A Projected-Search Interior Method for Nonlinear Optimization

Philip E. Gill* Minxin Zhang*

UCSD Center for Computational Mathematics
Technical Report CCoM-22-1
June 2022

Abstract

This paper concerns the formulation and analysis of a new interior method for general nonlinearly constrained optimization that combines a shifted primal-dual interior 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 eliminates the ill-conditioning of the associated linear equations that may occur when the dual variables are close to zero. 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 show that the proposed method requires fewer iterations than a conventional interior method, thereby reducing the number of times that a search direction must be computed. In particular, 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.

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

AMS subject classifications. 49J20, 49J15, 49M37, 49D37, 65F05, 65K05, 90C30

---

*Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (pgill@ucsd.edu, miz151@ucsd.edu) 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.

1. Introduction

This paper concerns the formulation and analysis of a new primal-dual interior method for solving nonlinear optimization problems of the form

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

where $c:\mathbb{R}^n\mapsto\mathbb{R}^m$ and $f:\mathbb{R}^n\mapsto\mathbb{R}$ are twice-continuously differentiable. (The slack variables s serve to convert the inequalities $c(x)\geq 0$ into a mixture of equalities and inequalities that do not require the need to know an initial point for which c is strictly positive.)

In [13], Gill, Kungurtsev and Robinson propose an algorithm for (NIP) based on using a shifted primal-dual penalty-barrier function as a merit function for a primal-dual path-following method. This function involves a primal-dual shifted penalty term (i.e., an augmented Lagrangian term) for the equality constraints $c(x)-s=0$ (see, e.g., Powell [20], Hestenes [18] and Gill and Robinson [15]), and an analogous primal-dual shifted barrier term for the inequalities $s\geq 0$. It is shown that a specific approximate Newton method for the unconstrained minimization of the merit function generates search directions that are identical to those associated with a variant of the conventional path-following method in which the perturbation of the complementarity condition does not need to go to zero.

The proposed method is based on an extension of the Gill, Kungurtsev and Robinson method that includes shifts on the dual variables as well as the slack variables s. (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 [16].) Shifts on the dual variables allow the method to be safely "warm started" from a good approximate solution and eliminates the ill-conditioning of the associated linear equations that may occur when the dual variables are close to zero.

The shifted primal-dual penalty-barrier function includes logarithmic barrier terms that create a singularity at the boundary of the primal-dual shifted feasible region, which implies that the variables are subject to implicit bound constraints during the minimization. We consider minimizing the merit function using a projected-search method that uses a flexible non-monotone *quasi-Armijo* line search. Unlike conventional interior methods, which impose an upper bound on the step size to prevent the variables from becoming infeasible, projected-search interior methods project the underlying search direction onto a subset of the feasible region defined by perturbing the 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 methods have the potential of requiring fewer iterations than a conventional interior method, thereby reducing the number of times that a search direction must be computed.

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 perturbed feasible region Ω_k. 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 point (KKT), i.e., it satisfies reasonable stopping criteria and is a KKT point under a complementary approximate KKT regularity condition (see Andreani, Martínez and Svaiter [2]). In particular, it is shown that the limit points of $\{v_k\}_{k=0}^{\infty}$ lie within the intersection of all the sets Ω_k, with $\Omega_k\to\Omega$.

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 slack variables. 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 (NIP) 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, $\min(a, b)$ 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 Lagrangian function associated with (NIP) is $L(x, y) = f(x) - c(x)^{\mathrm{T}} y$, where y is the m-vector of dual variables. The Hessian of the Lagrangian with respect to x is denoted by $H(x, y) = \nabla^2 f(x) - \sum_{i=1}^m y_i \nabla^2 c_i(x)$. Let $\{\alpha_j\}_{j\ge 0}$ be a sequence of scalars, vectors, or matrices and let $\{\beta_j\}_{j\ge 0}$ be a sequence of positive scalars. If there exists a positive constant γ such that $\|\alpha_j\| \le \gamma\beta_j$, we write $\alpha_j = O\big(\beta_j\big)$. If there exists a sequence $\{\gamma_j\} \to 0$ such that $\|\alpha_j\| \le \gamma_j \beta_j$, we say that $\alpha_j = o(\beta_j)$. If there exists a positive sequence $\{\sigma_j\} \to 0$ and a positive constant β such that $\beta_j > \beta\sigma_j$, we write $\beta_j = \Omega(\sigma_j)$.

2. Background

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

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y &= 0, & y - w &= 0,\\ c(x) - s &= 0, & s &\ge 0,\\ s \cdot w &= 0, & w &\ge 0, \end{aligned}\right\} \tag{2.1}$$

where the vectors y and w constitute the Lagrange multipliers for the equality constraint $c(x) - s = 0$ and nonnegativity constraint $s \ge 0$ respectively. Following standard practice, any point satisfying the conditions (2.1) 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 (NIP). Points on this path satisfy a system of nonlinear equations that represent the deviations from a perturbation of the first-order optimality conditions (2.1). In a conventional path-following approach, the perturbed optimality conditions correspond to replacing the equality constraints and complementarity conditions of (2.1) 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 (NIP). Penalty and barrier 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 [9], Frisch [11] and Fiacco [8]). 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 the Newton equations 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 [4], Wächter and Biegler [21], Forsgren and Gill [10], and Gertz and Gill [12].

When implemented with exact second derivatives, path-following interior methods often converge in few iterations–even for very large problems. As the dimension and zero/nonzero structure of the Jacobian matrix remains fixed, the Newton equations may be solved efficiently using advanced "off-the-shelf" linear algebra software. On the negative side, although conventional path-following interior methods are very effective for solving "one-off" problems, they are difficult to adapt to solving a sequence of related problems using so-called "warm starts", i.e., using the solution of one problem as an initial estimate of the solution of the next.

In a conventional path-following interior method, it is necessary to force $\mu \to 0$ to ensure that points near the path eventually satisfy the optimality conditions (2.1). 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 converge to the optimal multipliers. Based on this observation, the method of Gill, Kungurtsev and Robinson [13] is based on the perturbed optimality conditions

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y &= 0, & y - w &= 0, \\ c(x) - s &= \mu^P(y^E - y), & s &\ge 0, \\ s \cdot w &= \mu^B(w^E - w), & w &\ge 0, \end{aligned}\right\} \tag{2.2}$$

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 (2.2) 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 method based on the conditions (2.2) 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 (NIP) satisfying certain second-order optimality conditions, the Newton equations for a zero of the conditions (2.2) 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 [2]). The reader is referred to Gill, Kungurtsev and Robinson [13] for details.

In the following section, the Gill, Kungurtsev and Robinson algorithm is extended to include shifts on the dual variables w as well as 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

$$\left.\begin{aligned} \nabla f(x) - J(x)^{\mathrm{T}} y = 0, \qquad & y - w = 0, \\ c(x) - s = \mu^P (y^E - y), \qquad & s \ge 0, \\ s \cdot w = \mu^B (w^E - w) + \mu^B (s^E - s), \qquad & w \ge 0, \end{aligned}\right\} \tag{3.1}$$

where $y^E \in \mathbb{R}^m$ is an estimate of a Lagrange multiplier vector for the constraint $c(x) - s = 0$, $w^E \in \mathbb{R}^m$ is an estimate of a Lagrange multiplier for the constraint $s \ge 0$, $s^E \in \mathbb{R}^m$ is an estimate of the optimal slacks, and μ^P and μ^B are positive scalars. The last equation of (3.1) 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 function

$$F(x, s, y, w\,; s^E, y^E, w^E, \mu^P, \mu^B) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ c(x) - s + \mu^P (y - y^E) \\ s \cdot w - \mu^B (w^E - w + s^E - s) \end{pmatrix}, \tag{3.2}$$

then any point $(x,\ s,\ y,\ w)$ such that $F(x,\ s,\ y,\ w\ ;\ s^E,\ y^E,\ w^E,\ \mu^P,\ \mu^B) = 0$ must satisfy the perturbed optimality conditions (3.1). Let $F(v)$ denote the function at a given point $v = (x, s, y, w)$. The Newton equations for the step Δv are given by $F'(v)\Delta v = -F(v)$, i.e.,

$$\begin{pmatrix} H(x,y) & 0 & -J(x)^{\mathrm{T}} & 0 \\ 0 & 0 & I_m & -I_m \\ J(x) & -I_m & \mu^P I_m & 0 \\ 0 & W + \mu^B I_m & 0 & S + \mu^B I_m \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} y \\ y - w \\ c(x) - s + \mu^P (y - y^E) \\ s \cdot w - \mu^B (w^E - w + s^E - s) \end{pmatrix}. \tag{3.3}$$

We need 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 (3.3). Consider the shifted primal-dual penalty-barrier function

$$\left.\begin{aligned} M(x, s, y, w\,; s^E, y^E, w^E, \mu^P, \mu^B) = {} & \underbrace{f(x)}_{(A)} \underbrace{-(c(x) - s)^{\mathrm{T}} y^E}_{(B)} + \underbrace{\frac{1}{2\mu^P} \|c(x) - s\|^2}_{(C)} \\ & + \underbrace{\frac{1}{2\mu^P} \|c(x) - s + \mu^P (y - y^E)\|^2}_{(D)} \\ & \underbrace{-2 \sum_{i=1}^m \mu^B (w_i^E + s_i^E + \mu^B) \ln(s_i + \mu^B)}_{(E)} \\ & \underbrace{- \sum_{i=1}^m \mu^B (w_i^E + s_i^E + \mu^B) \ln(w_i + \mu^B)}_{(F)} \\ & + \underbrace{\sum_{i=1}^m w_i (s_i + \mu^B)}_{(G)} + \underbrace{2\mu^B \sum_{i=1}^m s_i}_{(H)}. \end{aligned}\right\} \tag{3.4}$$

Let S and W denote diagonal matrices with diagonal entries s_i and w_i such that $s_i + \mu^B > 0$ and $w_i + \mu^B > 0$, and let S^E denote the diagonal matrix with diagonal entries s_i^E. Similarly, let

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

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 ∇M may be written as

$$\nabla M = \begin{pmatrix} \nabla f(x) - J(x)^T\big(\pi^Y + (\pi^Y - y)\big) \\ (\pi^Y - y) + (\pi^Y - \pi^W) + (w - \pi^W) \\ -D_P(\pi^Y - y) \\ -D_B(\pi^W - w) \end{pmatrix}, \tag{3.5}$$

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

$$\begin{pmatrix} H + 2J(x)^T D_P^{-1} J(x) & -2J(x)^T D_P^{-1} & J(x)^T & 0 \\ -2D_P^{-1} J(x) & 2(D_P^{-1} + D_B^{-1} W_B^{-1} \mathit{\Pi}^W + \mu^B S_B^{-1}) & -I_m & I_m \\ J(x) & -I_m & D_P & 0 \\ 0 & I_m & 0 & D_B W_B^{-1} \mathit{\Pi}^W + \mu^B W_B^{-2} S_B \end{pmatrix}, \tag{3.6}$$

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

At the start of iteration k, given the 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

$$H_k^M \Delta v_k = -\nabla M(v_k), \tag{3.7}$$

where H_k^M is a positive-definite approximation of $\nabla^2 M(x_k,\, s_k,\, y_k,\, w_k)$. 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 (3.6). This gives

$$H^M = \begin{pmatrix} \widehat{H} + 2J(x)^T D_P^{-1} J(x) & -2J(x)^T D_P^{-1} & J(x)^T & 0 \\ -2D_P^{-1} J(x) & 2(D_P^{-1} + D_B^{-1}) & -I_m & I_m \\ J(x) & -I_m & D_P & 0 \\ 0 & I_m & 0 & D_B \end{pmatrix}, \tag{3.8}$$

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 [13] 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 approximate Newton equations (3.7) defined with H^M from (3.8) are not solved directly because of the potential for numerical instability. Instead, an *equivalent* transformed system is solved based on the transformation

$$UH^M \Delta v = -U\nabla M(v), \tag{3.9}$$

where U is a nonsingular matrix defined by

$$U = \begin{pmatrix} I_m & 0 & -2J(x)^T D_P^{-1} & 0 \\ 0 & I_m & 2D_P^{-1} & -2D_B^{-1} \\ 0 & 0 & I_m & 0 \\ 0 & 0 & 0 & W + \mu^B I_m \end{pmatrix}. \tag{3.10}$$

Upon multiplication and application of the identity $W_B D_B = S_B$, the equations (3.9) may be rewritten as

$$\begin{pmatrix} \widehat{H} & 0 & -J(x)^T & 0 \\ 0 & 0 & I_m & -I_m \\ J(x) & -I_m & D_P & 0 \\ 0 & W + \mu^B I_m & 0 & S + \mu^B I_m \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^T y \\ y - w \\ c(x) - s + \mu^P (y - y^E) \\ s \cdot w - \mu^B (w^E - w + s^E - s) \end{pmatrix}. \tag{3.11}$$

These equations are identical to the shifted path-following equations (3.3) when $\widehat{H} = H(x,y)$. The solution of (3.11) 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

$$\begin{pmatrix} \widehat{H} & J(x)^T \\ J(x) & -(D_P + D_B) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - J(x)^T y \\ D_P(y - \pi^Y) + D_B(y - \pi^W) \end{pmatrix}. \tag{3.12}$$

The matrix H^M in (3.8) 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 (3.12) 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).

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 (3.4) 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 [7]) that allows the acceptance of a step under a wider range of conditions. The generalization uses the idea of flexible line search proposed by Curtis and Nocedal [5], and also employs the relation 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 (3.2). In this case, we simplify the notation by writing $M(x, s, y, w\,; s^E, y^E, w^E, z^E, \mu^P, \mu^B)$ 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

$$\Omega_k = \big\{\, v : v \ge \min\{v_k - \sigma(v_k - \ell), 0\} \,\big\}, \tag{4.1}$$

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 $\mu^L \geq \mu^P$. At an iteration k, let $\psi_k(\alpha\,;\mu)$ and $\phi_k(\alpha\,;\mu)$ denote the piecewise-differentiable 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

$$\psi_k(\alpha_k\,;\mu^P) < \max\{\,\psi_k(0\,;\mu^P), M_{\max}\,\}, \tag{4.2a}$$

$$\psi_k(\alpha_k\,;\mu^L) < \max\{\,\psi_k(0\,;\mu^L), M_{\max}\,\}, \quad \text{and} \tag{4.2b}$$

$$\phi_k(\alpha_k\,;\mu^P) \leq \eta_F \min\{\,\phi_k(0\,;\mu^P), \eta_F^{m_k} F_{\max}\,\} \tag{4.2c}$$

are satisfied, or

$$\psi_k(\alpha_k\,;\mu_k^F) \leq \psi_k(0\,;\mu_k^F) + \alpha_k \eta_A \nabla M(v_k\,;\mu^P)^{\mathrm{T}} \Delta v_k, \tag{4.2d}$$

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 (4.2a)–(4.2c) were satisfied. Any α_k satisfying the conditions (4.2a)–(4.2c) or the condition (4.2d) is classified as a flexible quasi-Armijo step. Alternatively, an α_k that satisfies (4.2d) for $\mu_k^F = \mu^P$ is simply known as a quasi-Armijo step. The conditions (4.2a)–(4.2d) allow a step to be accepted if either (4.2a)–(4.2c) holds, which implies that α_k gives a sufficient decrease in the norm of the shifted path-following function F (3.2), or (4.2d) 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 (NIP) that uses Algorithm 1 as a basis for minimizing the underlying merit function. The slack-variable reset in Step 16 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 [14], and Gill, Kungurtsev and Robinson [13]. As defined in Step 15 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 15 and Step 16 of Algorithm 1 that the value of s_{k+1} computed in Step 16 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

$$c(x_{k+1}) - s_{k+1} \leq \mu_k^F\big(y^E + \tfrac{1}{2}(w_{k+1} - y_{k+1}) + \mu^B\big). \tag{4.3}$$

4.2. Convergence analysis

The following assumptions are made for the convergence analysis:

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

Assumption 4.2. *The sequence of matrices $\{H_k^M\}_{k\geq 0}$ used in (3.7) are chosen to be uniformly positive definite and bounded in norm.*

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, μ^P, μ^B, μ^L)
2:     Restrictions: s_0 + μ^B e > 0, w_0 + μ^B e > 0, s^E + w^E + μ^B e > 0, μ^L ≥ μ^P > 0, μ^B > 0;
3:     Constants: { η_A, γ_A, η_F } ∈ (0,1);
4:     Set v_0 ← (x_0, s_0, y_0, w_0);
5:     while ‖∇M(v_k)‖ > 0 do
6:         Choose H_k^M ≻ 0, and then compute the search direction Δv_k from (3.7);
7:         Set α_k ← 1;
8:         loop
9:             if (4.2a)–(4.2c) hold or (4.2d) holds for μ_k^F = μ^L or μ_k^F = μ^P then
10:                break;
11:            end if
12:            Set α_k ← γ_A α_k;
13:        end loop
14:        Set v_{k+1} ← proj_{Ω_k}(v_k + α_k Δv_k);
15:        Set ŝ_{k+1} ← c(x_{k+1}) − μ_k^F (y^E + ½(w_{k+1} − y_{k+1}) + μ^B);
16:        Perform a slack reset s_{k+1} ← max{s_{k+1}, ŝ_{k+1}};
17:        Set v_{k+1} ← (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1});
18:    end while
19: end procedure
```

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

If $M(v\,;\mu^F)$ is used to denote $M(x,\, s,\, y,\, w,\, z\,; s^E,\, y^E,\, w^E,\, z^E,\, \mu^F,\, \mu^B)$ with either $\mu^F = \mu^P$ or $\mu^F = \mu^L$, the first result shows that $M(v\,;\mu^F)$ is monotonically decreasing if μ^F is fixed.

Lemma 4.1. *Suppose that μ^F is fixed. The sequence of iterates $\{v_k\}$ computed by* Algorithm 1 *is such that $\{M(v_k\,;\mu^F)\}$ is bounded. In particular, if α_k is a step that satisfies* (4.2d)*, 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 $k \ge 0$, the vector Δv_k is a descent direction for M at v_k. This property, together with equations (4.2a) and (4.2b), 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 (4.2d) holds, then $M(v_{k+1}\,;\mu^F) < M(v_k\,;\mu^F)$. If follows that the only way the desired result can not hold is if the slack-reset procedure of Step 16 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 can not increase. Also, (A) 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 its components, which means that (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. *If μ^F is fixed, then 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*

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

(iii) *The sequences $\{\pi^{Y}(x_k, s_k)\}$, $\{\pi^{W}(s_k)\}$, and $\{\nabla M(v_k\,;\mu^{P})\}$ 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 (4.2c) 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

$$\lim_{k\in\mathcal{S}}\,[\,s_k\,]_i = \infty \;\text{ and }\; [\,s_k\,]_i \geq [\,s_k\,]_j \;\text{ for every } j \text{ and all } k\in\mathcal{S}. \tag{4.4}$$

Next it will be shown that M must go to infinity on $\mathcal{S}$. It follows from (4.4), Assumption 4.3, and the continuity of c that the term (A) in the definition of M is bounded below for all k, 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 previous argument proves that M converges to infinity on $\mathcal{S}$, which contradicts 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 contradicts Lemma 4.1. We have thus proved that $\{s_k\}$ is bounded, which is 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 prove the final bound in part (**i**), i.e., that $\{\,w_k\,\}$ is bounded. Using the boundedness of $\{\,x_k\,\}$, $\{\,s_k\,\}$ and $\{\,y_k\,\}$, we know that (A), (B), (C), (D) and (H) are bounded, (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 [13]. ∎

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* $k \in \mathcal{S}$.

(iii) *There exist a positive scalar* $\alpha_{\min}$ *such that, for all* $k \in \mathcal{S}$*, the condition* (4.2d) *is satisfied with* $\alpha_k \ge \alpha_{\min}$.

Proof. See the proof of Lemma 3.3 in [13]. ∎

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 (4.2a)–(4.2c) hold for all $k \in \mathcal{S}$. Then the line-search condition (4.2c) implies that $\lim_{k\in\mathcal{S}} \|F(v_k\,; \mu^P)\| = 0$. By (3.9), $F(v_k\,; \mu^P) = U_k \nabla M(v_k\,; \mu^P)$, where U_k is a matrix of the form (3.10). 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 $\mathcal{S}$ such that the line-search conditions (4.2a)–(4.2c) hold is finite. This implies that for all k sufficiently large, the line-search condition (4.2d) must hold. The proof that $\lim_{k\in\mathcal{S}} \nabla M(v_k\,; \mu^P) = 0$ is by contradiction. Suppose there exists a constant $\epsilon > 0$ and a subsequence $\mathcal{S}$ such that $\|\nabla M(v_k\,; \mu^P)\| \ge \epsilon$ for all $k \in \mathcal{S}$. 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 fact that the line-search condition (4.2d) 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 contradicts Lemma 4.3(**ii**) and Lemma 4.3(**iii**). ∎

5. Solving the Nonlinear Optimization Problem

In this section, a projected-search interior method for solving the nonlinear optimization problem (NIP) 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, which 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 (NIP). 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:

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

where

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

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

$$\chi(v, \mu) = \chi_{\text{feas}}(v) + \chi_{\text{stny}}(v) + \chi_{\text{comp}}(v, \mu). \tag{5.1}$$

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

$$\underset{v=(x,s,y,w)}{\text{minimize}} \ \ M(v\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B). \tag{5.2}$$

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

$$\|\nabla_x M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k, \tag{5.3a}$$

$$\|\nabla_s M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k, \tag{5.3b}$$

$$\|\nabla_y M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k \|D_{k+1}^P\|_\infty, \ \text{ and} \tag{5.3c}$$

$$\|\nabla_w M(v_{k+1}\,; s_k^E, y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k \|D_{k+1}^B\|_\infty, \tag{5.3d}$$

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 (5.2). The estimates s_{k+1}^E, y_{k+1}^E and w_{k+1}^E are defined by the safeguarded values

$$\left.\begin{aligned} s_{k+1}^E &= \min\big(\max(0, s_{k+1}), s_{\max} e\big), \\ y_{k+1}^E &= \max\big(-y_{\max} e, \min(y_{k+1}, y_{\max} e)\big), \\ w_{k+1}^E &= \min\big(w_{k+1}, w_{\max} e\big) \end{aligned}\right\} \tag{5.4}$$

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

$$\chi_{\text{feas}}(v_{k+1}) \le \tau_k \tag{5.5}$$

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

$$\chi_{\text{comp}}(v_{k+1}, \mu^B_k) \le \tau_k, \quad s_{k+1} \ge -\tau_k e, \quad \text{and} \quad w_{k+1} \ge -\tau_k e. \tag{5.6}$$

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 method.

1: **procedure pdProj**(x_0, s_0, y_0, w_0)
2: **Restrictions:** $s_0 > 0$ and $w_0 > 0$;
3: **Constants:** $\{\, \eta_A, \gamma_A \,\} \subset (0,1)$ and $\{\, y_{\max}, w_{\max}, s_{\max} \,\} \subset (0,\infty)$;
4: Choose w^E_0 and s^E_0 such that $w^E_0 + s^E_0 + \mu^B e > 0$;
5: Choose y^E_0; $\chi^{\max}_0 > 0$; $\{\, \mu^P_0, \mu^B_0 \,\} \subset (0,\infty)$; and $\mu^L_0 \ge \mu^P_0$;
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 6–17 of Algorithm 1;
10: **if** $\chi(v_{k+1}, \mu^B_k) \le \chi^{\max}_k$ **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 (5.3a)–(5.3d) **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 (5.4);
15: **if** $\chi_{\text{feas}}(v_{k+1}) \le \tau_k$ **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** $\chi_{\text{comp}}(v_{k+1}, \mu^B_k) \le \tau_k$, $s_{k+1} \ge -\tau_k e$ and $w_{k+1} \ge -\tau_k e$ **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 (5.7);
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 $s_i + \bar{\mu}^B \le 0$, a strategy is proposed in Section 5.4 of [13], 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

$$\mu_{k+1}^L = \begin{cases} \mu_k^L & \text{if } \psi_k(\alpha_k\,; \mu_k^L) \leq \psi_k(0\,; \mu_k^L) + \alpha_k \eta_A \delta_k \text{ and } \mu_{k+1}^P = \mu_k^P; \\ \max\left\{ \tfrac{1}{2}\mu_k^L, \mu_{k+1}^P \right\} & \text{otherwise.} \end{cases} \tag{5.7}$$

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 [13], which uses the properties of the *complementary approximate KKT (*CAKKT) condition proposed by Andreani, Martínez and Svaiter [2], 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

$$\{ \nabla f(x_j) - J(x_j)^{\mathrm{T}} u_j \} \to 0, \tag{5.8}$$

$$\{ u_j - z_j \} \to 0, \tag{5.9}$$

$$\{ z_j \} \geq 0, \quad \text{and} \tag{5.10}$$

$$\{ z_j \cdot s_j \} \to 0. \tag{5.11}$$

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

Theorem 5.1. (Andreani et al. [1, Theorem 4.3]) *If* (x^*, s^*) *is a* CAKKT point that satisfies CAKKT-regularity, then (x^*, s^*) is a first-order KKT point for (NIP).

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

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

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 $\{ \chi_{\text{feas}}(v_{k+1}) \}_{k\in\mathcal{K}} \to 0$, 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 $s^* \geq 0$ is proved componentwise. For any $1 \leq i \leq m$, define

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

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 $\{\,\chi_{\rm comp}(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \le \{\,\chi(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \le \{\,\chi^{\max}_k\,\}_{k\in\mathcal{K}} \to 0$ 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 $s^*_i = \lim_{\mathcal{K}\cap\mathcal{Q}_2}\, [s_{k+1}]_i = \lim_{\mathcal{K}\cap\mathcal{Q}_2}\, [s_{k+1}+\mu^B_k e]_i \ge 0$, where the second equality uses the limit $\{\,\mu^B_k\,\}_{k\in\mathcal{K}\cap\mathcal{Q}_2} \to 0$ 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

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

and

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

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, $\{\,\chi_{\rm comp}(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \to 0$ implies that $\liminf_{k\in\mathcal{K}}[\,w_{k+1}\,]_0 \ge 0$, 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 $\chi_{\rm comp}(v_{k+1},\mu^B_k)$, together with the limit $\{\,\chi_{\rm comp}(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \to 0$ imply that $\{\,\mu^B_k\,\} \to 0$, 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 $\{\,\chi(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \to 0$ established above that $\{\,\chi_{\rm stny}(v_{k+1})+\chi_{\rm comp}(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \le \{\,\chi(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \to 0$. 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 (5.8) and (5.9) hold. The nonnegativity of $\bar{w}_{k+1}$ for all k is obvious from its definition, which implies that (5.10) is satisfied for $\{\,w_k\,\}_{k\in\mathcal{K}}$. Finally, it must be shown that (5.11) 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, the definitions of $\bar{s}_{k+1}$, $q_1(v_{k+1})$ and $\chi_{\rm comp}(v_{k+1},\mu^B_k)$, together with the limit $\{\,\chi_{\rm comp}(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \to 0$ 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 $\bar{s}_{k+1}$, $q_2(v_{k+1},\mu^B_k)$ and $\chi_{\rm comp}(v_{k+1},\mu^B_k)$, together with the limits $\{\,\chi_{\rm comp}(v_{k+1},\mu^B_k)\,\}_{k\in\mathcal{K}} \to 0$ and $\{\,\bar{w}_{k+1}-w_{k+1}\,\}_{k\in\mathcal{K}} \to 0$ imply that $\{\,[\,\bar{w}_{k+1}\cdot\bar{s}_{k+1}\,]_i\,\}_{k\in\mathcal{K}\cap\mathcal{Q}_2} \to 0$. 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 (5.11) is satisfied. This concludes the proof that (x^*,s^*) is a CAKKT point. ∎

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 constraint $c(x)-s=0$ but solves the least-infeasibility problem

$$\underset{x,s}{\text{minimize}}\ \ \tfrac{1}{2}\|c(x)-s\|_2^2 \ \ \text{subject to}\ \ s\ge 0. \tag{5.12}$$

The first-order KKT conditions for problem (5.12) are

$$J(x^*)^{\rm T}\big(c(x^*)-s^*\big) = 0, \qquad s^*\ge 0, \tag{5.13a}$$

$$s^*\cdot\big(c(x^*)-s^*\big) = 0, \qquad c(x^*)-s^*\le 0. \tag{5.13b}$$

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* (5.13).

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

$$k \in \mathcal{F}, \;\; y_k^E = y^E, \;\; \text{and} \;\; (\tau_k, w_k^E, \mu_k^P, \mu_k^B, \mu_k^L) = (\tau, w^E, \mu^P, \mu^B, \mu^L) > 0 \tag{5.14}$$

for all $k \geq k_F$. The updating rule for $\{\, \mu_k^L \,\}$ implies that μ_k^F is fixed at the value $\max \{\, \mu^L, \mu^P \,\}$. 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 (5.3) 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 (5.14) because $\mathcal{F} \cap \mathcal{M} = \emptyset$. ∎

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

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

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

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

Proof. It follows from (3.5) and (5.3c) that

$$\|\pi_{k+1}^Y - y_{k+1}\| \leq \tau_k. \tag{5.15}$$

As $\|\mathcal{M}\| = \infty$ by assumption, Step 14 of Algorithm 2 implies that $\lim_{k \to \infty} \; \tau_k = 0$. Combining this with (5.15) establishes the first limit in the result.

Furthermore, if there exists a subsequence $\mathcal{K} \subset \mathcal{M}$ such that $\lim_{k \in \mathcal{K}} \; s_k = s^* \geq 0$, then the updating rule of Algorithm 2 for s_k^E implies that $\lim_{k \in \mathcal{K}} (s_k^E - s_k) = 0$. The limit $\lim_{k \to \infty} \; \tau_k = 0$ may then be combined with (3.5), (5.3b) and (5.3c) to show that

$$\lim_{k \in \mathcal{K}} \; \|\pi_{k+1}^W - w_{k+1}\| = 0 \quad \text{and} \quad \lim_{k \in \mathcal{K}} \; |\pi_{k+1}^Y - \pi_{k+1}^W| = 0. \tag{5.16}$$

Finally, as $\lim_{k \to \infty} \; \tau_k = 0$, it follows the bound (5.15) and limits (5.16) that

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

This establishes the last of the four limits. ∎

Lemma 5.4. *If* $|\mathcal{O}| < \infty$, *then every limit point* (x^*, s^*) *of the 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 [13] 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

$$\lim_{k\in\mathcal{K}} \|c(x_{k+1}) - s_{k+1}\| = 0. \tag{5.17}$$

First, we show that $s^* \ge 0$, which will imply that $(x^*,\ s^*)$ is feasible because of the assumption that $c(x^*) - s^* = 0$. The line search in Algorithm 1 gives $s_{k+1} + \mu_k^B e > 0$ for all k. If $\lim_{k\to\infty}\ \mu_k^B = 0$, then $s^* = \lim_{k\in\mathcal{K}}\ s_{k+1} \ge -\lim_{k\in\mathcal{K}} \mu_k^B e = 0$. On the other hand, if $\lim_{k\to\infty}\ \mu_k^B \ne 0$, then Step 19 of Algorithm 2 is executed a finite number of times, $\mu_k^B = \mu^B > 0$ and (5.6) holds for all $k \in \mathcal{M}$ sufficiently large. Taking limits over $k \in \mathcal{M}$ in (5.6) and using $\lim_{k\to\infty}\ \tau_k = 0$ gives $s^* \ge 0$.

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

$$\chi_k^{\max} = \chi^{\max} > 0 \ \text{ for all sufficiently large } k \in \mathcal{K}. \tag{5.18}$$

Using $|\mathcal{O}| < \infty$ together with Lemma 5.3, the fact that $\lim_{k\in\mathcal{K}}\ s_k = s^* \ge 0$, and Step 14 of the line search of Algorithm 1 gives

$$\lim_{k\in\mathcal{K}} \|y_{k+1} - w_{k+1}\| = 0, \ \text{ and } \ w_{k+1} + \mu_{k+1}^B > 0 \ \text{ for all } k \ge 0. \tag{5.19}$$

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

$$\lim_{k\in\mathcal{K}} \ \big\{\, \nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} y_{k+1} \,\big\} = 0. \tag{5.20}$$

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

Case 1: $\lim_{k\to\infty} \mu_k^B \ne 0$. In this case $\mu_k^B = \mu^B > 0$ for all sufficiently large k. Combining this with the limit $|\mathcal{M}| = \infty$ and the update to τ_k in Step 19 of Algorithm 2, it must be that (5.6) holds for all sufficiently large $k \in \mathcal{K}$, i.e., that $\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \le \tau_k$ 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 that

$$0 = \lim_{k\in\mathcal{K}} (S_{k+1} + \mu_k^B I)(\pi_{k+1}^W - w_{k+1}) = \lim_{k\in\mathcal{K}} \ \big(\mu_k^B w_k^E - (S_{k+1} + \mu_k^B I) w_{k+1}\big). \tag{5.21}$$

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 (5.4)). It then follows from (5.21), the uniform boundedness of $\{\, w_k^E \,\}$, and $\lim_{k\to\infty} \mu_k^B = 0$ that

$$0 = \lim_{k\in\mathcal{K}} \ \big([s_{k+1}]_i + \mu_k^B\big)[w_{k+1}]_i = \lim_{k\in\mathcal{K}} \ \big([s_{k+1}]_i + \mu_k^B\big)\big([w_{k+1}]_i + \mu_k^B\big). \tag{5.22}$$

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 (5.22) 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 $\lim_{k\to\infty} \mu_k^B = 0$ and (5.22), $w_{k+1} + \mu_k^B > 0$ and the limit $\lim_{k\in\mathcal{K}}[\, s_{k+1}\,]_i = s_i^* = 0$ that $\lim_{k\in\mathcal{K}}[q_2(v_{k+1}, \mu_k^B)]_i = 0$, which is the desired result for this case.

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

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

which completes the proof for Case 2.

Under the assumption $c(x^*) - s^* = 0$ it has been shown that (5.17), (5.19), (5.20), and the limit $\lim_{k\in\mathcal{K}}\ \chi_{\text{comp}}(v_{k+1}, \mu_k^B) = 0$ hold. Collectively, these results imply that $\lim_{k\in\mathcal{K}} \chi(v_{k+1}, \mu_k^B) = 0$. This limit, together with the inequality (5.18) 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 [13] 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 $\{\, y_k^E \,\}$ and $\{\, w_k^E \,\}$ to be bounded (see (5.4)). 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 $\{\, [\, \pi_{k+1}^Y\,]_i \,\}_{k\in\mathcal{K}}$ is unbounded. On the other hand, by (3.5), (5.3a), together with the limit $\lim_{k\to\infty} \tau_k = 0$ and Lemma 5.3,

$$\begin{aligned}
0 &= \lim_{k\in\mathcal{M}} \|\nabla_x M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\| \\
&= \lim_{k\in\mathcal{M}} \|\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} \pi_{k+1}^Y - J(x_{k+1})^{\mathrm{T}} (\pi_{k+1}^Y - y_{k+1})\| \\
&= \lim_{k\in\mathcal{M}} \|\nabla f(x_{k+1}) - J(x_{k+1})^{\mathrm{T}} \pi_{k+1}^Y\| = 0,
\end{aligned}$$

which contradicts the unboundedness of $\{\, [\, \pi_{k+1}^Y\,]_i \,\}_{k\in\mathcal{K}}$. 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 (5.13a). The test in Step 16 of Algorithm 2 (i.e., testing whether (5.6) holds) is checked infinitely often because $|\mathcal{M}| = \infty$. If (5.6) is satisfied finitely many times, then the update $\mu_{k+1}^B = \frac{1}{2}\mu_k^B$ forces $\{\, \mu_{k+1}^B \,\} \to 0$. Combining this with $s_{k+1} + \mu_k^B e > 0$ shows that $s^* \geq 0$, as claimed. On the other hand, if (5.6) is satisfied for all sufficiently large $k \in \mathcal{M}$, then $\mu_{k+1}^B = \mu^B > 0$ 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 $s^* \geq 0$, 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 (5.13a)–(5.13b)).

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 (5.5) 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 (4.3), the boundedness of $\{ y_k^E \}$, and Lemma 5.3, gives

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

This implies that $c(x^*) - s^* \le 0$ and the second condition in (5.13b) holds.

The rest of the proof is the same as in the proof of Lemma 4.8 in [13] ∎

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 (NIP). If, in addition, CAKKT-regularity holds at (x^*, s^*), then (x^*, s^*) is a KKT point for problem (NIP).

(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

6.1. Implementation Details

Numerical results were obtained for MATLAB implementations of three variants of the shifted interior-point method. Algorithm `pdb` is an implementation of the shifted primal-dual method of Gill, Kungurtsev and Robinson [13]; `pdbAll` is the primal-dual method with shifts on both the primal and dual variables; and `pdProj` is the projected-search interior method proposed in Sections 3 and 4. Algorithms `pdb` and `pdbAll` are implemented with a flexible Armijo line search in which the step length is chosen to satisfy the conditions (4.2a)–(4.2d) with $\psi_k(\alpha\,;\mu)$ and $\phi_k(\alpha\,;\mu)$ given by $M\big(v_k+\alpha\Delta v_k\,;\mu\big)$ and $\|F\big(v_k+\alpha\Delta v_k\,;\mu\big)\|$. Exact second derivatives were used for all the runs.

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

$$e_P(x,s) = \left\| \begin{pmatrix} \min\{\,0,\,s\,\} \\ \|c(x)-s\|_\infty / \max\{\,1,\,\|s\|_\infty\,\} \end{pmatrix} \right\|_\infty, \tag{6.1a}$$

and

$$e_D(x,s,y,w) = \left\| \begin{pmatrix} \|\nabla f(x) - J(x)^T y\|_\infty/\sigma \\ \|w-y\|_\infty \\ w\cdot\min\{\,1,\,s\,\} \end{pmatrix} \right\|_\infty, \tag{6.1b}$$

with $\sigma = \max\{\, 1,\ \|\nabla f(x)\|,\ \max\{\, 1,\ \|y\|\}\|J(x)\|_\infty \,\}$. 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 $e_I(x, s) \le \tau_{\mathrm{inf}}$, where

$$e_I(x, s) = \left\| J(x)^{\mathrm{T}}(c(x) - s) \cdot \min\{\, 1,\ s \,\} \right\|_\infty / \sigma. \tag{6.2}$$

6.2. Numerical results

The results were obtained for problems from the CUTEst test collection (see Bongartz et al. [3] and Gould, Orban and Toint [17]). The runs were done using MATLAB version R2020a on an iMac with a 3.0 GHz Intel Zeon W processor and 128 GB of 800 MHz DDR4 RAM running macOS, version 10.14.6 (64 bit). Results were obtained for five subsets of problems from the CUTEst test collection. The subsets consisted of 135 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); 124 problems formulated by Hock and Schittkowski ([19]) (problems HS); 372 problems with a general nonlinear objective, general linear and nonlinear constraints and bounds on the variables (problems NC); and 117 problems with a quadratic objective, general linear constraints and bounds on the variables (problems QP). 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 1000 or less. The same criterion was used to set the dimension of those problems for which the problem size can be specified. The nonsmooth problem `hs87` was excluded from the Hock-Schittkowski problems. Exact second derivatives were used for all the runs.

Each CUTEst problem may be written in the form

$$\underset{x}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; \begin{pmatrix} \ell^x \\ \ell^s \end{pmatrix} \le \begin{pmatrix} x \\ c(x) \end{pmatrix} \le \begin{pmatrix} u^x \\ u^s \end{pmatrix}, \tag{6.3}$$

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. In this format, a fixed variable or an equality constraint has the same value for its upper and lower bound. 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 (6.3) are derived by Gill and Zhang [16]. As is the case for problem (NIP) the principal work at each iteration is the solution of a reduced $(n+m)\times(n+m)$ KKT system analogous to (3.12). Each KKT matrix was factored using the MATLAB built-in command `LDL`, which uses the routine MA57 [6]. If this 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 [22, Algorithm IC, p. 36], which factors the KKT matrix with δI_n added to H. At any given iteration the δ is increased from zero if necessary until the inertia of the KKT matrix is correct.

All three MATLAB implementations were initialized with identical parameter values that were chosen based on the empirical performance on the entire collection of problems. A summary of the values is given in Table 1. 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. The iterates were terminated at the first point that satisfied the conditions (6.1a)–(6.1b) and (6.2) defined in terms of the constraints associated with problem (6.3).

Tables 1–5 give the performance profiles. The results 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.

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 for Algorithm 2	1.0e-4
μ_0^L	Initial flexible line-search penalty parameter for Algorithm 2	1.0
μ_0^B	Initial barrier parameter for Algorithm 2	1.0e-4
τ_0	Initial termination tolerance for specifying an M-iterate	0.5
τ_P	Primal feasibility tolerance (6.1a)	1.0e-4
τ_D	Dual feasibility tolerance (6.1b)	1.0e-4
$\tau_{\inf}$	Infeasible stationary point tolerance (6.2)	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 $\|F\|$	1.0e-2
γ_A	Line-search factor for reducing an Armijo step	1.0e-3
f_{unb}	Unbounded objective	1.0e-9
$M_{\max}$	Constants in line-search tolerance (4.2a) and (4.2b)	1.0e+12
$F_{\max}$	Constant in the line-search tolerance (4.2c)	1.0e+8
σ	Bound perturbation in the definition of Ω_k (4.1)	0.8
$k_{\max}$	Iteration limit for Algorithm 2	500

References

[1] R. Andreani, J. M. Martínez, A. Ramos, and P. J. S. Silva. Strict constraint qualifications and sequential optimality conditions for constrained optimization. *Math. Oper. Res.*, 43(3):693–717, 2018. 14

[2] R. Andreani, J. M. Martínez, and B. F. Svaiter. A new sequential optimality condition for constrained optimization and algorithmic consequences. *SIAM J. Optim.*, 20(6):3533–3554, 2010. 2, 4, 14

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software*, 21(1):123–160, 1995. 20

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

[5] F. E. Curtis and J. Nocedal. Flexible penalty functions for nonlinear constrained optimization. *IMA J. Numer. Anal.*, 28(4):749–769, 2008. 7

[6] I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Software*, 30(2):118–144, 2004. 20

[7] M. W. Ferry, P. E. Gill, E. Wong, and M. Zhang. A class of projected-search methods for bound-constrained optimization. Center for Computational Mathematics Report CCoM 20-07, Center for Computational Mathematics, University of California, San Diego, La Jolla, CA, 2020. 7

[8] A. V. Fiacco. Barrier methods for nonlinear programming. In A. Holzman, editor, *Operations Research Support Methodology*, pages 377–440, New York, 1979. Marcel Dekker. 3

[9] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, Inc., New York, 1968. 3

[10] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.*, 8:1132–1152, 1998. 4

[11] K. R. Frisch. The logarithmic potential method of convex programming. Memorandum of May 13, University Institute of Economics, Oslo, Norway, 1955. 3

[12] E. M. Gertz and P. E. Gill. A primal-dual trust-region algorithm for nonlinear programming. *Math. Program., Ser. B*, 100:49–94, 2004. 4

[13] P. E. Gill, V. Kungurtsev, and D. P. Robinson. A shifted primal-dual penalty-barrier method for nonlinear optimization. *SIAM J. Optim.*, 30(2):1067–1093, 2020. 2, 4, 6, 8, 11, 13, 14, 17, 18, 19

[14] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM J. Optim.*, 12(4):979–1006, 2002. 8

[15] P. E. Gill and D. P. Robinson. A primal-dual augmented Lagrangian. *Comput. Optim. Appl.*, 51:1–25, 2012. 2

[16] P. E. Gill and M. Zhang. 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, 20

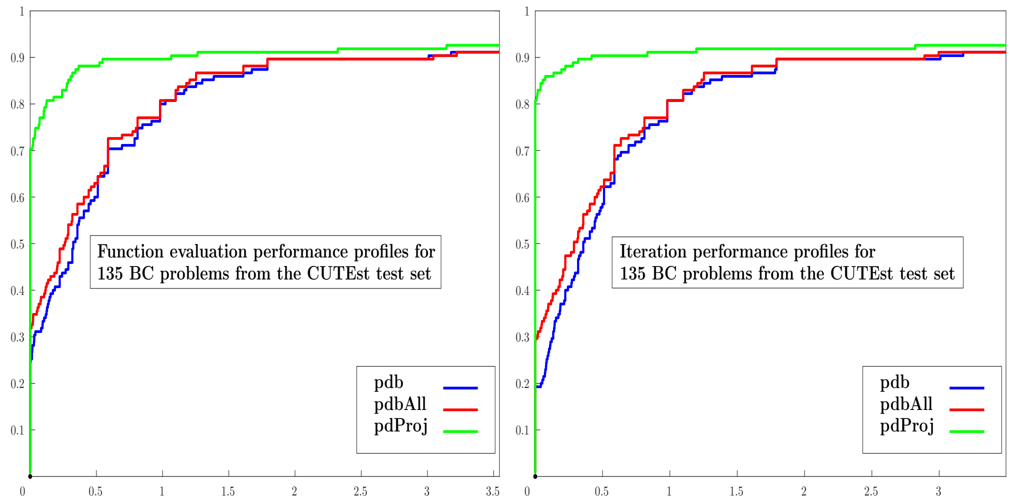

Figure 1: Performance profiles for the primal-dual interior algorithms `pdb`, `pdbAll` and `pdProj` applied to 135 bound-constrained (BC) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

[17] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29(4):373–394, 2003. 20

[18] M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969. 2

[19] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Lecture Notes in Econom. Math. Syst. 187. Springer-Verlag, Berlin, 1981. 20

[20] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298, London and New York, 1969. Academic Press. 2

[21] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM J. Optim.*, 16(1):1–31 (electronic), 2005. 4

[22] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1, Ser. A):25–57, 2006. 20

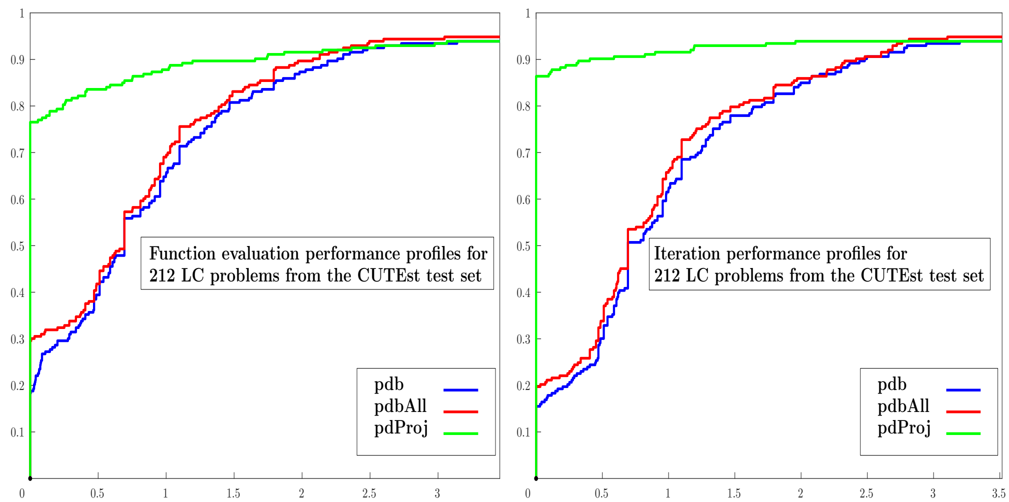

Figure 2: Performance profiles for the primal-dual interior algorithms `pdb`, `pdbAll` and `pdProj` applied to 212 linearly constrained (LC) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

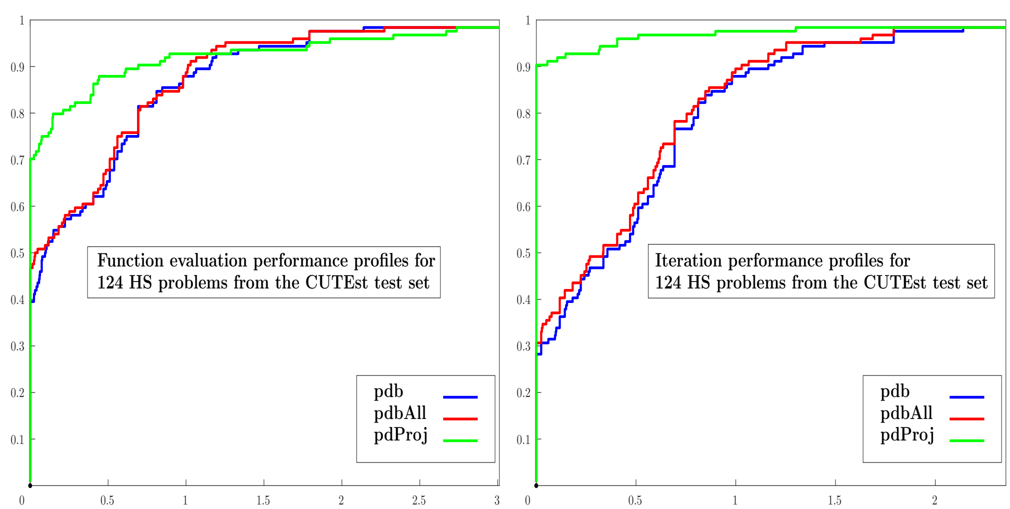

Figure 3: Performance profiles for the primal-dual interior algorithms `pdb`, `pdbAll` and `pdProj` applied to 124 Hock-Schittkowski (HS) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

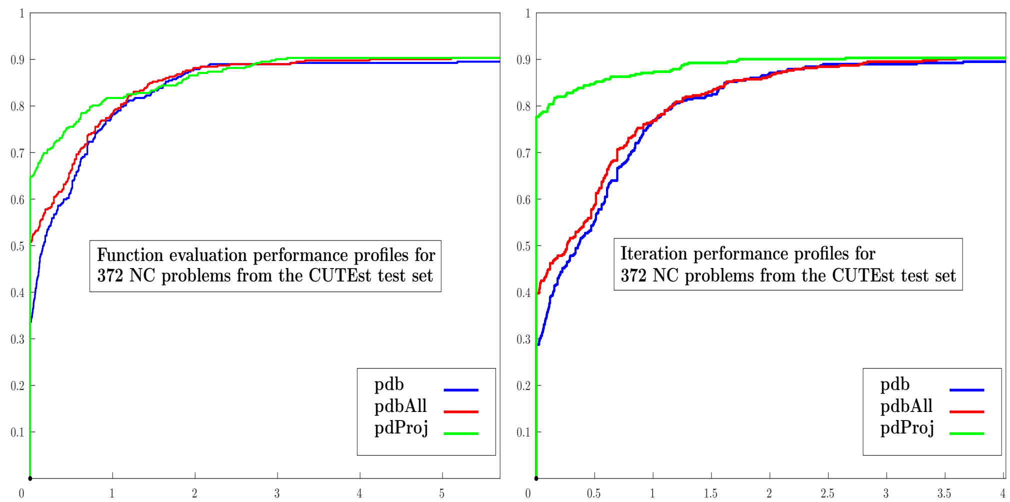

Figure 4: Performance profiles for the primal-dual interior algorithms **pdb**, **pdbAll** and **pdProj** applied to 372 nonlinearly constrained (NC) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.

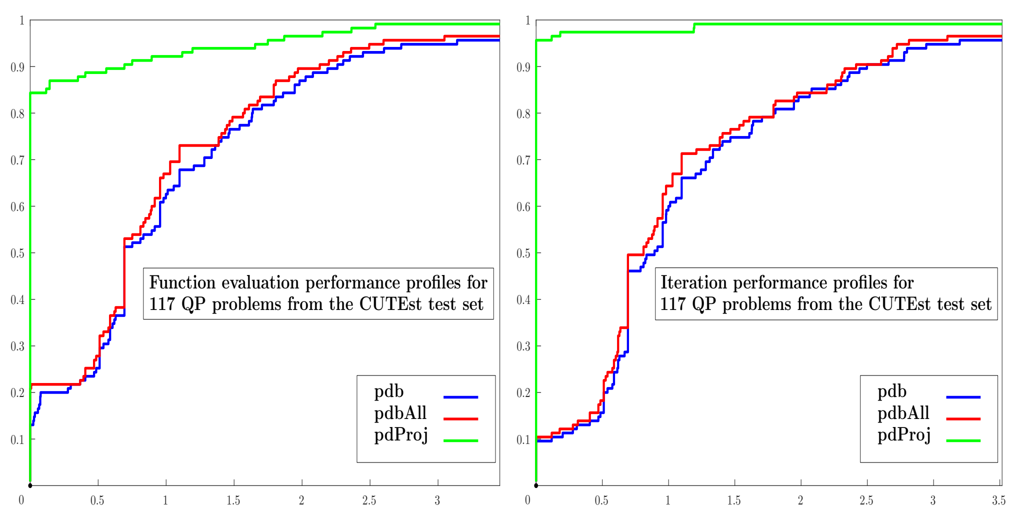

Figure 5: Performance profiles for the primal-dual interior algorithms **pdb**, **pdbAll** and **pdProj** applied to 117 quadratic programming (QP) problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of iterations.