RECENT ADVANCES IN BOUND CONSTRAINED OPTIMIZATION

W. W. Hager,[1] and H. Zhang[1]

[1] *PO Box 118105, Department of Mathematics, University of Florida, Gainesville, FL 32611-8105,{hager, hzhang}@math.ufl.edu, http://www.math.ufl.edu/∼hager, http://www.math.ufl.edu/∼hzhang**

Abstract A new active set algorithm (ASA) for large-scale box constrained optimization is introduced. The algorithm consists of a nonmonotone gradient projection step, an unconstrained optimization step, and a set of rules for switching between the two steps. Numerical experiments and comparisons are presented using box constrained problems in the CUTEr and MINPACK test problem libraries.

keywords: Nonmonotone gradient projection, box constrained optimization, active set algorithm, ASA, cyclic BB method, CBB, conjugate gradient method, CG_DESCENT, degenerate optimization

1. Introduction

We present a new active set algorithm for solving the box constrained optimization problem

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{B}\}, \tag{1}$$

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

$$\mathcal{B} = \{\mathbf{x} \in \Re^n : \mathbf{l} \le \mathbf{x} \le \mathbf{u}\}. \tag{2}$$

Here $\mathbf{l} < \mathbf{u}$; possibly, $l_i = -\infty$ or $u_i = \infty$. The following notation is used throughout the paper: $\|\cdot\|$ is the Euclidean norm of a vector, the subscript k is used for the iteration number, 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 and $^{\mathsf{T}}$ denotes transpose. The gradient at the iterate $\mathbf{x}_k$ is $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$, the Hessian of f at $\mathbf{x}$ is $\nabla^2 f(\mathbf{x})$, and the ball with center $\mathbf{x}$ and radius ρ is $B_\rho(\mathbf{x})$.

*This paper is based upon work supported by the National Science Foundation under Grant No. 0203270.

Please use the following format when citing this chapter:
Author(s) [insert Last name, First-name initial(s)], 2006, in IFIP International Federation for Information Processing, Volume 199, System Modeling and Optimization, eds. Ceragioli F., Dontchev A., Furuta H., Marti K., Pandolfi L., (Boston: Springer), pp. [insert page numbers].

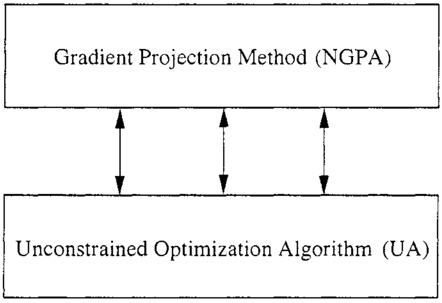
Gradient Projection Method (NGPA)
Unconstrained Optimization Algorithm (UA)

Figure 1. Structure of ASA.

The problem (1) may result from the discretization of a variational inequality such as the obstacle problem [42, 49]:

$$\begin{gathered}\min \int_\Omega \|\nabla u(x)\|^2 + 2f(x)u(x)dx \\ \text{subject to } u(x) \geq \psi(x) \quad a.e.\end{gathered}$$

It may come from the discretization of a control problem such as

$$\begin{gathered}\min \quad f(\mathbf{x}, \mathbf{u}) \\ \text{subject to } \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{u} \geq \mathbf{0} \quad a.e.,\end{gathered}$$

where $\mathbf{A}$ and $\mathbf{B}$ are operators and the dot denotes time derivative. It also appears as the subproblem in augmented Lagrangian or penalty methods [18, 4, 26, 27, 30, 32, 47]. For example, in an augmented Lagrangian approach to the nonlinear optimization

$$\min f(\mathbf{x}) \text{ subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \geq \mathbf{0},$$

we might solve the box constrained subproblem

$$\min f(\mathbf{x}) + \lambda^{\mathsf{T}}\mathbf{h}(\mathbf{x}) + p\|\mathbf{h}(\mathbf{x})\|^2 \text{ subject to } \mathbf{x} \geq \mathbf{0},$$

where p is the penalty parameter and λ is an approximation to a Lagrange multiplier for the equality constraint. Thus efficient algorithms for large-scale box constrained optimization problems are important, both in theory and practice.

2. Gradient projection methods

Our active set algorithm (ASA) has two phases as indicated in Figure 1 a gradient projection phase and an unconstrained optimization phase. For the unconstrained optimization phase, we exploit the box structure of the constraints

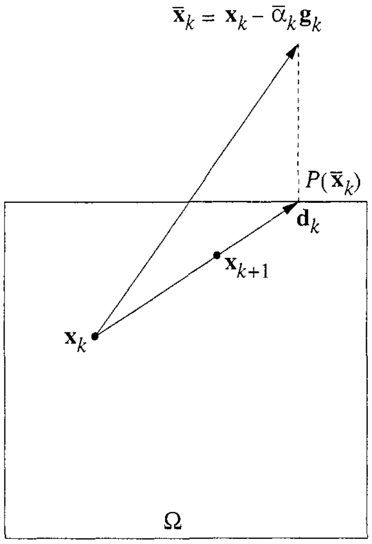

Figure 2. The gradient projection step.

in (1), while the gradient projection phase can be applied to any problem with a closed, convex feasible set. Hence, in this section, we consider a more general problem in which the box $\mathcal{B}$ is replaced by a nonempty, closed convex set Ω:

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \Omega\}. \tag{3}$$

Let P denote the projection onto Ω. The gradient projection step at iteration k is depicted in Figure 2. Starting from the current iterate $\mathbf{x}_k$, we take a positive step $\overline{\alpha}_k$ along the negative gradient arriving at $\bar{\mathbf{x}}_k = \mathbf{x}_k - \overline{\alpha}_k \mathbf{g}_k$. If $\bar{\mathbf{x}}_k$ is outside Ω, then we apply the projection P to obtain a point $P(\bar{\mathbf{x}}_k)$ on the boundary of Ω. The search direction $\mathbf{d}_k$ is along the line segment $[\mathbf{x}_k, P(\bar{\mathbf{x}}_k)]$. The new iterate $\mathbf{x}_{k+1}$ is obtained by a line search along the search direction.

A more precise statement of the gradient projection algorithm follows:

Nonmonotone Gradient Projection Algorithm (NGPA)

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

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 $f^r_k$ so that $f(\mathbf{x}_k) \le f^r_k \le \max\{f^r_{k-1}, f^{\max}_k\}$ and $f^r_k \le$
       $f^{\max}_k$ infinitely often.
```

3. Let f_R be either f_k^r or $\min\{f_k^{\max}, f_k^r\}$. If $f(\mathbf{x}_k + \mathbf{d}_k) \leq f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k$, 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

$$f(\mathbf{x}_k + \eta^j \mathbf{d}_k) \leq f_R + \eta^j \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k. \tag{4}$$

5. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ and $k = k + 1$.

End

The statement of NGPA involves the following parameters:

$\epsilon \in [0, \infty)$	–	convergence tolerance ($P(\mathbf{x}_k - \mathbf{g}_k) = \mathbf{x}_k$ if and only if $\mathbf{x}_k$ is a stationary point)
$[\alpha_{\min}, \alpha_{\max}] \subset (0, \infty)$	–	bound on the stepsize in Step 1
$f_k^{\max}$	–	$\max\{f(\mathbf{x}_{k-i}) : 0 \leq i \leq \min(k, M-1)\}$ (local maximum of function values near $\mathbf{x}_k$, $M > 0$)
$\delta, \eta \in (0, 1)$	–	parameters entering the Armijo line search in Step 4

In Step 2, the requirement that "$f_k^r \leq f_k^{\max}$ infinitely often" is needed for global convergence. This is a rather weak condition which can be satisfied by many strategies. For example, 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

$$f(\mathbf{x}_{k-L}) - f(\mathbf{x}_k) \leq \Delta.$$

Thus we set $f_k^r = f_k^{\max}$ when the function values decrease "too slowly."

For our numerical experiments, the initial step $\bar{\alpha}_k$ in Step 1 is generated by a cyclic Barzilai-Borwein method developed in [20]. The traditional Barzilai and Borwein stepsize [3] is

$$\overline{\alpha}_{k+1}^{BB} = \frac{\mathbf{s}_k^{\mathsf{T}} \mathbf{s}_k}{\mathbf{s}_k^{\mathsf{T}} \mathbf{y}_k}, \tag{5}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. If the same BB stepsize is repeated for several iterations, then even faster convergence can be achieved (see [20]). These schemes in which the same BB stepsize are repeated for several iterations are called cyclic BB schemes (CBB). The CBB update formula is

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

where m is the cycle length, and k is a multiple of m. The cycle length can be chosen in a adaptive way, as explained in [20].

Our line search along the search direction $\mathbf{d}_k$ is an Armijo type line search [1], which may be viewed as a relaxed version of the Grippo, Lampariello, and Lucidi nonmonotone line search [34] (denoted GLL). For NGPA, the GLL scheme corresponds to $f_k^r = f_k^{\max}$ for each k. In practice, we have obtained faster convergence results by allowing the reference function value f_k^r to decay more slowly on average than $f_k^{\max}$.

Our statement of the gradient projection algorithm employs a direction operator $\mathbf{d}^\alpha(\mathbf{x})$ given by

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

where α is a scalar. Some properties of $\mathbf{d}^\alpha$ are summarized below (see [37] for further details concerning these properties and other results presented in this paper):

Proposition 1 *P and $\mathbf{d}^\alpha$ have the following properties:*

P1. $\|P(\mathbf{x}) - P(\mathbf{y})\| \le \|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}$ *and* $\mathbf{y} \in \Re^n$.

P2. *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* (3).

P3. *Suppose* $\mathbf{x}^*$ *is a stationary point for* (3). *If for some* $\mathbf{x} \in \Re^n$, *there exist positive scalars* λ *and* γ *such that*

$$(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*))^{\mathsf{T}}(\mathbf{x} - \mathbf{x}^*) \ge \gamma\|\mathbf{x} - \mathbf{x}^*\|^2 \tag{6}$$

and

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*)\| \le \lambda\|\mathbf{x} - \mathbf{x}^*\|, \tag{7}$$

then we have

$$\|\mathbf{x} - \mathbf{x}^*\| \le \left(\frac{1+\lambda}{\gamma}\right)\|\mathbf{d}^1(\mathbf{x})\|. \tag{8}$$

P4. *Suppose that f is twice-continuously differentiable near a stationary point* $\mathbf{x}^*$ *of* (1) *satisfying the strong second-order sufficient optimality condition; that is, there exists* $\gamma > 0$ *such that*

$$\mathbf{d}^{\mathsf{T}}\nabla^2 f(\mathbf{x}^*)\mathbf{d} \ge \gamma\|\mathbf{d}\|^2 \tag{9}$$

for all $\mathbf{d} \in \Re^n$ *with the property that* $d_i = 0$ *when* $x_i = 0$ *and* $g_i(\mathbf{x}^*) > 0$. *Then there exists* $\rho > 0$ *with the following property:*

$$\|\mathbf{x} - \mathbf{x}^*\| \le \sqrt{1 + \left(\frac{(1+\lambda)^2}{.5\gamma}\right)^2}\,\|\mathbf{d}^1(\mathbf{x})\| \tag{10}$$

whenever $\mathbf{x} \in B_\rho(\mathbf{x}^*)$, *where* λ *is any Lipschitz constant for* ∇f *on* $B_\rho(\mathbf{x}^*)$.

In P3 we assume a convexity/monotonicity type condition at $\mathbf{x}$; for any $\mathbf{x}$ which satisfies (6) and the Lipschitz condition (7), we can estimate the error in $\mathbf{x}$ in accordance with (8). In P4, we make a convexity type assumption at $\mathbf{x}^*$ (the strong second-order sufficient optimality condition), and we have the error estimate (10) in a neighborhood of $\mathbf{x}^*$. Based on P3 and P4, the Lipschitz continuity of $\mathbf{d}^1(\cdot)$ implied by P1, and the fact P2 that $\mathbf{d}^1(\mathbf{x}) = \mathbf{0}$ if and only if $\mathbf{x}$ is a stationary point, the function $\mathbf{d}^1(\mathbf{x})$ can be used to measure the error in any iterate $\mathbf{x}_k$. In particular the convergence condition $\|P(\mathbf{x}_k - \mathbf{g}_k) - \mathbf{x}_k\| \le \epsilon$ in NGPA is equivalent to $\|\mathbf{d}^1(\mathbf{x}_k)\| \le \epsilon$.

Sufficient conditions for the global convergence of NGPA are given below.

THEOREM 1 *Let* $\mathcal{L}$ *be the level set defined by*

$$\mathcal{L} = \{\mathbf{x} \in \Omega : f(\mathbf{x}) \le f(\mathbf{x}_0)\}. \tag{11}$$

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 ∇f is Lipschitz continuous on $\bar{\mathcal{L}}$.*

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

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

When f is a strongly convex function, Theorem 1 can be strengthened as follows:

COROLLARY 2 *Suppose f is strongly convex and twice continuously differentiable on Ω, 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 NGPA with $\epsilon = 0$ converge to the global minimizer $\mathbf{x}^*$.*

3. Active Set Algorithm

In this section, we focus on the active set algorithm. Unlike the gradient projection algorithm where the feasible set can be any closed, convex set, we now restrict ourselves to box constraints. Moreover, to simplify the discussion, we consider (without loss of generality) the special case $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \infty$. In other words, the constraint is $\mathbf{x} \ge \mathbf{0}$.

Although the gradient projection scheme NGPA has an attractive global convergence theory, the convergence rate can be slow in a neighborhood of a local minimizer. We accelerate the convergence by exploiting a superlinearly convergent algorithm for unconstrained minimization. For the numerical experiments, we utilize the conjugate gradient code CG_DESCENT [35, 38, 36, 39] for the unconstrained algorithm (UA). In general, any UA satisfying the following conditions can be employed:

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

U3. If $x_{ji} > 0$ for $j \geq k$, then $\liminf\limits_{j \geq k} |g_i(\mathbf{x}_j)| = 0$.

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

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

where $\phi(\alpha) = f(P(\mathbf{x}_k - \alpha \mathbf{g}_I(\mathbf{x}_k)))$, $0 < \delta < \sigma < 1$, and $\mathbf{g}_I(\mathbf{x})$ is the part of the gradient associated with inactive constraints:

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

Conditions U1–U3 are sufficient for global convergence, while U1–U4 are sufficient for the local convergence results summarized below. U4 could be replaced by another descent condition for the initial line search, however, the local analysis in [37] has been carried out under U4.

The active set algorithm is based on a set of rules which determine when we switch between NGPA and UA. These rules correspond to the double arrows in Figure 1. Before presenting the switching rules, we give some motivation. A fundamental set embedded in our switching rules is the "undecided index set" $\mathcal{U}$:

$$\mathcal{U}(\mathbf{x}) = \{i \in [1, n] : |g_i(\mathbf{x})| \geq \|\mathbf{d}^1(\mathbf{x})\|^\alpha \text{ and } x_i \geq \|\mathbf{d}^1(\mathbf{x})\|^\beta\}, \tag{12}$$

where $\alpha \in (0, 1)$ and $\beta \in (1, 2)$ are fixed constants. In the numerical experiments, we take $\alpha = 1/2$ and $\beta = 3/2$. Observe that at a local minimizer $\mathbf{x}^*$, the only components of the gradient which do not vanish are associated with components of $\mathbf{x}^*$ at the boundary of the feasible set. The undecided index set consists of indices of large gradient components with large $\mathbf{x}$ components (in the sense of (12)).

We show [37] that if f is twice continuously differentiable, then for any algorithm converging to a stationary point where each iterate is generated by either NGPA or a UA satisfying U1–U4, the set $\mathcal{U}(\mathbf{x}_k)$ is empty for k sufficiently large. This result does not depend on the rules used to switch between NGPA and UA. When $\mathcal{U}(\mathbf{x}_k)$ becomes empty while performing NGPA, we feel that the strictly active constraints at a stationary point are almost identified and we may switch to UA to exploit its faster convergence.

Another quantity which enters into our switching rules is the ratio between the norm of the inactive gradient components $\|\mathbf{g}_I(\mathbf{x})\|$ and the error estimator $\|\mathbf{d}^1(\mathbf{x})\|$. By U3, $\mathbf{g}_I(\mathbf{x}_k)$ tends to zero as iterates are generated by the UA. By U2, UA does not free constraints; hence, any limit, say $\mathbf{y}^*$, of iterates typically does not solve the original problem (1). In other words, $\mathbf{d}^1(\mathbf{y}^*)$ may not be $\mathbf{0}$. We stop the UA and switch to the NGPA when $\|\mathbf{g}_I(\mathbf{x}_k)\|$ is sufficiently small relative to $\|\mathbf{d}^1(\mathbf{x}_k)\|$. More precisely, we introduce a parameter $\mu > 0$ and we branch from UA to NGPA when

$$\|\mathbf{g}_I(\mathbf{x}_k)\| \le \mu\|\mathbf{d}^1(\mathbf{x}_k)\|. \tag{13}$$

Unlike UA where bound components are fixed by U2, NGPA allows bound components of $\mathbf{x}_k$ to move into the interior of the feasible set. Hence, by switching from UA to NGPA, the iterates are able to move to a new face of the feasible set. In NGPA we may decrease μ, in which case the accuracy with which we solve subproblems in UA increases.

Assuming f is twice continuously differentiable, we show in [37] that for a local minimizer $\mathbf{x}^*$ satisfying the strong second-order sufficient optimality condition (9) and for any sequence of iterates generated by either NGPA or a UA satisfying U1–U4, there exists a scalar $\mu^* > 0$ such that

$$\|\mathbf{g}_I(\mathbf{x}_k)\| \ge \mu^*\|\mathbf{d}^1(\mathbf{x}_k)\| \tag{14}$$

for k sufficiently large. As a result, when μ becomes sufficiently small, condition (13) is never satisfied; hence, if the switch from UA to NGPA is dictated by (13), we conclude that the iterates will never leave the UA. In other words, we eventually solve (1) using the unconstrained optimization algorithm.

With these insights, we now state ASA, or equivalently, we give the switching rules:

Active Set Algorithm (ASA)

```
1. While $\|\mathbf{d}^1(\mathbf{x}_k)\| > \epsilon$ execute 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 $\mathcal{A}(\mathbf{x}_k) = \mathcal{A}(\mathbf{x}_{k-1}) = \ldots = \mathcal{A}(\mathbf{x}_{k-n_1})$, 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 UA and check the following:
     a. If $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$, then restart 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 UA at $\mathbf{x}_k$.
            Else restart NGPA.
     End
   End
```

In addition to the convergence tolerance ϵ introduced previously, ASA utilizes the following four parameters:

$\mu \in (0, 1)$	–	$\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$ implies the UA subproblem solved with sufficient accuracy
$\rho \in (0, 1)$	–	decay factor used to decrease μ in NGPA
$n_1, n_2 \in [1, n)$	–	integers connected with active set repetitions or change

A strong convergence theory can be developed for this algorithm. The following global convergence property holds:

THEOREM 3 *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 ∇f is Lipschitz continuous on $\bar{\mathcal{L}}$.*

A3. *UA satisfies* U1–U3.

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

$$\liminf_{k \to \infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0. \tag{15}$$

For strongly convex objective functions, the global convergence result can be strengthened as follows.

THEOREM 4 *If f is strongly convex and twice continuously differentiable on $\mathcal{B}$, and assumptions* A2 *and* A3 *of Theorem* 3 *are satisfied, then the iterates* $\mathbf{x}_k$ *of ASA with* $\epsilon = 0$ *converge to the global minimum.*

Under the hypotheses of the following theorem, ASA eventually reduces to the unconstrained algorithm with a fixed active constraint set. In other words, the constrained problem is eventually solved by the unconstrained algorithm.

THEOREM 5 *If f is twice-continuously differentiable and the iterates* $\mathbf{x}_k$ *generated by ASA with* $\epsilon = 0$ *converge to a stationary point satisfying the strong second-order sufficient optimality condition, then after a finite number of iterations, ASA performs only the UA without restarts.*

When f is a strongly convex quadratic function, the iterates $\mathbf{x}_k$ converge to the global minimizer $\mathbf{x}^*$ by Theorem 4. Thus, if the UA is based on the conjugate gradient method, it follows from Theorem 5 that ASA converges in a finite number of iterations, since the conjugate gradient method has finite convergence when applied to a convex quadratic.

In our analysis, summarized above, we never claim that the active indices at a stationary point $\mathbf{x}^*$ can be identified in a finite number of iterations. In fact, there is a fundamental difference between the gradient projection algorithm presented in this paper, and algorithms based on a "piecewise projected gradient" [11–13]. For our gradient projection algorithm, we perform a single projection, and then we back track towards the starting point. We are unable to show that the active constraints are identified in a finite number of iterations. In the piecewise projected gradient approach, where a series of projections may be performed, the active constraints can be identified in a finite number of iterations. Even though we do not identify the active constraints, we show in [37] that the components of $\mathbf{x}_k$ corresponding to the strictly active constraints are on the order of $||\mathbf{x}_k - \mathbf{x}^*||^2$. Moreover, in our experience, the single-projection approach is more efficient in practice.

4. Numerical Experiments

In this section, we compare the CPU time performance of ASA to the performance of other algorithms for box constrained optimization. We begin with a brief overview of algorithm development for box constrained optimization.

One important line of research focused on the development of conjugate gradient methods for box constrained problems with a quadratic objective function. Polyak's 1969 seminal work [50] 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 only added or dropped one constraint in each iteration, Dembo and Tulowitzki proposed [21] an algorithm CGP which could add and drop many constraints in an iteration. Later, Yang and Tolle [55] further developed this algorithm so as to obtain finite termination, even when the problem was degenerate at a local minimizer $\mathbf{x}^*$. That is, for some i, $x_i^* = 0$ and $g_i(\mathbf{x}^*) = 0$. Another variation of the CGP algorithm, for which there is a rigorous convergence theory, is developed by Wright [53]. Moré and Toraldo [49] 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 *et al.* [24, 25, 27] 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 [4, 14, 33, 44, 48] focused on gradient projection methods. To accelerate the convergence, more recent research has developed Newton and trust region methods. In [1, 13, 18, 29] superlinear and quadratic convergence is established for nondegenerate problems, while [31, 32, 43, 46] establish analogous convergence results, even for degenerate problems. Although computing a Newton step can be expensive computationally, approximation techniques, such as a sparse, incomplete Cholesky factorization [45], could be used to reduce the computational expense. Nonetheless, for large-dimensional problems or for problems where the initial guess is far from the solution, the Newton/trust region approach can be inefficient. In cases where the Newton step is unacceptable, a gradient projection step is preferred.

The affine-scaling interior point method of Coleman and Li [10, 15–17] is a different approach to (1), related to the trust region algorithm. More recent research on this strategy includes [22, 40, 41, 52, 56]. 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 [51] 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 [56] proposes 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 [28, 54].

We compare the performance of ASA to the following four codes:

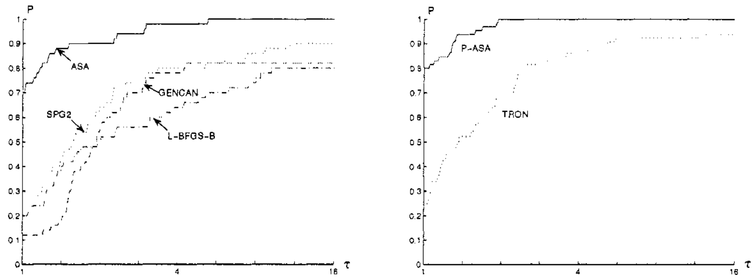

Figure 3. Performance profiles, 50 CUTEr test problems (left), 42 sparsest CUTEr problems, 23 MINPACK-2 problems (right)

- L-BFGS-B [57]: The limited memory quasi-Newton method of Zhu, Byrd, Nocedal (ACM Algorithm 778).
- SPG2 Version 2.1 [7, 8]: The nonmonotone spectral projected gradient method of Birgin, Martínez, and Raydan (ACM Algorithm 813).
- GENCAN [6]: The monotone active set method with spectral projected gradients developed by Birgin and Martínez.
- TRON Version 1.2 [46]: A Newton trust region method with incomplete Cholesky preconditioning developed by Lin and Moré.

These codes are all carefully written, high quality codes that reflect the different approaches to box constrained optimization summarized above. All codes are written in Fortran and compiled with f77 (default compiler settings) on a Sun workstation. The stopping condition was

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

where $\|\cdot\|_\infty$ denotes the sup-norm of a vector. In running any of these codes, default values were used for all parameters. Our test problem set consisted of all 50 box constrained problems in the CUTEr library [9] with dimensions between 50 and 15,625, and all 23 box constrained problems in the MINPACK-2 library [2] with dimension 2500. The performance of the algorithms, relative to CPU time, was evaluated using the performance profiles of Dolan and Moré [23]. That is, for each method, we plot the fraction P of problems for which the method is within a factor τ of the best time.

TRON is somewhat different from the other codes since it employs Hessian information and an incomplete Cholesky preconditioner, while the other codes only utilize gradient information. In Figure 3, left, we compare the performance of the four gradient based codes ASA, L-BFGS-B, SPG2, and GENCAN using

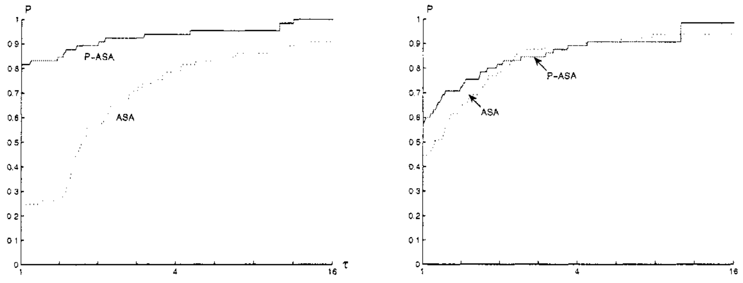

Figure 4. Performance comparison for P-ASA and ASA, $\epsilon = 10^{-6}$ (left), for $\epsilon = 10^{-2}\|d^1(x_0)\|_\infty$ (right)

the 50 CUTEr test problems. In a performance profile, the top curve corresponds to the method which solved the largest fraction of problems in a time within a factor τ of the best time. According to Figure 3, left, ASA achieves better CPU time performance than the other methods for this test set.

In order to compare ASA to the Hessian-based code TRON, we incorporated preconditioning in the conjugate gradient iteration. The preconditioner was the inverse of the incomplete Cholesky factorization of the Hessian at the current iterate. That is, we extracted the incomplete Cholesky factorization from TRON and used it in our code; hence, the two codes were using precisely the same approximation to the Hessian at each iterate. We let P-ASA denote this preconditioned version of ASA. Since TRON is targeted to large-sparse problems, such as the MINPACK problems, we compare P-ASA to TRON using the 23 MINPACK-2 problems and the 42 sparsest CUTEr problems (the number of nonzeros in the Hessian at most 1/5 the total number of entries in the Hessian). In Figure 3, right, we see that P-ASA has better CPU time performance than TRON in this test set.

In Figure 4, left, we compare the performance of P-ASA to that of ASA using the 42 sparsest CUTEr problems and the 23 MINPACK-2 problems. Clearly, the preconditioning was effective for this problem set and the convergence tolerance $\epsilon = 10^{-6}$. In Figure 4, right, the convergence tolerance is relaxed to $\epsilon = 10^{-2}\|d^1(x_0)\|_\infty$. With this relaxed convergence tolerance, there is not much difference between the preconditioned and the unconditioned codes.

References

[1] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J. Math.*, 16:1–3, 1966.

[2] B. M. Averick, R. G. Carter, J. J. Moré, and G. L. Xue. The MINPACK-2 test problem collection. Technical report, Mathematics and Computer Science Division, Argonne

National Laboratory, Argonne, IL, 1992.

[3] J. Barzilai and J. M. Borwein. Two point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[4] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automatic Control*, 21:174–184, 1976.

[5] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM J. Control Optim.*, 20:221–246, 1982.

[6] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Comput. Optim. Appl.*, 23:101–125, 2002.

[7] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods for convex sets. *SIAM J. Optim.*, 10:1196–1211, 2000.

[8] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813: SPG - software for convex-constrained optimization. *ACM Trans. Math. Software*, 27:340–349, 2001.

[9] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint. CUTE: constrained and unconstrained testing environments. *ACM Trans. Math. Software*, 21:123–160, 1995.

[10] 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.*, 21:1–23, 1999.

[11] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM J. Numer. Anal.*, 25:1197–1211, 1988.

[12] J. V. Burke and J. J. Moré. Exposing constraints. *SIAM J. Optim.*, 25:573–595, 1994.

[13] J. V. Burke, J. J. Moré, and G. Toraldo. Convergence properties of trust region methods for linear and convex constraints. *Math. Prog.*, 47:305–336, 1990.

[14] P. Calamai and J. Moré. Projected gradient for linearly constrained problems. *Math. Prog.*, 39:93–116, 1987.

[15] T. F. Coleman and Y. Li. On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Math. Prog.*, 67:189–224, 1994.

[16] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6:418–445, 1996.

[17] T. F. Coleman and Y. Li. A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints. Technical report, Cornell University, Ithaca, NY, 1997.

[18] 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.*, 25:433–460, 1988.

[19] 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.*, 28:545–572, 1991.

[20] Y. H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang. The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA J. Numer. Anal.*, submitted, 2005.

[21] R. S. Dembo and U. Tulowitzki. On the minimization of quadratic functions subject to box constraints. Technical report, School of Organization and Management, Yale University, New Haven, CT, 1983.

[22] 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.*, 36:1750–1794, 1998.

[23] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[24] Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM J. Optim.*, 7:871–887, 1997.

[25] Z. Dostál. A proportioning based algorithm for bound constrained quadratic programming with the rate of convergence. *Numer. Algorithms*, 34:293–302, 2003.

[26] Z. Dostál, A. Friedlander, and S. A. Santos. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemp. Math.*, 218:82–93, 1998.

[27] 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.*, 13:1120–1140, 2003.

[28] 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.*, 89:507–541, 1996.

[29] F. Facchinei, J. Júdice, and J. Soares. An active set Newton's algorithm for large-scale nonlinear programs with box constraints. *SIAM J. Optim.*, 8:158–186, 1998.

[30] F. Facchinei and S. Lucidi. A class of penalty functions for optimization problems with bound constraints. *Optimization*, 26:239–259, 1992.

[31] F. Facchinei, S. Lucidi, and L. Palagi. A truncated Newton algorithm for large-scale box constrained optimization. *SIAM J. Optim.*, 4:1100–1125, 2002.

[32] A. Friedlander, J. M. Martínez, and S. A. Santos. A new trust region algorithm for bound constrained minimization. *Appl. Math. Optim.*, 30:235–266, 1994.

[33] A .A. Goldstein. Convex programming in Hilbert space. *Bull. Amer. Math. Soc.*, 70:709–710, 1964.

[34] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM J. Numer. Anal.*, 23:707–716, 1986.

[35] W. W. Hager and H. Zhang. CG_DESCENT user's guide. Technical report, Dept. Math., Univ. Florida, 2004.

[36] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.*, 16:170–192, 2005.

[37] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. Optim.*, submitted, 2005.

[38] W. W. Hager and H. Zhang. CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Software*, to appear 2006.

[39] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific J. Optim.*, to appear 2006.

[40] 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. Prog.*, 86:615–635, 1999.

[41] C. Kanzow and A. Klug. On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints. *Comput. Optim. Appl.*, 2006, to appear.

[42] D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and their applications*, volume 31 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 2000.

[43] M. Lescrenier. Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold. *SIAM J. Numer. Anal.*, 28:476–495, 1991.

[44] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Comput. Math. Math. Physics*, 6:1–50, 1966.

[45] C. J. Lin and J. J. Moré. Incomplete cholesky factorizations with limited memory. *SIAM J. Sci. Comput.*, 21:24–45, 1999.

[46] C. J. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM J. Optim.*, 9:1100–1127, 1999.

[47] J. M. Martínez. BOX-QUACAN and the implementation of augmented Lagrangian algorithms for minimization with inequality constraints. *J. Comput. Appl. Math.*, 19:31–56, 2000.

[48] G. P. McCormick and R. A. Tapia. The gradient projection method under mild differentiability conditions. *SIAM J. Control*, 10:93–98, 1972.

[49] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optim.*, 1:93–113, 1991.

[50] B. T. Polyak. The conjugate gradient method in extremal problems. *USSR Comp. Math. Math. Phys.*, 9:94–112, 1969.

[51] S. M. Robinson. Strongly regular generalized equations. *Math. Oper. Res.*, 5:43–62, 1980.

[52] M. Ulbrich, S. Ulbrich, and M. Heinkenschloss. Global convergence of affine-scaling interior-point Newton methods for infinite-dimensional nonlinear problems with pointwise bounds. *SIAM J. Control Optim.*, 37:731–764, 1999.

[53] S. J. Wright. Implementing proximal point methods for linear programming. *J. Optim. Theory Appl.*, 65:531–554, 1990.

[54] H. Yamashita and H. Yabe. Superlinear and quadratic convergence of some primal-dual interior-point methods for constrained optimization. *Math. Prog.*, 75:377–397, 1996.

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

[56] Y. Zhang. Interior-point gradient methods with diagonal-scalings for simple-bound constrained optimization. Technical Report TR04-06, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 2004.

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