Industrial and
Systems Engineering

A SHIFTED PRIMAL-DUAL PENALTY-BARRIER METHOD FOR NONLINEAR OPTIMIZATION

Philip E. Gill[1], Vyacheslav Kungurtsev[2], and Daniel P. Robinson[3]

[1]Department of Mathematics, University of California, San Diego

[2]Agent Technology Center, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University

[3]Department of Industrial and Systems Engineering, Lehigh University

ISE Technical Report 19T-024

LEHIGH
UNIVERSITY

A SHIFTED PRIMAL-DUAL PENALTY-BARRIER METHOD FOR NONLINEAR OPTIMIZATION

Philip E. Gill[*] Vyacheslav Kungurtsev[†] Daniel P. Robinson[‡]

Lehigh University
Industrial and Systems Engineering
March 1, 2019

Abstract

In nonlinearly constrained optimization, penalty methods provide an effective strategy for handling equality constraints, while barrier methods provide an effective approach for the treatment of inequality constraints. A new algorithm for nonlinear optimization is proposed based on minimizing a shifted primal-dual penalty-barrier function. Certain global convergence properties are established. In particular, it is shown that a limit point of the sequence of iterates may always be found that is either an *infeasible stationary point* or a *complementary approximate Karush-Kuhn-Tucker point*, i.e., it satisfies reasonable stopping criteria and is a Karush-Kuhn-Tucker point under a regularity condition that is the weakest constraint qualification associated with sequential optimality conditions. It is also shown that under suitable additional assumptions, the method is equivalent to a shifted variant of the primal-dual path-following method in the neighborhood of a solution. Numerical examples are provided that illustrate the performance of the method compared to a widely-used conventional interior-point method.

Key words. nonlinear optimization, augmented Lagrangian methods, barrier methods, interior methods, path-following methods, regularized methods, primal-dual 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). Research supported in part by National Science Foundation grants DMS-1318480 and DMS-1361421.

[†] Agent Technology Center, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague. (vyacheslav.kungurtsev@fel.cvut.cz) Research supported by the OP VVV project CZ.02.1.01/0.0/0.0/16 019/0000765 “Research Center for Informatics”.

[‡] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (daniel.p.robinson@gmail.com).

1. Introduction

This paper presents a new primal-dual shifted penalty-barrier method for solving nonlinear optimization problems of the form

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

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $f : \mathbb{R}^n \mapsto \mathbb{R}$ are twice-continuously differentiable. Barrier methods are a class of methods for solving (NIP) that involve the minimization of a sequence of unconstrained barrier functions parameterized by a scalar barrier parameter μ (see, e.g., Frisch [18], Fiacco and McCormick [13], and Fiacco [12]). Each barrier function includes a logarithmic barrier term that creates a positive singularity at the boundary of the feasible region and enforces strict feasibility of the barrier function minimizers. Reducing μ to zero has the effect of allowing the barrier minimizers to approach a solution of (NIP) from the interior of the feasible region. However, as the barrier parameter decreases and the values of the constraints that are active at the solution approach zero, the linear equations associated with solving each barrier subproblem become increasingly ill-conditioned. Shifted barrier functions were introduced to avoid this ill-conditioning by implicitly shifting the constraint boundary so that the barrier minimizers approach a solution without the need for the barrier parameter to go to zero. This idea was first proposed in the context of penalty-function methods by Powell [35] and extended to barrier methods for linear programming by Gill et al. [22] (see also Freund [17]). Shifted barrier functions are defined in terms of Lagrange multiplier estimates and are analogous to augmented Lagrangian methods for equality constrained optimization. The advantages of an augmented Lagrangian function over the quadratic penalty function for equality-constrained optimization motivated the class of modified barrier methods, which were proposed independently for nonlinear optimization by Polyak [34]. Additional theoretical developments and numerical results were given by Jensen and Polyak [30], and Nash, Polyak and Sofer [32]. Conn, Gould and Toint [7, 8] generalized the modified barrier function by exploiting the close connection between shifted and modified barrier methods. Optimization problems with a mixture of equality and inequality constraints may be solved by combining a penalty or augmented Lagrangian method with a shifted/modified barrier method. In this context, a number of authors have proposed the use of an augmented Lagrangian method, see e.g., Conn, Gould and Toint [7, 8], Breitfeld and Shanno [4, 5], and Goldfarb, Polyak, Scheinberg and Yuzefovich [26].

It is well-known that conventional barrier methods are closely related to path-following interior methods (for a survey, see, e.g., Forsgren, Gill and Wright [16]). If $x(\mu)$ denotes a local minimizer of the barrier function with parameter μ, then under mild assumptions on f and c, $x(\mu)$ lies on a continuous path that approaches a solution of (NIP) from the interior of the feasible region as μ goes to zero. Points on this path satisfy a system of nonlinear equations that may be interpreted as a set of perturbed first-order optimality conditions for (NIP). Solving these equations using Newton's method provides an alternative to solving the ill-conditioned equations associated with a conventional barrier method. In this context, the 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 [6], Wächter and Biegler [37], Forsgren and Gill [15], and Gertz and Gill [19].

An important property of the path-following approach is that the barrier parameter μ serves an auxiliary role as an implicit regularization parameter in the Newton equations. This regularization plays a crucial role in the robustness of interior methods on ill-conditioned and ill-posed problems.

1.1. Contributions and organization of the paper

Several contributions are made to advance the state-of-the-art in the design of algorithms for nonlinear optimization. (i) A new shifted primal-dual penalty-barrier function is formulated and analyzed. (ii) An algorithm is proposed based on using the penalty-barrier function as a merit function for a primal-dual path-following method. It is shown that a specific modified Newton method for the unconstrained minimization of the shifted primal-dual penalty-barrier function generates search directions identical to those associated with a shifted variant of the conventional path-following method. (iii) Under mild assumptions (e.g., no Kurdyka-Łojasiewicz type assumption is needed), 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*. This regularity condition is the weakest constraint qualification associated with sequential optimality conditions. (iv) The method maintains the positivity of certain variables, but it does not require a *fraction-to-the-boundary rule*, which differentiates it from most other interior-point methods in the literature. (v) Shifted barrier methods have the disadvantage that a reduction in the shift necessary to ensure convergence may cause an iterate to become infeasible with respect to a shifted constraint. In the proposed method, infeasible shifts are returned to feasibility without any increase in the cost of an iteration.

The paper is organized in seven sections. The proposed primal-dual penalty-barrier function is introduced in Section 2. In Section 3, a line-search algorithm is presented for minimizing the shifted primal-dual penalty-barrier function for fixed penalty and barrier parameters. The convergence of this algorithm is established under certain assumptions. In Section 4, an algorithm for solving problem (NIP) is proposed that builds upon the work from Section 3. Global convergence results are also established. Section 5 focuses on the properties of a single iteration and the computation of the primal-dual search direction. In particular, it is shown that the computed direction is equivalent to the Newton step associated with a shifted variant of the conventional primal-dual path-following equations. In Section 6 an implementation of the method is discussed, as well as some numerical examples that illustrate the performance of the method. Finally, Section 7 gives some conclusions and topics for further work.

1.2. 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, $\min(a, b)$ is 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 vector $g(x)$ is used to denote $\nabla f(x)$, the gradient of $f(x)$. The matrix $J(x)$ denotes the $m \times n$ constraint Jacobian, which has ith row $\nabla c_i(x)^T$. The Lagrangian function associated with (NIP) is $L(x, y) = f(x) - c(x)^T y$, 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. A Shifted Primal-Dual Penalty-Barrier Function

In order to avoid the need to find a strictly feasible point for the constraints of (NIP), each inequality $c_i(x) \ge 0$ is written in terms of an equality and nonnegative slack variable $c_i(x) - s_i = 0$ and $s_i \ge 0$. This gives the equivalent problem

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

The vector (x^*, s^*, y^*, w^*) is called a first-order KKT point for problem (NIPs) when

$$c(x^*) - s^* = 0, \qquad s^* \ge 0, \tag{2.1a}$$

$$g(x^*) - J(x^*)^T y^* = 0, \qquad y^* - w^* = 0, \tag{2.1b}$$

$$s^* \cdot w^* = 0, \qquad w^* \ge 0. \tag{2.1c}$$

The vectors y^* and w^* constitute the Lagrange multiplier vectors for, respectively, the equality constraint $c(x) - s = 0$ and non-negativity constraint $s \ge 0$. The vector (x_k, s_k, y_k, w_k) will be used to denote the kth primal-dual iterate computed by the proposed algorithm, with the aim of giving limit points of $\big\{(x_k, s_k, y_k, w_k)\big\}_{k=0}^\infty$ that are first-order KKT points for problem (NIPs), i.e., limit points that satisfy (2.1).

An important concept related to the design of efficient algorithms for computing first-order KKT points for problem (NIPs) is that of perturbed optimality conditions.

An appropriate set of perturbed conditions for (2.1) is given by

$$\begin{aligned} g(x) - J(x)^T y &= 0, & y - w &= 0,\\ c(x) - s &= \mu^P (y^E - y), & s &\geq 0,\\ s \cdot w &= \mu^B (w^E - w), & w &\geq 0, \end{aligned} \tag{2.2}$$

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 $s \geq 0$, and the scalars μ^P and μ^B are positive penalty and barrier parameters, respectively. (The interpretation of μ^P and μ^B as penalty and barrier parameters is discussed below.) In the neighborhood of a first-order KKT point it is well-known that 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. At the same time, to ensure convergence to a first-order KKT point from an arbitrary starting point, an algorithm must include a strategy for deciding when one iterate is preferable to another. These considerations motivate the formulation of the new shifted primal-dual penalty-barrier function

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

It is shown in Section 5.3 that in the neighborhood of a minimizer of (NIPs) satisfying certain second-order optimality conditions, the Newton equations for a zero of the perturbed optimality conditions (2.2) are equivalent to the Newton equations for a minimizer of M. Also, it is shown in Section 3 that if the parameters y^E, w^E, μ^P, and μ^B are updated appropriately, then stationary points of M have properties that may be used in the formulation of a globally convergent algorithm for (NIPs).

Let S and W denote diagonal matrices with diagonal entries s and w (i.e., $S = \operatorname{diag}(s)$ and $W = \operatorname{diag}(w)$) such that $s_i + \mu^B > 0$ and $w_i > 0$. Define the positive-definite matrices

$$D_P = \mu^P I \quad \text{and} \quad D_B = (S + \mu^B I)W^{-1},$$

and auxiliary vectors

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

Then $\nabla M(x, s, y, w\,; y^E, w^E, \mu^P, \mu^B)$ may be written as

$$\nabla M = \begin{pmatrix} g - J^T\big(\pi^Y + (\pi^Y - y)\big) \\ \big(\pi^Y - y\big) + \big(\pi^Y - \pi^W\big) + \big(w - \pi^W\big)\big) \\ -D_P(\pi^Y - y) \\ -D_B(\pi^W - w) \end{pmatrix}, \tag{2.3}$$

with $g = g(x)$ and $J = J(x)$. The purpose of writing the gradient ∇M in this form is to highlight the quantities $\pi^Y - y$ and $\pi^W - w$, which are important in the analysis. Similarly, the penalty-barrier function Hessian $\nabla^2 M(x, s, y, w\,; y^E, w^E, \mu^P, \mu^B)$ is written in the form

$$\nabla^2 M = \begin{pmatrix} H + 2J^T D_P^{-1} J & -2J^T D_P^{-1} & J^T & 0 \\ -2D_P^{-1} J & 2(D_P^{-1} + D_B^{-1} W^{-1} \mathit{\Pi}^W) & -I & I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B W^{-1} \mathit{\Pi}^W \end{pmatrix}, \tag{2.4}$$

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

In developing algorithms, the goal is to achieve rapid convergence to a solution of (NIPs) without the need for μ^P and μ^B to go to zero. The underlying mechanism for ensuring convergence is the minimization of M for fixed parameters. A suitable line-search method is proposed in the next section.

3. Minimizing the Shifted Primal-Dual Penalty-Barrier Function

This section concerns the minimization of M for fixed parameters y^E, w^E, μ^P and μ^B. In this case the notation can be simplified by omitting the reference to y^E, w^E, μ^P and μ^B when writing M, ∇M and $\nabla^2 M$.

3.1. The algorithm

The method for minimizing M with fixed parameters is given as Algorithm 1. 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 system of equations

$$H_k^M \Delta v_k = -\nabla M(v_k), \tag{3.1}$$

where H_k^M is a positive-definite approximation of the matrix $\nabla^2 M(x_k, s_k, y_k, w_k)$. (The definition of H_k^M and the properties of the equations (3.1) are discussed in Section 5.) Once Δv_k has been computed, a line search is used to compute a step length α_k, such that the next iterate $v_{k+1} = v_k + \alpha_k \Delta v_k$ sufficiently decreases the function M and keeps important quantities positive (see Steps 7–12 of Algorithm 1).

The analysis of subsection 3.2 below establishes that under typical assumptions, limit points $(x^*,\ s^*,\ y^*,\ w^*)$ of the sequence $\big\{(x_k,\ s_k,\ y_k,\ w_k)\big\}_{k=0}^{\infty}$ generated by minimizing M for fixed y^E, w^E, μ^P, and μ^B satisfy $\nabla M(x^*, s^*, y^*, w^*) = 0$. However, the ultimate purpose is to use Algorithm 1 as the basis of a practical algorithm for the solution of problem (NIPs). The slack-variable reset used in Step 14 of Algorithm 1 plays a crucial role in the properties of this more general algorithm (an analogous slack-variable reset is used in Gill et al. [21]). The specific update can be motivated by noting that $\widehat{s}_{k+1}$, as defined in Step 13 of Algorithm 1, is the unique minimizer, with respect to s, of the sum of the terms (B), (C), (D), and (G) in the definition of the function M. In particular, it follows from Step 13 and Step 14 of Algorithm 1 that the value of s_{k+1} computed in Step 14 satisfies

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

which implies, after rearrangement, that

$$c(x_{k+1}) - s_{k+1} \leq \mu^P\big(y^E + \tfrac{1}{2}(w_{k+1} - y_{k+1})\big). \tag{3.2}$$

This inequality is crucial below when μ^P and y^E are modified. In this situation, the inequality (3.2) ensures that any limit point (x^*, s^*) of the sequence $\{(x_k, s_k)\}$ satisfies $c(x^*) - s^* \leq 0$ if y^E and $w_{k+1} - y_{k+1}$ are bounded and μ^P converges to zero. This is necessary to handle problems that are (locally) infeasible, which is a challenge for all methods for nonconvex optimization. The slack update never causes M to increase, which implies that M decreases monotonically (see Lemma 3.1).

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

```
 1: procedure MERIT($x_0, s_0, y_0, w_0$)
 2:     Restrictions: $s_0 + \mu^B e > 0$, $w_0 > 0$ and $w^E > 0$;
 3:     Constants: $\{\eta, \gamma\} \in (0,1)$;
 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 $\Delta v_k$ from (3.1);
 7:         Set $\alpha_k \leftarrow 1$;
 8:         loop
 9:             if $s_k + \alpha_k \Delta s_k + \mu^B e > 0$ and $w_k + \alpha_k \Delta w_k > 0$ then
10:                 if $M(v_k + \alpha_k \Delta v_k) \leq M(v_k) + \eta \alpha_k \nabla M(v_k)^T \Delta v_k$ then break;
11:             Set $\alpha_k \leftarrow \gamma \alpha_k$;
12:         Set $v_{k+1} \leftarrow v_k + \alpha_k \Delta v_k$;
13:         Set $\widehat{s}_{k+1} \leftarrow c(x_{k+1}) - \mu^P\big(y^E + \frac{1}{2}(w_{k+1} - y_{k+1})\big)$;
14:         Perform a slack reset $s_{k+1} \leftarrow \max\{s_{k+1}, \widehat{s}_{k+1}\}$;
15:         Set $v_{k+1} \leftarrow (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$;
```

3.2. Convergence analysis

The convergence analysis of Algorithm 1 requires assumptions on the differentiability of f and c, the properties of the positive-definite matrix sequence $\{H_k^M\}$ in (3.1), and the sequence of computed iterates $\{x_k\}$.

Assumption 3.1. *The functions f and c are twice continuously differentiable.*

Assumption 3.2. *The sequence of matrices $\{H_k^M\}_{k\geq 0}$ used in (3.1) are chosen to be uniformly positive definite and bounded in norm.*

Assumption 3.3. *The sequence of iterates $\{x_k\}$ is contained in a bounded set.*

The first result shows that the merit function is monotonically decreasing. It is assumed throughout this section that Algorithm 1 generates an infinite sequence, i.e., $\nabla M(v_k) \neq 0$ for all $k \geq 0$.

Lemma 3.1. *The sequence of iterates $\{v_k\}$ satisfies $M(v_{k+1}) < M(v_k)$ for all k.*

Proof. The vector Δv_k is a descent direction for M at v_k, i.e., $\nabla M(v_k)^T \Delta v_k < 0$, if $\nabla M(v_k)$ is nonzero and the matrix H_k^M is positive definite. Since H_k^M is positive definite by Assumption 3.2 and $\nabla M(v_k)$ 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 implies that the line search performed in Algorithm 1 produces an α_k such that the new point $v_{k+1} = v_k + \alpha_k \Delta v_k$ satisfies $M(v_{k+1}) < M(v_k)$. If follows that the only way the desired result can not hold is if the slack-reset procedure of Step 14 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), and (G) defining the function M, 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) 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 3.2. *The sequence of iterates $\{v_k\} = \big\{(x_k, s_k, y_k, w_k)\big\}$ 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 all i it holds that*

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

(iii) *The sequences $\{\pi^Y(x_k, s_k)\}$, $\{\pi^W(s_k)\}$, and $\{\nabla M(v_k)\}$ are bounded.*

(iv) *There exists a scalar M_{low} such that $M(x_k, s_k, y_k, w_k) \geq M_{\text{low}} > -\infty$ for all k.*

Proof. For a proof by contradiction, assume that $\{s_k\}$ is unbounded. Since $s_k + \mu^B e > 0$ by construction, there exists a subsequence $\mathcal{S}$ and component i such that

$$\lim_{k\in\mathcal{S}}\, [\, s_k\,]_i = \infty \;\text{ and }\; [\, s_k\,]_i \geq [\, s_k\,]_j \;\text{ for all } j \text{ and } k \in \mathcal{S}. \tag{3.3}$$

Next it will be shown that M must go to infinity on $\mathcal{S}$. It follows from (3.3), Assumption 3.3, 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 $-\ln([\, s_k\,]_i + \mu^B)$. Next, note that (G) 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 3.1. Otherwise, if (F) goes to $-\infty$ on $\mathcal{S}$ there must exist a subsequence $\mathcal{S}_1 \subseteq \mathcal{S}$ and a component j (say) such that

$$\lim_{k\in\mathcal{S}_1} [\, s_k + \mu^B e\,]_j [\, w_k\,]_j = \infty \quad \text{and} \tag{3.4}$$

$$[\, s_k + \mu^B e\,]_j [\, w_k\,]_j \ge [\, s_k + \mu^B e\,]_l [\, w_k\,]_l \;\; \text{for all } l \text{ and } k \in \mathcal{S}_1. \tag{3.5}$$

Using these properties and the fact that $w_k > 0$ and $s_k + \mu^B e > 0$ for all k by construction in Step 9 of Algorithm 1, it follows that (G) converges to ∞ faster than (F) converges to $-\infty$. Thus, M converges to ∞ on $\mathcal{S}_1$, which contradicts Lemma 3.1. 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 3.3.

Next, the third bound in part (i) will be established, i.e., $\{y_k\}$ is bounded. For a proof by contradiction, assume that there exists some subsequence $\mathcal{S}$ and component i such that

$$\lim_{k\in\mathcal{S}} \big|[\, y_k\,]_i\big| = \infty \;\; \text{and} \;\; \big|[\, y_k\,]_i\big| \ge \big|[\, y_k\,]_j\big| \;\; \text{for all } j \text{ and } k \in \mathcal{S}. \tag{3.6}$$

Using arguments as in the previous paragraph and 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 it has already been shown that $\{s_k\}$ is bounded. Using the uniform boundedness of $\{s_k\}$ a second time and $w^E > 0$, it may be deduced that (E) is bounded below. If (F) is bounded below on $\mathcal{S}$, then as (G) is bounded below by zero we would conclude, in totality, that $\lim_{k\in\mathcal{S}} M(v_k) = \infty$, which contradicts Lemma 3.1. Thus, (F) must converge to $-\infty$, which guarantees the existence of a subsequence $\mathcal{S}_1 \subseteq \mathcal{S}$ and a component, say j, that satisfies (3.4) and (3.5). For such $k \in \mathcal{S}_1$ and j it holds that (G) converges to ∞ faster than (F) converges to $-\infty$, so that $\lim_{k\in\mathcal{S}_1} M(v_k) = \infty$ on $\mathcal{S}_1$, which contradicts Lemma 3.1. Thus, $\{y_k\}$ is bounded.

We now prove the final bound in part (i), i.e., that $\{w_k\}$ is bounded. For a proof by contradiction, assume that the set is unbounded, which implies—using that $w_k > 0$ holds by construction of the line search in Step 9 of Algorithm 1—the existence of a subsequence $\mathcal{S}$ and a component i such that

$$\lim_{k\in\mathcal{S}} [\, w_k\,]_i = \infty \;\; \text{and} \;\; [\, w_k\,]_i \ge [\, w_k\,]_j \;\; \text{for all } j \text{ and } k \in \mathcal{S}. \tag{3.7}$$

It follows that there exists a subsequence $\mathcal{S}_1 \subseteq \mathcal{S}$ and set $\mathcal{J} \subseteq \{1,\, 2,\, \ldots,\, m\}$ satisfying

$$\lim_{k\in\mathcal{S}_1} [\, w_k\,]_j = \infty \;\; \text{for all } j \in \mathcal{J} \;\; \text{and} \;\; \big\{[\, w_k\,]_j : j \notin \mathcal{J} \;\; \text{and} \;\; k \in \mathcal{S}_1\big\} \text{ is bounded.} \tag{3.8}$$

Next, using similar arguments as above and boundedness of $\{y_k\}$, we know that (A), (B), (C), and (D) are bounded. Next, the sum of (E) and (F) is

$$\text{(E)} + \text{(F)} = -\mu^B \sum_{j=1}^{m} w_j^E \big(2 \ln([\, s_k + \mu^B e \,]_j) + \ln([\, w_k \,]_j) \big). \tag{3.9}$$

Combining this with the definition of (G) and the result of Lemma 3.1, shows that

$$[\, w_k \,]_j [\, s_k + \mu^B e \,]_j = O\big(\ln([\, w_k \,]_i) \big) \quad \text{for all } 1 \le j \le m, \tag{3.10}$$

which can be seen to hold as follows. It follows from (3.7), the boundedness of $\{s_k\}$, $w^E > 0$, and (3.9) that (E) + (F) is bounded below by $-\mu^B w_i^E \ln([\, w_k \,]_i)$ for all sufficiently large $k \in \mathcal{S}$. Combining this with the boundedness of (A), (B), (C), and (D), implies that (3.10) must hold, because otherwise the merit function M would converge to infinity on $\mathcal{S}$, contradicting Lemma 3.1. Thus, (3.10) holds.

Using $w_k > 0$ (which holds by construction) and the monotonicity of $\ln(\cdot)$, it follows from (3.10) that there exists a positive constant κ_1 such that

$$\ln\big([\, s_k + \mu^B e \,]_j \big) \le \ln\left(\frac{\kappa_1 \ln([\, w_k \,]_i)}{[\, w_k \,]_j} \right) = \ln(\kappa_1) + \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_j) \tag{3.11}$$

for all $1 \le j \le m$ and sufficiently large k. Then, a combination of (3.9), the boundedness of $\{s_k\}$, (3.8), $w^E > 0$, and the bound (3.11) implies the existence of positive constants κ_2 and κ_3 satisfying

$$\begin{aligned} \text{(E)} + \text{(F)} &\ge -\kappa_2 - \mu^B \sum_{j \in \mathcal{J}} w_j^E \big(2 \ln([\, s_k + \mu^B e \,]_j) + \ln([\, w_k \,]_j) \big) \\ &\ge -\kappa_2 - \mu^B \sum_{j \in \mathcal{J}} w_j^E \big(2 \ln(\kappa_i) + 2 \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_j) \big) \\ &\ge -\kappa_3 - \mu^B \sum_{j \in \mathcal{J}} w_j^E \big(2 \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_j) \big) \end{aligned} \tag{3.12}$$

for all sufficiently large k. With the aim of bounding the summation in (3.12), define

$$\alpha = \frac{[\, w^E \,]_i}{4 \| w^E \|_1} > 0,$$

which is well-defined because $w^E > 0$. It follows from (3.7) and (3.8) that

$$2 \ln\big(\ln\big([\, w_k \,]_i \big) \big) - \ln\big([\, w_k \,]_j \big) \le \alpha \ln\big([\, w_k \,]_i \big)$$

for all $j \in \mathcal{J}$ and sufficiently large $k \in \mathcal{S}_1$. This bound, (3.12), and $w^E > 0$ imply that

$$\begin{aligned} &\text{(E)} + \text{(F)} \\ &\ge -\kappa_3 - \mu^B w_i^E \big(2 \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_i) \big) - \mu^B \sum_{j \in \mathcal{J}, j \ne i} w_j^E \big(2 \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_j) \big) \\ &\ge -\kappa_3 - \mu^B w_i^E \big(2 \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_i) \big) - \mu^B \sum_{j \in \mathcal{J}, j \ne i} w_j^E \alpha \ln([\, w_k \,]_i) \\ &\ge -\kappa_3 - \mu^B w_i^E \big(2 \ln\big(\ln([\, w_k \,]_i) \big) - \ln([\, w_k \,]_i) \big) - \mu^B \alpha \ln([\, w_k \,]_i) \| w^E \|_1 \end{aligned}$$

for all sufficiently large $k \in \mathcal{S}_1$. Combining this inequality with the choice of α and

$$2\ln\big(\ln([\,w_k\,]_i)\big) - \ln([\,w_k\,]_i) \le -\tfrac{1}{2}\ln([\,w_k\,]_i)$$

for all sufficiently large $k \in \mathcal{S}$ (this follows from (3.7)), we obtain

$$\begin{aligned}
\text{(E)} + \text{(F)} &\ge -\kappa_3 + \tfrac{1}{2}\mu^{B} w_i^{E}\ln([\,w_k\,]_i) - \mu^{B}\alpha\ln([\,w_k\,]_i)\|w^{E}\|_1 \\
&\ge -\kappa_3 + \mu^{B}\left(\tfrac{1}{2}w_i^{E} - \alpha\|w^{E}\|_1\right)\ln([\,w_k\,]_i) \\
&= -\kappa_3 + \tfrac{1}{4}\mu^{B}\ln([\,w_k\,]_i)
\end{aligned}$$

for all sufficiently large $k \in \mathcal{S}_1$. In particular, this inequality and (3.7) together give

$$\lim_{k\in\mathcal{S}_1} \text{(E)} + \text{(F)} = \infty.$$

It has already been established that the terms (A), (B), (C), and (D) are bounded, and it is clear that (G) is bounded below by zero. It follows that M converges to infinity on $\mathcal{S}_1$. As this contradicts Lemma 3.1, it must hold that $\{w_k\}$ is bounded.

Part (ii) is also proved by contradiction. Suppose that $\big\{[\,s_k + \mu^{B}e\,]_i\big\} \to 0$ on some subsequence $\mathcal{S}$ and for some component i. As before, (A), (B), (C), and (D) are all bounded from below over all k. We may also use $w^{E} > 0$ 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 $\mathcal{S}$. Also, as already shown, the term (G) is bounded below. In summary, it has been shown that $\lim_{k\in\mathcal{S}} M(v_k) = \infty$, which contradicts Lemma 3.1, and therefore establishes that $\liminf\,[\,s_k + \mu e\,]_i > 0$ for all i. A similar argument may be used to prove that $\liminf\,[\,w_k\,]_i > 0$ for all i, which completes the proof.

Consider part (iii). The sequence $\big\{\pi^{Y}(x_k, s_k)\big\}$ is bounded as a consequence of part (i) and the fact that y^{E} and μ^{P} are fixed. Similarly, the sequence $\big\{\pi^{W}(s_k)\big\}$ is bounded as a consequence of part (ii) and the fact that w^{E} and μ^{B} are fixed. Lastly, the sequence $\big\{\nabla M(x_k, s_k, y_k)\big\}$ is bounded as a consequence of parts (i) and (ii), the uniform boundedness just established for $\big\{\pi^{Y}(x_k, s_k)\big\}$ and $\big\{\pi^{W}(s_k)\big\}$, Assumption 3.1, Assumption 3.3, and the fact that y^{E}, w^{E}, μ^{P}, and μ^{B} are fixed.

For part (iv) it will be shown that each term in the definition of M is bounded below. Term (A) is bounded below because of Assumption 3.1 and Assumption 3.2. Term (B) is bounded below as a consequence of part (i) and the fact that y^{E} is kept fixed. Terms (C) and (D) are both nonnegative, hence, trivially bounded below. Terms (E) and (F) are bounded below because μ^{B} and $w^{E} > 0$ are held fixed, and part (i). Finally, it follows from part (ii) that (G) is positive. The existence of the lower bound M_{low} now follows. ∎

Certain results hold when the gradients of M are bounded away from zero.

Lemma 3.3. *If there exists a positive scalar ϵ and a subsequence $\mathcal{S}$ satisfying*

$$\|\nabla M(v_k)\| \ge \epsilon \;\; \textit{for all} \;\; k \in \mathcal{S}, \tag{3.13}$$

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* $\nabla M(v_k)^T\Delta v_k \le -\delta$ *for all* $k\in\mathcal{S}$*.*

(iii) *There exist a positive scalar* $\alpha_{\min}$ *such that, for all* $k\in\mathcal{S}$*, the Armijo condition in* Step 10 *of* Algorithm 1 *is satisfied with* $\alpha_k \ge \alpha_{\min}$*.*

Proof. Part (i) follows from (3.13), Assumption 3.2, Lemma 3.2(iii), and the fact that Δv_k is computed from (3.1). For part (ii), first observe from (3.1) that

$$\nabla M(v_k)^T\Delta v_k = -\Delta v_k^T H_k^M\Delta v_k \le -\lambda_{\min}(H_k^M)\|\Delta v_k\|_2^2. \tag{3.14}$$

The existence of δ in part (ii) now follows from (3.14), Assumption 3.2, and part (i).

For part (iii), a standard result of unconstrained optimization [33] is that the Armijo condition is satisfied for all

$$\alpha_k = \Omega\left(\frac{-\nabla M(v_k)^T\Delta v_k}{\|\Delta v_k\|^2}\right). \tag{3.15}$$

This result requires the Lipschitz continuity of $\nabla M(v)$, which holds as a consequence of Assumption 3.1 and Lemma 3.2(ii). The existence of the positive $\alpha_{\min}$ of part (iii) now follows from (3.15), and parts (i) and (ii). ∎

The main convergence result follows.

Theorem 3.1. *Under* Assumptions 3.1–3.3*, the sequence of iterates* $\{v_k\}$ *satisfies* $\lim_{k\to\infty}\nabla M(v_k)=0$.

Proof. The proof is by contradiction. Suppose there exists a constant $\epsilon>0$ and a subsequence $\mathcal{S}$ such that $\|\nabla M(v_k)\|\ge\epsilon$ for all $k\in\mathcal{S}$. It follows from Lemma 3.1 and Lemma 3.2(iv) that $\lim_{k\to\infty}M(v_k)=M_{\min}>-\infty$. Using this result and the fact that the Armijo condition is satisfied for all k (see Step 10 in Algorithm 1), it must follow that

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

which implies that $\lim_{k\in\mathcal{S}}\alpha_k=0$ from Lemma 3.3(ii). This result and Lemma 3.3(iii) imply that the inequality constraints enforced in Step 9 of Algorithm 1 must have restricted the step length. In particular, there must exist a subsequence $\mathcal{S}_1\subseteq\mathcal{S}$ and a component i such that either

$$[\,s_k+\alpha_k\Delta s_k+\mu^B e\,]_i>0 \;\text{ and }\; [\,s_k+(1/\gamma)\alpha_k\Delta s_k+\mu^B e\,]_i\le 0 \;\text{ for } k\in\mathcal{S}_1$$

or

$$[\,w_k+\alpha_k\Delta w_k\,]_i>0 \;\text{ and }\; [\,w_k+(1/\gamma)\alpha_k\Delta w_k\,]_i\le 0 \;\text{ for } k\in\mathcal{S}_1, \tag{3.16}$$

where $\gamma\in(0,1)$ is the Armijo parameter of Algorithm 1. As the argument used for both cases is the same, it may be assumed, without loss of generality, that (3.16) occurs. It follows from Lemma 3.2(ii) that there exists some positive ϵ such that

$$\epsilon < w_{k+1} = w_k+\alpha_k\Delta w_k = w_k+(1/\gamma)\alpha_k\Delta w_k-(1/\gamma)\alpha_k\Delta w_k+\alpha_k\Delta w_k$$

for all sufficiently large k, so that with (3.16) it must hold that

$$w_k + (1/\gamma)\alpha_k \Delta w_k > \epsilon + (1/\gamma)\alpha_k \Delta w_k - \alpha_k \Delta w_k = \epsilon + \alpha_k \Delta w_k (1/\gamma - 1) > 0$$

for all sufficiently large $k \in \mathcal{S}_1$, where the last inequality follows from $\lim_{k\in\mathcal{S}} \alpha_k = 0$ and Lemma 3.3(i). This contradicts (3.16) for all sufficiently large $k \in \mathcal{S}_1$. ∎

4. Solving the Nonlinear Optimization Problem

In this section a method for solving the nonlinear optimization problem (NIPs) is formulated and analyzed. The method builds upon the algorithm presented in Section 3 for minimizing the shifted primal-dual penalty-barrier function.

4.1. The algorithm

The proposed method is given in Algorithm 2. It combines Algorithm 1 with strategies for adjusting the parameters that define the merit function, which were fixed in Algorithm 1. The proposed strategy uses the distinction between 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 at $v_{k+1} = (x_{k+1},\, s_{k+1},\, y_{k+1},\, w_{k+1})$ is defined in terms of the following feasibility, stationarity, and complementarity measures:

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

where

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

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

$$\chi(v, \mu) = \chi_{\rm feas}(v) + \chi_{\rm stny}(v) + \chi_{\rm comp}(v, \mu). \tag{4.1}$$

With these definitions in hand, 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 10). 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

$$\operatorname*{minimize}_{v=(x,s,y,w)} \;\; M(v\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B). \tag{4.2}$$

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

$$\|\nabla_x M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k, \tag{4.3a}$$

$$\|\nabla_s M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k, \tag{4.3b}$$

$$\|\nabla_y M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k \|D_{k+1}^P\|_\infty, \quad \text{and} \tag{4.3c}$$

$$\|\nabla_w M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\|_\infty \le \tau_k \|D_{k+1}^B\|_\infty, \tag{4.3d}$$

where τ_k is a positive tolerance, $D_{k+1}^P = \mu_k^P I$, and $D_{k+1}^B = (S_{k+1} + \mu_k^B I)W_{k+1}^{-1}$. (See Lemma 4.3 for a justification of (4.3).) In this case v_{k+1} is called an M-iterate because it is an approximate first-order solution of (4.2). The multiplier estimates y_{k+1}^E and w_{k+1}^E are defined by the safeguarded values

$$y_{k+1}^E = \max\big(-y_{\max}e, \min(y_{k+1}, y_{\max}e)\big) \quad \text{and} \quad w_{k+1}^E = \min(w_{k+1}, w_{\max}e) \tag{4.4}$$

for some positive constants $y_{\max}$ and $w_{\max}$. Next, Step 13 checks if the condition

$$\chi_{\text{feas}}(v_{k+1}) \le \tau_k \tag{4.5}$$

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 constraint $c(x) - s = 0$ in subsequent iterations. Similarly, Step 17 checks the inequalities

$$\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \le \tau_k \quad \text{and} \quad s_{k+1} \ge -\tau_k e. \tag{4.6}$$

If these conditions hold, then $\mu_{k+1}^B \leftarrow \mu_k^B$; otherwise, $\mu_{k+1}^B \leftarrow \frac{1}{2}\mu_k^B$ 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 merit function parameters are changed, so that progress is measured solely in terms of the reduction in the merit function.

4.2. Convergence analysis

Convergence of the iterates is established using the properties of the *complementary approximate KKT (CAKKT) condition* proposed by Andreani, Martínez and Svaiter [2], as described next.

Definition 4.1. (CAKKT condition) *A feasible point (x^*, s^*) (i.e., a point such that $s^* \ge 0$ 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*

$$\{g(x_j) - J(x_j)^T u_j\} \to 0, \tag{4.7a}$$

$$\{u_j - z_j\} \to 0, \tag{4.7b}$$

$$\{z_j\} \ge 0, \quad \textit{and} \tag{4.7c}$$

$$\{z_j \cdot s_j\} \to 0. \tag{4.7d}$$

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

Algorithm 2 A shifted primal-dual penalty-barrier method.

```
procedure PDB($x_0, s_0, y_0, w_0$)
    Restrictions: $s_0 > 0$ and $w_0 > 0$;
    Constants: $\{\eta, \gamma\} \subset (0,1)$ and $\{y_{\max}, w_{\max}\} \subset (0,\infty)$;
    Choose $y_0^E$, $w_0^E > 0$;  $\chi_0^{\max} > 0$; and $\{\mu_0^P, \mu_0^B\} \subset (0,\infty)$;
    Set $v_0 = (x_0, s_0, y_0, w_0)$;  $k \leftarrow 0$;
    while $\|\nabla M(v_k)\| > 0$ do
        $(y^E, w^E, \mu^P, \mu^B) \leftarrow (y_k^E, w_k^E, \mu_k^P, \mu_k^B)$;
        Compute $v_{k+1} = (x_{k+1}, s_{k+1}, y_{k+1}, w_{k+1})$ in Steps 6–15 of Algorithm 1;
        if $\chi(v_{k+1}, \mu_k^B) \le \chi_k^{\max}$ then                              [O-iterate]
            $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\frac{1}{2}\chi_k^{\max}, y_{k+1}, w_{k+1}, \mu_k^P, \mu_k^B, \tau_k)$;
        else if $v_{k+1}$ satisfies (4.3) then                                      [M-iterate]
            Set $(\chi_{k+1}^{\max}, \tau_{k+1}) = (\chi_k^{\max}, \frac{1}{2}\tau_k)$;  Set $y_{k+1}^E$ and $w_{k+1}^E$ using (4.4);
            if $\chi_{\text{feas}}(v_{k+1}) \le \tau_k$ then $\mu_{k+1}^P \leftarrow \mu_k^P$ else $\mu_{k+1}^P \leftarrow \frac{1}{2}\mu_k^P$ end if
            if $\chi_{\text{comp}}(v_{k+1}, \mu_k^B) \le \tau_k$ and $s_{k+1} \ge -\tau_k e$ then
                $\mu_{k+1}^B \leftarrow \mu_k^B$;
            else
                $\mu_{k+1}^B \leftarrow \frac{1}{2}\mu_k^B$;  Reset $s_{k+1}$ so that $s_{k+1} + \mu_{k+1}^B e > 0$;
        else                                                                        [F-iterate]
            $(\chi_{k+1}^{\max}, y_{k+1}^E, w_{k+1}^E, \mu_{k+1}^P, \mu_{k+1}^B, \tau_{k+1}) \leftarrow (\chi_k^{\max}, y_k^E, w_k^E, \mu_k^P, \mu_k^B, \tau_k)$;
```

The CAKKT condition is a sequential optimality condition that holds for every local minimizer. Compared to other sequential conditions, it is relatively tight, i.e., there are relatively few CAKKT points that are not local minimizers. The mechanism for relating a CAKKT point to a KKT point is given by CAKKT-regularity, which is the weakest known constraint qualification that ensures the following result holds.

Theorem 4.1. (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 (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

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

The first part of the analysis establishes that limit points of the sequence of O-iterates are CAKKT points.

Lemma 4.1. *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 3.3 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 3.1 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 $\chi(v_{k+1},\mu_k^B)\le\chi_k^{\max}$ for all $k\in\mathcal{K}\subseteq\mathcal{O}$, and $\chi_{\text{feas}}(v_{k+1})\le\chi(v_{k+1},\mu_k^B)$ 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 $s^*\ge 0$ is proved componentwise. Let $i\in\{1,\,2,\,\ldots,\,m\}$, and 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 χ_{comp}. If the set $\mathcal{K}\cap\mathcal{Q}_1$ is infinite, then it follows from the inequalities $\{\chi_{\text{comp}}(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\le\{\chi(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\le\{\chi_k^{\max}\}_{k\in\mathcal{K}}\to 0$ 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 $[\,s^*\,]_i=\lim_{\mathcal{K}\cap\mathcal{Q}_2}[\,s_{k+1}\,]_i=\lim_{\mathcal{K}\cap\mathcal{Q}_2}[\,s_{k+1}+\mu_k^Be\,]_i\ge 0$, 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. Consider the sequence $(x_{k+1},\,\bar{s}_{k+1},\,y_{k+1},\,w_{k+1})_{k\in\mathcal{K}}$ as a candidate for the sequence used in Definition 4.1 to verify that (x^*,s^*) is a CAKKT point, where

$$[\,\bar{s}_{k+1}\,]_i=\begin{cases}[\,s_{k+1}\,]_i & \text{if } k\in\mathcal{Q}_1,\\ [\,s_{k+1}+\mu_k^Be\,]_i & \text{if } k\in\mathcal{Q}_2,\end{cases}\tag{4.8}$$

for each $i\in\{1,\,2,\,\ldots,\,m\}$. 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 $\{[\,\bar{s}_{k+1}\,]_i\}_{k\in\mathcal{K}}\to[\,s^*\,]_i$. On the other hand, if $\mathcal{O}\cap\mathcal{Q}_2$ is infinite, then the definitions of $\mathcal{Q}_2$ and $\chi_{\text{comp}}(v_{k+1},\mu_k^B)$, together with the limit $\{\chi_{\text{comp}}(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\to 0$ imply that $\{\mu_k^B\}\to 0$, giving $\{[\,\bar{s}_{k+1}\,]_i\}_{k\in\mathcal{K}}\to[\,s^*\,]_i$. As the choice of i was arbitrary, these cases taken together imply that $\{\bar{s}_{k+1}\}_{k\in\mathcal{K}}\to s^*$.

The next step is to show that $\{(x_{k+1},\bar{s}_{k+1},y_{k+1},w_{k+1})\}_{k\in\mathcal{K}}$ satisfies the conditions required by Definition 4.1. It follows from the limit $\{\chi(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\to 0$ established above that $\{\chi_{\text{stny}}(v_{k+1})+\chi_{\text{comp}}(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\le\{\chi(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\to 0$. This implies that $\{g_{k+1}-J_{k+1}^Ty_{k+1}\}_{k\in\mathcal{K}}\to 0$ and $\{y_{k+1}-w_{k+1}\}_{k\in\mathcal{K}}\to 0$, which establishes that conditions (4.7a) and (4.7b) hold. Step 9 of Algorithm 1 enforces the nonnegativity of w_{k+1} for all k, which implies that (4.7c) is satisfied for $\{w_k\}_{k\in\mathcal{K}}$. Finally, it must be shown that (4.7d) holds, i.e., that $\{w_{k+1}\cdot\bar{s}_{k+1}\}_{k\in\mathcal{K}}\to 0$. Consider the ith components of s_k, $\bar{s}_k$ and 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 $\chi_{\text{comp}}(v_{k+1},\mu_k^B)$, together with the limit $\{\chi_{\text{comp}}(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\to 0$ imply that $\{[\,w_{k+1}\cdot\bar{s}_{k+1}\,]_i\}_{\mathcal{K}\cap\mathcal{Q}_1}\to 0$. 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 $\chi_{\text{comp}}(v_{k+1},\mu_k^B)$, together with the limit $\{\chi_{\text{comp}}(v_{k+1},\mu_k^B)\}_{k\in\mathcal{K}}\to 0$ imply that $\{[\,w_{k+1}\cdot\bar{s}_{k+1}\,]_i\}_{k\in\mathcal{K}\cap\mathcal{Q}_2}\to 0$. These two cases lead to the conclusion that $\{w_{k+1} \cdot \bar{s}_{k+1}\}_{k\in\mathcal{K}} \to 0$, which implies that condition (4.7d) is satisfied. This concludes the proof that (x^*, s^*) is a CAKKT point. ∎

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

$$\underset{x,s}{\text{minimize}} \;\; \tfrac{1}{2}\|c(x) - s\|_2^2 \;\; \text{subject to} \;\; s \geq 0. \tag{4.9}$$

The first-order KKT conditions for problem (4.9) are

$$J(x^*)^T\big(c(x^*) - s^*\big) = 0, \qquad s^* \geq 0, \tag{4.10a}$$

$$s^* \cdot \big(c(x^*) - s^*\big) = 0, \qquad c(x^*) - s^* \leq 0. \tag{4.10b}$$

These conditions define an infeasible stationary point.

Definition 4.2. (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 (4.10).*

The first result shows that the set of M-iterations is infinite whenever the set of O-iterations is finite.

Lemma 4.2. *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, which implies that there must exist an iteration index k_F such that

$$k \in \mathcal{F}, \;\; y_k^E = y^E, \;\; \text{and} \;\; (\tau_k, w_k^E, \mu_k^P, \mu_k^B) = (\tau, w^E, \mu^P, \mu^B) > 0 \tag{4.11}$$

for all $k \geq k_F$. This means that the iterates computed by Algorithm 2 are the same as those computed by Algorithm 1 for all $k \geq k_F$. In this case Theorem 3.1, Lemma 3.2(i), and Lemma 3.2(ii) can be applied to show that (4.3) is satisfied for all k sufficiently large. This would mean, in view of Step 11 of Algorithm 2, that $k \in \mathcal{M}$ for all sufficiently large $k \geq k_F$, which contradicts (4.11) since $\mathcal{F} \cap \mathcal{M} = \emptyset$. ∎

The next lemma justifies the use of the quantities on the right-hand side of (4.3). In order to simplify the notation, we introduce the quantities

$$\pi_{k+1}^Y = y_k^E - \frac{1}{\mu_k^P}\big(c(x_{k+1}) - s_{k+1}\big) \;\; \text{and} \;\; \pi_{k+1}^W = \mu_k^B(S_{k+1} + \mu_k^B I)^{-1} w_k^E \tag{4.12}$$

with $S_{k+1} = \operatorname{diag}(s_{k+1})$ associated with the gradient of the merit function in (2.3).

Lemma 4.3. *If* $|\mathcal{M}| = \infty$ *then*

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

Proof. It follows from (2.3), (4.3c) and (4.3d) that

$$|\pi^{Y}_{k+1} - y_{k+1}| \le \tau_k \quad \text{and} \quad |\pi^{W}_{k+1} - w_{k+1}| \le \tau_k. \tag{4.13}$$

As $|\mathcal{M}| = \infty$ by assumption, Step 12 of Algorithm 2 implies that $\lim_{k\to\infty} \tau_k = 0$. Combining this with (4.13) establishes the first two limits in the result. The limit $\lim_{k\to\infty} \tau_k = 0$ may then be combined with (2.3), (4.13) and (4.3b) to show that

$$\lim_{k\in\mathcal{M}} \; |\pi^{Y}_{k+1} - \pi^{W}_{k+1}| = 0, \tag{4.14}$$

which is the third limit in the result. Finally, as $\lim_{k\to\infty} \tau_k = 0$, it follows from the limit (4.14) and bounds (4.13) that

$$\begin{aligned} 0 &= \lim_{k\in\mathcal{M}} \; |\pi^{Y}_{k+1} - \pi^{W}_{k+1}| = \lim_{k\in\mathcal{M}} \; |(\pi^{Y}_{k+1} - y_{k+1}) + (y_{k+1} - w_{k+1}) + (w_{k+1} - \pi^{W}_{k+1})| \\ &= \lim_{k\in\mathcal{M}} \; |y_{k+1} - w_{k+1}|. \end{aligned}$$

This establishes the last of the four limits. ∎

The next lemma shows that if the set of O-iterations is finite, then any limit point of the sequence $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{M}}$ is infeasible with respect to $c(x) - s = 0$.

Lemma 4.4. *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* $c(x^*) - s^* \ne 0$.

Proof. 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

$$\lim_{k\in\mathcal{K}} \; \|c(x_{k+1}) - s_{k+1}\| = 0. \tag{4.15}$$

A combination of the assumption that $|\mathcal{O}| < \infty$, the result of Lemma 4.2, and the updates of Algorithm 2, establishes that $\lim_{k\to\infty} \tau_k = 0$ and

$$\chi^{\max}_k = \chi^{\max} > 0 \;\text{ for all sufficiently large } k \in \mathcal{K}. \tag{4.16}$$

Using $|\mathcal{O}| < \infty$ together with Lemma 4.3, the fact that $\mathcal{K} \subseteq \mathcal{M}$, and Step 9 of the line search of Algorithm 1 gives

$$\lim_{k\in\mathcal{K}} \; \|y_{k+1} - w_{k+1}\| = 0, \;\text{ and }\; w_{k+1} > 0 \;\text{ for all } k \ge 0. \tag{4.17}$$

Next, it can be observed from the definitions of π^{Y}_{k+1} and $\nabla_{\!x} M$ that

$$\begin{aligned} g_{k+1} - J^T_{k+1} y_{k+1} &= g_{k+1} - J^T_{k+1}(2\pi^{Y}_{k+1} + y_{k+1} - 2\pi^{Y}_{k+1}) \\ &= g_{k+1} - J^T_{k+1}\big(2\pi^{Y}_{k+1} - y_{k+1}\big) - 2J^T_{k+1}(y_{k+1} - \pi^{Y}_{k+1}) \\ &= \nabla_{\!x} M(v_{k+1}; y^E_k, w^E_k, \mu^P_k, \mu^B_k) - 2J^T_{k+1}(y_{k+1} - \pi^{Y}_{k+1}), \end{aligned}$$

which combined with $\{x_{k+1}\}_{k\in\mathcal{K}} \to x^*$, $\lim_{k\to\infty} \tau_k = 0$, (4.3a), and Lemma 4.3 gives

$$\lim_{k\in\mathcal{K}} (g_{k+1} - J_{k+1}^T y_{k+1}) = 0. \tag{4.18}$$

Next, 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 (Algorithm 1, Steps 7–12) gives $s_{k+1} + \mu_k^B e > 0$ for all k. If $\lim_{k\to\infty} \mu_k^B = 0$, then $s^* = \lim_{k\in\mathcal{K}} s_{k+1} \geq -\lim_{k\in\mathcal{K}} \mu_k^B e = 0$. On the other hand, if $\lim_{k\to\infty} \mu_k^B \neq 0$, then Step 17 of Algorithm 2 is executed a finite number of times, $\mu_k^B = \mu^B > 0$ and (4.6) holds for all $k \in \mathcal{M}$ sufficiently large. Taking limits over $k \in \mathcal{M}$ in (4.6) and using $\lim_{k\to\infty} \tau_k = 0$ gives $s^* \geq 0$.

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 τ_k in Step 17 of Algorithm 2, it must be that (4.6) 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 $\lim_{k\to\infty} \tau_k = 0$, we have $\lim_{k\in\mathcal{K}} \chi_{\text{comp}}(v_{k+1}, \mu_k^B) = 0$.

Case 2: $\lim_{k\to\infty} \mu_k^B = 0$. Lemma 4.3 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

$$0 = \lim_{k\in\mathcal{K}} (S_{k+1} + \mu_k^B I)(\pi_{k+1}^W - w_{k+1}) = \lim_{k\in\mathcal{K}} \big(\mu_k^B w_k^E - (S_{k+1} + \mu_k^B I) w_{k+1}\big). \tag{4.19}$$

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 (4.4)). It then follows from (4.19), the uniform boundedness of $\{w_k^E\}$, and $\lim_{k\to\infty} \mu_k^B = 0$ that

$$0 = \lim_{k\in\mathcal{K}} \big([\, s_{k+1} \,]_i + \mu_k^B\big)[\, w_{k+1} \,]_i. \tag{4.20}$$

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 (4.20) 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 (4.20), $w_{k+1} > 0$ (see Step 9 of Algorithm 1), 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 (4.15), (4.17), (4.18), 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 (4.16) and the condition checked in Step 9 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 $c(x^*) - s^* \neq 0$. ∎

The next result shows that if the number of O-iterations is finite then all limit points of the set of M-iterations are infeasible stationary points.

Lemma 4.5. *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 4.2.

Proof. If $|\mathcal{O}| < \infty$ then Lemma 4.2 implies that $|\mathcal{M}| = \infty$. Moreover, the updating strategy of Algorithm 2 forces $\{y_k^E\}$ and $\{w_k^E\}$ to be bounded (see (4.4)). 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 $\{[\, s_{k+1}\,]_i\}_{k\in\mathcal{K}} \to \infty$. This implies that $\{[\, \pi_{k+1}^W\,]_i\}_{k\in\mathcal{K}} \to 0$ (see (4.12)) because $\{w_k^E\}$ is bounded and $\{\mu_k^B\}$ is positive and monotonically decreasing. These results together with Lemma 4.3 give $\{[\, \pi_{k+1}^Y\,]_i\}_{k\in\mathcal{K}} \to 0$. However, this limit, together with the boundedness of $\{y_k^E\}$ and the assumption that $\{[\, s_{k+1}\,]_i\}_{k\in\mathcal{K}} \to \infty$ implies that $\{[\, c(x_{k+1})\,]_i\}_{k\in\mathcal{K}} \to \infty$, which is impossible when Assumption 3.3 and Assumption 3.1 hold. Thus, it must be the case that $\{s_{k+1}\}_{k\in\mathcal{M}}$ is bounded.

The boundedness of $\{s_{k+1}\}_{k\in\mathcal{M}}$ and Assumption 3.3 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 (4.10a)–(4.10b)).

As $|\mathcal{O}| < \infty$, it follows from Lemma 4.4 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 (4.5) of Step 13 of Algorithm 2 will not hold for all sufficiently large $k \in \mathcal{K} \subseteq \mathcal{M}$. The subsequent updates ensure that $\{\mu_k^P\} \to 0$, which, combined with (3.2), the boundedness of $\{y_k^E\}$, and Lemma 4.3, gives

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

This implies that $c(x^*) - s^* \le 0$ and the second condition in (4.10b) holds.

The next part of the proof is to establish that $s^* \ge 0$, which is the inequality condition of (4.10a). The test in Step 14 of Algorithm 2 (i.e., testing whether (4.6) holds) is checked infinitely often because $|\mathcal{M}| = \infty$. If (4.6) 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 $s_{k+1} + \mu_k^B e > 0$, which is enforced by Step 9 of Algorithm 1, shows that $s^* \ge 0$, as claimed. On the other hand, if (4.6) 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 $s^* \ge 0$, as claimed.

For a proof of the equality condition of (4.10a) observe that the gradients must satisfy $\{\nabla_{\!x} M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)\}_{k\in\mathcal{K}} \to 0$ because condition (4.3) is satisfied for all $k \in \mathcal{M}$ (cf. Step 11 of Algorithm 2). Multiplying $\nabla_{\!x} M(v_{k+1}\,; y_k^E, w_k^E, \mu_k^P, \mu_k^B)$ by μ_k^P, and applying the definition of π_{k+1}^Y from (4.12) yields

$$\bigl\{\mu_k^P g(x_{k+1}) - J(x_{k+1})^T\bigl(\mu_k^P \pi_{k+1}^Y + \mu_k^P(\pi_{k+1}^Y - y_{k+1})\bigr)\bigr\}_{k\in\mathcal{K}} \to 0.$$

Combining this with $\{x_{k+1}\}_{k\in\mathcal{K}} \to x^*$, $\{\mu_k^P\} \to 0$, and the result of Lemma 4.3 yields

$$\bigl\{-J(x_{k+1})^T(\mu_k^P \pi_{k+1}^Y)\bigr\}_{k\in\mathcal{K}} = \bigl\{-J(x_{k+1})^T(\mu_k^P y_k^E - c(x_{k+1}) + s_{k+1})\bigr\}_{k\in\mathcal{K}} \to 0.$$

Using this limit in conjunction with the boundedness of $\{y_k^E\}$, the fact that $\{\mu_k^P\} \to 0$, and $\{(x_{k+1}, s_{k+1}\}_{k\in\mathcal{K}} \to (x^*, s^*)$ establishes that the first condition of (4.10a) holds.

It remains to show that the complementarity condition of (4.10b) holds. From Lemma 4.3 it must be the case that $\{\pi_{k+1}^W - \pi_{k+1}^Y\}_{k\in\mathcal{K}} \to 0$. Also, the limiting value does not change if the sequence is multiplied (term by term) by the bounded sequence $\{\mu_k^P(S_{k+1} + \mu_k^B I)\}_{k\in\mathcal{K}}$ (recall that $\{s_{k+1}\}_{k\in\mathcal{K}} \to s^*$). This yields

$$\big\{\mu_k^B\mu_k^P w_k^E - \mu_k^P(S_{k+1} + \mu_k^B I)y_k^E + (S_{k+1} + \mu_k^B I)(c(x_{k+1}) - s_{k+1})\big\}_{k\in\mathcal{K}} \to 0.$$

This limit, together with the limits $\{\mu_k^P\} \to 0$ and $\{s_{k+1}\}_{k\in\mathcal{K}} \to s^*$, and the boundedness of $\{y_k^E\}$ and $\{w_k^E\}$ implies that

$$\big\{(S_{k+1} + \mu_k^B I)(c(x_{k+1}) - s_{k+1})\big\}_{k\in\mathcal{K}} \to 0. \tag{4.21}$$

As $c(x^*) - s^* \neq 0$, there must exist a constraint index i such that $[\,c(x^*) - s^*\,]_i \neq 0$. Combining this with $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{K}} \to (x^*, s^*)$ and (4.21) shows that $\{[\,s_{k+1}\,]_i + \mu_k^B\}_{k\in\mathcal{K}} \to 0$. As s^* is nonnegative, it follows that $\{\mu_k^B\}_{k\in\mathcal{K}} \to 0$, However, as $\{\mu_k^B\}$ is a monotonically decreasing sequence, it must hold that $\{\mu_k^B\} \to 0$. Using this fact, (4.21), and $\{(x_{k+1}, s_{k+1})\}_{k\in\mathcal{K}} \to (x^*, s^*)$ it follows that $s^* \cdot \big(c(x^*) - s^*\big) = 0$, and the first condition in (4.10b) holds. This completes the proof. ∎

The overall convergence result can now be established.

Theorem 4.2. *Under* Assumptions 3.1–3.3*, 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 4.1 and Theorem 4.1. Part (ii) follows from Lemma 4.5. Also, it is clear that only one of these two cases must occur. ∎

5. The Modified-Newton Equations

This section concerns the properties of the modified-Newton equations $H_k^M \Delta v_k = -\nabla M(v_k)$ of (3.1). Subsection 5.1 focuses on the properties of the modified-Newton matrix, while subsection 5.2 discusses an efficient method for solving the resulting modified-Newton equations for the primal-dual search direction. Finally, subsection 5.3 establishes the relationship between the computed search direction and a shifted variant of the conventional primal-dual path-following equations. As this section is concerned with details of only a single iteration, the notation is simplified by omitting the dependence on the iteration k. In particular, we write $v = v_k$, $y^E = y_k^E$, $w^E = w_k^E$, $\pi^Y = \pi_k^Y$, $\pi^W = \pi_k^W$, $\Delta v = \Delta v_k$, $c = c(x_k)$, $J = J(x_k)$, $g = g(x_k)$, $D_P = \mu_k^P I$, $D_B = (S_k + \mu_k^B I)W_k^{-1}$, and $H^M = H_k^M$.

5.1. Definition of the modified-Newton matrix

The choice of H^M in the equations $H^M \Delta v = -\nabla M(v)$ is based on making two modifications to $\nabla^2 M$. The first involves substituting y for π^Y and w for π^W in (2.4). (Lemma 4.3 and the discussion of subsection 5.3 below provide justification for this choice.) The second modification is to replace the modified Hessian $H(x, y)$ by a symmetric $\widehat{H}$ such that $\widehat{H} \approx H(x, y)$ and H^M is positive definite. These modifications give an H^M in the form

$$H^M = \begin{pmatrix} \widehat{H} + 2J^T D_P^{-1} J & -2J^T D_P^{-1} & J^T & 0 \\ -2D_P^{-1} J & 2(D_P^{-1} + D_B^{-1}) & -I & I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B \end{pmatrix}. \tag{5.1}$$

Practical conditions for the choice of a positive-definite $\widehat{H}$ are based on the next result.

Theorem 5.1. *The matrix H^M in (5.1) is positive definite if and only if*

$$\text{In}(K) = \text{In}(n, m, 0), \quad \textit{where} \;\; K = \begin{pmatrix} \widehat{H} & J^T \\ J & -(D_B + D_P) \end{pmatrix}, \tag{5.2}$$

which holds if and only if $\widehat{H} + J^T (D_P + D_B)^{-1} J^T$ is positive definite.

Proof. Let $\bar{H}$, $\bar{J}$, $\bar{D}$ and H_{ind} denote the matrices

$$\bar{H} = \begin{pmatrix} \widehat{H} & 0 \\ 0 & 0 \end{pmatrix}, \quad \bar{J} = \begin{pmatrix} J & -I \\ 0 & I \end{pmatrix}, \quad \bar{D} = \begin{pmatrix} D_P & 0 \\ 0 & D_B \end{pmatrix}, \quad \text{and} \;\; H_{\text{ind}} = \begin{pmatrix} \bar{H} & \bar{J}^T \\ \bar{J} & -\bar{D} \end{pmatrix}. \tag{5.3}$$

Defining the nonsingular matrices T_1 and T_2 such that

$$T_1 = \begin{pmatrix} I & 0 \\ -\bar{D}^{-1}\bar{J} & I \end{pmatrix} \quad \text{and} \quad T_2 = \begin{pmatrix} 0 & I \\ I & \bar{D}^{-1}\bar{J} \end{pmatrix},$$

and using Sylvester's law of inertia, yields

$$\text{In}(H^M) = \text{In}(T_1^T H^M T_1) = \text{In} \begin{pmatrix} \bar{H} + \bar{J}^T \bar{D}^{-1} \bar{J} & 0 \\ 0 & \bar{D} \end{pmatrix} = \text{In}(\bar{H} + \bar{J}^T \bar{D}^{-1} \bar{J}) + (2m, 0, 0)$$

and

$$\text{In}(H_{\text{ind}}) = \text{In}(T_2^T H_{\text{ind}} T_2) = \text{In} \begin{pmatrix} -\bar{D} & 0 \\ 0 & \bar{H} + \bar{J}^T \bar{D}^{-1} \bar{J} \end{pmatrix} = \text{In}(\bar{H} + \bar{J}^T \bar{D}^{-1} \bar{J}) + (0, 2m, 0).$$

These identities imply that H^M is positive definite if and only if the inertia of H_{ind} is $(m + n, 2m, 0)$. The inertia of H_{ind} may be determined from the factorization $H_{\text{ind}} = S \Omega S^T$, where

$$S = \begin{pmatrix} 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ -D_B & -I & 0 & I \\ 0 & I & 0 & 0 \end{pmatrix} \quad \text{and} \quad \Omega = \begin{pmatrix} 0 & I & 0 & 0 \\ I & -D_B & 0 & 0 \\ 0 & 0 & \widehat{H} & J^T \\ 0 & 0 & J & -(D_B + D_P) \end{pmatrix}.$$

The matrix S is nonsingular, and Sylvester's law of inertia and [15, Lemma 4.1] give

$$\operatorname{In}(H_{\text{ind}}) = \operatorname{In}\begin{pmatrix} 0 & I \\ I & -D_B \end{pmatrix} + \operatorname{In}\begin{pmatrix} \widehat{H} & J^T \\ J & -(D_P + D_B) \end{pmatrix} = (m, m, 0) + \operatorname{In}(K).$$

If $\operatorname{In}(H_{\text{ind}}) = (m+n, 2m, 0)$ then it must be the case that $\operatorname{In}(K) = (n, m, 0)$, as required. The condition on the inertia of $\widehat{H} + J^T(D_P + D_B)^{-1}J^T$ follows from the identity

$$\operatorname{In}(K) = \operatorname{In}\big(\widehat{H} + J^T(D_P + D_B)^{-1}J^T\big) + (0, m, 0) \tag{5.4}$$

(see Forsgren and Gill [15, Lemma 4.1, p. 1143] for details). ∎

There are a number of alternative approaches for choosing $\widehat{H}$ based on computing a factorization of the $(n+m)$ by $(n+m)$ matrix K (5.2) (see, e.g., Gill and Robinson [24, Section 4], Forsgren [14], Forsgren and Gill [15]), Gould [27], Gill and Wong [25], and Wächter and Biegler [37]). All of these methods use $\widehat{H} = H(x,y)$ if this gives a sufficiently positive-definite H^M. The next result shows that $\widehat{H} = H(x,y)$ gives a positive-definite H^M in a sufficiently small neighborhood of a solution satisfying second-order sufficient optimality conditions and strict complementarity.

Theorem 5.2. *The matrix H^M in (5.1) with the choice $\widehat{H} = H(x,y)$ is positive definite for all $u = (x, s, y, w, y^E, w^E, \mu^P, \mu^B)$ sufficiently close to $u^* = (x^*,\ s^*,\ y^*,\ w^*,\ y^*,\ w^*,\ 0,\ 0)$, when $(x^*,\ s^*,\ y^*,\ w^*)$ is a solution of problem (NIPs) that satisfies second-order sufficient optimality conditions and strict complementarity.*

Proof. Let H denote the matrix $H(x,y)$. The aim is to show that $H + J^T(D_P + D_B)^{-1}J$ is positive definite under the assumptions made in the statement of the theorem. Let $(x^*,\ s^*,\ y^*,\ w^*)$ be a solution of problem (NIPs) that satisfies strict complementarity and second-order sufficiency optimality condition. It follows that

$$\max\{s^*, w^*\} > 0 \quad \text{and} \tag{5.5}$$

$$p^T H(x^*, y^*) p > 0 \quad \text{for all } p \neq 0 \text{ satisfying } J_{\mathcal{A}}(x^*) p = 0, \tag{5.6}$$

where $\mathcal{A} = \big\{i : [\,s^*\,]_i = 0\big\} = \big\{i : [\,c(x^*)\,]_i = 0\big\}$ and $J_{\mathcal{A}}(x^*)$ denotes the submatrix of $J(x^*)$ consisting of the rows with indices in $\mathcal{A}$. If the rows of J are partitioned according to the active and inactive constraints, then

$$\begin{aligned} H + J^T(D_P + D_B)^{-1}J = H &+ J_{\mathcal{A}}^T \left(\mu^P I + (S_{\mathcal{A}} + \mu^B I)W_{\mathcal{A}}^{-1}\right)^{-1} J_{\mathcal{A}} \\ &+ J_{\mathcal{I}}^T \left(\mu^P I + (S_{\mathcal{I}} + \mu^B I)W_{\mathcal{I}}^{-1}\right)^{-1} J_{\mathcal{I}}, \end{aligned} \tag{5.7}$$

where $\mathcal{I} = \{1,\ 2,\ \ldots,\ m\} \setminus \mathcal{A}$, and $S_{\mathcal{A}}$ and $S_{\mathcal{I}}$ are the submatrices of S consisting of the rows and columns from the index set $\mathcal{A}$ and $\mathcal{I}$, respectively, with a similar meaning for $W_{\mathcal{A}}$ and $W_{\mathcal{I}}$. If u and u^* are the quantities defined in the statement of the theorem, then the following limits hold:

$$\lim_{u \to u^*} J(x) = J(x^*), \qquad \lim_{u \to u^*} H(x,y) = H(x^*, y^*), \quad \text{and} \tag{5.8}$$

$$\lim_{u \to u^*} \left[\mu^P I + (S_{\mathcal{A}} + \mu^B I)W_{\mathcal{A}}^{-1}\right]_{ii}^{-1} = \infty. \tag{5.9}$$

for all $1 \le i \le m$. Using (5.8), (5.9), (5.6), and the same proof as in [23, Theorem 3.1], we can conclude that $H + J_{\mathcal{A}}^T \left(\mu^P I + (S_{\mathcal{A}} + \mu^B I) W_{\mathcal{A}}^{-1}\right)^{-1} J_{\mathcal{A}}$ is positive definite for all u sufficiently close to u^*. Combining this with the fact that the matrix $J_{\mathcal{I}}^T \left(\mu^P I + (S_{\mathcal{I}} + \mu^B I) W_{\mathcal{I}}^{-1}\right)^{-1} J_{\mathcal{I}}$ is positive semidefinite, the definition (5.7) implies that $H + J^T(D_P + D_B)^{-1} J$ is positive definite for all u sufficiently close to u^*. Finally, combining this fact with (5.4) shows that the matrix H_{ind} of (5.3) satisfies $\text{In}(H_{\text{ind}}) = (n+m, 2m, 0)$ for all u sufficiently close to u^*, which, together with Theorem 5.2, implies that H^M is positive definite for all u sufficiently close to u^*. ■

5.2. Solving the modified-Newton system

The modified-Newton system (3.1) defined with H^M from (5.1) should *not* be solved directly because of the potential for numerical instability. Instead, an *equivalent* transformed system should be solved based on the transformation

$$T = \begin{pmatrix} I & 0 & -2J^T D_P^{-1} & 0 \\ 0 & I & 2D_P^{-1} & -2D_B^{-1} \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & W \end{pmatrix}.$$

As T is nonsingular, the modified-Newton direction Δv from (3.1) satisfies

$$TH^M \Delta v = -T \nabla M(x, s, y, w; y^E, w^E, \mu^P, \mu^B),$$

which, upon multiplication and application of the identity $WD_B = S + \mu^B I$, yields

$$\begin{pmatrix} \widehat{H} & 0 & -J^T & 0 \\ 0 & 0 & I & -I \\ J & -I & D_P & 0 \\ 0 & W & 0 & S + \mu^B I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} g - J^T y \\ y - w \\ c - s + \mu^P (y - y^E) \\ s \cdot w + \mu^B (w - w^E) \end{pmatrix}. \tag{5.10}$$

The solution of this transformed system may be found by solving two sets of equations, one diagonal and the other of order $n + m$. To see this, first observe that the equations (5.10) may be written in the form

$$\begin{pmatrix} \widehat{H} & 0 & -J^T & 0 \\ 0 & 0 & I & -I \\ J & -I & D_P & 0 \\ 0 & I & 0 & D_B \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta w \end{pmatrix} = - \begin{pmatrix} g - J^T y \\ y - w \\ c - s + \mu^P (y - y^E) \\ W^{-1}\big(s \cdot w + \mu^B (w - w^E)\big) \end{pmatrix}. \tag{5.11}$$

The solution of (5.11) is given by

$$\Delta w = y - w + \Delta y \quad \text{and} \quad \Delta s = -W^{-1}\big(s \cdot (y + \Delta y) + \mu^B (y + \Delta y - w^E)\big), \tag{5.12}$$

where Δx and Δy satisfy the equations

$$\begin{pmatrix} \widehat{H} & -J^T \\ J & D_P + D_B \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} g - J^T y \\ c - s + \mu^P (y - y^E) + W^{-1}\big(s \cdot y + \mu^B (y - w^E)\big) \end{pmatrix},$$

or, equivalently, the symmetric equations

$$\begin{pmatrix} \widehat{H} & J^T \\ J & -(D_P + D_B) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = - \begin{pmatrix} g - J^T y \\ D_P(y - \pi^Y) + D_B\big(y - \pi^W\big) \end{pmatrix}. \tag{5.13}$$

Solving this $(n+m)\times(n+m)$ symmetric system is the dominant cost of an iteration. The identity $w + \Delta w = y + \Delta y$ implies that if the initial values satisfy $y_0 = w_0$ and $y_0^E = w_0^E$, and the positive safeguarding values in (4.4) satisfy $y_{\max} = w_{\max}$, then all subsequent iterates will satisfy $w = y$.

5.3. Relationship to primal-dual path-following

Consider the perturbed optimality conditions (2.2) and their associated primal-dual path-following equations

$$F(x, s, y, w\,; y^E, w^E, \mu^P, \mu^B) = \begin{pmatrix} g(x) - J(x)^T y \\ y - w \\ c(x) - s + \mu^P(y - y^E) \\ s \cdot w + \mu^B(w - w^E) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5.14}$$

A zero (x, s, y, w) of F satisfying $s > 0$ and $w > 0$ approximates a solution to problem (NIPs), with the approximation becoming increasingly accurate as both $\mu^P(y - y^E) \to 0$ and $\mu^B(w - w^E) \to 0$. If $v = (x, s, y, w)$ is a given approximate zero of F such that $s + \mu^B e > 0$ and $w > 0$, the Newton equations for the change in variables $\Delta v = (\Delta x, \Delta s, \Delta y, \Delta w)$ are given by $F'(v)\Delta v = -F(v)$, i.e.,

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

These equations are identical to the modified-Newton equations (5.10) for minimizing M when $\widehat{H} = H(x,y)$. Theorem 5.2 shows that the choice $\widehat{H} = H(x,y)$ is allowed in the neighborhood of a solution satisfying certain second-order optimality conditions, and it follows that the modified-Newton direction used in the proposed method is equivalent asymptotically to the shifted primal-dual path-following directions.

5.4. Infeasible shifted constraints

In Algorithm 2 it is necessary to reduce the value of the barrier parameter μ^B during an M-iteration if the slacks are not sufficiently feasible or the complementarity condition is not sufficiently satisfied (see Step 17 of Algorithm 2). In addition, as the initial values of μ^P and μ^B may be larger than the minimum values needed to give a positive-definite H_k^M (5.1) at a solution, it is prudent to reduce μ^P and μ^B if a sequence of iterations occurs in which H_k^M is not positive definite. However, reducing the value of μ^B reduces the value of the constraint shift, which may cause a slack variable to become infeasible with respect to its shifted bound. In this section we define a minor modification of the method that treats this situation. For reasons discussed below, it is assumed that a barrier parameter μ_i^B is associated with every constraint $s_i \geq 0$, i.e., μ^B is an m-vector with positive components. Suppose that μ_i^B and $\bar{\mu}_i^B$ denote a shift before and after it is reduced, with $s_i + \mu_i^B > 0$ and $s_i + \bar{\mu}_i^B \leq 0$. The variable s_i can be returned to feasibility by imposing a temporary equality constraint $s_i = 0$. This constraint is enforced by the primal-dual augmented Lagrangian term until $|c_i(x)|$ is sufficiently small that $c_i(x) > -\bar{\mu}_i^B$, at which point s_i is assigned the value $s_i = c_i(x)$ and allowed to move. On being freed, the value of w_i is reinitialized as $\max\{y_i, \epsilon\}$, where ϵ is a small positive constant. At a given iteration, if m_X slacks are fixed, then $m_F = m - m_X$ slacks are free to move. In those iterations for which some of the slack variables are fixed, the problem being solved has the form

$$\underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) - s = 0, \quad L_X s = 0, \quad L_F s \geq 0, \tag{5.15}$$

where L_X and L_F are $m_X \times m$ and $m_F \times m$ matrices formed from rows of the identity matrix I_m in such a way that $L_X s$ and $L_F s$ give the "fixed" and "free" components of s. While a slack is fixed, its associated barrier term is omitted from the shifted primal-dual merit function.

The shifted primal-dual modified-Newton equations for problem (5.15) are given in (5.16) and (5.17) below (for details on how the equations are derived, see Gill, Kungurtsev and Robinson [20]). In the following discussion, μ^B denotes a vector of shifts with the appropriate values of μ_i^B or $\bar{\mu}_i^B$. Any feasible s can be written uniquely as $s = L_F^T s_F$, where s_F is the n_F vector of free slacks. If w_F and w_X denote Lagrange multipliers for the constraints $L_X s = 0$ and $L_F s \geq 0$, given x and s such that $[\, s_F + \mu^B\,]_i > 0$, the solution of the modified-Newton equations for problem (5.15) can be written in terms of the quantities

$$D_P = \mu^P I, \qquad \pi^Y = y^E - \frac{1}{\mu^P}\big(c(x) - s\big),$$
$$D_B = (S_F + D_\mu^B)W_F^{-1}, \qquad \pi_F^W = \mu^B \cdot (S_F + D_\mu^B)^{-1} w_F^E,$$

where $D_\mu^B = \operatorname{diag}(\mu^B)$, $S_F = \operatorname{diag}(s_F)$, $W_F = \operatorname{diag}(w_F)$, and I_F is the identity matrix of order n_F. Given these definitions, the equations for Δs, Δw_F and Δw_X analogous to (5.12) and (5.13) are given by

$$\widehat{y} = y + \Delta y, \qquad \Delta s_F = -D_B\big(L_F\widehat{y} - \pi_F^W\big), \qquad \Delta s = L_F^T \Delta s_F, \tag{5.16a}$$
$$\Delta w_X = \;\; L_X\widehat{y} - w_X, \tag{5.16b}$$
$$\widehat{s} = s + \Delta s, \qquad \Delta w_F = -(S_F + D_\mu^B)^{-1}\big(w_F \cdot (L_F\widehat{s} + \mu^B) - \mu^B \cdot w_F^E\big), \tag{5.16c}$$

where Δx and Δy satisfy the equations

$$\begin{pmatrix} H & J^T \\ J & -(D_P + \bar{D}_B) \end{pmatrix} \begin{pmatrix} \Delta x \\ -\Delta y \end{pmatrix} = -\begin{pmatrix} g - J^T y \\ D_P(y - \pi^Y) + \bar{D}_B\big(y - L_F^T \pi_F^W\big) \end{pmatrix}, \tag{5.17}$$

with $\bar{D}_B = L_F^T D_B L_F$. Since the matrix $D_P + \bar{D}_B$ is diagonal, the treatment of an infeasible shifted constraint requires no significant additional computation (cf. (5.13)).

6. Implementation Details and Numerical Testing

Numerical results are given for a simple MATLAB implementation of `PDB` (Algorithm 2). Results were obtained for 140 problems from the CUTEst test collection (see Bongartz et al. [3] and Gould, Orban and Toint [28]). The problems consist of the CUTEst implementations of all but two of the 126 problems from the Hock and Schittkowski (HS) test set [29], and 16 problems from the COPS test set [9,11]. The two excluded problems are `hs87`, which is nonsmooth and `hs99exp`, which is poorly scaled.

6.1. The implementation

Each CUTEst problem may be written in the form

$$\underset{x}{\text{minimize}}\;\; f(x) \quad \text{subject to} \quad \begin{pmatrix} \ell^X \\ \ell^S \end{pmatrix} \le \begin{pmatrix} x \\ c(x) \end{pmatrix} \le \begin{pmatrix} u^X \\ u^S \end{pmatrix}, \tag{6.1}$$

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 $\pm 10^{20}$. For Algorithm `PDB`, each problem was converted to the equivalent form

$$\begin{array}{lllll} \underset{x\in\mathbb{R}^n, s\in\mathbb{R}^m}{\text{minimize}} & f(x) & & & \\ \text{subject to} & c(x) - s = 0, & L_X s = h_X, & \ell^S \le L_L s, & L_U s \le u^S, \\ & & E_X x = b_X, & \ell^X \le E_L x, & E_U x \le u^X, \end{array} \tag{6.2}$$

where s is a vector of slack variables. The quantity E_X denotes an $n_X \times n$ matrix formed from n_X independent rows of I_n. Similarly, E_L and E_U denote matrices formed from subsets of I_n such that $E_X^T E_L = 0$, $E_X^T E_U = 0$, i.e., a variable is either fixed or free to move, possibly bounded by an upper or lower bound. Note that a variable x_j need not be subject to a lower or upper bound, or may be bounded below and above, in which case e_j is not a row of E_X, E_L or E_U. Analogous definitions hold for L_X, L_L and L_U as subsets of rows of I_m although a given s_j must be either fixed or restricted by an upper or lower bound, i.e., there are no unrestricted slacks. The bound constraints involving E_X and L_X are enforced explicitly as in Section 5.4. The modified-Newton equations for problem (6.2) are derived by Gill, Kungurtsev and Robinson [20]. As is the case for problem (5.15) the principal work at each iteration is the solution of a perturbed reduced KKT system analogous to (5.17).

The problem format (6.2) must be extended to allow for the possibility of a variable or slack becoming infeasible with respect to its shifted bound. An infeasible slack variable is treated as in the previous section by temporarily fixing it on its bound. An infeasible variable is treated by imposing the bound indirectly using the primal-dual augmented Lagrangian. If x_j is infeasible with respect to $\ell_j^X - \mu_j^B$, the constraint $x_j - \ell_j^X = 0$ is included as a temporary penalty term in M, i.e.,

$$-v_j^E(x_j - \ell_j^X) + \frac{1}{2\mu_j^A}(x_j - \ell_j^X)^2 + \frac{1}{2\mu_j^A}\big(x_j - \ell_j^X + \mu_j^A(v_j - v_j^E)\big)^2,$$

where v_j^E is an estimate of the multiplier for the constraint $x_j = \ell_j^X$, and μ_j^A is a penalty parameter chosen so that $\mu_j^A < \bar{\mu}_j^B$. The initial values of v_j and v_j^E are $v_j = z_j$ and $v_j^E = z_j^E$, where $z_j > 0$ is the dual variable associated with the constraint $x_j \geq \ell_j^X$. (These quantities appear in the perturbed primal-dual optimality conditions associated with problem format (6.2)). While x_j is infeasible, its associated barrier term is omitted from the shifted primal-dual merit function. Once x_j returns to feasibility for the shifted bound, the shifted barrier term replaces the temporary penalty term in the definition of M with z_j and z_j^E initialized from v_j and v_j^E. For the purposes of deriving the KKT equations, this scheme implies that additional constraints $Ax - b = 0$ are imposed, where A is a matrix of positive and negative rows of I_n and b_j is either ℓ_j^X or $-u_j^X$. The effect of imposing the constraints $Ax - b = 0$ is to add a diagonal matrix $A^T D_\mu^A A = A^T \operatorname{diag}(\mu^A) A$ to the H block of the reduced KKT equations analogous to (5.13) (see Gill, Kungurtsev and Robinson [20] for more details).

Two alternative methods were used to modify the H-block of a KKT matrix with fewer than n positive eigenvalues, with the choice of method depending on the size of the problem. For the HS problems, H was modified during the calculation of the LDL^T factorization using the inertia controlling LDL^T factorization of Forsgren [14] and Forsgren and Gill [15]. For the COPS problems the Hessian was modified using the method of Wächter and Biegler [38, 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. Each (possibly perturbed) KKT matrix was factored using the MATLAB built-in command `LDL`, which uses the routine MA57.

6.2. Algorithm parameters and termination conditions

The MATLAB implementation was initialized with parameter values given in Table 1, which were chosen based on the empirical performance on the entire collection of problems. The primal-dual vector (x_0, y_0) was the default values supplied by CUTEst, although the code immediately projects x_0 onto the feasible region to ensure feasibility with respect to the bounds on x. The iterates were terminated at a point satisfying the condition

$$\|\chi(v_k)\|_\infty < \tau_{\text{stop}}, \tag{6.3}$$

where $\chi(v)$ is the optimality measure (4.1) defined in terms of problem (6.2).

Table 1: Control parameters and initial values for Algorithm `PDB`.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$y_{\max}/w_{\max}$	1.0e+5	τ_{stop}	1.0e-4	μ_0^P	1.0	$\chi_0^{\max}$	1.0e+3
η	1.0e-2	τ_0	0.5	μ_0^B	1.0e-4	γ	1.0e-3

6.3. Numerical results

Figure 1 gives the performance profiles and bar graphs that compare the number of function evaluations needed by PDB and the interior-point solver IPOPT [36, 38, 39] on the CUTEst HS and COPS test problems. In each case, the left figure gives performance profiles for the total number of functions evaluations. (For a description of how performance profiles should be interpreted, see Dolan and Moré [10].) The right figure gives the "outperforming factor" bar graphs proposed by Morales [31]. On the x-axis, each bar corresponds to a particular test problem, with the problems listed in ascending order for the HS problems and alphabetical order camshape, catmix, chain, channel, elec, gasoil, glider, marine, methanol, minsurfo, pinene, polygon, robotarm, rocket, steering, and torsion1 for the COPS problems. The y-axis indicates the factor ($\log_2$ scaled) by which one solver outperformed the other. A bar in the positive region indicates that PDB outperformed IPOPT. A negative dark gray bar means IPOPT performed better. A negative light gray bar denotes that PDB was unable to satisfy the termination criteria in 500 iterations. The results indicate that, overall, the simple MATLAB code PDB usually requires fewer function evaluations than IPOPT, but is slightly less robust. Algorithm PDB was able to satisfy the optimality measure for 137 (98%) of the 140 test problems. In each of the three COPS "failures", glider, robotarm and rocket, the iterates were terminated at a point where the KKT matrix was nearly singular. In these three cases, respectively 100%, 99% and 98% of the iterations required the Hessian to be modified. For the 124 HS problems, a grand total of 90% of the iterations computed were O-iterates, and 7% of the iterations computed were F-iterates. An M-iterate was computed in only 55 of the iterations required to solve all 124 HS problems. Overall, 41 of the 124 problems required the Hessian of the Lagrangian to be modified. For the COPS problems a grand total of 72% of the iterations computed were O-iterates, 26% computed were F-iterates, and there were 16 M-iterations. The Hessian was modified in 73% of the iterates. The results illustrate the crucial importance of an effective modification scheme when the KKT matrix does not have the correct inertia.

Tables 2 and 3 give the results of running algorithm PDB on the 158 CUTEst test problems. For each problem the table lists the number of general constraint functions ("m"), the number of x variables ("n"), the number of function evaluations ("fe"), and the number of outer iterations ("itns"). A run is considered to have failed if PDB could not satisfy the optimality condition (6.3) in 500 iterations. The function and iteration entries for these "failed" runs are marked with an "f". The column with the heading "H mods" gives the percentage of iterations for which the Hessian $H(x, y)$ was modified to ensure the positive definiteness of H^M (see 5.1). The columns "O-itns" and "F-itns" give the percentages of O-iterates and F-iterates required (see Algorithm 2). The column with heading "M-itns" gives the total number of M-iterates required for each problem. (The M-iterates are not listed as a percentage of the total number of iterations because they generally constitute significantly less than 1% of the total iterations.)

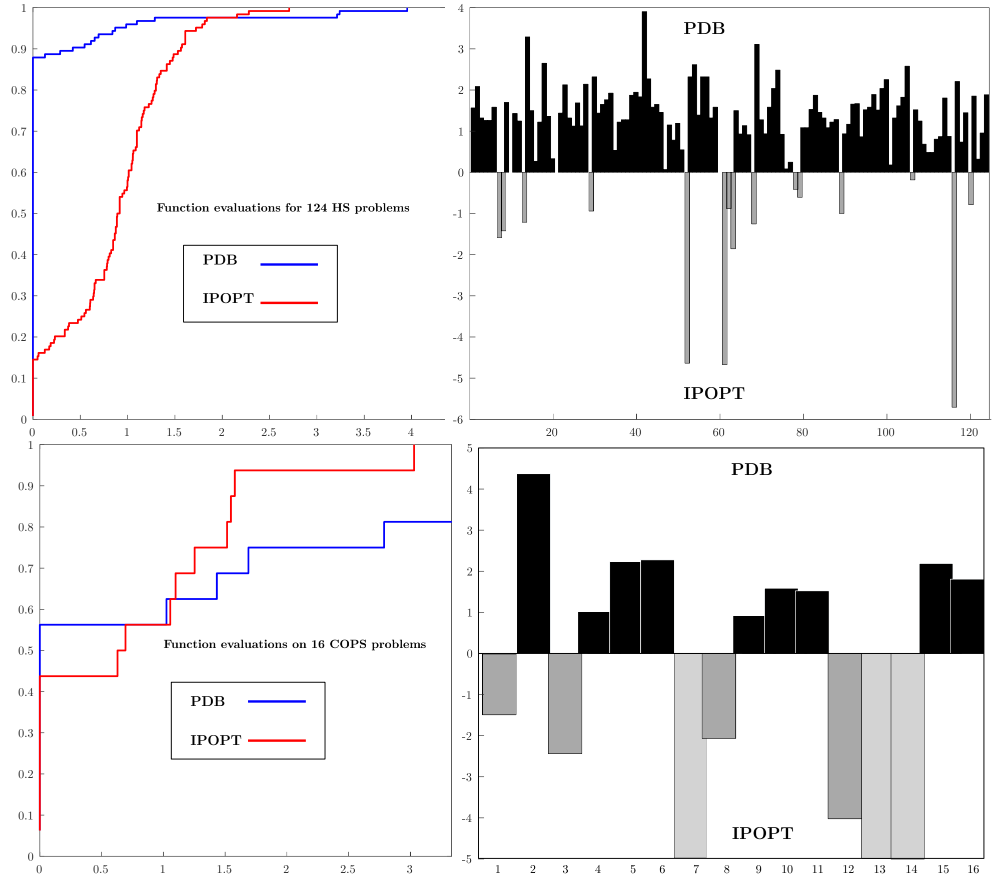

Figure 1: Performance profiles and outperforming factors for function evaluations.

Table 2: Results for Algorithm `PDB` on 124 Hock-Schittkowski problems

Problem	m	n	fe	itns	H mods	O-itns	F-itns	M-itns
`hs1`	0	2	26	23	0%	100%	0%	0
`hs2`	0	2	8	8	0%	100%	0%	0
`hs3`	0	2	4	4	0%	100%	0%	0
`hs3mod`	0	2	5	5	0%	100%	0%	0
`hs4`	0	2	5	5	0%	100%	0%	0
`hs5`	0	2	6	6	0%	100%	0%	0
`hs6`	1	2	42	12	0%	100%	0%	0
`hs7`	1	2	233	49	87%	28%	71%	0
`hs8`	2	2	4	4	0%	100%	0%	0
`hs9`	1	2	11	4	25%	100%	0%	0
`hs10`	1	2	10	10	0%	100%	0%	0
`hs11`	1	2	8	8	0%	100%	0%	0
`hs12`	1	2	44	15	0%	100%	0%	0

Table 2: Results for 124 Hock-Schittkowski test problems (continued)

Problem	m	n	fe	itns	H mods	O-itns	F-itns	M-itns
hs13	1	2	14	14	0%	100%	0%	0
hs14	2	2	6	6	0%	100%	0%	0
hs15	2	2	34	33	27%	48%	48%	1
hs16	2	2	9	9	0%	100%	0%	0
hs17	2	2	7	7	0%	100%	0%	0
hs18	2	2	14	14	0%	100%	0%	0
hs19	2	2	23	23	0%	73%	13%	3
hs20	3	2	19	19	52%	100%	0%	0
hs21	1	2	7	7	0%	100%	0%	0
hs21mod	1	7	8	8	0%	100%	0%	0
hs22	2	2	6	6	0%	100%	0%	0
hs23	5	2	11	11	0%	100%	0%	0
hs24	3	2	9	9	22%	100%	0%	0
hs25	0	3	37	37	70%	64%	29%	2
hs26	1	3	12	12	8%	100%	0%	0
hs27	1	3	451	43	0%	34%	60%	2
hs28	1	3	1	1	0%	100%	0%	0
hs29	1	3	7	6	33%	100%	0%	0
hs30	1	3	7	7	0%	100%	0%	0
hs31	1	3	5	5	0%	100%	0%	0
hs32	2	3	10	10	0%	100%	0%	0
hs33	2	3	20	12	50%	100%	0%	0
hs34	2	3	9	9	0%	100%	0%	0
hs35	1	3	7	7	0%	100%	0%	0
hs35i	1	3	7	7	0%	100%	0%	0
hs35mod	1	3	9	9	0%	100%	0%	0
hs36	1	3	7	7	0%	100%	0%	0
hs37	2	3	7	7	0%	100%	0%	0
hs38	0	4	9	8	0%	100%	0%	0
hs39	2	4	6	6	16%	100%	0%	0
hs40	3	4	3	3	0%	100%	0%	0
hs41	1	4	7	7	0%	100%	0%	0
hs42	2	4	4	4	0%	100%	0%	0
hs43	3	4	20	13	0%	100%	0%	0
hs44	6	4	18	18	44%	94%	5%	0
hs44new	6	4	18	18	44%	94%	5%	0
hs45	0	5	7	7	28%	100%	0%	0
hs46	2	5	28	16	0%	100%	0%	0
hs47	3	5	1045	107	52%	14%	85%	0
hs48	2	5	1	1	0%	100%	0%	0
hs49	2	5	7	7	0%	100%	0%	0
hs50	3	5	8	7	0%	100%	0%	0
hs51	3	5	1	1	0%	100%	0%	0
hs52	3	5	1	1	0%	100%	0%	0

Table 2: Results for 124 Hock-Schittkowski test problems (continued)

Problem	m	n	fe	itns	H mods	O-itns	F-itns	M-itns
hs53	3	5	6	6	0%	100%	0%	0
hs54	1	6	11	11	0%	100%	0%	0
hs55	6	6	9	9	22%	100%	0%	0
hs56	4	7	587	58	32%	24%	75%	0
hs57	1	2	96	92	22%	22%	75%	2
hs59	3	2	424	67	53%	28%	68%	2
hs60	1	3	6	6	0%	100%	0%	0
hs61	2	3	11	7	28%	100%	0%	0
hs62	1	3	10	10	0%	100%	0%	0
hs63	2	3	9	9	11%	100%	0%	0
hs64	1	3	93	34	17%	67%	8%	8
hs65	1	3	14	14	0%	100%	0%	0
hs66	2	3	7	7	0%	100%	0%	0
hs67	14	3	12	12	0%	100%	0%	0
hs68	2	4	16	16	6%	100%	0%	0
hs69	2	4	9	9	0%	100%	0%	0
hs70	1	4	15	15	13%	100%	0%	0
hs71	2	4	10	10	0%	100%	0%	0
hs72	2	4	33	33	0%	69%	0%	10
hs73	3	4	16	16	0%	100%	0%	0
hs74	5	4	28	22	0%	81%	13%	1
hs75	5	4	32	26	0%	76%	15%	2
hs76	3	4	8	8	0%	100%	0%	0
hs76i	3	4	8	8	0%	100%	0%	0
hs77	2	5	9	8	0%	100%	0%	0
hs78	3	5	3	3	0%	100%	0%	0
hs79	3	5	4	4	0%	100%	0%	0
hs80	3	5	6	6	0%	100%	0%	0
hs81	3	5	8	8	12%	100%	0%	0
hs83	3	5	15	15	0%	100%	0%	0
hs84	3	5	16	16	0%	100%	0%	0
hs85	21	5	88	82	0%	24%	75%	0
hs86	10	5	12	12	0%	100%	0%	0
hs88	1	2	16	16	0%	100%	0%	0
hs89	1	3	19	19	15%	78%	0%	4
hs90	1	4	16	16	18%	100%	0%	0
hs91	1	5	17	15	20%	100%	0%	0
hs92	1	6	16	16	18%	100%	0%	0
hs93	2	6	7	7	0%	100%	0%	0
hs95	4	6	14	14	7%	100%	0%	0
hs96	4	6	14	14	7%	100%	0%	0
hs97	4	6	9	9	11%	100%	0%	0
hs98	4	6	9	9	11%	100%	0%	0
hs99	2	7	15	10	0%	100%	0%	0

Table 2: Results for 124 Hock-Schittkowski test problems (continued)

Problem	m	n	fe	itns	H mods	O-itns	F-itns	M-itns
hs100	4	7	14	10	0%	100%	0%	0
hs100lnp	2	7	14	10	50%	100%	0%	0
hs100mod	4	7	11	8	0%	100%	0%	0
hs101	5	7	56	53	16%	39%	56%	2
hs102	5	7	67	67	17%	31%	65%	2
hs103	5	7	47	47	6%	44%	51%	2
hs104	5	8	8	8	0%	100%	0%	0
hs105	1	8	23	23	47%	78%	4%	4
hs106	6	8	25	13	0%	100%	0%	0
hs107	6	9	15	15	0%	100%	0%	0
hs108	13	9	20	14	14%	100%	0%	0
hs109	10	9	41	12	0%	100%	0%	0
hs110	0	10	4	4	0%	100%	0%	0
hs111	3	10	18	14	14%	100%	0%	0
hs111lnp	3	10	1723	108	87%	14%	84%	1
hs112	3	10	8	8	0%	100%	0%	0
hs113	8	10	15	15	0%	100%	0%	0
hs114	11	10	15	15	0%	100%	0%	0
hs116	14	13	88	58	18%	39%	48%	7
hs117	5	15	13	13	0%	100%	0%	0
hs118	17	15	20	20	0%	100%	0%	0
hs119	8	16	17	17	0%	100%	0%	0
hs268	5	5	10	10	0%	100%	0%	0

Table 3: Results for Algorithm PDB on the COPS problems

Problem	m	n	fe	itns	H mods	O-itns	F-itns	M-itns
camshape	203	100	170	170	0%	14%	82%	5
catmix	200	303	7	7	0%	100%	0%	0
chain	51	102	92	24	4%	91%	0%	2
channel	448	450	8	3	100%	100%	0%	0
elec	100	300	123	118	100%	19%	79%	1
gasoil	1298	1303	13	13	0%	100%	0%	0
glider	608	664	779^f	500^f	100%	1%	98%	0
marine	1392	1415	415	204	99%	8%	90%	2
methanol	1497	1505	16	13	0%	100%	0%	0
minsurfo	0	731	10	10	10%	100%	0%	0
pinene	1095	1105	8	8	100%	100%	0%	0
polygon	324	50	1511	421	100%	5%	93%	6
robotarm	402	562	618^f	500^f	99%	2%	97%	2
rocket	502	607	501^f	500^f	98%	2%	97%	1
steering	400	506	9	9	0%	100%	0%	0
torsion1	0	484	8	8	0%	100%	0%	0

7. Conclusions

A new primal-dual shifted penalty-barrier function has been formulated and analyzed for solving inequality constrained nonlinear optimization problems. This function is proposed as a merit function for a primal-dual algorithm for nonlinear optimization with favorable convergence properties. In particular, it has been shown that a limit point of the sequence of iterates may always be found that is either an infeasible stationary point or a complementary approximate Karush-Kuhn-Tucker point, i.e., it satisfies reasonable stopping criteria and is a Karush-Kuhn-Tucker point under the cone continuity property, which is the weakest constraint qualification associated with sequential optimality conditions. At each step of the algorithm, a regularized KKT system is solved to obtain a descent direction for the merit function. Under suitable additional assumptions the method is equivalent to a shifted variant of the primal-dual path-following method in the neighborhood of a solution. Preliminary numerical experiments indicate that the primal-dual shifted penalty-barrier function provides an effective way of ensuring global convergence. The results also illustrate the crucial importance of an effective modification scheme when the KKT matrix does not have the correct inertia.

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. 15

[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. 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. 27

[4] M. G. Breitfeld and D. F. Shanno. Preliminary computational experience with modified log-barrier functions for large-scale nonlinear programming. In W. W. Hager, D. W. Hearn, and P. M. Pardalos, editors, *Large Scale Optimization: State of the Art*, pages 45–67. Springer US, Boston, MA, 1994. 2

[5] M. G. Breitfeld and D. F. Shanno. Computational experience with penalty-barrier methods for nonlinear programming. *Ann. Oper. Res.*, 62:439–463, 1996. 2

[6] 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. 3

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. Technical Report 92/07, Département de Mathématique, Facultés Universitaires de Namur, 1992. 2

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Math. Comput.*, 66(217):261–288, S1–S11, 1997. 2

[9] E. D. Dolan and J. J. Moré. Benchmarking optimization software with COPS. Technical Memorandum ANL/MCS-TM-246, Argonne National Laboratory, Argonne, IL, 2000. 27

[10] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002. 29

[11] E. D. Dolan, J. J. Moré, and T. S. Munson. Benchmarking optimization software with COPS 3.0. Technical Memorandum ANL/MCS-TM-273, Argonne National Laboratory, Argonne, IL, February 2004. 27

[12] A. V. Fiacco. Barrier methods for nonlinear programming. In A. Holzman, editor, *Operations Research Support Methodology*, pages 377–440, New York, 1979. Marcel Dekker. 2

[13] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, Inc., New York-London-Sydney, 1968. 2

[14] A. Forsgren. Inertia-controlling factorizations for optimization algorithms. *Appl. Num. Math.*, 43:91–107, 2002. 23, 28

[15] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.*, 8:1132–1152, 1998. 3, 23, 28

[16] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Rev.*, 44:525–597, 2002. 2

[17] R. M. Freund. Theoretical efficiency of a shifted-barrier-function algorithm for linear programming. *Linear Algebra Appl.*, 152:19–41, 1991. 2

[18] K. R. Frisch. The logarithmic potential method of convex programming. Memorandum of May 13, University Institute of Economics, Oslo, Norway, 1955. 2

[19] E. M. Gertz and P. E. Gill. A primal-dual trust-region algorithm for nonlinear programming. *Math. Program., Ser. B*, 100:49–94, 2004. 3

[20] P. E. Gill, V. Kungurtsev, and D. P. Robinson. Note on the formulation of a shifted primal-dual penalty-barrier method for nonlinear optimization. Center for Computational Mathematics Report CCoM 19-04, University of California, San Diego, 2019. 26, 27, 28

[21] 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. 7

[22] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Shifted barrier methods for linear programming. Report SOL 87-9, Department of Operations Research, Stanford University, Stanford, CA, 1987. 2

[23] P. E. Gill and D. P. Robinson. A primal-dual augmented Lagrangian. *Comput. Optim. Appl.*, 51:1–25, 2012. 24

[24] P. E. Gill and D. P. Robinson. A globally convergent stabilized SQP method. *SIAM J. Optim.*, 23(4):1983–2010, 2013. 23

[25] P. E. Gill and E. Wong. Methods for convex and general quadratic programming. *Math. Prog. Comp.*, 7:71–112, 2015. 23

[26] D. Goldfarb, R. A. Polyak, K. Scheinberg, and I. Yuzefovich. A modified barrier-augmented Lagrangian method for constrained minimization. *Comput. Optim. Appl.*, 14(1):55–74, 1999. 2

[27] N. I. M. Gould. On modified factorizations for large-scale linearly constrained optimization. *SIAM J. Optim.*, 9:1041–1063, 1999. 23

[28] 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. 27

[29] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Lecture Notes in Economics and Mathematical Systems 187. Springer Verlag, Berlin, Heidelberg and New York, 1981. 27

[30] D. L. Jensen and R. A. Polyak. The convergence of a modified barrier method for convex programming. *IBM J. Res. Dev.*, 38(3):307–321, 1994. 2

[31] J. L. Morales. A numerical study of limited memory BFGS methods. *Appl. Math. Lett.*, 15(4):481–487, 2002. 29

[32] S. G. Nash, R. Polyak, and A. Sofer. Numerical comparison of barrier and modified-barrier methods for large-scale bound-constrained optimization. In D. W. Hearn and P. M. Pardalos, editors, *Large-Scale Optimization: State of the Art*, pages 319–338. Kluwer, Dordrecht, 1994. 2

[33] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, New York, 1970. 12

[34] R. A. Polyak. Modified barrier functions (theory and methods). *Math. Program.*, 54(2, Ser. A):177–222, 1992. 2

[35] 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

[36] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: local convergence. *SIAM J. Optim.*, 16(1):32–48 (electronic), 2005. 29

[37] 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. 3, 23

[38] 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. 28, 29

[39] A. Wächter, L. T. Biegler, Y.-D. Lang, and A. Raghunathan. IPOPT: An interior point algorithm for large-scale nonlinear optimization. https://projects.coin-or.org/Ipopt, 2002. 29