose w^{E}_{0} and s^{E}_{0} such that $w^{E}_{0} + s^{E}_{0} + \mu^{B}_{0} e > 0$;
6:     Set $v_0 = (x_0, s_0, y_0, w_0)$; $k \leftarrow 0$;
7:     **while** $\|\nabla M(v_k)\| > 0$ **do**
8:         $(s^{E}, y^{E}, w^{E}, \mu^{P}, \mu^{B}) \leftarrow (s^{E}_{k}, y^{E}_{k}, w^{E}_{k}, \mu^{P}_{k}, \mu^{B}_{k})$;
9:         Compute v_{k+1} in Steps 6–19 of Algorithm 1;
10:         **if** $\chi(v_{k+1}, \mu^{B}_{k}) \leq \chi^{\max}_{k}$ **then** [O-iterate]
11:             $(\chi^{\max}_{k+1}, y^{E}_{k+1}, w^{E}_{k+1}, \mu^{P}_{k+1}, \mu^{B}_{k+1}, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi^{\max}_{k}, y_{k+1}, w_{k+1}, \mu^{P}_{k}, \mu^{B}_{k}, \tau_k)$;
12:             $s^{E}_{k+1} \leftarrow \max\{\, 0, s_{k+1} \,\}$;
13:         **else if** v_{k+1} satisfies (25a)–(25d) **then** [M-iterate]
14:             Set $(\chi^{\max}_{k+1}, \tau_{k+1}) = (\chi^{\max}_{k}, \frac{1}{2}\tau_k)$; Set s^{E}_{k+1}, y^{E}_{k+1} and w^{E}_{k+1} using (26);
15:             **if** $\chi_{\text{feas}}(v_{k+1}) \leq \tau_k$ **then** $\mu^{P}_{k+1} \leftarrow \mu^{P}_{k}$ **else** $\mu^{P}_{k+1} \leftarrow \frac{1}{2}\mu^{P}_{k}$ **end if**
16:             **if** $\chi_{\text{comp}}(v_{k+1}, \mu^{B}_{k}) \leq \tau_k$, $s_{k+1} \geq -\tau_k e$ and $w_{k+1} \geq -\tau_k e$ **then**
17:                 $\mu^{B}_{k+1} \leftarrow \mu^{B}_{k}$;
18:             **else**
19:                 $\mu^{B}_{k+1} \leftarrow \frac{1}{2}\mu^{B}_{k}$;
20:                 Reset s_{k+1} and w_{k+1} so that $s_{k+1} + \mu^{B}_{k+1} e > 0$ and $w_{k+1} + \mu^{B}_{k+1} e > 0$;
21:             **end if**
22:         **else** [F-iterate]
23:             $(\chi^{\max}_{k+1}, s^{E}_{k+1}, y^{E}_{k+1}, w^{E}_{k+1}, \mu^{P}_{k+1}, \mu^{B}_{k+1}, \tau_{k+1}) \leftarrow (\chi^{\max}_{k}, s^{E}_{k}, y^{E}_{k}, w^{E}_{k}, \mu^{P}_{k}, \mu^{B}_{k}, \tau_k)$;
24:         **end if**
25:         Update μ^{L}_{k+1} as in (29);
26:     **end while**
27: **end procedure**

Reducing the barrier parameter μ^{B} in Step 19 of Algorithm 2 may cause a slack variable s_i or a dual variable w_i to become infeasible with respect to its shifted bounds. In Step 20, if a multiplier w_i becomes infeasible after μ^{B} is reduced, it is reinitialized as $\max\{\, y_i, \frac{1}{2}w_i \,\}$. To remedy the infeasibility of a slack variable s_i, suppose μ^{B} and $\bar{\mu}^{B}$ denote a shift before and after it is reduced, with $s_i + \mu^{B} > 0$ and $s_i + \bar{\mu}^{B} \leq 0$, a strategy is proposed in Section 5.4 of [2], which temporarily imposes an equality constraint $s_i = 0$. This constraint is enforced by the primal-dual augmented Lagrangian term until the nonlinear constraint value $c_i(x)$ becomes larger than $\bar{\mu}^{B}$, at which point s_i is assigned the value $s_i = c_i(x)$ and allowed to move. On being freed, the corresponding Lagrange multiplier w_i is reinitialized as $\max\{\, y_i, \epsilon \,\}$, where ϵ is a small positive constant.

Given an initial value $\mu_0^L \geq \mu_0^P$, in Step 25 of Algorithm 2, the line-search parameter μ_k^L is updated as

$$\mu_{k+1}^L = \begin{cases} \mu_k^L & \text{if } \psi_k(\alpha_k\,; \mu_k^L) \leq \psi_k(0\,; \mu_k^L) + \alpha_k \eta_A \delta_k \text{ and } \mu_{k+1}^P = \mu_k^P; \\ \max\{\, \tfrac{1}{2}\mu_k^L, \mu_{k+1}^P \,\} & \text{otherwise,} \end{cases} \tag{29}$$

where $\delta_k = \nabla M(v_k\,; \mu^P)^{\mathrm{T}} \Delta v_k$. This updating rule guarantees that $\mu_k^L \geq \mu_k^P$ for all k.

5.2 Convergence analysis

Convergence analysis for Algorithm 2 follows a similar procedure as in Section 4.2 of [2], which uses the properties of the complementary approximate KKT (CAKKT) condition proposed by Andreani, Martínez and Svaiter [3], as described below.

Definition 5.1 (CAKKT condition)**.** *A feasible point* (x^*, s^*) *(i.e., a point such that* $s^* \geq 0$ *and* $c(x^*) - s^* = 0$*) is said to satisfy the* CAKKT condition if there exists a sequence $\{\, (x_j, s_j, u_j, z_j) \,\}$ with $\{\, x_j \,\} \to x^*$ and $\{\, s_j \,\} \to s^*$ such that

$$\{\, \nabla f(x_j) - J(x_j)^{\mathrm{T}} u_j \,\} \to 0, \tag{30}$$

$$\{\, u_j - z_j \,\} \to 0, \tag{31}$$

$$\{\, z_j \,\} \geq 0, \quad \text{and} \tag{32}$$

$$\{\, z_j \cdot s_j \,\} \to 0. \tag{33}$$

Any (x^*, s^*) satisfying these conditions is called a CAKKT point.

Theorem 5.1 (Andreani et al. [23, Theorem 4.3])**.** *If* (x^*, s^*) *is a* CAKKT point that satisfies CAKKT-regularity, then (x^*, s^*) is a first-order KKT point for (NIPs).

The first part of the analysis concerns the conditions under which limit points of the sequence $\{\, (x_k, s_k) \,\}$ are CAKKT points. As the results are tied to the different iteration types, to facilitate referencing of the iterations during the analysis we define

$$\mathcal{O} = \{\, k : \text{iteration } k \text{ is an O-iteration} \,\},$$
$$\mathcal{M} = \{\, k : \text{iteration } k \text{ is an M-iteration} \,\}, \quad \text{and}$$
$$\mathcal{F} = \{\, k : \text{iteration } k \text{ is an F-iteration} \,\}.$$

Lemma 5.1. *If* $|\mathcal{O}| = \infty$ *there exists at least one limit point* (x^*, s^*) *of the infinite sequence* $\{\, (x_{k+1}, s_{k+1}) \,\}_{k \in \mathcal{O}}$ *and any such limit point is a* CAKKT point.

Proof. Assumption 4.3 implies that there must exist at least one limit point of $\{\, x_{k+1} \,\}_{k \in \mathcal{O}}$. If x^* is such a limit point, Assumption 4.1 implies the existence of $\mathcal{K} \subseteq \mathcal{O}$ such that $\{\, x_{k+1} \,\}_{k \in \mathcal{K}} \to x^*$ and $\{\, c(x_{k+1}) \,\}_{k \in \mathcal{K}} \to c(x^*)$. As $|\mathcal{O}| = \infty$, the updating strategy of Algorithm 2 gives $\{\, \chi_k^{\max} \,\} \to 0$. Furthermore, as $\chi(v_{k+1}, \mu_k^B) \leq \chi_k^{\max}$ for all $k \in \mathcal{K} \subseteq \mathcal{O}$, and $\chi_{\text{feas}}(v_{k+1}) \leq \chi(v_{k+1}, \mu_k^B)$ for all k, it follows that

$\{\ \chi_{\rm feas}(v_{k+1})\ \}_{k\in\mathcal{K}} \to 0$, i.e., $\{\ c(x_{k+1}) - s_{k+1}\ \}_{k\in\mathcal{K}} \to 0$. With the definition $s^* = c(x^*)$, it follows that $\{\ s_{k+1}\ \}_{k\in\mathcal{K}} \to \lim_{k\in\mathcal{K}} c(x_{k+1}) = c(x^*) = s^*$, which implies that (x^*, s^*) is feasible for the general constraints because $c(x^*) - s^* = 0$. The remaining feasibility condition $s^* \ge 0$ is proved componentwise. For any $1 \le i \le m$, define

$$\mathcal{Q}_1 = \{\ k : [q_1(v_{k+1})]_i \le [q_2(v_{k+1}, \mu_k^B)]_i\ \} \ \text{ and } \ \mathcal{Q}_2 = \{\ k : [q_2(v_{k+1}, \mu_k^B)]_i < [q_1(v_{k+1})]_i\ \},$$

where q_1 and q_2 are used in the definition of $\chi_{\rm comp}$. If the set $\mathcal{K} \cap \mathcal{Q}_1$ is infinite, then it follows from the inequalities $\{\ \chi_{\rm comp}(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \le \{\ \chi(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \le \{\ \chi_k^{\max}\ \}_{k\in\mathcal{K}} \to 0$ that $s_i^* = \lim_{\mathcal{K}\cap\mathcal{Q}_1} [s_{k+1}]_i \ge 0$. Using a similar argument, if the set $\mathcal{K} \cap \mathcal{Q}_2$ is infinite, then $s_i^* = \lim_{\mathcal{K}\cap\mathcal{Q}_2} [s_{k+1}]_i = \lim_{\mathcal{K}\cap\mathcal{Q}_2} [s_{k+1} + \mu_k^B e]_i \ge 0$, where the second equality uses the limit $\{\ \mu_k^B\ \}_{k\in\mathcal{K}\cap\mathcal{Q}_2} \to 0$ that follows from the definition of $\mathcal{Q}_2$. Combining these two cases implies that $s_i^* \ge 0$, as claimed. It follows that the limit point (x^*, s^*) is feasible.

It remains to show that (x^*, s^*) is a CAKKT point. Let

$$[\bar{s}_{k+1}]_i = \begin{cases} [s_{k+1}]_i & \text{if } k \in \mathcal{Q}_1; \\ [s_{k+1} + \mu_k^B e]_i & \text{if } k \in \mathcal{Q}_2, \end{cases}$$

and

$$[\bar{w}_{k+1}]_i = \begin{cases} \max\{\ [w_{k+1}]_i, 0\ \} & \text{if } k \in \mathcal{Q}_1; \\ [w_{k+1} + \mu_k^B e]_i & \text{if } k \in \mathcal{Q}_2, \end{cases}$$

for every $1 \le i \le m$, and consider the sequence $(x_{k+1},\ \bar{s}_{k+1},\ y_{k+1},\ \bar{w}_{k+1})_{k\in\mathcal{K}}$ as a candidate for the sequence used in Definition 5.1 to verify that (x^*, s^*) is a CAKKT point. If $\mathcal{O} \cap \mathcal{Q}_2$ is finite, then it follows from the definition of $\bar{s}_{k+1}$ and the limit $\{\ s_{k+1}\ \}_{k\in\mathcal{K}} \to s^*$ that $\{\ [\ \bar{s}_{k+1}\ ]_i\ \}_{k\in\mathcal{K}} \to s_i^*$; also, $\{\ \chi_{\rm comp}(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \to 0$ implies that $\liminf_{k\in\mathcal{K}} [\ w_{k+1}\ ]_i \ge 0$, therefore $\{\ [\ \bar{w}_{k+1} - w_{k+1}\ ]_i\ \}_{k\in\mathcal{K}} \to 0$. On the other hand, if $\mathcal{O} \cap \mathcal{Q}_2$ is infinite, then the definitions of $\mathcal{Q}_2$ and $\chi_{\rm comp}(v_{k+1}, \mu_k^B)$, together with the limit $\{\ \chi_{\rm comp}(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \to 0$ imply that $\{\ \mu_k^B\ \} \to 0$, giving $\{\ [\ \bar{s}_{k+1}\ ]_i\ \}_{k\in\mathcal{K}} \to s_i^*$ and $\{\ [\ \bar{w}_{k+1} - w_{k+1}\ ]_i\ \}_{k\in\mathcal{K}} \to 0$. As the choice of i was arbitrary, these cases taken together imply that $\{\ \bar{s}_{k+1}\ \}_{k\in\mathcal{K}} \to s^*$ and $\{\ \bar{w}_{k+1} - w_{k+1}\ \}_{k\in\mathcal{K}} \to 0$.

The next step is to show that $\{\ (x_{k+1}, \bar{s}_{k+1}, y_{k+1}, \bar{w}_{k+1})\ \}_{k\in\mathcal{K}}$ satisfies the conditions required by Definition 5.1. It follows from the limit $\{\ \chi(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \to 0$ established above that $\{\ \chi_{\rm stny}(v_{k+1}) + \chi_{\rm comp}(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \le \{\ \chi(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \to 0$. This, together with the limit $\{\ \bar{w}_{k+1} - w_{k+1}\ \}_{k\in\mathcal{K}} \to 0$, implies that $\{\ \nabla f(x_{k+1}) - J(x_{k+1})^{\rm T} y_{k+1}\ \}_{k\in\mathcal{K}} \to 0$ and $\{\ y_{k+1} - w_{k+1}\ \}_{k\in\mathcal{K}} \to 0$, which establishes that conditions (30) and (31) hold. The nonnegativity of $\bar{w}_{k+1}$ for all k is obvious from its definition, which implies that (32) is satisfied for $\{\ \bar{w}_k\ \}_{k\in\mathcal{K}}$. Finally, it must be shown that (33) holds, i.e., that $\{\ \bar{w}_{k+1} \cdot \bar{s}_{k+1}\ \}_{k\in\mathcal{K}} \to 0$. Consider the ith components of $\bar{s}_k$ and $\bar{w}_k$. If the set $\mathcal{K} \cap \mathcal{Q}_1$ is infinite, then the definitions of $\bar{s}_{k+1}$, $q_1(v_{k+1})$ and $\chi_{\rm comp}(v_{k+1}, \mu_k^B)$, together with the limit $\{\ \chi_{\rm comp}(v_{k+1}, \mu_k^B)\ \}_{k\in\mathcal{K}} \to 0$, imply that $\{\ [\ \bar{w}_{k+1} \cdot \bar{s}_{k+1}\ ]_i\ \}_{\mathcal{K}\cap\mathcal{Q}_1} \to 0$. Similarly, if the set $\mathcal{K} \cap \mathcal{Q}_2$ is infinite, then the definitions of $\bar{s}_{k+1}$, $q_2(v_{k+1}, \mu_k^B)$ and $\chi_{\rm comp}(v_{k+1}, \mu_k^B)$, together with the limits $\{\, \chi_{\text{comp}}(v_{k+1}, \mu_k^B) \,\}_{k \in \mathcal{K}} \to 0$ and $\{\, \bar{w}_{k+1} - w_{k+1} \,\}_{k \in \mathcal{K}} \to 0$, imply that $\{\, [\, \bar{w}_{k+1} \cdot \bar{s}_{k+1} \,]_i \,\}_{k \in \mathcal{K} \cap \mathcal{Q}_2} \to 0$. Thus, these two cases lead to the conclusion that $\{\, \bar{w}_{k+1} \cdot \bar{s}_{k+1} \,\}_{k \in \mathcal{K}} \to 0$, which implies that condition (33) is satisfied. This completes the proof that (x^*, s^*) is a CAKKT point. $\square$

In the complementary case where $|\mathcal{O}| < \infty$, it will be shown that every limit point of the iteration subsequence $\{\, (x_{k+1}, s_{k+1}) \,\}_{k \in \mathcal{M}}$ is infeasible with respect to the constraints $c(x) - s = 0$ but solves the least-infeasibility problem

$$\underset{x,s}{\text{minimize}} \;\; \tfrac{1}{2}\|c(x) - s\|_2^2 \;\; \text{subject to} \;\; s \geq 0. \tag{34}$$

The first-order KKT conditions for problem (34) are

$$J(x^*)^{\mathrm{T}}\big(c(x^*) - s^*\big) = 0, \qquad s^* \geq 0, \tag{35a}$$
$$s^* \cdot \big(c(x^*) - s^*\big) = 0, \qquad c(x^*) - s^* \leq 0. \tag{35b}$$

These conditions define an infeasible stationary point.
Definition 5.2 (Infeasible stationary point)**.** *The pair (x^*, s^*) is an* infeasible stationary point *if $c(x^*) - s^* \neq 0$ and (x^*, s^*) satisfies the optimality conditions (35).*
Lemma 5.2. *If $|\mathcal{O}| < \infty$, then $|\mathcal{M}| = \infty$.*

Proof. The proof is by contradiction. Suppose that $|\mathcal{M}| < \infty$, in which case $|\mathcal{O} \cup \mathcal{M}| < \infty$. It follows from the definition of Algorithm 2 that $k \in \mathcal{F}$ for all k sufficiently large, i.e., there must exist an iteration index k_F such that

$$k \in \mathcal{F}, \;\; y_k^E = y^E, \;\; \text{and} \;\; (\tau_k, w_k^E, \mu_k^P, \mu_k^B) = (\tau, w^E, \mu^P, \mu^B) > 0 \tag{36}$$

for all $k \geq k_F$. The updating rule for $\{\, \mu_k^L \,\}$ implies that μ_k^L will be fixed at some $\mu^L \geq \mu^P$, and μ_k^F is then fixed at the value μ^L for all k sufficiently large. It follows from Theorem 4.1 that there exists a subsequence of iterates $\mathcal{S}$ such that

$$\lim_{k \to \mathcal{S}} \|\nabla M(v_k)\| = 0.$$

Then Lemma 4.2(**i**) and Lemma 4.2(**ii**) can be applied to show that (25) is satisfied for all $k \in \mathcal{S}$. This would mean, in view of Step 13 of Algorithm 2, that $\mathcal{S} \in \mathcal{M}$ with $|\mathcal{S}| = \infty$, which contradicts (36) because $\mathcal{F} \cap \mathcal{M} = \emptyset$. $\square$

For the next lemma, we introduce the quantities

$$\pi_{k+1}^Y = y_k^E - \frac{1}{\mu_k^P}\big(c(x_{k+1}) - s_{k+1}\big) \;\; \text{and} \;\; \pi_{k+1}^W = \mu_k^B (S_{k+1} + \mu_k^B I)^{-1}(w_k^E - s_{k+1} + s_k^E),$$

with $S_{k+1} = \operatorname{diag}(s_{k+1})$ associated with the gradient of the merit function in (7).

Lemma 5.3. *If* $|\mathcal{M}| = \infty$ *then*

$$\lim_{k\in\mathcal{M}} \|\pi^{Y}_{k+1} - y_{k+1}\| = 0.$$

Moreover, if there exists a subsequence of iterates $\mathcal{K} \subseteq \mathcal{M}$ *such that* $\lim_{k\in\mathcal{K}}\ s_k = s^* \geq 0$*, then*

$$\lim_{k\in\mathcal{K}}\ \|\pi^{W}_{k+1} - w_{k+1}\| = \lim_{k\in\mathcal{K}}\ \|\pi^{Y}_{k+1} - \pi^{W}_{k+1}\| = \lim_{k\in\mathcal{K}}\ \|y_{k+1} - w_{k+1}\| = 0.$$

Proof. It follows from (7) and (25c) that

$$\|\pi^{Y}_{k+1} - y_{k+1}\| \leq \tau_k. \tag{37}$$

As $|\mathcal{M}| = \infty$ by assumption, Step 14 of Algorithm 2 implies that $\lim_{k\to\infty}\ \tau_k = 0$. Combining this with (37) establishes the first limit in the result.

Furthermore, if there exists a subsequence $\mathcal{K} \subseteq \mathcal{M}$ such that $\lim_{k\in\mathcal{K}}\ s_k = s^* \geq 0$, then the updating rule of Algorithm 2 for s^{E}_k implies that $\lim_{k\in\mathcal{K}}\ (s^{E}_k - s_k) = 0$. The limit $\lim_{k\to\infty}\ \tau_k = 0$ may then be combined with (7), (25b) and (25c) to show that

$$\lim_{k\in\mathcal{K}}\ \|\pi^{W}_{k+1} - w_{k+1}\| = 0 \quad \text{and} \quad \lim_{k\in\mathcal{K}}\ \|\pi^{Y}_{k+1} - \pi^{W}_{k+1}\| = 0. \tag{38}$$

Finally, as $\lim_{k\to\infty}\ \tau_k = 0$, it follows from the bound (37) and limits (38) that

$$\lim_{k\in\mathcal{K}}\ \|y_{k+1} - w_{k+1}\| = \lim_{k\in\mathcal{K}}\ \|(y_{k+1} - \pi^{Y}_{k+1}) + (\pi^{Y}_{k+1} - \pi^{W}_{k+1}) + (\pi^{W}_{k+1} - w_{k+1})\| = 0.$$

This establishes the last of the four limits. ☐

Lemma 5.4. *If* $|\mathcal{O}| < \infty$*, then every limit point* (x^*, s^*) *of the iterate subsequence* $\{\,(x_{k+1}, s_{k+1})\,\}_{k\in\mathcal{M}}$ *satisfies* $c(x^*) - s^* \neq 0$.

Proof. The proof is similar to the proof of Lemma 4.7 in [2] but with some modified technical details.

Let (x^*, s^*) be a limit point of (the necessarily infinite) sequence $\mathcal{M}$, i.e., there exists a subsequence $\mathcal{K} \subseteq \mathcal{M}$ such that $\lim_{k\in\mathcal{K}}\ (x_{k+1}, s_{k+1}) = (x^*, s^*)$. For a proof by contradiction, assume that $c(x^*) - s^* = 0$, which implies that

$$\lim_{k\in\mathcal{K}}\ \|c(x_{k+1}) - s_{k+1}\| = 0. \tag{39}$$

First, we show that $s^* \geq 0$, which will imply that $(x^*,\ s^*)$ is feasible because of the assumption that $c(x^*) - s^* = 0$. The line search in Algorithm 1 gives $s_{k+1} + \mu^{B}_k e > 0$ for all k. If $\lim_{k\to\infty}\ \mu^{B}_k = 0$, then $s^* = \lim_{k\in\mathcal{K}}\ s_{k+1} \geq -\lim_{k\in\mathcal{K}} \mu^{B}_k e = 0$. On the other hand, if $\lim_{k\to\infty}\ \mu^{B}_k \neq 0$, then Step 19 of Algorithm 2 is executed a finite number of times, $\mu^{B}_k = \mu^{B} > 0$ and (28) holds for all $k \in \mathcal{M}$ sufficiently large. A combination of the assumption that $|\mathcal{O}| < \infty$, the result of Lemma 5.2, and the updates of Algorithm 2, establishes that $\lim_{k\to\infty} \tau_k = 0$ and

$$\chi_k^{\max} = \chi^{\max} > 0 \text{ for all sufficiently large } k \in \mathcal{K}. \tag{40}$$

Taking limits over $k \in \mathcal{M}$ in (28) and using $\lim_{k\to\infty} \tau_k = 0$ gives $s^* \geq 0$.

Using $|\mathcal{O}| < \infty$ together with Lemma 5.3, the fact that $\lim_{k\in\mathcal{K}} s_k = s^* \geq 0$ with $\mathcal{K} \subseteq \mathcal{M}$, and Step 16 of the line search of Algorithm 1 gives

$$\lim_{k\in\mathcal{K}} \|y_{k+1} - w_{k+1}\| = 0, \text{ and } w_{k+1} + \mu_{k+1}^{B} > 0 \text{ for all } k \geq 0. \tag{41}$$

Next, it can be observed from the definitions of π_{k+1}^{Y} and $\nabla_x M$ that

$$\begin{aligned}\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1} &= \nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}(2\pi_{k+1}^{Y} + y_{k+1} - 2\pi_{k+1}^{Y}) \\ &= \nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}\big(2\pi_{k+1}^{Y} - y_{k+1}\big) - 2J(x_{k+1})^{\mathrm{T}}(y_{k+1} - \pi_{k+1}^{Y}) \\ &= \nabla_x M(v_{k+1}\,; y_k^{E}, w_k^{E}, \mu_k^{P}, \mu_k^{B}) - 2J(x_{k+1})^{\mathrm{T}}(y_{k+1} - \pi_{k+1}^{Y}),\end{aligned}$$

which combined with $\{\, x_{k+1} \,\}_{k\in\mathcal{K}} \to x^*$, $\lim_{k\to\infty} \tau_k = 0$, (25a), and Lemma 5.3 gives

$$\lim_{k\in\mathcal{K}} \left\{\, \nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1} \,\right\} = 0. \tag{42}$$

The proof that $\lim_{k\in\mathcal{K}} \chi_{\text{comp}}(v_{k+1}, \mu_k^{B}) = 0$ involves two cases.

Case 1: $\lim_{k\to\infty} \mu_k^{B} \neq 0$. In this case $\mu_k^{B} = \mu^{B} > 0$ for all sufficiently large k. Combining this with $|\mathcal{M}| = \infty$ and the update to μ_k^{B} in Step 19 of Algorithm 2, it must be that (28) holds for all sufficiently large $k \in \mathcal{K}$, i.e., that $\chi_{\text{comp}}(v_{k+1}, \mu_k^{B}) \leq \tau_k$ for all sufficiently large $k \in \mathcal{K}$. As $\lim_{k\to\infty} \tau_k = 0$, it must hold that $\lim_{k\in\mathcal{K}} \chi_{\text{comp}}(v_{k+1}, \mu_k^{B}) = 0$.

Case 2: $\lim_{k\to\infty} \mu_k^{B} = 0$. Lemma 5.3 implies that $\lim_{k\in\mathcal{K}} (\pi_{k+1}^{W} - w_{k+1}) = 0$. The sequence $\{\, S_{k+1} + \mu_k^{B} I \,\}_{k\in\mathcal{K}}$ is bounded because $\{\, \mu_k^{B} \,\}$ is positive and monotonically decreasing and $\lim_{k\in\mathcal{K}} s_{k+1} = s^*$, which means by the definition of π_{k+1}^{W} and the updating rule for s_{k+1}^{E} in (26),

$$0 = \lim_{k\in\mathcal{K}} (S_{k+1} + \mu_k^{B} I)(\pi_{k+1}^{W} - w_{k+1}) = \lim_{k\in\mathcal{K}} \big(\mu_k^{B} w_k^{E} - (S_{k+1} + \mu_k^{B} I) w_{k+1}\big). \tag{43}$$

Moreover, as $|\mathcal{O}| < \infty$ and $w_k > 0$ for all k by construction, the updating strategy for w_k^{E} in Algorithm 2 guarantees that $\{\, w_k^{E} \,\}$ is bounded over all k (see (26)). It then follows from (43), the uniform boundedness of $\{\, w_k^{E} \,\}$, and $\lim_{k\to\infty} \mu_k^{B} = 0$ that

$$0 = \lim_{k\in\mathcal{K}} \big([s_{k+1}]_i + \mu_k^{B}\big)[w_{k+1}]_i = \lim_{k\in\mathcal{K}} \big([s_{k+1}]_i + \mu_k^{B}\big)\big([w_{k+1}]_i + \mu_k^{B}\big). \tag{44}$$

There are two subcases.

Subcase 2a: $s_i^* > 0$ for some i. As $\lim_{k\in\mathcal{K}} [\, s_{k+1} \,]_i = s_i^* > 0$ and $\lim_{k\to\infty} \mu_k^{B} = 0$, it follows from (44) that $\lim_{k\in\mathcal{K}} [\, w_{k+1} \,]_i = 0$. Combining these limits allows us to conclude that $\lim_{k\in\mathcal{K}} [\, q_1(v_{k+1}) \,]_i = 0$, which is the desired result for this case.

Subcase 2b: $s^*_i = 0$ for some i. In this case, it follows from the limits $\lim_{k\to\infty} \mu^B_k = 0$ and (44), $w_{k+1} + \mu^B_k > 0$ and the limit $\lim_{k\in\mathcal{K}}[\, s_{k+1}\,]_i = s^*_i = 0$ that $\lim_{k\in\mathcal{K}}[q_2(v_{k+1}, \mu^B_k)]_i = 0$, which is the desired result for this case.

As one of the two subcases above must occur for each component i, it follows that

$$\lim_{k\in\mathcal{K}} \chi_{\text{comp}}(v_{k+1}, \mu^B_k) = 0,$$

which completes the proof for Case 2.

Under the assumption $c(x^*) - s^* = 0$ it has been shown that (39), (41), (42), and the limit $\lim_{k\in\mathcal{K}}\ \chi_{\text{comp}}(v_{k+1}, \mu^B_k) = 0$ hold. Collectively, these results imply that $\lim_{k\in\mathcal{K}} \chi(v_{k+1}, \mu^B_k) = 0$. This limit, together with the inequality (40) and the condition checked in Step 10 of Algorithm 2, gives $k \in \mathcal{O}$ for all $k \in \mathcal{K} \subseteq \mathcal{M}$ sufficiently large. This is a contradiction because $\mathcal{O} \cap \mathcal{M} = \emptyset$, which establishes the desired result that $c(x^*) - s^* \neq 0$. ☐

Lemma 5.5. *If $|\mathcal{O}| < \infty$, then there exists at least one limit point (x^*, s^*) of the infinite sequence $\{\,(x_{k+1}, s_{k+1})\,\}_{k\in\mathcal{M}}$, and any such limit point is an infeasible stationary point as given by* Definition 5.2*.*

Proof. The proof is similar to the proof of Lemma 4.8 in [2] but with some modified technical details.

If $|\mathcal{O}| < \infty$ then Lemma 5.2 implies that $|\mathcal{M}| = \infty$. Moreover, the updating strategy of Algorithm 2 forces $\{\, y^E_k \,\}$ and $\{\, w^E_k \,\}$ to be bounded (see (26)). The next step is to show that $\{\, s_{k+1} \,\}_{k\in\mathcal{M}}$ is bounded.

For a proof by contradiction, suppose that $\{\, s_{k+1} \,\}_{k\in\mathcal{M}}$ is unbounded. It follows that there must be a component i and a subsequence $\mathcal{K} \subseteq \mathcal{M}$ for which $\{\,[\, s_{k+1}]_i \,\}_{k\in\mathcal{K}} \to \infty$. When Assumption 4.3 and Assumption 4.1 hold, $\{\, c(x_{k+1}) \,\}_{k\in\mathcal{K}}$, $\{\, \nabla f(x_{k+1}) \,\}_{k\in\mathcal{K}}$ and $\{\, J(x_{k+1}) \,\}_{k\in\mathcal{K}}$ must be bounded. This implies that $\{\,[\, \pi^Y_{k+1}]_i \,\}_{k\in\mathcal{K}}$ is unbounded. On the other hand, by (7), (25a), together with the limit $\lim_{k\to\infty} \tau_k = 0$ and Lemma 5.3,

$$\begin{aligned} 0 &= \lim_{k\in\mathcal{M}} \|\nabla_{\!x} M(v_{k+1}\,; y^E_k, w^E_k, \mu^P_k, \mu^B_k)\| \\ &= \lim_{k\in\mathcal{M}} \|\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}\pi^Y_{k+1} - J(x_{k+1})^{\mathrm{T}}(\pi^Y_{k+1} - y_{k+1})\| \\ &= \lim_{k\in\mathcal{M}} \|\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}}\pi^Y_{k+1}\| = 0, \end{aligned}$$

which contradicts the unboundedness of $\{\,[\, \pi^Y_{k+1}]_i \,\}_{k\in\mathcal{K}}$. Thus, it must be the case that $\{\, s_{k+1} \,\}_{k\in\mathcal{M}}$ is bounded.

The next part of the proof is to establish that $s^* \geq 0$, which is the inequality condition of (35a). The test in Step 16 of Algorithm 2 (i.e., testing whether (28) holds) is checked infinitely often because $|\mathcal{M}| = \infty$. If (28) is satisfied finitely many times, then the update $\mu^B_{k+1} = \frac{1}{2}\mu^B_k$ forces $\{\, \mu^B_{k+1} \,\} \to 0$. Combining this with $s_{k+1} + \mu^B_k e > 0$ shows that $s^* \geq 0$, as claimed. On the other hand, if (28) is satisfied for all sufficiently large $k \in \mathcal{M}$, then $\mu^B_{k+1} = \mu^B > 0$ for all sufficiently large k and

$\lim_{k\in\mathcal{K}} \chi_{\text{comp}}(v_{k+1}, \mu_k^B) = 0$ because $\{\tau_k\} \to 0$. It follows from these two facts that $s^* \ge 0$, as claimed.

The boundedness of $\{s_{k+1}\}_{k\in\mathcal{M}}$ and Assumption 4.3 ensure the existence of at least one limit point of $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$. If (x^*, s^*) is any such limit point, there must be a subsequence $\mathcal{K} \subseteq \mathcal{M}$ such that $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{K}} \to (x^*, s^*)$. It remains to show that (x^*, s^*) is an infeasible stationary point (i.e., that (x^*, s^*) satisfies the optimality conditions (35a)–(35b)).

As $|\mathcal{O}| < \infty$, it follows from Lemma 5.4 that $c(x^*) - s^* \neq 0$. Combining this with $\{\tau_k\} \to 0$, which holds because $\mathcal{K} \subseteq \mathcal{M}$ is infinite (on such iterations $\tau_{k+1} \leftarrow \frac{1}{2}\tau_k$), it follows that the condition (27) of Step 15 of Algorithm 2 will not hold for all sufficiently large $k \in \mathcal{K} \subseteq \mathcal{M}$. The subsequent updates ensure that $\{\mu_k^P\} \to 0$, hence $\{\mu_k^F\} \to 0$ by the updating rule for $\{\mu_k^L\}$, which, combined with (17), the boundedness of $\{y_k^E\}$, and Lemma 5.3, gives

$$\{c(x_{k+1}) - s_{k+1}\}_{k\in\mathcal{K}} \le \left\{\mu_k^F\left(y_k^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) + \mu_k^B\right)\right\}_{k\in\mathcal{K}} \to 0.$$

This implies that $c(x^*) - s^* \le 0$ and the second condition in (35b) holds.

The rest of the proof is the same as in the proof of Lemma 4.8 in [2]. □

Theorem 5.2. *Under* Assumptions 4.1–4.3, *one of the following occurs:*

- **(i)** $|\mathcal{O}| = \infty$, *limit points of* $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{O}}$ *exist, and every such limit point* (x^*, s^*) *is a* CAKKT point for problem (NIPs). If, in addition, CAKKT-regularity holds at (x^*, s^*), then (x^*, s^*) is a KKT point for problem (NIPs).
- **(ii)** $|\mathcal{O}| < \infty$, $|\mathcal{M}| = \infty$, limit points of $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$ exist, and every such limit point (x^*, s^*) is an infeasible stationary point.

Proof. Part (**i**) follows from Lemma 5.1 and Theorem 5.1. Part (**ii**) follows from Lemma 5.5. Also, the exclusive conditions on $|\mathcal{O}|$ imply that only one of these two cases must occur. □

6 Numerical Experiments

Numerical results were obtained for Algorithm `pdProj`, which is a MATLAB implementation of the projected-search primal-dual interior-point method proposed in Sections 3–5. For comparison purposes, results are also given for two primal-dual interior-point methods that do not use projection. The first is Algorithm `pdbAll`, which is a method that shifts both the primal and dual variables. The second is Algorithm `pdb`, which is an extension of the primal-shifted method of Gill, Kungurtsev and Robinson [2].

Algorithms `pdb` and `pdbAll` are implemented with a flexible Armijo line search in which the step length is chosen to satisfy the conditions (16a)–(16d) with $\psi_k(\alpha\,;\mu)$ and $\phi_k(\alpha\,;\mu)$ given by $M\left(v_k + \alpha\Delta v_k\,;\mu\right)$ and $\|F\left(v_k + \alpha\Delta v_k\,;\mu\right)\|$. Exact second derivatives were used for all the runs.

6.1 Implementation details

The iterates were terminated at the first point that satisfied the conditions $e_P(x,s) < \tau_P$ and $e_D(x,s,y,w) < \tau_D$, where e_P and e_D are the primal and dual infeasibilities

$$e_P(x,s) = \left\| \begin{pmatrix} \min\{0,\ s\} \\ \|c(x)-s\|_\infty / \max\{1,\ \|s\|_\infty\} \end{pmatrix} \right\|_\infty, \tag{45a}$$

and

$$e_D(x,s,y,w) = \left\| \begin{pmatrix} \|\nabla f(x) - J(x)^{\mathrm{T}} y\|_\infty / \sigma \\ \|w - y\|_\infty \\ w \cdot \min\{1,\ s\} \end{pmatrix} \right\|_\infty, \tag{45b}$$

with $\sigma = \max\{1,\ \|\nabla f(x)\|,\ \max\{1,\ \|y\|\}\|J(x)\|_\infty\}$. These quantities provide a measure of the scaled distance to the primal and dual optimality conditions (1). Similarly, the iterates were terminated at an infeasible stationary point (x,s) if $e_P(x,s) > \tau_P$, $\min\{0,\ s\} \le \tau_P$ and $e_I(x,s) \le \tau_{\text{inf}}$, where

$$e_I(x,s) = \left\| J(x)^{\mathrm{T}}(c(x)-s) \cdot \min\{1,\ s\} \right\|_\infty / \sigma. \tag{46}$$

All three MATLAB implementations were initialized with identical control parameters that were chosen based on the empirical performance on the entire collection of problems. A summary of the values is given in Table 6.1.

Table 1 Control parameters for Algorithms **pdb**, **pdbAll** and **pdProj**.

Parameter	Description	Value
$s_{\max}$, $y_{\max}$, $w_{\max}$	Maximum allowed y^E, w^E, s^E	1.0e+6
μ_0^P	Initial penalty parameter	1.0e-4
μ_0^B	Initial barrier parameter	1.0e-4
μ_0^L	Initial flexible line-search penalty parameter	1.0
τ_0	Initial termination tolerance for specifying an M-iterate	0.5
τ_P	Primal feasibility tolerance (45a)	1.0e-4
τ_D	Dual feasibility tolerance (45b)	1.0e-4
τ_{inf}	Infeasible stationary point tolerance (46)	1.0e-4
$\chi_0^{\max}$	Initial target for an O-iteration	1.0e+3
η_A	Line-search Armijo sufficient reduction	1.0e-2
η_F	Line-search sufficient reduction for $\|F\|$	0.9
γ_A	Line-search factor for reducing an Armijo step	0.5
f_{unb}	Unbounded objective	-1.0e+12
$M_{\max}$	Constants in line-search tolerance (16a) and (16b)	1.0e+12
$F_{\max}$	Constant in the line-search tolerance (16c)	1.0e+8
σ	Bound perturbation in the definition of Ω_k (15)	0.8
$k_{\max}$	Iteration limit	500

The results were obtained for optimization problems from the CUTEst test collection (see Bongartz et al. [24] and Gould, Orban and Toint [25]). Results were obtained for five subsets of problems from the CUTEst test collection. The subsets consisted of all 126 problems formulated by Hock and Schittkowski ([26]) (problems HS); 139 problems with a general nonlinear objective and upper and lower bounds on the variables (problems BC); 212 problems with a general nonlinear objective, general linear constraints and bounds on the variables (problems LC); 648 problems with a general nonlinear objective, general linear and nonlinear constraints and bounds on the variables (problems NC); and 141 problems with a quadratic objective, general linear constraints and bounds on the variables (problems QP). The NC problems include 264 feasibility problems, i.e., problems with nonlinear constraints but a constant objective function. In an attempt to create a unique solution for comparison purposes, all the feasibility problems were modified to find the feasible point of least Euclidean length. For example, in terms of the problem format (NIPs) the constant objective function was replaced by $\frac{1}{2}\|x\|^2$.

The BC, LC, NC and QP subsets were selected based on the number of variables and general constraints. In particular, a problem was chosen if the associated KKT system was of the order of 2000 or less. The same criterion was used to set the dimension of those problems for which the problem size can be specified. The only eligible problem omitted from the test-set was `lhaifam`, which generated a floating-point exception when computed at the initial point. A complete list of the problems tested, together with additional details of the number of function evaluations and iterations needed for each problem is given by Gill and Zhang [27].

Each CUTEst problem may be written in the general form (NIP). In this format, a fixed variable or an equality constraint has the same value for its upper and lower bounds. A variable or constraint with no upper or lower limit is indicated by a bound of $\pm 10^{20}$. The approximate Newton equations for problem (NIP) are derived by Gill and Zhang [1]. As is the case for problem (NIPs) the principal work at each iteration is the solution of a reduced $(n+m)\times(n+m)$ KKT system analogous to (14). Each KKT matrix was factored using the MATLAB built-in command `LDL`, which uses the routine MA57 [28]. If the inertia of this matrix was incorrect, i.e., the matrix was singular or had more than m negative eigenvalues, the Hessian of the Lagrangian H was modified using the method of Wächter and Biegler [29, Algorithm IC, p. 36], which factors the KKT matrix with δI_n added to H. At any given iteration the value of δ is increased from zero if necessary until the inertia of the KKT matrix is correct.

The initial primal-dual estimate $(x_0,\, y_0)$ was based on the default initial values supplied by CUTEst. If necessary, x_0 was projected onto the set $\{\, x : \ell^x \le x \le u^x \,\}$ to ensure feasibility with respect to the bounds on x. An algorithm was considered to have converged if the iterates were terminated at a point that satisfied the conditions (45a)–(45b) and (46) defined in terms of the constraints associated with problem (NIP).

We note that an interior-point method that does not use shifts would require a strictly interior starting point, which implies that the choice of default CUTEst starting point would not be possible in this situation. Moreover, this choice of starting point illustrates the potential benefits of using shifts for performing a warm-start. The CUTEst QP problems `ferrisdc` and `linspanh` both have a solution at the initial point. All three algorithms `pdb`, `pdbAll` and `pdProj` recognize the initial point as being optimal and terminate immediately.

6.2 Performance profiles

The runs were done using Matlab version R2022b on an iMac Pro with a 3.0 GHz Intel Xeon W processor and 128 GB of 800 MHz DDR4 RAM running macOS, version 12.6.8 (64 bit). The overall cpu-time required by a constrained optimization method is dominated by the time needed to solve the KKT equations and the time to evaluate the problem functions (i.e., the objective and constraint functions and their derivatives). Given the difficulty of accurately measuring cpu time in a multiprocessor and multiuser computing environment, function and iteration performance profiles provide a clear, accurate and concise way to display the relative efficiencies of methods. Performance profiles are particularly effective when comparing methods on problems for which the cpu time is negligible.

Performance profiles were proposed by Dolan and Moré [30]. Let $\mathcal{P}$ denote a set of problems used for a given numerical experiment. For each method s we define the function $\pi_s : [0, r_M] \mapsto \mathbb{R}^+$ such that

$$\pi_s(\tau) = \frac{1}{n_p} \left| \left\{ \, p \in \mathcal{P} : \log_2(r_{p,s}) \le \tau \, \right\} \right|,$$

where n_p is the number of problems in the test set and $r_{p,s}$ denotes the ratio of the number of function evaluations needed to solve problem p with method s and the least number of function evaluations needed to solve problem p. If method s failed for problem p, then $r_{p,s}$ is set to be twice the maximal ratio. The parameter r_M is the maximum value of $\log_2(r_{p,s})$. Figures 1–5 give the function-evaluation and iteration performance profiles for the HS, BC, LC, NC and QP test-sets respectively. The profiles show the benefits of shifting both primal and dual variables, as well as using a projected-search method based on the primal-dual search direction. The proposed method `pdProj` outperforms the other two solvers in terms of both efficiency and robustness. In particular, the number of times that the search direction must be computed is substantially reduced. This reduction is most pronounced when the problem is a quadratic program.

7 Conclusions

A new projected-search primal-dual interior-point method has been formulated and analyzed for constrained optimization problems. The method is based on combining a new primal-dual interior-point method with a projected-search method for bound-constrained optimization that uses a flexible non-monotone quasi-Armijo line search. The projected-search method projects the underlying search direction onto a superset of the feasible region defined by perturbing the constraint bounds. With this approach the direction of the search path may change multiple times along the boundary of the perturbed feasible region at the cost of computing a single direction. The direction for the projected search is an approximate Newton direction associated with minimizing a shifted primal-dual penalty-barrier function. This function involves a primal-dual shifted penalty term for the equality constraints in conjunction with an analogous primal-dual shifted barrier term for enforcing the inequality constraints

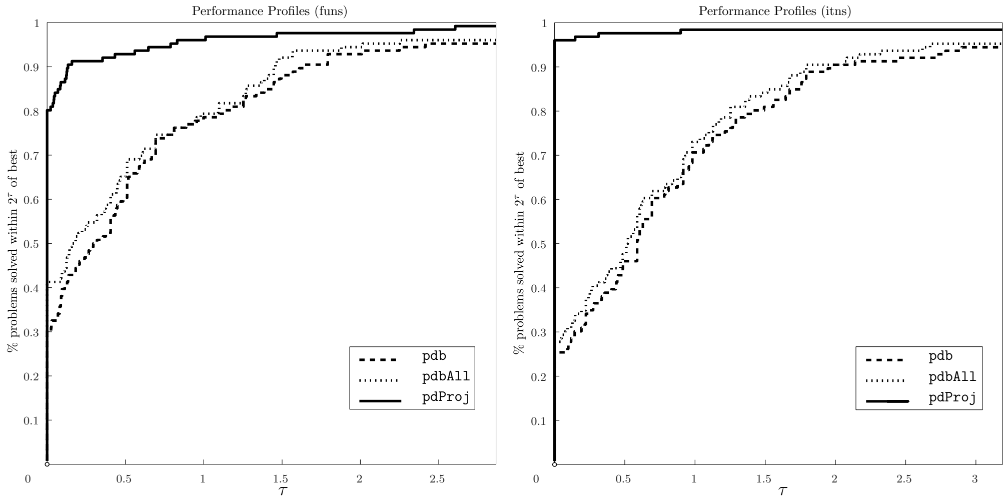

Fig. 1 Performance profiles for the primal-dual interior-point algorithms `pdb`, `pdbAll` and `pdProj` applied to all 126 Hock-Schittkowski (HS) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

and the nonnegativity constraints on their associated multipliers. It is shown that a specific approximate Newton method for the unconstrained minimization of the penalty-barrier function generates directions that are identical to those associated with a variant of the conventional path-following method. In this context the penalty-barrier function is used as a merit function for assessing points generated by Newton's method for a zero of the path-following equations. Numerical results from a large number of test problems from the CUTEst test collection indicate that the use of a projected search can significantly reduce the number of iterations, thereby reducing the number of times that a search direction must be computed. In particular, the numerical results indicates that the method is particularly well-suited to solving the quadratic programming subproblem in an SQP method. In this context the work per iteration is dominated by the cost of solving a large symmetric indefinite system of equations for the search direction. Moreover, the shifts on the primal and dual variables allow the method to be safely "warm started" from the solution of the preceding QP subproblem.

Future work will consider the implementation of the method within an SQP solver, the integration of the method with iterative KKT solvers and the extension of the method to a stochastic setting.

Acknowledgments

The authors would like to thank the referees for constructive comments that significantly improved the presentation.

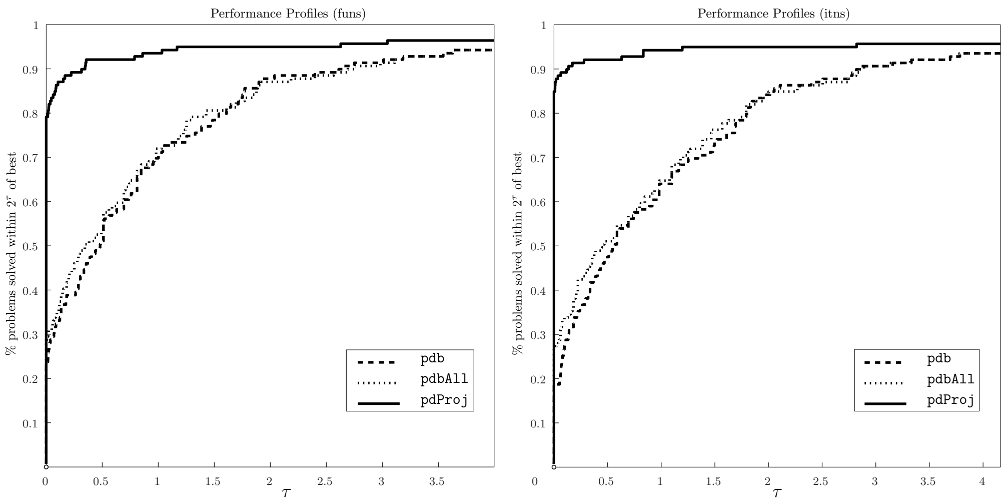

Fig. 2 Performance profiles for the primal-dual interior-point algorithms `pdb`, `pdbAll` and `pdProj` applied to 139 bound-constrained (BC) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

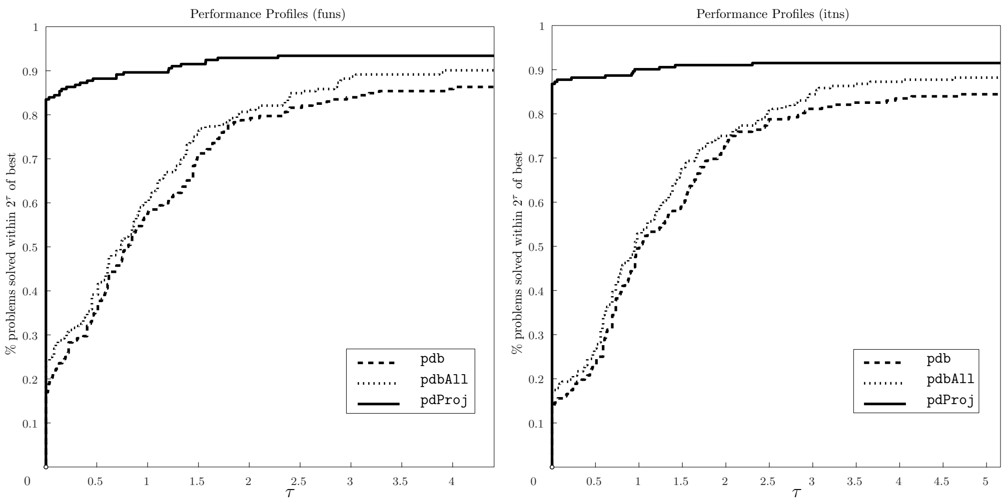

Fig. 3 Performance profiles for the primal-dual interior-point algorithms `pdb`, `pdbAll` and `pdProj` applied to 212 linearly constrained (LC) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

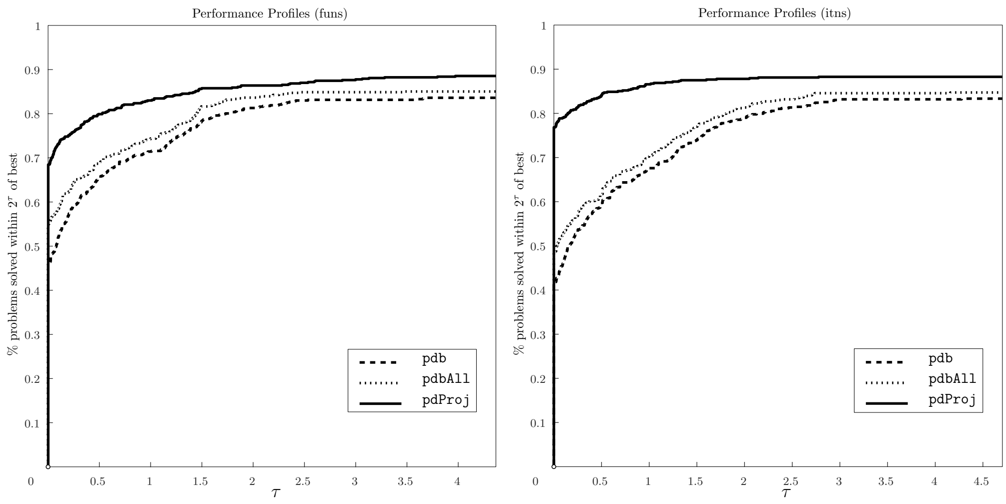

Fig. 4 Performance profiles for the primal-dual interior-point algorithms `pdb`, `pdbAll` and `pdProj` applied to 648 nonlinearly constrained (NC) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

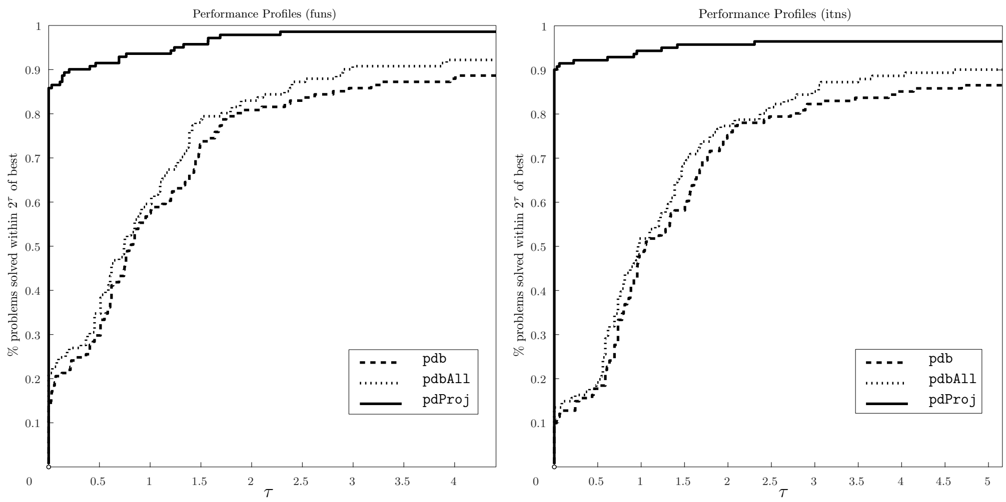

Fig. 5 Performance profiles for the primal-dual interior-point algorithms `pdb`, `pdbAll` and `pdProj` applied to 141 quadratic programming (QP) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

Declarations

Funding Research supported in part by National Science Foundation grants DMS-0915220 and DMS-1318480. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.
Competing interests The authors have no competing interests to declare that are relevant to the content of this article.
Data availability The data that support the findings of this study are available from the corresponding author upon request.

References

[1] Gill, P.E., Zhang, M.: Equations for a projected-search path-following method for nonlinear optimization. Center for Computational Mathematics Report CCoM 22-02, Center for Computational Mathematics, University of California, San Diego, La Jolla, CA (2022)

[2] Gill, P.E., Kungurtsev, V., Robinson, D.P.: A shifted primal-dual penalty-barrier method for nonlinear optimization. SIAM J. Optim. **30**(2), 1067–1093 (2020)

[3] Andreani, R., Martínez, J.M., Svaiter, B.F.: A new sequential optimality condition for constrained optimization and algorithmic consequences. SIAM J. Optim. **20**(6), 3533–3554 (2010)

[4] Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: Fletcher, R. (ed.) Optimization, pp. 283–298. Academic Press, London and New York (1969)

[5] Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**, 303–320 (1969)

[6] Gill, P.E., Saunders, M.A., Wong, E.: On the performance of SQP methods for nonlinear programming. In: Defourny, B., Terlaky, T. (eds.) Modeling and Optimization: Theory and Applications. Springer Proceedings in Mathematics and Statistics, vol. 147, pp. 95–123 (2015)

[7] Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM Rev. **47**, 99–131 (2005)

[8] Fiacco, A.V., McCormick, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques, p. 210. John Wiley and Sons, Inc., New York (1968)

[9] Frisch, K.R.: The logarithmic potential method of convex programming. Memorandum of May 13, University Institute of Economics, Oslo, Norway (1955)

[10] Fiacco, A.V.: Barrier methods for nonlinear programming. In: Holzman, A. (ed.)

Operations Research Support Methodology, pp. 377–440. Marcel Dekker, New York (1979)

[11] Byrd, R.H., Hribar, M.E., Nocedal, J.: An interior point algorithm for large-scale nonlinear programming. SIAM J. Optim. **9**(4), 877–900 (1999)

[12] Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM J. Optim. **16**(1), 1–31 (2005)

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

[14] Gertz, E.M., Gill, P.E.: A primal-dual trust-region algorithm for nonlinear programming. Math. Program., Ser. B **100**, 49–94 (2004)

[15] Forsgren, A., Gill, P.E., Shinnerl, J.R.: Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization. SIAM J. Matrix Anal. Appl. **17**, 187–211 (1996)

[16] Ponceleón, D.B.: Barrier methods for large-scale quadratic programming. Report SOL 91-2, Department of Operations Research, Stanford University, Stanford, CA (1991). PhD thesis

[17] Wright, S.J.: Stability of linear equations solvers in interior-point methods. SIAM J. Matrix Anal. Appl. **16**, 1287–1307 (1995)

[18] Wright, S.J.: Stability of augmented system factorizations in interior point methods. SIAM J. Matrix Anal. Appl. **18**, 191–222 (1997)

[19] Ferry, M.W., Gill, P.E., Wong, E., Zhang, M.: A class of projected-search methods for bound-constrained optimization. Optimization Methods and Software **0**(0), 1–30 (2023) https://doi.org/10.1080/10556788.2023.2241769

[20] Zhang, M.: Projected-search methods for constrained optimization. PhD thesis, Department of Mathematics, University of California San Diego, La Jolla, CA (2023)

[21] Curtis, F.E., Nocedal, J.: Flexible penalty functions for nonlinear constrained optimization. IMA J. Numer. Anal. **28**(4), 749–769 (2008)

[22] Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM J. Optim. **12**(4), 979–1006 (2002)

[23] Andreani, R., Martínez, J.M., Ramos, A., Silva, P.J.S.: Strict constraint qualifications and sequential optimality conditions for constrained optimization. Math. Oper. Res. **43**(3), 693–717 (2018)

[24] Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, Ph.L.: CUTE: Constrained and

unconstrained testing environment. ACM Trans. Math. Software **21**(1), 123–160 (1995)

[25] Gould, N.I.M., Orban, D., Toint, Ph.L.: CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. ACM Trans. Math. Software **29**(4), 373–394 (2003)

[26] Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes. Lecture Notes in Econom. Math. Syst. 187. Springer, Berlin (1981)

[27] Gill, P.E., Zhang, M.: Numerical results for a projected-search interior-point method. Center for Computational Mathematics Report CCoM 22-03, Center for Computational Mathematics, University of California, San Diego, La Jolla, CA (2022)

[28] Duff, I.S.: MA57—a code for the solution of sparse symmetric definite and indefinite systems. ACM Trans. Math. Software **30**(2), 118–144 (2004)

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

[30] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2, Ser. A), 201–213 (2002)