SIAM J. OPTIM.
Vol. 17, No. 2, pp. 526–557


# A NEW ACTIVE SET ALGORITHM FOR BOX CONSTRAINED OPTIMIZATION*

WILLIAM W. HAGER† AND HONGCHAO ZHANG†

**Abstract.** An active set algorithm (ASA) for box constrained optimization is developed. The algorithm consists of a nonmonotone gradient projection step, an unconstrained optimization step, and a set of rules for branching between the two steps. Global convergence to a stationary point is established. For a nondegenerate stationary point, the algorithm eventually reduces to unconstrained optimization without restarts. Similarly, for a degenerate stationary point, where the strong second-order sufficient optimality condition holds, the algorithm eventually reduces to unconstrained optimization without restarts. A specific implementation of the ASA is given which exploits the recently developed cyclic Barzilai–Borwein (CBB) algorithm for the gradient projection step and the recently developed conjugate gradient algorithm CG_DESCENT for unconstrained optimization. Numerical experiments are presented using box constrained problems in the CUTEr and MINPACK-2 test problem libraries.

**Key words.** nonmonotone gradient projection, box constrained optimization, active set algorithm, ASA, cyclic BB method, CBB, conjugate gradient method, CG_DESCENT, degenerate optimization

**AMS subject classifications.** 90C06, 90C26, 65Y20

**DOI.** 10.1137/050635225

## 1. Introduction.

We develop an active set method for the box constrained optimization problem

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{B}\}, \tag{1.1}$$

where $f$ is a real-valued, continuously differentiable function defined on the set

$$\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \le \mathbf{x} \le \mathbf{u}\}. \tag{1.2}$$

Here $\mathbf{l} < \mathbf{u}$, and possibly $l_i = -\infty$ or $u_i = \infty$.

The box constrained optimization problem appears in a wide range of applications, including the obstacle problem [67], the elastic-plastic torsion problem [47], optimal design problems [7], journal bearing lubrication [20], inversion problems in elastic wave propagation [6], and molecular conformation analysis [48]. Problem (1.1) is often a subproblem of augmented Lagrangian or penalty schemes for general constrained optimization (see [24, 25, 37, 38, 43, 46, 52, 53, 65]). Thus the development of numerical algorithms to efficiently solve (1.1), especially when the dimension is large, is important in both theory and applications.

We begin with an overview of the development of active set methods. A seminal paper is Polyak's 1969 paper [68] which considers a convex, quadratic cost function. The conjugate gradient method is used to explore a face of the feasible set, and the negative gradient is used to leave a face. Since Polyak's algorithm added or dropped

*Received by the editors July 5, 2005; accepted for publication (in revised form) February 9, 2006; published electronically August 16, 2006. This material is based upon work supported by the National Science Foundation under grant 0203270.

http://www.siam.org/journals/siopt/17-2/63522.html

†Department of Mathematics, University of Florida, P.O. Box 118105, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager; hzhang@math.ufl.edu, http://www.math.ufl.edu/∼hzhang).

only one constraint in each iteration, Dembo and Tulowitzki proposed [32] the conjugate gradient projection (CGP) algorithm which could add and drop many constraints in an iteration. Later, Yang and Tolle [79] further developed this algorithm to obtain finite termination, even when the problem was degenerate at a local minimizer $\mathbf{x}^*$. That is, for some $i$, $x_i^* = l_i$ or $x_i^* = u_i$ and $\nabla f(\mathbf{x}^*)_i = 0$. Another variation of the CGP algorithm, for which there is a rigorous convergence theory, is developed by Wright [77]. Moré and Toraldo [67] point out that when the CGP scheme starts far from the solution, many iterations may be required to identify a suitable working face. Hence, they propose using the gradient projection method to identify a working face, followed by the conjugate gradient method to explore the face. Their algorithm, called GPCG, has finite termination for nondegenerate quadratic problems. Recently, adaptive conjugate gradient algorithms have been developed by Dostál [35, 36] and Dostál, Friedlander, and Santos [38] which have finite termination for a strictly convex quadratic cost function, even when the problem is degenerate.

For general nonlinear functions, some of the earlier research [3, 19, 49, 61, 66, 71] focused on gradient projection methods. To accelerate the convergence, more recent research has developed Newton and trust region methods (see [26] for an in-depth analysis). In [4, 17, 24, 42] superlinear and quadratic convergence is established for nondegenerate problems, while [44, 46, 60, 63] establish analogous convergence results, even for degenerate problems. Although computing a Newton step can be computationally expensive, approximation techniques, such as a sparse, incomplete Cholesky factorization [62], could be used to reduce the computational expense. Nonetheless, for large-dimensional problems or for problems in which the initial guess is far from the solution, the Newton/trust region approach can be inefficient. In cases when the Newton step is unacceptable, a gradient projection step is preferred.

The affine-scaling interior-point method of Coleman and Li [21, 22, 23] (also see Branch, Coleman, and Li [14]) is a different approach to (1.1), related to the trust region algorithm. More recent research on this strategy includes [33, 58, 59, 76, 83]. These methods are based on a reformulation of the necessary optimality conditions obtained by multiplication with a scaling matrix. The resulting system is often solved by Newton-type methods. Without assuming strict complementarity (i.e., for degenerate problems), the affine-scaling interior-point method converges superlinearly or quadratically, for a suitable choice of the scaling matrix, when the strong second-order sufficient optimality condition [70] holds. When the dimension is large, forming and solving the system of equations at each iteration can be time consuming, unless the problem has special structure. Recently, Zhang [83] proposed an interior-point gradient approach for solving the system at each iteration. Convergence results for other interior-point methods applied to more general constrained optimization appear in [39, 40, 78].

The method developed in this paper is an active set algorithm (ASA) which consists of a nonmonotone gradient projection step, an unconstrained optimization step, and a set of rules for branching between the steps. Global convergence to a stationary point is established. For a nondegenerate stationary point, the ASA eventually reduces to unconstrained optimization without restarts. Similarly, for a degenerate stationary point, where the strong second-order sufficient optimality condition holds, the ASA eventually reduces to unconstrained optimization without restarts. If strict complementarity holds and all the constraints are active at a stationary point, then convergence occurs in a finite number of iterations. In general, our analysis does not show that the strictly active constraints are identified in a finite number of iterations;

instead, when the strong second-order sufficient optimality condition holds, we show that the ASA eventually branches to the unconstrained optimization step, and henceforth, the active set does not change. Thus in the limit, the ASA reduces to unconstrained optimization without restarts. Furthermore, if the $i$th constraint in (1.1) is strictly active at a stationary point $\mathbf{x}^*$ (i.e., $\nabla f(\mathbf{x}^*)_i \neq 0$) and the iterates $\mathbf{x}_k$ converge to $\mathbf{x}^*$, then the distance between the $i$th component of $\mathbf{x}_k$ and the associated limit, either $l_i$ or $u_i$, is on the order of the square of the distance between $\mathbf{x}_k$ and $\mathbf{x}^*$.

A specific implementation of the ASA is given, which utilizes our recently developed cyclic Barzilai–Borwein (CBB) algorithm [30] for the gradient projection step and our recently developed conjugate gradient algorithm CG_DESCENT [54, 55, 56, 57] for the unconstrained optimization step. Recent numerical results [27, 45, 50, 51, 74, 81] indicate that in some cases, a nonmonotone line search is superior to a monotone line search. Moreover, gradient methods based on a Barzilai–Borwein (BB) step [2] have exhibited impressive performance in a variety of applications [7, 10, 28, 29, 48, 64, 72]. The BB methods developed in [8, 9, 10, 11, 12, 69] are all based on a Grippo–Lampariello–Lucidi (GLL) type of line search [50]. We have obtained better performance using an adaptive, nonmonotone line search which originates from [31, 75]. Using the adaptive nonmonotone line search, more constraints can be added or dropped in a single iteration. In addition, the cyclic implementation of the BB step [30], in which the same BB stepsize is reused for several iterations, performs better than the original BB step. Hence, in the gradient projection phase of the ASA, we use the CBB scheme of [30] and an adaptive nonmonotone line search.

After detecting a suitable working face, the ASA branches to the unconstrained optimization algorithm, which operates in a lower-dimensional space since some components of $\mathbf{x}$ are fixed. For the numerical experiments, we implement this step using our conjugate gradient algorithm CG_DESCENT. An attractive feature of this algorithm is that the search directions are always sufficient descent directions; furthermore, when the cost function is a strongly convex quadratic, the ASA converges in a finite number of iterations, even when strict complementary slackness does not hold.

Our paper is organized as follows. In section 2 we present the nonmonotone gradient projection algorithm (NGPA) and analyze its global convergence properties. Section 3 presents the ASA and specifies the requirements of the unconstrained optimization algorithm. Section 4 establishes global convergence results for the ASA, while section 5 analyzes local convergence. Section 6 presents numerical comparisons using box constrained problems in the CUTEr [13] and MINPACK-2 [1] test problem libraries. Finally, the appendix gives a specific implementation of the nonmonotone gradient projection method based on our CBB method.

Throughout this paper, we use the following notation. For any set $\mathcal{S}$, $|\mathcal{S}|$ stands for the number of elements (cardinality) of $\mathcal{S}$, while $\mathcal{S}^c$ is the complement of $\mathcal{S}$. $\|\cdot\|$ is the Euclidean norm of a vector. The subscript $k$ is often used to denote the iteration number in an algorithm, while $x_{ki}$ stands for the $i$th component of the iterate $\mathbf{x}_k$. The gradient $\nabla f(\mathbf{x})$ is a row vector, while $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})^\mathsf{T}$ is a column vector; here $^\mathsf{T}$ denotes transpose. The gradient at the iterate $\mathbf{x}_k$ is $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$. We let $\nabla^2 f(\mathbf{x})$ denote the Hessian of $f$ at $\mathbf{x}$. The ball with center $\mathbf{x}$ and radius $\rho$ is denoted $B_\rho(\mathbf{x})$.

**2. Nonmonotone gradient projection algorithm.** In this section, we consider a generalization of (1.1) in which the box $\mathcal{B}$ is replaced with a nonempty, closed convex set $\Omega$:

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \Omega\}. \tag{2.1}$$

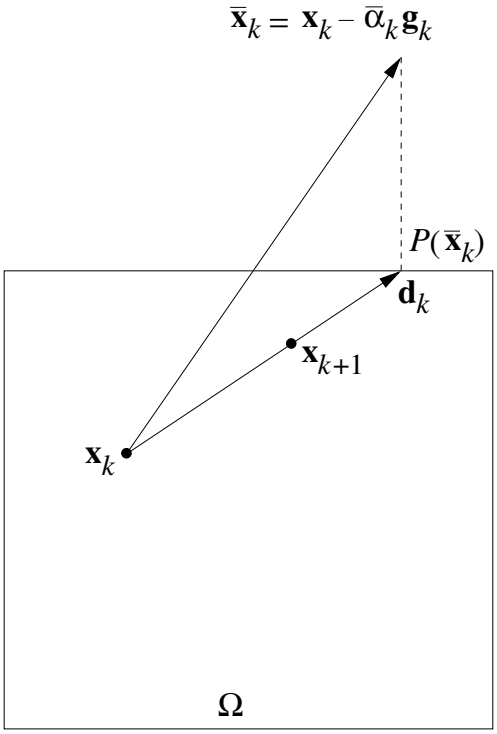


FIG. 2.1. *The gradient projection step.*

We begin with an overview of our gradient projection algorithm. Step $k$ in our algorithm is depicted in Figure 2.1. Here $P$ denotes the projection onto $\Omega$:

$$P(\mathbf{x}) = \arg \min_{\mathbf{y} \in \Omega} \|\mathbf{x} - \mathbf{y}\|. \tag{2.2}$$

Starting at the current iterate $\mathbf{x}_k$, we compute an initial iterate $\bar{\mathbf{x}}_k = \mathbf{x}_k - \bar{\alpha}_k \mathbf{g}_k$. The only constraint on the initial steplength $\bar{\alpha}_k$ is that $\bar{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, where $\alpha_{\min}$ and $\alpha_{\max}$ are fixed, positive constants, independent of $k$. Since the nominal iterate may lie outside $\Omega$, we compute its projection $P(\bar{\mathbf{x}}_k)$ onto $\Omega$. The search direction is $\mathbf{d}_k = P(\bar{\mathbf{x}}_k) - \mathbf{x}_k$, similar to the choice made in SPG2 [11]. Using a nonmonotone line search along the line segment connecting $\mathbf{x}_k$ and $P(\bar{\mathbf{x}}_k)$, we arrive at the new iterate $\mathbf{x}_{k+1}$.

In the statement of the NGPA given below, $f_k^r$ denotes the "reference" function value. A monotone line search corresponds to the choice $f_k^r = f(\mathbf{x}_k)$. The nonmonotone GLL scheme takes $f_k^r = f_k^{\max}$, where

$$f_k^{\max} = \max\{f(\mathbf{x}_{k-i}) : 0 \le i \le \min(k, M-1)\}. \tag{2.3}$$

Here $M > 0$ is a fixed integer, the memory. In the appendix, we give a procedure for choosing the reference function value based on our CBB scheme.

NGPA PARAMETERS.

- $\epsilon \in [0, \infty)$, error tolerance
- $\delta \in (0, 1)$, descent parameter used in Armijo line search

- $\eta \in (0,1)$, decay factor for stepsize in Armijo line search
- $[\alpha_{\min}, \alpha_{\max}] \subset (0,\infty)$, interval containing initial stepsize

NONMONOTONE GRADIENT PROJECTION ALGORITHM (NGPA).

Initialize $k = 0$, $\mathbf{x}_0$ = starting guess, and $f_{-1}^r = f(\mathbf{x}_0)$.

While $\|P(\mathbf{x}_k - \mathbf{g}_k) - \mathbf{x}_k\| > \epsilon$

1. Choose $\overline{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$ and set $\mathbf{d}_k = P(\mathbf{x}_k - \overline{\alpha}_k \mathbf{g}_k) - \mathbf{x}_k$.
2. Choose $f_k^r$ so that $f(\mathbf{x}_k) \le f_k^r \le \max\{f_{k-1}^r, f_k^{\max}\}$ and $f_k^r \le f_k^{\max}$ infinitely often.
3. Let $f_R$ be either $f_k^r$ or $\min\{f_k^{\max}, f_k^r\}$. If $f(\mathbf{x}_k + \mathbf{d}_k) \le f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k$, then $\alpha_k = 1$.
4. If $f(\mathbf{x}_k + \mathbf{d}_k) > f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k$, then $\alpha_k = \eta^j$, where $j > 0$ is the smallest integer such that
$$f(\mathbf{x}_k + \eta^j \mathbf{d}_k) \le f_R + \eta^j \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k. \tag{2.4}$$
5. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ and $k = k + 1$.

End

The condition $f(\mathbf{x}_k) \le f_k^r$ guarantees that the Armijo line search in step 4 can be satisfied. The requirement that "$f_k^r \le f_k^{\max}$ infinitely often" in step 2 is needed for the global convergence result, Theorem 2.2. This is a rather weak requirement which can be satisfied by many strategies. For example, at every $L$ iteration, we could simply set $f_k^r = f_k^{\max}$. Another strategy, closer in spirit to the one used in the numerical experiments, is to choose a decrease parameter $\Delta > 0$ and an integer $L > 0$ and set $f_k^r = f_k^{\max}$ if $f(\mathbf{x}_{k-L}) - f(\mathbf{x}_k) \le \Delta$.

To begin the convergence analysis, recall that $\mathbf{x}^*$ is a stationary point for (2.1) if the first-order optimality condition holds:

$$\nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \ge 0 \quad \text{for all } \mathbf{x} \in \Omega. \tag{2.5}$$

Let $\mathbf{d}^\alpha(\mathbf{x})$, $\alpha \in \mathbb{R}$, be defined in terms of the gradient $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})^{\mathsf{T}}$ as follows:

$$\mathbf{d}^\alpha(\mathbf{x}) = P(\mathbf{x} - \alpha \mathbf{g}(\mathbf{x})) - \mathbf{x}.$$

In the NGPA, the search direction is $\mathbf{d}_k = \mathbf{d}^{\overline{\alpha}_k}(\mathbf{x}_k)$. For unconstrained optimization, $\mathbf{d}^\alpha(\mathbf{x})$ points along the negative gradient at $\mathbf{x}$ when $\alpha > 0$. Some properties of $P$ and $\mathbf{d}^\alpha$ are summarized below.

PROPOSITION 2.1 (Properties of $P$ and $\mathbf{d}^\alpha$).

P1. $(P(\mathbf{x}) - \mathbf{x})^{\mathsf{T}}(\mathbf{y} - P(\mathbf{x})) \ge 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \Omega$.

P2. $(P(\mathbf{x}) - P(\mathbf{y}))^{\mathsf{T}}(\mathbf{x} - \mathbf{y}) \ge \|P(\mathbf{x}) - P(\mathbf{y})\|^2$ *for all* $\mathbf{x}$ *and* $\mathbf{y} \in \mathbb{R}^n$.

P3. $\|P(\mathbf{x}) - P(\mathbf{y})\| \le \|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}$ *and* $\mathbf{y} \in \mathbb{R}^n$.

P4. $\|\mathbf{d}^\alpha(\mathbf{x})\|$ *is nondecreasing in* $\alpha > 0$ *for any* $\mathbf{x} \in \Omega$.

P5. $\|\mathbf{d}^\alpha(\mathbf{x})\|/\alpha$ *is nonincreasing in* $\alpha > 0$ *for any* $\mathbf{x} \in \Omega$.

P6. $\mathbf{g}(\mathbf{x})^{\mathsf{T}} \mathbf{d}^\alpha(\mathbf{x}) \le -\|\mathbf{d}^\alpha(\mathbf{x})\|^2/\alpha$ *for any* $\mathbf{x} \in \Omega$ *and* $\alpha > 0$.

P7. *For any* $\mathbf{x} \in \Omega$ *and* $\alpha > 0$, $\mathbf{d}^\alpha(\mathbf{x}) = \mathbf{0}$ *if and only if* $\mathbf{x}$ *is a stationary point for* (2.1).

P8. *Suppose* $\mathbf{x}^*$ *is a stationary point for* (2.1). *If for some* $\mathbf{x} \in \mathbb{R}^n$ *there exist positive scalars* $\lambda$ *and* $\gamma$ *such that*
$$(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*))^{\mathsf{T}}(\mathbf{x} - \mathbf{x}^*) \ge \gamma \|\mathbf{x} - \mathbf{x}^*\|^2 \tag{2.6}$$
*and*
$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*)\| \le \lambda \|\mathbf{x} - \mathbf{x}^*\|, \tag{2.7}$$

*then we have*

$$\|\mathbf{x}-\mathbf{x}^*\| \le \left(\frac{1+\lambda}{\gamma}\right)\|\mathbf{d}^1(\mathbf{x})\|.$$

*Proof.* P1 is the first-order optimality condition associated with the solution of (2.2). Replacing $\mathbf{y}$ with $P(\mathbf{y})$ in P1 gives

$$(P(\mathbf{x})-\mathbf{x})^{\mathsf{T}}(P(\mathbf{y})-P(\mathbf{x})) \ge 0.$$

Adding this to the corresponding inequality obtained by interchanging $\mathbf{x}$ and $\mathbf{y}$ yields P2 (see [80]). P3 is the nonexpansive property of a projection (for example, see [5, Prop. 2.1.3]). P4 is given in [73]. For P5, see [5, Lem. 2.3.1]. P6 is obtained from P1 by replacing $\mathbf{x}$ with $\mathbf{x}-\alpha\mathbf{g}(\mathbf{x})$ and replacing $\mathbf{y}$ with $\mathbf{x}$. If $\mathbf{x}^*$ is a stationary point satisfying (2.5), then P6 with $\mathbf{x}$ replaced by $\mathbf{x}^*$ yields $\mathbf{d}^\alpha(\mathbf{x}^*)=\mathbf{0}$. Conversely, if $\mathbf{d}^\alpha(\mathbf{x}^*)=\mathbf{0}$, then by P1 with $\mathbf{x}$ replaced by $\mathbf{x}^*-\alpha\mathbf{g}(\mathbf{x}^*)$, we obtain

$$0 \le \alpha\mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{y}-P(\mathbf{x}^*-\alpha\mathbf{g}(\mathbf{x}^*)) = \alpha\mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{y}-\mathbf{x}^*),$$

which implies that $\mathbf{x}^*$ is a stationary point (see [5, Fig. 2.3.2]).

Finally, let us consider P8. Replacing $\mathbf{x}$ with $\mathbf{x}-\mathbf{g}(\mathbf{x})$ and replacing $\mathbf{y}$ with $\mathbf{x}^*$ in P1 gives

$$[P(\mathbf{x}-\mathbf{g}(\mathbf{x}))-\mathbf{x}+\mathbf{g}(\mathbf{x})]^{\mathsf{T}}[\mathbf{x}^*-P(\mathbf{x}-\mathbf{g}(\mathbf{x}))] \ge 0. \tag{2.8}$$

By the definition of $\mathbf{d}^\alpha(\mathbf{x})$, (2.8) is equivalent to

$$[\mathbf{d}^1(\mathbf{x})+\mathbf{g}(\mathbf{x})]^{\mathsf{T}}[\mathbf{x}^*-\mathbf{x}-\mathbf{d}^1(\mathbf{x})] \ge 0.$$

Rearranging this and utilizing (2.6) gives

$$\begin{aligned}\mathbf{d}^1(\mathbf{x})^{\mathsf{T}}(\mathbf{x}^*-\mathbf{x})-\mathbf{g}(\mathbf{x})^{\mathsf{T}}\mathbf{d}^1(\mathbf{x})-\|\mathbf{d}^1(\mathbf{x})\|^2 &\ge \mathbf{g}(\mathbf{x})^{\mathsf{T}}(\mathbf{x}-\mathbf{x}^*)\\ &\ge \gamma\|\mathbf{x}-\mathbf{x}^*\|^2+\mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{x}-\mathbf{x}^*).\end{aligned} \tag{2.9}$$

Focusing on the terms involving $\mathbf{g}$ and utilizing (2.7), we have

$$\begin{aligned}\mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{x}^*-\mathbf{x})-\mathbf{g}(\mathbf{x})^{\mathsf{T}}\mathbf{d}^1(\mathbf{x}) &\le \lambda\|\mathbf{x}-\mathbf{x}^*\|\,\|\mathbf{d}^1(\mathbf{x})\|+\mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{x}^*-\mathbf{x}-\mathbf{d}^1(\mathbf{x}))\\ &= \lambda\|\mathbf{x}-\mathbf{x}^*\|\,\|\mathbf{d}^1(\mathbf{x})\|+\mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}[\mathbf{x}^*-P(\mathbf{x}-\mathbf{g}(\mathbf{x}))]\\ &\le \lambda\|\mathbf{x}-\mathbf{x}^*\|\,\|\mathbf{d}^1(\mathbf{x})\|\end{aligned} \tag{2.10}$$

by (2.5), since $P(\mathbf{x}-\mathbf{g}(\mathbf{x})) \in \Omega$. Combining (2.9) and (2.10), the proof is complete. ∎

Next, we establish a convergence result for the NGPA.

THEOREM 2.2. *Let $\mathcal{L}$ be the level set defined by*

$$\mathcal{L}=\{\mathbf{x}\in\Omega : f(\mathbf{x}) \le f(\mathbf{x}_0)\}. \tag{2.11}$$

*We assume the following conditions hold:*

G1. *$f$ is bounded from below on $\mathcal{L}$ and $d_{\max}=\sup_k\|\mathbf{d}_k\|<\infty$.*

G2. *If $\bar{\mathcal{L}}$ is the collection of $\mathbf{x}\in\Omega$ whose distance to $\mathcal{L}$ is at most $d_{\max}$, then $\nabla f$ is Lipschitz continuous on $\bar{\mathcal{L}}$.*

*Then either the NGPA with $\epsilon = 0$ terminates in a finite number of iterations at a stationary point, or we have*

$$\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0.$$

*Proof.* By P6, the search direction $\mathbf{d}_k$ generated in step 1 of the NGPA is a descent direction. Since $f_k^r \geq f(\mathbf{x}_k)$ and $\delta < 1$, the Armijo line search condition (2.4) is satisfied for $j$ sufficiently large. We now show that $\mathbf{x}_k \in \mathcal{L}$ for each $k$. Since $f_0^{\max} = f_{-1}^r = f(\mathbf{x}_0)$, step 2 of the NGPA implies that $f_0^r \leq f(\mathbf{x}_0)$. Proceeding by induction, suppose that for some $k \geq 0$, we have

$$f_j^r \leq f(\mathbf{x}_0) \quad \text{and} \quad f_j^{\max} \leq f(\mathbf{x}_0) \tag{2.12}$$

for all $j \in [0, k]$. Again, since the search direction $\mathbf{d}_k$ generated in step 1 of the NGPA is a descent direction, it follows from steps 3 and 4 of the NGPA and the induction hypothesis that

$$f(\mathbf{x}_{k+1}) \leq f_k^r \leq f(\mathbf{x}_0). \tag{2.13}$$

Hence, $f_{k+1}^{\max} \leq f(\mathbf{x}_0)$ and $f_{k+1}^r \leq \max\{f_k^r, f_{k+1}^{\max}\} \leq f(\mathbf{x}_0)$. This completes the induction. Thus (2.12) holds for all $j$. Consequently, we have $f_R \leq f(\mathbf{x}_0)$ in steps 3 and 4 of the NGPA. Again, since the search direction $\mathbf{d}_k$ generated in step 1 of the NGPA is a descent direction, it follows from steps 3 and 4 that $f(\mathbf{x}_k) \leq f(\mathbf{x}_0)$, which implies that $\mathbf{x}_k \in \mathcal{L}$ for each $k$.

Let $\lambda$ be the Lipschitz constant for $\nabla f$ on $\bar{\mathcal{L}}$. As in [81, Lem. 2.1], we have

$$\alpha_k \geq \min\left\{1, \left(\frac{2\eta(1-\delta)}{\lambda}\right)\frac{|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}\right\} \tag{2.14}$$

for all $k$. By P6,

$$|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k| \geq \frac{\|\mathbf{d}_k\|^2}{\overline{\alpha}_k} \geq \frac{\|\mathbf{d}_k\|^2}{\alpha_{\max}}.$$

It follows from (2.14) that

$$\alpha_k \geq \min\left\{1, \left(\frac{2\eta(1-\delta)}{\lambda\alpha_{\max}}\right)\right\} := c. \tag{2.15}$$

By steps 3 and 4 of the NGPA and P6, we conclude that

$$f(\mathbf{x}_{k+1}) \leq f_k^r + \delta c\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \leq f_k^r - \delta c\|\mathbf{d}_k\|^2/\overline{\alpha}_k \leq f_k^r - \delta c\|\mathbf{d}_k\|^2/\alpha_{\max}. \tag{2.16}$$

We now prove that $\liminf_{k\to\infty} \|\mathbf{d}_k\| = 0$. Suppose, to the contrary, that there exists a constant $\gamma > 0$ such that $\|\mathbf{d}_k\| \geq \gamma$ for all $k$. By (2.16), we have

$$f(\mathbf{x}_{k+1}) \leq f_k^r - \tau, \quad \text{where } \tau = \delta c\gamma^2/\alpha_{\max}. \tag{2.17}$$

Let $k_i$, $i = 0, 1, \ldots$, denote an increasing sequence of integers with the property that $f_j^r \leq f_j^{\max}$ for $j = k_i$ and $f_j^r \leq f_{j-1}^r$ when $k_i < j < k_{i+1}$. Such a sequence exists by the requirement on $f_k^r$ given in step 2 of the NGPA. Hence, we have

$$f_j^r \leq f_{k_i}^r \leq f_{k_i}^{\max} \quad \text{when } k_i \leq j < k_{i+1}. \tag{2.18}$$

By (2.17) it follows that

$$f(\mathbf{x}_j) \le f_{j-1}^r - \tau \le f_{k_i}^{\max} - \tau \quad \text{when } k_i < j \le k_{i+1}. \tag{2.19}$$

It follows that

$$f_{k_{i+1}}^r \le f_{k_{i+1}}^{\max} \le f_{k_i}^{\max}. \tag{2.20}$$

Hence, if $a = k_{i_1}$ and $b = k_{i_2}$, where $i_1 > i_2$ and $a - b > M$, then by (2.18)–(2.20) we have

$$f_a^{\max} = \max_{0 \le j < M} f(\mathbf{x}_{a-j}) \le \max_{1 \le j \le M} f_{a-j}^r - \tau \le f_b^{\max} - \tau.$$

Since the sequence $k_i$, $i = 0, 1, \ldots$, is infinite, this contradicts the fact that $f$ is bounded from below. Consequently, $\liminf_{k\to\infty} \|\mathbf{d}_k\| = 0$. By P4 and P5, it follows that

$$\|\mathbf{d}_k\| \ge \min\{\alpha_{\min}, 1\} \|\mathbf{d}^1(\mathbf{x}_k)\|.$$

Thus $\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0$. □

Recall that $f$ is strongly convex on $\Omega$ if there exists a scalar $\gamma > 0$ such that

$$f(\mathbf{x}) \ge f(\mathbf{y}) + \nabla f(\mathbf{y})(\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{y}\|^2 \tag{2.21}$$

for all $\mathbf{x}$ and $\mathbf{y} \in \Omega$. Interchanging $\mathbf{x}$ and $\mathbf{y}$ in (2.21) and adding, we obtain the (usual) monotonicity condition

$$(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))(\mathbf{y} - \mathbf{x}) \ge \gamma \|\mathbf{y} - \mathbf{x}\|^2. \tag{2.22}$$

For a strongly convex function, (2.1) has a unique minimizer $\mathbf{x}^*$, and the conclusion of Theorem 2.2 can be strengthened as follows.

COROLLARY 2.3. *Suppose $f$ is strongly convex and twice continuously differentiable on $\Omega$, and there is a positive integer $L$ with the property that for each $k$, there exists $j \in [k, k + L)$ such that $f_j^r \le f_j^{\max}$. Then the iterates $\mathbf{x}_k$ of the NGPA with $\epsilon = 0$ converge to the global minimizer $\mathbf{x}^*$.*

*Proof.* As shown at the start of the proof of Theorem 2.2, $f(\mathbf{x}_k) \le f(\mathbf{x}_0)$ for each $k$. Hence, $\mathbf{x}_k$ lies in the level set $\mathcal{L}$ defined in (2.11). Since $f$ is strongly convex, $\mathcal{L}$ is a bounded set; since $f$ is twice continuously differentiable, $\|\nabla f(\mathbf{x}_k)\|$ is bounded uniformly in $k$. For any $\mathbf{x} \in \Omega$, we have $P(\mathbf{x}) = \mathbf{x}$. By P3, it follows that

$$\|\mathbf{d}^\alpha\| = \|P(\mathbf{x} - \alpha \mathbf{g}(\mathbf{x})) - \mathbf{x}\| = \|P(\mathbf{x} - \alpha \mathbf{g}(\mathbf{x})) - P(\mathbf{x})\| \le \alpha \|\mathbf{g}(\mathbf{x})\|.$$

Since $\bar{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, $d_{\max} = \sup_k \|\mathbf{d}_k\| < \infty$. Consequently, the set $\bar{\mathcal{L}}$ defined in G2 is bounded. Again, since $f$ is twice continuously differentiable, $\nabla f$ is Lipschitz continuous on $\bar{\mathcal{L}}$. By assumption, $f_k^r \le f_k^{\max}$ infinitely often. Consequently, the hypotheses of Theorem 2.2 are satisfied, and either the NGPA with $\epsilon = 0$ terminates in a finite number of iterations at a stationary point, or we have

$$\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0. \tag{2.23}$$

Since $f$ is strongly convex on $\Omega$, $\mathbf{x}^*$ is the unique stationary point for (2.1). Hence, when the iterates converge in a finite number of steps, they converge to $\mathbf{x}^*$. Otherwise,

(2.23) holds, in which case there exists an infinite sequence $l_1 < l_2 < \cdots$ such that $\|\mathbf{d}^1(\mathbf{x}_{l_j})\|$ approaches zero as $j$ tends to $\infty$. Since (2.22) holds, it follows from P8 that $\mathbf{x}_{l_j}$ approaches $\mathbf{x}^*$ as $j$ tends to $\infty$. By P4 and P5, we have

$$\|\mathbf{d}^\alpha(\mathbf{x})\| \le \max\{1, \alpha\}\|\mathbf{d}^1(\mathbf{x})\|.$$

Since $\bar{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, it follows that

$$\|\mathbf{d}_k\| \le \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k)\|.$$

Since the stepsize $\alpha_k \in (0, 1]$, we deduce that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| = \alpha_k\|\mathbf{d}_k\| \le \|\mathbf{d}_k\| \le \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k)\|. \tag{2.24}$$

By P3, $P$ is continuous; consequently, $\mathbf{d}^\alpha(\mathbf{x})$ is a continuous function of $\mathbf{x}$. The continuity of $\mathbf{d}^\alpha(\cdot)$ and $f(\cdot)$ combined with (2.24) and the fact that $\mathbf{x}_{l_j}$ converges to $\mathbf{x}^*$ implies that for any $\delta > 0$ and for $j$ sufficiently large, we have

$$f(\mathbf{x}_k) \le f(\mathbf{x}^*) + \delta \quad \text{for all } k \in [l_j, l_j + M + L].$$

By the definition of $f_k^{\max}$,

$$f_k^{\max} \le f(\mathbf{x}^*) + \delta \quad \text{for all } k \in [l_j + M, l_j + M + L]. \tag{2.25}$$

As in the proof of Theorem 2.2, let $k_i$, $i = 0, 1, \ldots$, denote an increasing sequence of integers with the property that $f_j^r \le f_j^{\max}$ for $j = k_i$ and $f_j^r \le f_{j-1}^r$ when $k_i < j < k_{i+1}$. As shown in (2.20),

$$f_{k_{i+1}}^{\max} \le f_{k_i}^{\max} \tag{2.26}$$

for each $i$. The assumption that for each $k$, there exists $j \in [k, k + L)$ such that $f_j^r \le f_j^{\max}$, implies that

$$k_{i+1} - k_i \le L. \tag{2.27}$$

Combining (2.25) and (2.27), for each $l_j$, there exists some $k_i \in [l_j + M, l_j + M + L]$ and

$$f_{k_i}^{\max} \le f(\mathbf{x}^*) + \delta. \tag{2.28}$$

Since $\delta$ was arbitrary, it follows from (2.26) and (2.28) that

$$\lim_{i\to\infty} f_{k_i}^{\max} = f(\mathbf{x}^*); \tag{2.29}$$

the convergence is monotone by (2.26). By the choice of $k_i$ and by the inequality $f(\mathbf{x}_k) \le f_k^r$ in step 2, we have

$$f(\mathbf{x}_k) \le f_k^r \le f_{k_i}^{\max} \quad \text{for all } k \ge k_i. \tag{2.30}$$

Combining (2.29) and (2.30),

$$\lim_{k\to\infty} f(\mathbf{x}_k) = f(\mathbf{x}^*). \tag{2.31}$$

Together, (2.5) and (2.21) yield

$$f(\mathbf{x}_k) \ge f(\mathbf{x}^*) + \frac{\gamma}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2. \tag{2.32}$$

Combining this with (2.31), the proof is complete. ∎

**3. The active set algorithm.** Starting with this section, we focus on the box constrained problem (1.1). To simplify the exposition, we consider the special case when $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \boldsymbol{\infty}$:

$$\min \{f(\mathbf{x}) : \mathbf{x} \geq \mathbf{0}\}.$$

We emphasize that the analysis and algorithm apply to the general box constrained problem (1.1) with both upper and lower bounds.

Although the gradient projection scheme of the NGPA has an attractive global convergence theory, the convergence rate can be slow in a neighborhood of a local minimizer. In contrast, for unconstrained optimization, the conjugate gradient algorithm often exhibits superlinear convergence in a neighborhood of a local minimizer. We develop an ASA which uses the NGPA to identify active constraints, and which uses an unconstrained optimization algorithm, such as the CG_DESCENT scheme in [54, 55, 57, 56], to optimize $f$ over a face identified by the NGPA.

We begin with some notation. For any $\mathbf{x} \in \Omega$, let $\mathcal{A}(\mathbf{x})$ and $\mathcal{I}(\mathbf{x})$ denote the active and inactive indices, respectively:

$$\mathcal{A}(\mathbf{x}) = \{i \in [1, n] : x_i = 0\},$$
$$\mathcal{I}(\mathbf{x}) = \{i \in [1, n] : x_i > 0\}.$$

The active indices are further subdivided into those indices satisfying strict complementarity and the degenerate indices:

$$\mathcal{A}_+(\mathbf{x}) = \{i \in \mathcal{A}(\mathbf{x}) : g_i(\mathbf{x}) > 0\},$$
$$\mathcal{A}_0(\mathbf{x}) = \{i \in \mathcal{A}(\mathbf{x}) : g_i(\mathbf{x}) = 0\}.$$

We let $\mathbf{g}_I(\mathbf{x})$ denote the vector whose components associated with the set $\mathcal{I}(\mathbf{x})$ are identical to those of $\mathbf{g}(\mathbf{x})$, while the components associated with $\mathcal{A}(\mathbf{x})$ are zero:

$$g_{Ii}(\mathbf{x}) = \begin{cases} 0 & \text{if } x_i = 0, \\ g_i(\mathbf{x}) & \text{if } x_i > 0. \end{cases}$$

An important feature of our algorithm is that we try to distinguish between active constraints satisfying strict complementarity and active constraints that are degenerate using an identification strategy, which is related to the idea of an identification function introduced in [41]. Given fixed parameters $\alpha \in (0, 1)$ and $\beta \in (1, 2)$, we define the (undecided index) set $\mathcal{U}$ at $\mathbf{x} \in \mathcal{B}$ as follows:

$$\mathcal{U}(\mathbf{x}) = \{i \in [1, n] : |g_i(\mathbf{x})| \geq \|\mathbf{d}^1(\mathbf{x})\|^\alpha \text{ and } x_i \geq \|\mathbf{d}^1(\mathbf{x})\|^\beta\}.$$

In the numerical experiments, we take $\alpha = 1/2$ and $\beta = 3/2$. In practice, $\mathcal{U}$ is almost always empty when we reach a neighborhood of a minimizer, and the specific choice of $\alpha$ and $\beta$ does not have a significant effect on convergence. The introduction of the $\mathcal{U}$ set leads to a strong local convergence theory developed in section 5.

The indices in $\mathcal{U}$ correspond to components of $\mathbf{x}$ for which the associated gradient component $g_i(\mathbf{x})$ is relatively large, while $x_i$ is not close to 0 (in the sense that $x_i \geq \|\mathbf{d}^1(\mathbf{x})\|^\beta$). When the set $\mathcal{U}$ of uncertain indices is empty, we feel that the indices with large associated gradient components are almost identified. In this case we prefer the unconstrained optimization algorithm.

Although our numerical experiments are based on the conjugate gradient code CG_DESCENT, a broad class of unconstrained optimization algorithms (UAs) can

be applied. The following requirements for the UA are sufficient for establishing the convergence results that follow. Conditions U1–U3 are sufficient for global convergence, while U1–U4 are sufficient for the local convergence analysis. Condition U4 could be replaced with another descent condition for the initial line search; however, the analysis of section 5 has been carried out under U4.

Unconstrained algorithm (UA) requirements.

U1. $\mathbf{x}_k \geq \mathbf{0}$ and $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ for each $k$.

U2. $\mathcal{A}(\mathbf{x}_k) \subset \mathcal{A}(\mathbf{x}_{k+1})$ for each $k$.

U3. If $\mathcal{A}(\mathbf{x}_{j+1}) = \mathcal{A}(\mathbf{x}_j)$ for $j \geq k$, then $\liminf_{j\to\infty} \|\mathbf{g}_I(\mathbf{x}_j)\| = 0$.

U4. Whenever the UA is started, $\mathbf{x}_{k+1} = P(\mathbf{x}_k - \alpha_k \mathbf{g}_I(\mathbf{x}_k))$, where $\alpha_k$ is obtained from a Wolfe line search. That is, $\alpha_k$ is chosen to satisfy

$$\phi(\alpha_k) \leq \phi(0) + \delta\alpha_k\phi'(0) \quad \text{and} \quad \phi'(\alpha_k) \geq \sigma\phi'(0), \tag{3.1}$$

where

$$\phi(\alpha) = f(P(\mathbf{x}_k - \alpha\mathbf{g}_I(\mathbf{x}_k))), \quad 0 < \delta < \sigma < 1. \tag{3.2}$$

Condition U1 implies that the UA is a monotone algorithm, so that the cost function can only decrease in each iteration. Condition U2 concerns how the algorithm behaves when an infeasible iterate is generated. Condition U3 describes the global convergence of the UA when the active set does not change. In U4, $\phi'(\alpha)$ is the derivative from the right side of $\alpha$; $\alpha_k$ exists since $\phi$ is piecewise smooth with a finite number of discontinuities in its derivative, and $\phi'(\alpha)$ is continuous at $\alpha = 0$.

Our ASA is presented in Figure 3.1. In the first step of the algorithm, we execute the NGPA until we feel that the active constraints satisfying strict complementarity have been identified. In step 2, we execute the UA until a subproblem has been solved (step 2a). When new constraints become active in step 2b, we may decide to restart either the NGPA or the UA. By restarting the NGPA, we mean that $\mathbf{x}_0$ in the NGPA is identified with the current iterate $\mathbf{x}_k$. By restarting the UA, we mean that iterates are generated by the UA using the current iterate as the starting point.

**4. Global convergence.** We begin with a global convergence result for the ASA.

Theorem 4.1. *Let $\mathcal{L}$ be the level set defined by*

$$\mathcal{L} = \{\mathbf{x} \in \mathcal{B} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}.$$

*Assume the following conditions hold:*

A1. *$f$ is bounded from below on $\mathcal{L}$ and $d_{\max} = \sup_k \|\mathbf{d}_k\| < \infty$.*

A2. *If $\bar{\mathcal{L}}$ is the collection of $\mathbf{x} \in \mathcal{B}$ whose distance to $\mathcal{L}$ is at most $d_{\max}$, then $\nabla f$ is Lipschitz continuous on $\bar{\mathcal{L}}$.*

A3. *The UA satisfies* U1–U3.

*Then either the ASA with $\epsilon = 0$ terminates in a finite number of iterations at a stationary point, or we have*

$$\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0. \tag{4.1}$$

*Proof.* If only the NGPA is performed for large $k$, then (4.1) follows from Theorem 2.2. If only the UA is performed for large $k$, then by U2, the active sets $\mathcal{A}(\mathbf{x}_k)$ must approach a limit. Since $\mu$ does not change in the UA, it follows from U3 and the condition $\|\mathbf{g}_I(\mathbf{x}_k)\| \geq \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$ that (4.1) holds. Finally, suppose that the NGPA

ASA PARAMETERS.

- $\epsilon \in [0, \infty)$, error tolerance, stop when $\|\mathbf{d}^1(\mathbf{x}_k)\| \leq \epsilon$
- $\mu \in (0, 1)$, $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu \|\mathbf{d}^1(\mathbf{x}_k)\|$ implies subproblem solved
- $\rho \in (0, 1)$, decay factor for $\mu$ tolerance
- $n_1 \in [1, n)$, number of repeated $\mathcal{A}(\mathbf{x}_k)$ before switch from the NGPA to the UA
- $n_2 \in [1, n)$, used in switch from the UA to the NGPA

ACTIVE SET ALGORITHM (ASA).

1. While $\|\mathbf{d}^1(\mathbf{x}_k)\| > \epsilon$ execute the NGPA and check the following:
   a. If $\mathcal{U}(\mathbf{x}_k) = \emptyset$, then
      If $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu \|\mathbf{d}^1(\mathbf{x}_k)\|$, then $\mu = \rho\mu$.
      Otherwise, goto step 2.
   b. Else if $\mathcal{A}(\mathbf{x}_k) = \mathcal{A}(\mathbf{x}_{k-1}) = \cdots = \mathcal{A}(\mathbf{x}_{k-n_1})$, then
      If $\|\mathbf{g}_I(\mathbf{x}_k)\| \geq \mu \|\mathbf{d}^1(\mathbf{x}_k)\|$, then goto step 2.

   End

2. While $\|\mathbf{d}^1(\mathbf{x}_k)\| > \epsilon$ execute the UA and check the following:
   a. If $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu \|\mathbf{d}^1(\mathbf{x}_k)\|$, then restart the NGPA (step 1).
   b. If $|\mathcal{A}(\mathbf{x}_{k-1})| < |\mathcal{A}(\mathbf{x}_k)|$, then
      If $\mathcal{U}(\mathbf{x}_k) = \emptyset$ or $|\mathcal{A}(\mathbf{x}_k)| > |\mathcal{A}(\mathbf{x}_{k-1})| + n_2$, restart the UA at $\mathbf{x}_k$.
      Else restart the NGPA.
      End

   End

FIG. 3.1. *Statement of the ASA.*

is restarted an infinite number of times at $k_1 < k_2 < \cdots$ and that it terminates at $k_1 + l_1 < k_2 + l_2 < \cdots$, respectively. Thus $k_i < k_i + l_i \leq k_{i+1}$ for each $i$. If (4.1) does not hold, then by (2.19) and (2.20), we have

$$f(\mathbf{x}_{k_i + l_i}) \leq f(\mathbf{x}_{k_i}) - \tau. \tag{4.2}$$

By U1,

$$f(\mathbf{x}_{k_{i+1}}) \leq f(\mathbf{x}_{k_i + l_i}). \tag{4.3}$$

Combining (4.2) and (4.3), we have $f(\mathbf{x}_{k_{i+1}}) \leq f(\mathbf{x}_{k_i}) - \tau$, which contradicts the assumption that $f$ is bounded from below. □

When $f$ is strongly convex, the entire sequence of iterates converges to the global minimizer $\mathbf{x}^*$, as stated in the following corollary. Since the proof of this result relies on the local convergence analysis, the proof is delayed until the end of section 5.

COROLLARY 4.2. *If $f$ is strongly convex and twice continuously differentiable on $\mathcal{B}$, and assumption* A3 *of Theorem* 4.1 *is satisfied, then the iterates $\mathbf{x}_k$ of the ASA with $\epsilon = 0$ converge to the global minimizer $\mathbf{x}^*$.*

**5. Local convergence.** In the next series of lemmas, we analyze local convergence properties of the ASA. We begin by focusing on nondegenerate stationary points; that is, stationary points $\mathbf{x}^*$ with the property that $g_i(\mathbf{x}^*) > 0$ whenever $x_i^* = 0$.

**5.1. Nondegenerate problems.** In this case, it is relatively easy to show that the ASA eventually performs only the UA without restarts. The analogous result for degenerate problems is established in section 5.2.

THEOREM 5.1. *If $f$ is continuously differentiable, $0 < \mu \le 1$, and the iterates $\mathbf{x}_k$ generated by the ASA with $\epsilon = 0$ converge to a nondegenerate stationary point $\mathbf{x}^*$, then after a finite number of iterations, the ASA performs only the UA without restarts.*

*Proof.* Since $\mathbf{x}^*$ is a nondegenerate stationary point and $f$ is continuously differentiable, there exists $\rho > 0$ with the property that for all $\mathbf{x} \in \mathcal{B}_\rho(\mathbf{x}^*)$, we have

$$g_i(\mathbf{x}) > 0 \text{ if } i \in \mathcal{A}(\mathbf{x}^*) \quad \text{and} \quad x_i > 0 \text{ if } i \in \mathcal{A}(\mathbf{x}^*)^c. \tag{5.1}$$

Let $k_+$ be chosen large enough that $\mathbf{x}_k \in \mathcal{B}_\rho(\mathbf{x}^*)$ for all $k \ge k_+$. If $k \ge k_+$ and $x_{ki} = 0$, then $d_{ki} = 0$ in step 1 of the NGPA. Hence, $x_{k+1,i} = 0$ if $\mathbf{x}_{k+1}$ is generated by the NGPA. By U2, the UA cannot free a bound constraint. It follows that if $k \ge k_+$ and $x_{ki} = 0$, then $x_{ji} = 0$ for all $j \ge k$. Consequently, there exists an index $K \ge k_+$ with the property that $\mathcal{A}(\mathbf{x}_k) = \mathcal{A}(\mathbf{x}_j)$ for all $j \ge k \ge K$.

For any index $i$, $|d_i^1(\mathbf{x})| \le |g_i(\mathbf{x})|$. Suppose $\mathbf{x} \in \mathcal{B}_\rho(\mathbf{x}^*)$; by (5.1), $d_i^1(\mathbf{x}) = 0$ if $x_i = 0$. Hence,

$$\|\mathbf{d}^1(\mathbf{x})\| \le \|\mathbf{g}_I(\mathbf{x})\| \tag{5.2}$$

for all $\mathbf{x} \in \mathcal{B}_\rho(\mathbf{x}^*)$. If $k > K + n_1$, then in step 1b of the ASA, it follows from (5.2) and the assumption $\mu \in (0, 1]$ that the NGPA will branch to step 2 (UA). In step 2, the condition "$\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$" of step 2a is never satisfied by (5.2). Moreover, the condition "$|\mathcal{A}(\mathbf{x}_{k-1})| < |\mathcal{A}(\mathbf{x}_k)|$" of step 2b is never satisfied since $k > K$. Hence, the iterates never branch from the UA to the NPGA and the UA is never restarted. ∎

**5.2. Degenerate problems.** We now focus on degenerate problems and show that a result analogous to Theorem 5.1 holds under the strong second-order sufficient optimality condition. We begin with a series of preliminary results.

LEMMA 5.2. *If $f$ is twice-continuously differentiable and there exists an infinite sequence of iterates $\mathbf{x}_k$ generated by the ASA with $\epsilon = 0$ converging to a stationary point $\mathbf{x}^*$, $\mathbf{x}_k \ne \mathbf{x}^*$ for each $k$, then for each $i \in \mathcal{A}_+(\mathbf{x}^*)$ we have*

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} < \infty. \tag{5.3}$$

*Proof.* Assume that $\mathcal{A}_+(\mathbf{x}^*)$ is nonempty; otherwise there is nothing to prove. Let $k_+$ be chosen large enough that $g_i(\mathbf{x}_k) > 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$ and $k \ge k_+$. Since $f$ is twice-continuously differentiable, $\nabla f$ is Lipschitz continuous in a neighborhood of $\mathbf{x}^*$. Choose $\rho > 0$ and let $\lambda$ be the Lipschitz constant for $\nabla f$ in the ball $B_\rho(\mathbf{x}^*)$ with center $\mathbf{x}^*$ and radius $\rho$. Since $\mathbf{d}^1(\mathbf{x}^*) = \mathbf{0}$, it follows from the continuity of $\mathbf{d}^1(\cdot)$ that $\mathbf{d}_k$ tends to $\mathbf{0}$ (see (2.24)). Choose $k_+$ large enough that the ball with center $\mathbf{x}_k$ and radius $\|\mathbf{d}_k\|$ is contained in $B_\rho(\mathbf{x}^*)$ for all $k \ge k_+$. If $x_{li} = 0$ for some $i \in \mathcal{A}_+(\mathbf{x}^*)$ and $l \ge k_+$, then by the definition of $\mathbf{d}_k$ in the NGPA, we have $d_{ki} = 0$ for all $k \ge l$. Hence, $x_{ki} = 0$ for each $k \ge l$ in the NGPA. Likewise, in the UA it follows from U2 that $x_{ji} = 0$ for $j \ge k$ when $x_{ki} = 0$; that is, the UA does not free an active constraint. In other words, when an index $i \in \mathcal{A}_+(\mathbf{x}^*)$ becomes active at iterate $\mathbf{x}_k$, $k \ge k_+$, it remains active for all the subsequent iterations. Thus (5.3) holds trivially for any $i \in \mathcal{A}_+(\mathbf{x}^*)$ with the property that $x_{ki} = 0$ for some $k \ge k_+$.

Now, let us focus on the nontrivial indices in $\mathcal{A}_+(\mathbf{x}^*)$. That is, suppose that there exists $l \in \mathcal{A}_+(\mathbf{x}^*)$ and $x_{kl} > 0$ for all $k \geq k_+$. By the analysis given in the previous paragraph, when $k_+$ is sufficiently large,

$$\text{either } x_{ki} > 0 \quad \text{or} \quad x_{ki} = 0 \tag{5.4}$$

for all $k \geq k_+$ and $i \in \mathcal{A}_+(\mathbf{x}^*)$ (since an index $i \in \mathcal{A}_+(\mathbf{x}^*)$, which becomes active at iterate $\mathbf{x}_k$, remains active for all the subsequent iterations). We consider the following possible cases.

*Case* 1. For an infinite number of iterations $k$, $\mathbf{x}_k$ is generated by the UA, and the UA is restarted a finite number of times.

In this case, the ASA eventually performs only the UA, without restarts. By U2 and U3, we have $\liminf_{k\to\infty} \|\mathbf{g}_I(\mathbf{x}_k)\| = 0$. On the other hand, by assumption, $l \in \mathcal{I}(\mathbf{x}_k)$ for $k \geq k_+$ and $g_l(\mathbf{x}^*) > 0$, which is a contradiction since $g_l(\mathbf{x}_k)$ converges to $g_l(\mathbf{x}^*)$.

*Case* 2. For an infinite number of iterations $k$, $\mathbf{x}_k$ is generated by the UA, and the UA is restarted an infinite number of times.

In this case, we will show that after a finite number of iterations, $x_{ki} = 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$. Suppose, to the contrary, that there exists an $l \in \mathcal{A}_+(\mathbf{x}^*)$ such that $x_{kl} > 0$ for all $k \geq k_+$. By U4, each time the UA is restarted, we perform a Wolfe line search. By the second half of (3.1), we have

$$\phi'(\alpha_k) - \phi'(0) \geq (\sigma - 1)\phi'(0). \tag{5.5}$$

It follows from the definition (3.2) of $\phi(\alpha)$ that

$$\phi'(0) = -\sum_{i\in\mathcal{I}(x_k)} g_{ki}^2 = -\|\mathbf{g}_I(\mathbf{x}_k)\|^2 \quad \text{and} \tag{5.6}$$

$$\begin{aligned}\phi'(\alpha_k) &= -\sum_{i\in\mathcal{I}(x_{k+1})} g_{ki}g_{k+1,i} \\ &= -\sum_{i\in\mathcal{I}(x_k)} g_{ki}g_{k+1,i} + \sum_{i\in\mathcal{A}(x_{k+1})\setminus\mathcal{A}(x_k)} g_{ki}g_{k+1,i}. \end{aligned} \tag{5.7}$$

By the Lipschitz continuity of $\nabla f$ and P3, we have

$$\begin{aligned}\|\mathbf{g}(\mathbf{x}_k) - \mathbf{g}(\mathbf{x}_{k+1})\| &= \|\mathbf{g}(P(\mathbf{x}_k)) - \mathbf{g}(P(\mathbf{x}_k - \alpha_k\mathbf{g}_I(\mathbf{x}_k)))\| \\ &\leq \lambda\alpha_k\|\mathbf{g}_I(\mathbf{x}_k)\|.\end{aligned}$$

Hence, by the Schwarz inequality,

$$\left|\sum_{i\in\mathcal{I}(\mathbf{x}_k)} g_{ki}(g_{ki} - g_{k+1,i})\right| \leq \lambda\alpha_k\|\mathbf{g}_I(\mathbf{x}_k)\|^2. \tag{5.8}$$

Since $\mathcal{A}(\mathbf{x}_{k+1}) \setminus \mathcal{A}(\mathbf{x}_k) \subset \mathcal{I}(\mathbf{x}_k)$, the Schwarz inequality also gives

$$\sum_{i\in\mathcal{A}(x_{k+1})\setminus\mathcal{A}(x_k)} g_{ki}g_{k+1,i} \leq \|\mathbf{g}_I(\mathbf{x}_k)\|\|\mathbf{g}_{k+1}\|_{\mathcal{N}}, \tag{5.9}$$

where

$$\|\mathbf{g}_{k+1}\|_{\mathcal{N}}^2 = \sum_{i\in\mathcal{A}(x_{k+1})\setminus\mathcal{A}(x_k)} g_{k+1,i}^2.$$

Here $\mathcal{N} = \mathcal{A}(\mathbf{x}_{k+1}) \setminus \mathcal{A}(\mathbf{x}_k)$ corresponds to the set of constraints that are newly activated as we move from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$. Combining (5.5)–(5.9),

$$\alpha_k \geq \frac{1-\sigma}{\lambda} - \frac{\|\mathbf{g}_{k+1}\|_{\mathcal{N}}}{\lambda \|\mathbf{g}_I(\mathbf{x}_k)\|}, \quad \text{where } \|\mathbf{g}_{k+1}\|_{\mathcal{N}}^2 = \sum_{i \in \mathcal{A}(x_{k+1}) \setminus \mathcal{A}(x_k)} g_{k+1,i}^2. \tag{5.10}$$

For $k$ sufficiently large, (5.4) implies that the newly activated constraints $\mathcal{A}(\mathbf{x}_{k+1}) \setminus \mathcal{A}(\mathbf{x}_k)$ exclude all members of $\mathcal{A}_+(\mathbf{x}^*)$. Since the $\mathbf{x}_k$ converge to $\mathbf{x}^*$, $\|\mathbf{g}_{k+1}\|_{\mathcal{N}}$ tends to zero. On the other hand, $\|\mathbf{g}_I(\mathbf{x}_k)\|$ is bounded away from zero since the index $l$ is contained in $\mathcal{I}(\mathbf{x}_k)$. Hence, the last term in (5.10) tends to 0 as $k$ increases, and the lower bound for $\alpha_k$ approaches $(1-\sigma)/\lambda$. Since $x_l^* = 0$, it follows that $x_{kl}$ approaches 0. Since the lower bound for $\alpha_k$ approaches $(1-\sigma)/\lambda$, $g_l(\mathbf{x}^*) > 0$, and $\mathbf{x}_k$ converges to $\mathbf{x}^*$, we conclude that

$$x_{k+1,l} = x_{kl} - \alpha_k g_{kl} < 0$$

for $k$ sufficiently large. This contradicts the initial assumption that constraint $l$ is inactive for $k$ sufficiently large. Hence, in a finite number of iterations, $x_{ki} = 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$.

*Case* 3. The UA is executed a finite number of iterations.

In this case, the iterates are generated by the NGPA for $k$ sufficiently large. Suppose that (5.3) is violated for some $l \in \mathcal{A}_+(\mathbf{x}^*)$. We show that this leads to a contradiction. By (5.4), $x_{kl} > 0$ for all $k \geq k_+$. Since $\mathbf{x}_k$ converges to $\mathbf{x}^*$, $\mathbf{x}_l^* = 0$, and $g_l(\mathbf{x}^*) > 0$, it is possible to choose $k$ larger, if necessary, so that

$$x_{kl} - g_{kl}\alpha_{\min} < 0. \tag{5.11}$$

Since (5.3) is violated and $\mathbf{x}_k$ converges to $\mathbf{x}^*$, we can choose $k$ larger, if necessary, so that

$$\frac{x_{kl}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \geq \frac{\lambda(2+\lambda)^2 \max\{1, \alpha_{\max}\}^2}{2(1-\delta)g_{kl}}, \tag{5.12}$$

where $0 < \delta < 1$ is the parameter appearing in step 3 of the NGPA, and $\lambda$ is the Lipschitz constant for $\nabla f$. We will show that for this $k$, we have

$$f(\mathbf{x}_k + \mathbf{d}_k) \leq f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k, \tag{5.13}$$

where $f_R$ is specified in step 3 of the NGPA. According to step 3 of the NGPA, when (5.13) holds, $\alpha_k = 1$, which implies that

$$x_{k+1,l} = x_{kl} + d_{kl}. \tag{5.14}$$

Since (5.11) holds and $\bar{\alpha}_k \geq \alpha_{\min}$, we have

$$d_{kl} = \max\{x_{kl} - \bar{\alpha}_k g_{kl}, 0\} - x_{kl} = -x_{kl}. \tag{5.15}$$

This substitution in (5.14) gives $x_{k+1,l} = 0$, which contradicts the fact that $x_{kl} > 0$ for all $k \geq k_+$.

To complete the proof, we need to show that when (5.12) holds, (5.13) is satisfied. Expanding in a Taylor series around $\mathbf{x}_k$ and utilizing (5.15) gives

$$\begin{aligned} f(\mathbf{x}_k + \mathbf{d}_k) &= f(\mathbf{x}_k) + \int_0^1 f'(\mathbf{x}_k + t\mathbf{d}_k)dt \\ &= f(\mathbf{x}_k) + \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + \int_0^1 (\nabla f(\mathbf{x}_k + t\mathbf{d}_k) - \mathbf{g}_k^{\mathsf{T}})\mathbf{d}_k dt \\ &\le f(\mathbf{x}_k) + \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + \frac{\lambda}{2}\|\mathbf{d}_k\|^2 \\ &= f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + (1-\delta)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + \frac{\lambda}{2}\|\mathbf{d}_k\|^2 \\ &\le f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + (1-\delta)g_{kl}d_{kl} + \frac{\lambda}{2}\|\mathbf{d}_k\|^2 \qquad \text{(5.16a)} \\ &= f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k - (1-\delta)g_{kl}x_{kl} + \frac{\lambda}{2}\|\mathbf{d}_k\|^2. \qquad \text{(5.16b)} \end{aligned}$$

The inequality (5.16a) is due to the fact that $g_{ki}d_{ki} \le 0$ for each $i$. By P3, P4, P5, and P7, and by the Lipschitz continuity of $\nabla f$, we have

$$\begin{aligned} \|\mathbf{d}_k\| &\le \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k)\| \\ &= \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k) - \mathbf{d}^1(\mathbf{x}^*)\| \\ &= \max\{1, \alpha_{\max}\}\|P(\mathbf{x}_k - \mathbf{g}_k) - \mathbf{x}_k - P(\mathbf{x}^* - \mathbf{g}(\mathbf{x}^*)) + \mathbf{x}^*\| \\ &\le \max\{1, \alpha_{\max}\}(\|\mathbf{x}_k - \mathbf{x}^*\| + \|P(\mathbf{x}_k - \mathbf{g}_k) - P(\mathbf{x}^* - \mathbf{g}(\mathbf{x}^*))\|) \\ &\le \max\{1, \alpha_{\max}\}(\|\mathbf{x}_k - \mathbf{x}^*\| + \|\mathbf{x}_k - \mathbf{g}_k - (\mathbf{x}^* - \mathbf{g}(\mathbf{x}^*))\|) \\ &\le \max\{1, \alpha_{\max}\}(2\|\mathbf{x}_k - \mathbf{x}^*\| + \|\mathbf{g}_k - \mathbf{g}(\mathbf{x}^*)\|) \\ &\le \max\{1, \alpha_{\max}\}(2 + \lambda)\|\mathbf{x}_k - \mathbf{x}^*\|. \end{aligned}$$

Combining this upper bound for $\|\mathbf{d}_k\|$ with the lower bound (5.12) for $x_{kl}$, we conclude that

$$\begin{aligned} \frac{\lambda}{2}\|\mathbf{d}_k\|^2 &\le \frac{\lambda}{2}\max\{1, \alpha_{\max}\}^2(2+\lambda)^2\|\mathbf{x}_k - \mathbf{x}^*\|^2 \\ &\le \frac{1}{2}\left(\frac{2(1-\delta)x_{kl}g_{kl}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2}\right)\|\mathbf{x}_k - \mathbf{x}^*\|^2 \\ &= (1-\delta)x_{kl}g_{kl}. \end{aligned}$$

Hence, by (5.16b) and by the choice for $f_R$ specified in step 3 of the NGPA, we have

$$f(\mathbf{x}_k + \mathbf{d}_k) \le f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le f_R + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k. \qquad \text{(5.17)}$$

This completes the proof of (5.13). □

There is a fundamental difference between the gradient projection algorithm presented in this paper and algorithms based on a "piecewise projected gradient" [15, 16, 17]. For our gradient projection algorithm, we perform a single projection, and then we backtrack towards the starting point. Thus we are unable to show that the active constraints are identified in a finite number of iterations; in contrast, with the piecewise project gradient approach, where a series of projections may be performed, the active constraints can be identified in a finite number of iterations. In Lemma 5.2 we show that even though we do not identify the active constraints, the

components of $\mathbf{x}_k$ corresponding to the strictly active constraints are on the order of the error in $\mathbf{x}_k$ squared.

If all the constraints are active at a stationary point $\mathbf{x}^*$ and strict complementarity holds, then convergence is achieved in a finite number of iterations.

COROLLARY 5.3. *If $f$ is twice-continuously differentiable, the iterates $\mathbf{x}_k$ generated by the ASA with $\epsilon = 0$ converge to a stationary point $\mathbf{x}^*$, and $|\mathcal{A}_+(\mathbf{x}^*)| = n$, then $\mathbf{x}_k = \mathbf{x}^*$ after a finite number of iterations.*

*Proof.* Let $\mathbf{x}_{k,\max}$ denote the largest component of $\mathbf{x}_k$. Since $\|\mathbf{x}_k\|^2 \le n\mathbf{x}_{k,\max}^2$, we have

$$\frac{\mathbf{x}_{k,\max}}{\|\mathbf{x}_k\|^2} \ge \frac{1}{n\mathbf{x}_{k,\max}}. \tag{5.18}$$

Since all the constraints are active at $\mathbf{x}^*$, $\mathbf{x}_{k,\max}$ tends to zero. By (5.18) the conclusion (5.3) of Lemma 5.2 does not hold. Hence, after a finite number of iterations, $\mathbf{x}_k = \mathbf{x}^*$. ∎

Recall [70] that for any stationary point $\mathbf{x}^*$ of (1.1), the strong second-order sufficient optimality condition holds if there exists $\gamma > 0$ such that

$$\mathbf{d}^{\mathsf{T}}\nabla^2 f(\mathbf{x}^*)\mathbf{d} \ge \gamma\|\mathbf{d}\|^2 \quad \text{whenever} \quad d_i = 0 \text{ for all } i \in \mathcal{A}_+(\mathbf{x}^*). \tag{5.19}$$

Using P8, we establish the following.

LEMMA 5.4. *If $f$ is twice-continuously differentiable near a stationary point $\mathbf{x}^*$ of (1.1) satisfying the strong second-order sufficient optimality condition, then there exists $\rho > 0$ with the following property:*

$$\|\mathbf{x} - \mathbf{x}^*\| \le \sqrt{1 + \left(\frac{(1+\lambda)^2}{.5\gamma}\right)^2}\,\|\mathbf{d}^1(\mathbf{x})\| \tag{5.20}$$

*for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$, where $\lambda$ is any Lipschitz constant for $\nabla f$ over $B_\rho(\mathbf{x}^*)$.*

*Proof.* By the continuity of the second derivative of $f$, it follows from (5.19) that for $\rho > 0$ sufficiently small,

$$(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*))^{\mathsf{T}}(\mathbf{x} - \mathbf{x}^*) \ge .5\gamma\|\mathbf{x} - \mathbf{x}^*\|^2 \tag{5.21}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$ with $x_i = 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$. Choose $\rho$ smaller if necessary so that

$$x_i - g_i(\mathbf{x}) \le 0 \text{ for all } i \in \mathcal{A}_+(\mathbf{x}^*) \text{ and } \mathbf{x} \in B_\rho(\mathbf{x}^*). \tag{5.22}$$

Let $\bar{\mathbf{x}}$ be defined as follows:

$$\bar{x}_i = \begin{cases} 0 \text{ if } i \in \mathcal{A}_+(\mathbf{x}^*), \\ x_i \text{ otherwise.} \end{cases} \tag{5.23}$$

Since (5.22) holds, it follows that

$$\|\mathbf{x} - \bar{\mathbf{x}}\| \le \|\mathbf{d}^1(\mathbf{x})\| \tag{5.24}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$. Also, by (5.22), we have

$$[P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}]_i = 0 \quad \text{and} \quad \mathbf{d}^1(\mathbf{x})_i = [P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x}]_i = -x_i$$

for all $i \in \mathcal{A}_+(\mathbf{x}^*)$, while

$$[P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}]_i = \mathbf{d}^1(\mathbf{x})_i = [P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x}]_i$$

for $i \notin \mathcal{A}_+(\mathbf{x}^*)$. Hence, we have

$$\|P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}\| \leq \|\mathbf{d}^1(\mathbf{x})\| \tag{5.25}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$. By the Lipschitz continuity of $\mathbf{g}$, (5.24), (5.25), and P3, it follows that

$$\begin{aligned} \|\mathbf{d}^1(\bar{\mathbf{x}})\| &= \|P(\bar{\mathbf{x}} - \mathbf{g}(\bar{\mathbf{x}})) - P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) + P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}\| \\ &\leq \lambda\|\bar{\mathbf{x}} - \mathbf{x}\| + \|\mathbf{d}^1(\mathbf{x})\| \\ &\leq (1+\lambda)\|\mathbf{d}^1(\mathbf{x})\| \end{aligned} \tag{5.26}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$. By P8, (5.21), and (5.26), we have

$$\|\bar{\mathbf{x}} - \mathbf{x}^*\| \leq \left(\frac{1+\lambda}{.5\gamma}\right)\|\mathbf{d}^1(\bar{\mathbf{x}})\| \leq \left(\frac{(1+\lambda)^2}{.5\gamma}\right)\|\mathbf{d}^1(\mathbf{x})\|. \tag{5.27}$$

Since $\|\mathbf{x} - \bar{\mathbf{x}}\|^2 + \|\bar{\mathbf{x}} - \mathbf{x}^*\|^2 = \|\mathbf{x} - \mathbf{x}^*\|^2$, the proof is completed by squaring and adding (5.27) and (5.24). □

We now show that the undecided index set $\mathcal{U}$ becomes empty as the iterates approach a stationary point, where the strong second-order sufficient optimality condition holds.

LEMMA 5.5. *Suppose $f$ is twice-continuously differentiable, $\mathbf{x}^*$ is a stationary point of* (1.1) *satisfying the strong second-order sufficient optimality condition, and $\mathbf{x}_k$, $k = 0, 1, \ldots$, is an infinite sequence of feasible iterates for* (1.1) *converging to $\mathbf{x}^*$, $\mathbf{x}_k \neq \mathbf{x}^*$ for each $k$. If there exists a constant $\xi$ such that*

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq \xi < \infty \tag{5.28}$$

*for all $i \in \mathcal{A}_+(\mathbf{x}^*)$, then $\mathcal{U}(\mathbf{x}_k)$ is empty for $k$ sufficiently large.*

*Proof.* To prove that $\mathcal{U}(\mathbf{x})$ is empty, we must show that for each $i \in [1, n]$, one of the following inequalities is violated:

$$|g_i(\mathbf{x})| \geq \|\mathbf{d}^1(\mathbf{x})\|^\alpha \text{ or} \tag{5.29}$$

$$x_i \geq \|\mathbf{d}^1(\mathbf{x})\|^\beta. \tag{5.30}$$

By Lemma 5.4, there exists a constant $c$ such that $\|\mathbf{x} - \mathbf{x}^*\| \leq c\|\mathbf{d}^1(\mathbf{x})\|$ for all $\mathbf{x}$ near $\mathbf{x}^*$. If $i \in \mathcal{A}_+(\mathbf{x}^*)$, then by (5.28), we have

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\beta} \leq \limsup_{k\to\infty} \frac{\xi\|\mathbf{x}_k - \mathbf{x}^*\|^2}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\beta} \leq \limsup_{k\to\infty} \xi c^2 \|\mathbf{d}^1(\mathbf{x}_k)\|^{2-\beta} = 0$$

since $\beta \in (1, 2)$. Hence, for each $i \in \mathcal{A}_+(\mathbf{x}^*)$, (5.30) is violated for $k$ sufficiently large.

If $i \notin \mathcal{A}_+(\mathbf{x}^*)$, then $g_i(\mathbf{x}^*) = 0$. By Lemma 5.4, we have

$$\begin{aligned} \limsup_{k\to\infty} \frac{|g_i(\mathbf{x}_k)|}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha} &= \limsup_{k\to\infty} \frac{|g_i(\mathbf{x}_k) - g_i(\mathbf{x}^*)|}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha} \\ &\leq \limsup_{k\to\infty} \frac{\lambda\|\mathbf{x}_k - \mathbf{x}^*\|}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha} \\ &\leq \limsup_{k\to\infty} \lambda c \|\mathbf{d}^1(\mathbf{x}_k)\|^{1-\alpha} = 0, \end{aligned}$$

since $\alpha \in (0,1)$. Here, $\lambda$ is a Lipschitz constant for $\mathbf{g}$ in a neighborhood of $\mathbf{x}^*$. Hence, (5.29) is violated if $i \notin \mathcal{A}_+(\mathbf{x}^*)$. ∎

*Remark.* If $i \in \mathcal{A}_+(\mathbf{x}^*)$ and the iterates $\mathbf{x}_k$ converge to a stationary point $\mathbf{x}^*$, then $g_i(\mathbf{x}_k)$ is bounded away from 0 for $k$ sufficiently large. Since $\mathbf{d}^1(\mathbf{x}_k)$ tends to zero, the inequality $|g_i(\mathbf{x}_k)| \geq \|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha$ is satisfied for $k$ sufficiently large. Hence, if $\mathcal{U}(\mathbf{x}_k)$ is empty and $i \in \mathcal{A}_+(\mathbf{x}^*)$, then $x_{ki} < \|\mathbf{d}^1(\mathbf{x}_k)\|^\beta$ where $\beta \in (1,2)$. In other words, when $\mathcal{U}(\mathbf{x}_k)$ is empty, the components of $\mathbf{x}_k$ associated with strictly active indices $\mathcal{A}_+(\mathbf{x}^*)$ are going to zero faster than the error $\|\mathbf{d}^1(\mathbf{x}_k)\|$.

LEMMA 5.6. *Suppose $f$ is twice-continuously differentiable, $\mathbf{x}^*$ is a stationary point of* (1.1) *satisfying the strong second-order sufficient optimality condition, and $\mathbf{x}_k$, $k = 0, 1, \ldots$, is an infinite sequence of feasible iterates for* (1.1) *converging to $\mathbf{x}^*$, $\mathbf{x}_k \neq \mathbf{x}^*$ for each $k$. If there exists a constant $\xi$ such that*

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq \xi < \infty \tag{5.31}$$

*for all $i \in \mathcal{A}_+(\mathbf{x}^*)$, then there exist $\mu^* > 0$ such that*

$$\|\mathbf{g}_I(\mathbf{x}_k)\| \geq \mu^* \|\mathbf{d}^1(\mathbf{x}_k)\| \tag{5.32}$$

*for $k$ sufficiently large.*

*Proof.* Choose $\rho > 0$, and let $\lambda$ be the Lipschitz constant for $\nabla f$ in $B_\rho(\mathbf{x}^*)$. As in (5.23), let $\bar{\mathbf{x}}$ be defined by $\bar{x}_i = 0$ if $i \in \mathcal{A}_+(\mathbf{x}^*)$ and $\bar{x}_i = x_i$ otherwise. If $\mathbf{x}_k \in B_\rho(\mathbf{x}^*)$, we have

$$\begin{aligned} \|\mathbf{d}^1(\mathbf{x}_k)\| &\leq \|\mathbf{d}^1(\mathbf{x}_k) - \mathbf{d}^1(\mathbf{x}^*)\| \\ &\leq \|\mathbf{d}^1(\mathbf{x}_k) - \mathbf{d}^1(\bar{\mathbf{x}}_k)\| + \|\mathbf{d}^1(\bar{\mathbf{x}}_k) - \mathbf{d}^1(\mathbf{x}^*)\| \\ &\leq (2+\lambda)(\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \|\bar{\mathbf{x}}_k - \mathbf{x}^*\|). \end{aligned} \tag{5.33}$$

Utilizing (5.31) gives

$$\begin{aligned} \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| &\leq \sum_{i=1}^n |\bar{x}_{ki} - x_{ki}| \\ &= \sum_{i \in \mathcal{A}_+(\mathbf{x}^*)} x_{ki} \leq n\xi \|\mathbf{x}_k - \mathbf{x}^*\|^2 \\ &\leq n\xi \|\mathbf{x}_k - \mathbf{x}^*\|(\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \|\bar{\mathbf{x}}_k - \mathbf{x}^*\|). \end{aligned}$$

Since $\mathbf{x}_k$ converges to $\mathbf{x}^*$, it follows that for any $\epsilon > 0$,

$$\|\bar{\mathbf{x}}_k - \mathbf{x}_k\| \leq \epsilon \|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \tag{5.34}$$

when $k$ is sufficiently large. Combining (5.33) and (5.34), there exists a constant $c > 0$ such that

$$\|\mathbf{d}^1(\mathbf{x}_k)\| \leq c \|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \tag{5.35}$$

for $k$ sufficiently large.

Let $k$ be chosen large enough that

$$\|\mathbf{x}_k - \mathbf{x}^*\| < \min\{x_i^* : i \in \mathcal{I}(\mathbf{x}^*)\}. \tag{5.36}$$

Suppose, in this case, that $i \in \mathcal{A}(\mathbf{x}_k)$. If $x_i^* > 0$, then $\|\mathbf{x}_k - \mathbf{x}^*\| \geq x_i^*$, which contradicts (5.36). Hence, $\bar{x}_{ki} = x_i^* = 0$. Moreover, if $i \in \mathcal{A}_+(\mathbf{x}^*)$, then by the definition (5.23), $\bar{x}_{ki} = x_i^* = 0$. In summary,

$$\begin{cases} \bar{x}_{ki} = x_i^* = 0 & \text{for each } i \in \mathcal{A}(\mathbf{x}_k) \cup \mathcal{A}_+(\mathbf{x}^*), \\ g_i(\mathbf{x}^*) = 0 & \text{for each } i \in \mathcal{A}_+(\mathbf{x}^*)^c, \end{cases} \tag{5.37}$$

where $\mathcal{A}_+(\mathbf{x}^*)^c$ is the complement of $\mathcal{A}_+(\mathbf{x}^*)$. Define $\mathcal{Z} = \mathcal{A}(\mathbf{x}_k)^c \cap \mathbf{A}_+(\mathbf{x}^*)^c$.

By the strong second-order sufficient optimality condition and for $\mathbf{x}$ near $\mathbf{x}^*$, we have

$$\begin{aligned} \frac{\gamma}{2}\|\bar{\mathbf{x}} - \mathbf{x}^*\|^2 &\leq [\bar{\mathbf{x}} - \mathbf{x}^*]^\mathsf{T} \int_0^1 \nabla^2 f(\mathbf{x}^* + t(\bar{\mathbf{x}} - \mathbf{x}^*))dt \;\; [\bar{\mathbf{x}} - \mathbf{x}^*] \\ &= (\bar{\mathbf{x}} - \mathbf{x}^*)^\mathsf{T}(\mathbf{g}(\bar{\mathbf{x}}) - \mathbf{g}(\mathbf{x}^*)). \end{aligned} \tag{5.38}$$

We substitute $\mathbf{x} = \mathbf{x}_k$ in (5.38) and utilize (5.37) to obtain

$$\begin{aligned} (\bar{\mathbf{x}}_k - \mathbf{x}^*)^\mathsf{T}(\mathbf{g}(\bar{\mathbf{x}}_k) - \mathbf{g}(\mathbf{x}^*)) &= \sum_{i=1}^n (\bar{x}_{ki} - x_i^*)(g_i(\bar{\mathbf{x}}_k) - g_i(\mathbf{x}^*)) \\ &= \sum_{i \in \mathcal{Z}} (\bar{x}_{ki} - x_i^*) g_i(\bar{\mathbf{x}}_k) \\ &\leq \|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \left( \sum_{i \in \mathcal{I}(\mathbf{x}_k)} g_i(\bar{\mathbf{x}}_k)^2 \right)^{1/2}, \end{aligned} \tag{5.39}$$

since $\mathcal{Z} \subset \mathcal{A}(\mathbf{x}_k)^c = \mathcal{I}(\mathbf{x}_k)$. Exploiting the Lipschitz continuity of $\nabla f$, (5.39) gives

$$(\bar{\mathbf{x}}_k - \mathbf{x}^*)^\mathsf{T}(\mathbf{g}(\bar{\mathbf{x}}_k) - \mathbf{g}(\mathbf{x}^*)) \leq \|\bar{\mathbf{x}}_k - \mathbf{x}^*\|(\|\mathbf{g}_I(\mathbf{x}_k)\| + \lambda\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|). \tag{5.40}$$

Combining (5.34), (5.38), and (5.40), we conclude that for $k$ sufficiently large,

$$\frac{\gamma}{4}\|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \leq \|\mathbf{g}_I(\mathbf{x}_k)\|. \tag{5.41}$$

Combining (5.35) and (5.41), the proof is complete. ∎

*Remark.* If $\mathbf{x}_k$ is a sequence converging to a nondegenerate stationary point $\mathbf{x}^*$, then (5.32) holds with $\mu^* = 1$, without assuming either the strong second-order sufficient optimality condition or (5.31)—see Theorem 5.1. In Lemma 5.6, the optimization problem could be degenerate.

We now show that after a finite number of iterations, the ASA will perform only the UA with a fixed active constraint set.

THEOREM 5.7. *If $f$ is twice-continuously differentiable and the iterates $\mathbf{x}_k$ generated by the ASA with $\epsilon = 0$ converge to a stationary point $\mathbf{x}^*$ satisfying the strong second-order sufficient optimality condition, then after a finite number of iterations, the ASA performs only the UA without restarts.*

*Proof.* By Lemma 5.2, the hypotheses (5.28) and (5.31) of Lemmas 5.5 and 5.6 are satisfied. Hence, for $k$ sufficiently large, the undecided set $\mathcal{U}(\mathbf{x}_k)$ is empty and the lower bound (5.32) holds. In step 1a, if $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$, then $\mu$ is multiplied by the factor $\rho < 1$. When $\mu < \mu^*$, Lemma 5.6 implies that $\|\mathbf{g}_I(\mathbf{x}_k)\| \geq \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$. Hence, step 1a of the ASA branches to step 2, while step 2 cannot branch to step 1 since the condition $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$ is never satisfied in step 2a and $\mathcal{U}(\mathbf{x}_k)$

is empty in step 2b for $k$ sufficiently large. Since the UA only adds constraints, we conclude that after a finite number of iterations, the active set does not change. ∎

*Remark.* If $f$ is a strongly convex quadratic function, then by Corollary 4.2, the iterates $\mathbf{x}_k$ converge to the global minimizer $\mathbf{x}^*$. If the UA is based on the conjugate gradient method for which there is finite convergence when applied to a convex quadratic, it follows from Theorem 5.7 that the ASA converges in a finite number of iterations.

We now give the proof of Corollary 4.2; that is, when $f$ is strongly convex and twice-continuously differentiable on $\mathcal{B}$, and assumption A3 of Theorem 4.1 is satisfied, then the entire sequence of iterates generated by the ASA converges to the global minimizer $\mathbf{x}^*$. Note that the assumptions of Corollary 4.2 are weaker than those of Corollary 2.3 (global convergence of the NGPA) since Corollary 4.2 requires only that $f_k^r \le f_k^{\max}$ infinitely often in the NGPA.

*Proof.* For a strongly convex function, A1 and A2 always hold. Since all the assumptions of Theorem 4.1 are satisfied, there exists a subsequence $\mathbf{x}_{k_j}$, $j = 1, 2, \ldots$, of the iterates such that

$$\lim_{j\to\infty} \|\mathbf{d}^1(\mathbf{x}_{k_j})\| = 0.$$

Since the UA is monotone and since the NGPA satisfies (2.12) and (2.13), it follows from the strong convexity of $f$ that the $\mathbf{x}_{k_j}$ are contained in a bounded set. Since $\mathbf{d}^1(\cdot)$ is continuous, there exists a subsequence, also denoted $\mathbf{x}_{k_j}$, converging to a limit $\mathbf{x}^*$ with $\mathbf{d}^1(\mathbf{x}^*) = \mathbf{0}$. Since the unique stationary point of a strongly convex function is its global minimizer, $\mathbf{x}^*$ is the global solution of (1.1).

*Case A.* There exists an infinite subsequence, also denoted $\{\mathbf{x}_{k_j}\}$, with the property that $\mathbf{x}_{k_j+1}$ is generated by the UA.

In this case, we are done since the UA is monotone and the inequality

$$f(\mathbf{x}_k) \le f(\mathbf{x}_{k_j}) \tag{5.42}$$

holds for all $k \ge k_j$ (see (2.12) and (2.13)). Since $\mathbf{x}_{k_j}$ converges to $\mathbf{x}^*$, it follows that $f(\mathbf{x}_{k_j})$ converges to $f(\mathbf{x}^*)$, and hence, by (5.42) and (2.32), the entire sequence converges to $\mathbf{x}^*$.

*Case B.* There exists an infinite subsequence, also denoted $\{\mathbf{x}_{k_j}\}$, with the property that $\mathbf{x}_{k_j+1}$ is generated by the NGPA.

Either

$$\limsup_{j\to\infty} \frac{(\mathbf{x}_{k_j})_i}{\|\mathbf{x}_{k_j} - \mathbf{x}^*\|^2} < \infty \quad \text{for all } i \in \mathcal{A}_+(\mathbf{x}^*) \tag{5.43}$$

holds or (5.43) is violated. By the analysis given in Case 3 of the proof of Lemma 5.2, when (5.43) is violated, (5.13) holds, from which it follows that for $j$ sufficiently large,

$$\mathbf{x}_{k_j+1,i} = 0 \quad \text{for all } i \in \mathcal{A}_+(\mathbf{x}^*). \tag{5.44}$$

Hence, either the sequence $\mathbf{x}_{k_j}$ satisfies (5.43) or the sequence $\mathbf{x}_{k_j+1}$ satisfies (5.44). In this latter case, it follows from (5.17) that

$$f(\mathbf{x}_{k_j+1}) \le f(\mathbf{x}_{k_j}).$$

Since $f(\mathbf{x}_{k_j})$ converges to $f(\mathbf{x}^*)$, we conclude that $f(\mathbf{x}_{k_j+1})$ converges to $f(\mathbf{x}^*)$, and $\mathbf{x}_{k_j+1}$ converges to $\mathbf{x}^*$.

In either case (5.43) or (5.44), there exists a sequence $K_j$ (either $K_j = k_j$ or $K_j = k_j + 1$) with the property that $\mathbf{x}_{K_j}$ converges to $\mathbf{x}^*$ and

$$\limsup_{j\to\infty} \frac{(\mathbf{x}_{K_j})_i}{\|\mathbf{x}_{K_j} - \mathbf{x}^*\|^2} < \infty \quad \text{for all } i \in \mathcal{A}_+(\mathbf{x}^*).$$

By Lemma 5.5, $\mathcal{U}(\mathbf{x}_{K_j})$ is empty for $j$ sufficiently large. By Lemma 5.6, there exists $\mu^* > 0$ such that

$$\|\mathbf{g}_I(\mathbf{x}_{K_j})\| \geq \mu^* \|\mathbf{d}^1(\mathbf{x}_{K_j})\|$$

for $j$ sufficiently large. As in the proof of Theorem 5.7, at iteration $K_j$ for $j$ sufficiently large, the ASA jumps from step 1 to the UA in step 2. Hence, for $j$ sufficiently large, $\mathbf{x}_{K_j+1}$ is generated by the UA, which implies that Case A holds. □

**6. Numerical experiments.** This section compares the CPU time performance of the ASA, implemented using the nonlinear conjugate gradient code CG_DESCENT for the UA and the CBB method (see the appendix) for the NGPA, to the performance of the following codes:

- L-BFGS-B [18, 84]: The limited memory quasi-Newton method of Zhu, Byrd, and Nocedal (ACM algorithm 778).
- SPG2 version 2.1 [10, 11]: The nonmonotone spectral projected gradient method of Birgin, Martínez, and Raydan (ACM algorithm 813).
- GENCAN [9]: The monotone active set method with spectral projected gradients developed by Birgin and Martínez.
- TRON version 1.2 [63]: A Newton trust region method with incomplete Cholesky preconditioning developed by Lin and Moré.

A detailed description of our implementation of the ASA is given in the appendix.

L-BFGS-B was downloaded from Jorge Nocedal's Web page (http://www.ece.northwestern.edu/∼nocedal/lbfgsb.html); TRON was downloaded from Jorge Moré's Web page (http://www-unix.mcs.anl.gov/∼more/tron/); and SPG2 and GENCAN were downloaded on June 28, 2005, from the TANGO Web page maintained by Ernesto Birgin (http://www.ime.usp.br/∼egbirgin/tango/downloads.php). All codes are written in Fortran and compiled with f77 (default compiler settings) on a Sun workstation. The stopping condition was

$$\|P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x}\|_\infty \leq 10^{-6},$$

where $\|\cdot\|_\infty$ denotes the sup-norm of a vector. In running any of these codes, default values were used for all parameters. In the NGPA, we chose the following parameter values:

$$\alpha_{\min} = 10^{-20}, \quad \alpha_{\max} = 10^{+20}, \quad \eta = .5, \quad \delta = 10^{-4}, \quad M = 8.$$

Here $M$ is the memory used to evaluate $f_k^{\max}$ (see (2.3)). In the ASA the parameter values were as follows:

$$\mu = .1, \quad \rho = .5, \quad n_1 = 2, \quad n_2 = 1.$$

In the CBB method (see the appendix), the parameter values were the following:

$$\theta = .975, \quad L = 3, \quad A = 40, \quad m = 4, \quad \gamma_1 = M/L, \quad \gamma_2 = A/M.$$

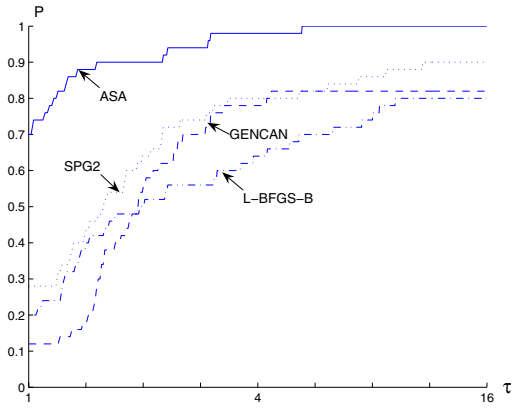


FIG. 6.1. *Performance profiles, CPU time metric, 50 CUTEr test problems, gradient-based methods.*

The separation parameter $\Delta$ in condition R4 of the appendix was the natural separation between floating point numbers. That is, R4 was satisfied when the floating point version of $f_{k+1}$ was strictly less than the floating point version of $f_k^{\min}$.

The test set consisted of all 50 box constrained problems in the CUTEr library [13] with dimensions between 50 and 15,625, and all 23 box constrained problems in the MINPACK-2 library [1] with dimension 2500. TRON is somewhat different from the other codes since it employs Hessian information and an incomplete Cholesky preconditioner, while the codes ASA, L-BFGS-B, SPG2, and GENCAN utilize only gradient information. When we compare our code to TRON, we use the same Lin–Moré preconditioner [62] used by TRON for our unconstrained algorithm. The preconditioned ASA code is called P-ASA. Since TRON is targeted to large-sparse problems, we compare our code to TRON using the 23 MINPACK-2 problems and the 42 sparsest CUTEr problems (the number of nonzeros in the Hessian was at most 1/5 the total number of entries). The codes L-BFGS-B, SPG2, and GENCAN were implemented for the CUTEr test problems, while ASA and TRON were implemented for both test sets CUTEr and MINPACK-2.

The CPU time in seconds and the number of iterations, function evaluations, gradient evaluations, and Hessian evaluations for each of the methods are posted at the following Web site: http://www.math.ufl.edu/∼hager/papers/CG. In running the numerical experiments, we checked whether different codes converged to different local minimizers; when comparing the codes, we restricted ourselves to test problems in which all codes converged to the same local minimizer, and where the running time of the fastest code exceeded .01 seconds. The numerical results are now analyzed.

The performance of the algorithms, relative to CPU time, was evaluated using the performance profiles of Dolan and Moré [34]. That is, for each method, we plot the fraction P of problems for which the method is within a factor $\tau$ of the best time. In Figure 6.1, we compare the performance of the four codes ASA, L-BFGS-B, SPG2, and GENCAN using the 50 CUTEr test problems. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that were successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor $\tau$ of the best time. Since the top curve in Figure 6.1 corresponds

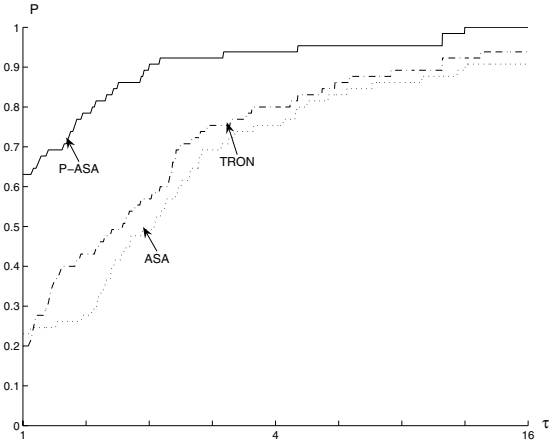


FIG. 6.2. *Performance profiles, CPU time metric,* 42 *sparsest CUTEr problems,* 23 *MINPACK-2 problems,* $\epsilon = 10^{-6}$.

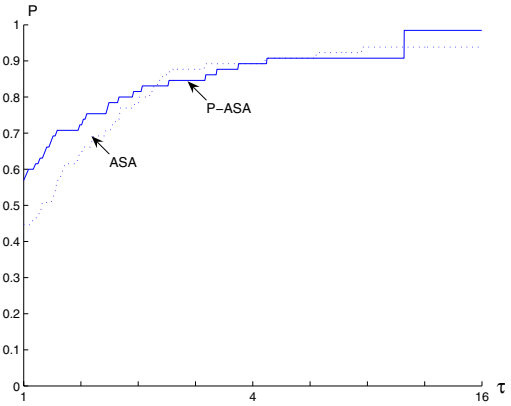


FIG. 6.3. *Performance profiles, CPU time metric,* $\epsilon = 10^{-2}\|\mathbf{d}^1(\mathbf{x}_0)\|_\infty$.

to the ASA, this algorithm is clearly fastest for this set of 50 test problems with dimensions ranging from 50 to 15,625. The relative difference in performance between the ASA and the competing methods seen in Figure 6.1 is greater than the relative difference in performance between CG_DESCENT and the competing methods, as seen in the figures given in [55, 57]. Hence, both the gradient projection algorithm and the conjugate gradient algorithm are contributing to the better performance of the ASA.

In Figure 6.2 we compare the performance of TRON to P-ASA and ASA for the 42 sparsest CUTEr test problems and the 23 MINPACK-2 problems. Observe that P-ASA has the top performance, and that ASA, which utilizes only the gradient, performs almost as well as the Hessian-based code TRON. The number of conjugate gradient iterations performed by the P-ASA code is much less than the number of conjugate gradient iterations performed by the ASA code. Finally, in Figure 6.3 we

compare the performance of P-ASA to ASA for the relaxed convergence tolerance $\epsilon = 10^{-2}\|\mathbf{d}^1(\mathbf{x}_0)\|_\infty$. Based on Figures 6.2 and 6.3, the preconditioned ASA scheme is more efficient than unconditioned ASA for the more stringent stopping criterion, while the unconditioned and preconditioned schemes are equally effective for a more relaxed stopping criterion. Although the performance profile for ASA is beneath 1 in Figure 6.2, it reaches 1 as $\tau$ increases—there are some problems in which P-ASA is more than 16 times faster than ASA. Due to these difficult problems, the ASA profile is still beneath 1 for $\tau = 16$.

When we solve an optimization problem, the solution time consists of two parts, as follows:

T1. The time associated with the evaluation of the function or its gradient or its Hessian.

T2. The remaining time, which is often dominated by the time used in the linear algebra.

The CPU time performance profile measures a mixture of T1 and T2 for a set of test problems. In some applications, T1 (the evaluation time) may dominate. In order to assess how the algorithms may perform in the limit, when T2 is negligible compared to T1, we could ignore T2 and compare the algorithms based on T1. In the next set of experiments, we explore how the algorithms perform in the limit, as T1 becomes infinitely large relative to T2.

Typically, the time to evaluate the gradient of a function is greater than the time to evaluate the function itself. Also, the time to evaluate the Hessian is greater than the time to evaluate the gradient. If the time to evaluate the function is 1, then the average time to evaluate the gradient and Hessian for the CUTEr bound constrained test set is as follows:

$$\text{function } = 1, \quad \text{gradient } = 2.6, \quad \text{Hessian } = 21.0.$$

Similarly, for the MINPACK-2 test set, the relative evaluation times are

$$\text{function } = 1, \quad \text{gradient } = 2.0, \quad \text{Hessian } = 40.5$$

on average.

For each method and for each test problem, we compute an "evaluation time" where the time for a function evaluation is 1, the time for a gradient evaluation is either 2.6 (CUTEr) or 2.0 (MINPACK-2), and the time for a Hessian evaluation is either 21.0 (CUTEr) or 40.5 (MINPACK-2). In Figure 6.4 we compare the performance of gradient-based methods, and in Figure 6.5 we compare the performance of the gradient-based ASA and the method which exploits the Hessian (P-ASA or TRON).

In Figure 6.4 we see that for the evaluation metric and $\tau$ near 1, L-BFGS-B performs better than ASA, but as $\tau$ increases, ASA dominates L-BFGS-B. In other words, in the evaluation metric, there are more problems in which L-BFGS-B is faster than the other methods; however, ASA is not much slower than L-BFGS-B. When $\tau$ reaches 1.5, ASA starts to dominate L-BFGS-B.

In Figure 6.5 we see that P-ASA dominates TRON in the evaluation metric. Hence, even though TRON uses far fewer function evaluations, it uses many more Hessian evaluations. Since the time to evaluate the Hessian is much greater than the time to evaluate the function, P-ASA has better performance. In summary, by neglecting the time associated with the linear algebra, the relative gap between P-ASA and TRON decreases, while the relative gap between TRON and ASA increases, as seen in Figure 6.5. Nonetheless, in the evaluation metric, the performance profile for P-ASA is still above the profile for TRON.

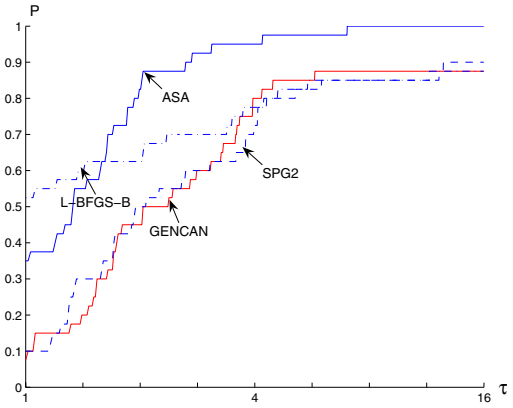


FIG. 6.4. *Performance profiles, evaluation metric,* 50 *CUTEr test problems, gradient-based methods.*

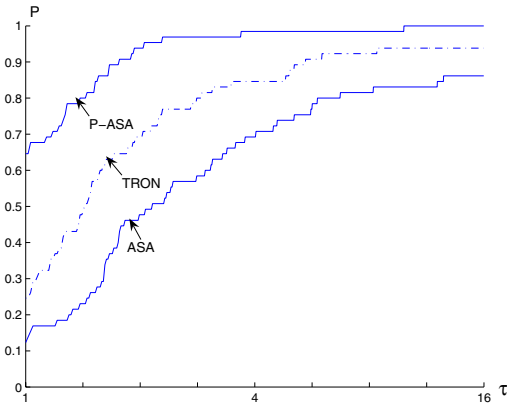


FIG. 6.5. *Performance profiles, evaluation metric,* 42 *sparsest CUTEr problems,* 23 *MINPACK-2 problems.*

**7. Conclusions.** We have presented a new ASA for solving box constrained optimization problems. The algorithm consists of a nonmonotone gradient projection phase and an unconstrained optimization phase. Rules are given for deciding when to branch from one phase to the other. The branching criteria are based on whether the set of undecided indices is empty or the active set subproblem is solved with sufficient accuracy. We show that for a nondegenerate stationary point, the algorithm eventually reduces to unconstrained optimization without restarts. The analogous result for a degenerate stationary point is established under the strong second-order sufficient optimality condition.

For an implementation of the ASA which uses the CBB method [30] for the nonmonotone gradient projection and which uses CG_DESCENT [54, 55, 56, 57] for unconstrained optimization, we obtained higher CPU time performance profiles than

those of L-BFGS-B, SPG2, GENCAN, and TRON for a test set consisting of all 50 CUTEr [13] box constrained problems with dimension greater than 50, and all 23 MINPACK-2 [1] box constrained problems.

**Appendix. An implementation of the ASA.** For the numerical results in section 6, our choice for the UA is the conjugate gradient algorithm CG_DESCENT [54, 55, 57, 56]. When an iterate lands outside the feasible set, we may increase the size of the active set using an approach similar to that in [9]. Roughly, we perform an approximate line search for the function

$$\phi(\alpha) = f(P(\mathbf{x}_k + \alpha \mathbf{d}_k))$$

along the current search direction $\mathbf{d}_k$, and any components of $\mathbf{x}_{k+1} = P(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$ which reach the boundary are added to the current active set.

The initial stepsize $\bar{\alpha}_k$ in the NGPA is generated using the CBB method [30]. In the remainder of this section, we explain in detail the initial stepsize computation and choice for the reference function value $f_k^r$ in the NGPA (see [82] for preliminary numerical results based on a closely related initial stepsize and reference function value). We show that these choices satisfy the hypotheses of Theorem 2.2.

The BB stepsize [2] is given by

$$\overline{\alpha}_{k+1}^{BB} = \frac{\mathbf{s}_k^{\mathsf{T}} \mathbf{s}_k}{\mathbf{s}_k^{\mathsf{T}} \mathbf{y}_k}, \tag{A.1}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. An attractive feature of the BB stepsize is that for unconstrained optimization and without a line search, linear convergence is achieved [30] for a starting guess in a neighborhood of the local minimizer with a positive definite Hessian. Moreover, if the same BB stepsize is repeated for several iterations, then even faster convergence is often achieved (see [30]). We refer to schemes that employ the same BB stepsize for several iterations as cyclic BB (CBB) schemes. From an asymptotic perspective, either BB or CBB schemes are inferior to conjugate gradient schemes, for which the convergence rate can be superlinear. On the other hand, for a bound constrained optimization problem, where the active constraints at an optimal solution are unknown, the asymptotic convergence rate is irrelevant until the active constraints are identified. A nonmonotone BB or CBB iteration yields an efficient strategy for identifying active constraints.

When possible, the initial stepsize $\overline{\alpha}_k$ is given by the CBB formula

$$\overline{\alpha}_{k+j} = \overline{\alpha}_k^{BB} \quad \text{for } j = 0, \ldots, m-1,$$

where the BB step appears in (A.1) and $m$ is the number of times the BB step is reused. When $\overline{\alpha}_k^{BB} \notin [\alpha_{\min}, \alpha_{\max}]$, we project it on the interval $[\alpha_{\min}, \alpha_{\max}]$.

We now provide a more detailed statement of our algorithm for computing the initial stepsize. The integer $j$ counts the number of times the current BB step has been reused, while the parameter $m$ is the CBB memory (the maximum number of times the BB step will be reused).

Initial stepsize.

I0. If $k = 0$, choose $\overline{\alpha}_0 \in [\alpha_{\min}, \alpha_{\max}]$ and a parameter $\theta < 1$ near 1; set $j = 0$ and $flag = 1$. If $k > 0$, set $flag = 0$.
I1. If $0 < |d_{ki}| < \bar{\alpha}_k |g_{ki}|$ for some $i$, then set $flag = 1$.
I2. If $\alpha_k = 1$ in the NGPA, then set $j = j + 1$.
I3. If $\alpha_k < 1$ in the NGPA, then set $flag = 1$.

I4. If $j \geq m$ or $flag = 1$ or $\mathbf{s}_k^T\mathbf{y}_k/(\|\mathbf{s}_k\|\|\mathbf{y}_k\|) \geq \theta$, then
  a. If $\mathbf{s}_k^T\mathbf{y}_k \leq 0$, then
    1. If $j \geq 1.5m$, then set $t = \min\{\|\mathbf{x}_k\|_\infty, 1\}/\|\mathbf{d}^1(\mathbf{x}_k)\|_\infty$, $\overline{\alpha}_{k+1} = \min\{\alpha_{\max}, \max[t, \alpha_k]\}$, and $j = 0$.
    2. Else set $\overline{\alpha}_{k+1} = \overline{\alpha}_k$.
  b. Else set $\overline{\alpha}_{k+1} = \min\{\alpha_{\max}, \max[\alpha_{\min}, \mathbf{s}_k^T\mathbf{s}_k/\mathbf{s}_k^T\mathbf{y}_k]\}$ and $j = 0$.

Since this procedure always generates an initial stepsize $\overline{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, it complies with the requirement in step 1 of the NGPA. If the original BB step is truncated (see I1), or an Armijo line search is performed (see I3), or the cycle number $j$ reaches $m$ (see I4), or $\mathbf{s}_k^T\mathbf{y}_k/(\|\mathbf{s}_k\|\|\mathbf{y}_k\|)$ is close to 1 (see I4), then we try to compute a new BB step. The BB stepsize computation appears in step I4b. One motivation for computing a new BB step when $\mathbf{s}_k^T\mathbf{y}_k/(\|\mathbf{s}_k\|\|\mathbf{y}_k\|)$ is close to 1 is given in [30]; when $f$ is a quadratic, this condition is satisfied when the step $\mathbf{s}_k$ is close to an eigenvector of the Hessian. When $\mathbf{s}_k^T\mathbf{y}_k \leq 0$ (see I4a), the function is not convex on the line segment connecting $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, and a relatively large stepsize is used in the next iteration. A rationale for the step taken in this case appears in [57].

Now consider the reference function value $f_k^r$. Let $f_k$ denote $f(\mathbf{x}_k)$. In the algorithm which follows, the integer $a$ counts the number of consecutive iterations that $\alpha_k = 1$ in the NGPA (and the Armijo line search in step 4 is skipped). The integer $l$ counts the number of iterations since the function value is strictly decreased by an amount $\Delta > 0$.

Reference function value.

R0. If $k = 0$, choose parameters $A > L > 0$, $\gamma_1 > 1$, $\gamma_2 > 1$, and $\Delta > 0$; initialize $a = l = 0$ and $f_0^{\min} = f_0^{\max\min} = f_0^r = f_{-1}^r = f_0$.

R1. Update $f_k^r$ as follows:
  a. If $l = L$, then set $l = 0$ and
$$f_k^r = \begin{cases} f_k^{\max\min} & \text{if } \frac{f_k^{\max} - f_k^{\min}}{f_k^{\max\min} - f_k^{\min}} \geq \gamma_1, \\ f_k^{\max} & \text{otherwise.} \end{cases}$$
  b. Else if $a > A$, then set
$$f_k^r = \begin{cases} f_k^{\max} & \text{if } f_k^{\max} > f_k \text{ and } \frac{f_{k-1}^r - f_k}{f_k^{\max} - f_k} \geq \gamma_2, \\ f_{k-1}^r & \text{otherwise.} \end{cases}$$
  c. Otherwise, $f_k^r = f_{k-1}^r$.

R2. Set $f_R$ as follows in step 3 of the NGPA:
  a. If $j = 0$ (first iterate in a CBB cycle), then $f_R = f_k^r$.
  b. If $j > 0$, then $f_R = \min\{f_k^{\max}, f_k^r\}$.

  If $\alpha_k < 1$ in the NGPA, then set $a = 0$.

R3. If $\alpha_k = 1$ in the NGPA, then set $a = a + 1$.

R4. If $f_{k+1} \leq f_k^{\min} - \Delta$, then set $f_{k+1}^{\max\min} = f_{k+1}^{\min} = f_{k+1}$ and $l = 0$; otherwise, put $l = l + 1$, $f_{k+1}^{\min} = f_k^{\min}$, and $f_{k+1}^{\max\min} = \max\{f_k^{\max\min}, f_{k+1}\}$.

The variable $f_k^{\max}$, defined in (2.3), stores the maximum of recent function values. The variable $f_k^{\min}$ stores the minimum function value to within the tolerance $\Delta$. The variable $f_k^{\max\min}$ stores the maximum function value since the last new minimum was recorded in $f_k^{\min}$. More explanations concerning the choice of the reference function value are given in [30, 31]. Now, let us check that the choice for $f_k^r$ given above satisfies the requirements in step 2 of the NGPA.

*Proof that* $f_k \leq f_k^r$. In R1, we set

(i) $f_k^r = f_k^{\max\min}$ or
(ii) $f_k^r = f_k^{\max}$ or
(iii) $f_k^r = f_{k-1}^r$.

By R4, $f_k^{\max\min} \geq f_k$. In case (ii), $f_k^{\max} \geq f_k$ by the definition of $f_k^{\max}$. In steps 3 and 4 of the NGPA, we have $f_k \leq f_R \leq f_{k-1}^r$. Hence, in each of the cases (i)–(iii), we have $f_k \leq f_k^r$. □

*Proof that* $f_k^r \leq \max\{f_{k-1}^r, f_k^{\max}\}$. In R1a, $f_k^r$ is equal to either $f_k^{\max}$ or $f_k^{\max\min}$. Since $\gamma_1 > 1$, we set only $f_k^r = f_k^{\max\min}$ when $f_k^{\max\min} \leq f_k^{\max}$. Hence, in R1a, $f_k^r \leq f_k^{\max}$. In R1b, $f_k^r$ is equal to either $f_k^{\max}$ or $f_{k-1}^r$. Since $\gamma_2 > 1$, we set only $f_k^r = f_{k-1}^r$ when $f_{k-1}^r \geq f_k^{\max}$. Hence, in R1b, $f_k^r \leq f_k^{\max}$. In R1c, we set $f_k^r = f_{k-1}^r$. Combining these observations, $f_k^r \leq \max\{f_{k-1}^r, f_k^{\max}\}$ in R1a–R1c. □

*Proof that* $f_k^r \leq f_k^{\max}$ *infinitely often.* The condition $f_{k+1} \leq f_k^{\min} - \Delta$ in R4 is satisfied only a finite number of times when $f$ is bounded from below. Thus for $k$ sufficiently large, $f_k^r$ is updated in R1a every $L$ iterations. In this case, since $\gamma_1 > 1$, $f_k^r = f_k^{\max\min}$ only when $f_k^{\max\min} \leq f_k^{\max}$, which implies that $f_k^r \leq f_k^{\max}$. Hence, for large $k$, $f_k^r \leq f_k^{\max}$ every $L$ iterations. □

**Acknowledgments.** Constructive comments by the referees are gratefully acknowledged. In particular, the idea of splitting the local convergence analysis into the nondegenerate case (where there is no need to assume the strong second-order sufficient optimality condition), followed by the degenerate case, was suggested by one of the referees.

## REFERENCES

[1] B. M. Averick, R. G. Carter, J. J. Moré, and G. L. Xue, *The MINPACK-2 Test Problem Collection*, Tech. report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992.
[2] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.
[3] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–184.
[4] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.
[5] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1999.
[6] E. G. Birgin, R. Biloti, M. Tygel, and L. T. Santos, *Restricted optimization: A clue to a fast and accurate implementation of the common reflection surface stack method*, J. Appl. Geophys., 42 (1999), pp. 143–155.
[7] E. G. Birgin, I. Chambouleyron, and J. M. Martínez, *Estimation of the optical constants and the thickness of thin films using unconstrained optimization*, J. Comput. Phys., 151 (1999), pp. 862–880.
[8] E. G. Birgin and J. M. Martínez, *A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients*, in Topics in Numerical Analysis, Comput. Suppl. 15, Springer, Vienna, 2001, pp. 49–60.
[9] E. G. Birgin and J. M. Martínez, *Large-scale active-set box-constrained optimization method with spectral projected gradients*, Comput. Optim. Appl., 23 (2002), pp. 101–125.
[10] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. Optim., 10 (2000), pp. 1196–1211.
[11] E. G. Birgin, J. M. Martínez, and M. Raydan, *Algorithm* 813: *SPG—software for convex-constrained optimization*, ACM Trans. Math. Software, 27 (2001), pp. 340–349.
[12] E. G. Birgin, J. M. Martínez, and M. Raydan, *Inexact spectral projected gradient methods on convex sets*, IMA J. Numer. Anal., 23 (2003), pp. 539–559.
[13] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, *CUTE: Constrained and unconstrained testing environments*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.
[14] M. A. Branch, T. F. Coleman, and Y. Li, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM J. Sci. Comput., 21 (1999), pp. 1–23.

[15] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.
[16] J. V. Burke and J. J. Moré, *Exposing constraints*, SIAM J. Optim., 4 (1994), pp. 573–595.
[17] J. V. Burke, J. J. Moré, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Program., 47 (1990), pp. 305–336.
[18] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208.
[19] P. Calamai and J. Moré, *Projected gradient for linearly constrained problems*, Math. Program., 39 (1987), pp. 93–116.
[20] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North–Holland, Amsterdam, 1978.
[21] T. F. Coleman and Y. Li, *On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds*, Math. Program., 67 (1994), pp. 189–224.
[22] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.
[23] T. F. Coleman and Y. Li, *A Trust Region and Affine Scaling Interior Point Method for Nonconvex Minimization with Linear Inequality Constraints*, Tech. report, Cornell University, Ithaca, NY, 1997.
[24] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.
[25] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.
[26] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, SIAM, Philadelphia, 2000.
[27] Y. H. Dai, *On the nonmonotone line search*, J. Optim. Theory Appl., 112 (2002), pp. 315–330.
[28] Y. H. Dai and R. Fletcher, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numer. Math., 100 (2005), pp. 21–47.
[29] Y. H. Dai and R. Fletcher, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Math. Program., 106 (2006), pp. 403–421.
[30] Y. H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang, *The cyclic Barzilai-Borwein method for unconstrained optimization*, IMA J. Numer. Anal., to appear.
[31] Y. H. Dai and H. Zhang, *An adaptive two-point stepsize gradient algorithm*, Numer. Algorithms, 27 (2001), pp. 377–385.
[32] R. S. Dembo and U. Tulowitzki, *On the Minimization of Quadratic Functions Subject to Box Constraints*, Tech. report, School of Organization and Management, Yale University, New Haven, CT, 1983.
[33] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-region interior-point algorithms for a class of nonlinear programming problems*, SIAM J. Control Optim., 36 (1998), pp. 1750–1794.
[34] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
[35] Z. Dostál, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Optim., 7 (1997), pp. 871–887.
[36] Z. Dostál, *A proportioning based algorithm for bound constrained quadratic programming with the rate of convergence*, Numer. Algorithms, 34 (2003), pp. 293–302.
[37] Z. Dostál, A. Friedlander, and S. A. Santos, *Solution of coercive and semicoercive contact problems by FETI domain decomposition*, Contemp. Math., 218 (1998), pp. 82–93.
[38] Z. Dostál, A. Friedlander, and S. A. Santos, *Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints*, SIAM J. Optim., 13 (2003), pp. 1120–1140.
[39] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of the primal-dual Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.
[40] A. S. El-Bakry, R. A. Tapia, and Y. Zhang, *On the convergence rate of Newton interior-point methods in the absence of strict complementarity*, Comput. Optim. Appl., 6 (1996), pp. 157–167.
[41] F. Facchinei, A. Fischer, and C. Kanzow, *On the accurate identification of active constraints*, SIAM J. Optim., 9 (1998), pp. 14–32.
[42] F. Facchinei, J. Júdice, and J. Soares, *An active set Newton algorithm for large-scale nonlinear programs with box constraints*, SIAM J. Optim., 8 (1998), pp. 158–186.

[43] F. Facchinei and S. Lucidi, *A class of penalty functions for optimization problems with bound constraints*, Optimization, 26 (1992), pp. 239–259.
[44] F. Facchinei, S. Lucidi, and L. Palagi, *A truncated Newton algorithm for large scale box constrained optimization*, SIAM J. Optim., 12 (2002), pp. 1100–1125.
[45] R. Fletcher, *On the Barzilai-Borwein Method*, Tech. report, Department of Mathematics, University of Dundee, Dundee, Scotland, 2001.
[46] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim., 30 (1994), pp. 235–266.
[47] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, New York, 1984.
[48] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114–120.
[49] A. A. Goldstein, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc., 70 (1964), pp. 709–710.
[50] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.
[51] L. Grippo and M. Sciandrone, *Nonmonotone globalization techniques for the Barzilai-Borwein gradient method*, Comput. Optim. Appl., 23 (2002), pp. 143–169.
[52] W. W. Hager, *Dual techniques for constrained optimization*, J. Optim. Theory Appl., 55 (1987), pp. 37–71.
[53] W. W. Hager, *Analysis and implementation of a dual algorithm for constrained optimization*, J. Optim. Theory Appl., 79 (1993), pp. 427–462.
[54] W. W. Hager and H. Zhang, *CG_DESCENT User's Guide*, Tech. report, Department of Mathematics, University of Florida, Gainesville, FL, 2004.
[55] W. W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim., 16 (2005), pp. 170–192.
[56] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pacific J. Optim., 2 (2006), pp. 35–58.
[57] W. W. Hager and H. Zhang, *Algorithm* 851: *CG_DESCENT, a conjugate gradient method with guaranteed descent*, ACM Trans. Math. Software, 32 (2006), pp. 113–137.
[58] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption*, Math. Program., 86 (1999), pp. 615–635.
[59] C. Kanzow and A. Klug, *On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints*, Comput. Optim. Appl., to appear.
[60] M. Lescrenier, *Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold*, SIAM J. Numer. Anal., 28 (1991), pp. 476–495.
[61] E. S. Levitin and B. T. Polyak, *Constrained minimization problems*, USSR Comput. Math. Math. Physics, 6 (1966), pp. 1–50.
[62] C.-J. Lin and J. J. Moré, *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21 (1999), pp. 24–45.
[63] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.
[64] W. B. Liu and Y. H. Dai, *Minimization algorithms based on supervisor and searcher cooperation*, J. Optim. Theory Appl., 111 (2001), pp. 359–379.
[65] J. M. Martínez, *BOX-QUACAN and the implementation of augmented Lagrangian algorithms for minimization with inequality constraints*, J. Comput. Appl. Math., 19 (2000), pp. 31–56.
[66] G. P. McCormick and R. A. Tapia, *The gradient projection method under mild differentiability conditions*, SIAM J. Control, 10 (1972), pp. 93–98.
[67] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.
[68] B. T. Polyak, *The conjugate gradient method in extremal problems*, USSR Comput. Math. Math. Phys., 9 (1969), pp. 94–112.
[69] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., 7 (1997), pp. 26–33.
[70] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.
[71] A. Schwartz and E. Polak, *Family of projected descent methods for optimization problems with simple bounds*, J. Optim. Theory Appl., 92 (1997), pp. 1–31.
[72] T. Serafini, G. Zanghirati, and L. Zanni, *Gradient projection methods for quadratic programs and applications in training support vector machines*, Optim. Methods Softw., 20 (2005), pp. 353–378.

[73] P. L. Toint, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[74] P. L. Toint, *An assessment of nonmonotone line search techniques for unconstrained optimization*, SIAM J. Sci. Comput., 17 (1996), pp. 725–739.

[75] P. L. Toint, *A non-monotone trust region algorithm for nonlinear optimization subject to convex constraints*, Math. Program., 77 (1997), pp. 69–94.

[76] M. Ulbrich, S. Ulbrich, and M. Heinkenschloss, *Global convergence of trust-region interior-point algorithms for infinite-dimensional nonconvex minimization subject to pointwise bounds*, SIAM J. Control Optim., 37 (1999), pp. 731–764.

[77] S. J. Wright, *Implementing proximal point methods for linear programming*, J. Optim. Theory Appl., 65 (1990), pp. 531–554.

[78] H. Yamashita and H. Yabe, *Superlinear and quadratic convergence of some primal-dual interior-point methods for constrained optimization*, Math. Program., 75 (1996), pp. 377–397.

[79] E. K. Yang and J. W. Tolle, *A class of methods for solving large convex quadratic programs subject to box constraints*, Math. Program., 51 (1991), pp. 223–228.

[80] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory*, in Contributions to Nonlinear Functional Analysis, E. H. Zarantonello, ed., Academic Press, New York, 1971, pp. 237–424.

[81] H. Zhang and W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim., 14 (2004), pp. 1043–1056.

[82] H. Zhang and W. W. Hager, *PACBB: A projected adaptive cyclic Barzilai-Borwein method for box constrained optimization*, in Multiscale Optimization Methods and Applications, William W. Hager, Shu-Jen Huang, Panos M. Pardalos, and Oleg A. Prokopyev, eds., Springer, New York, 2005, pp. 387–392.

[83] Y. Zhang, *Interior-point Gradient Methods with Diagonal-scalings for Simple-bound Constrained Optimization*, Tech. report TR04-06, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2004.

[84] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, ACM Trans. Math. Software, 23 (1997), pp. 550–560.