

An Active Set Modified Polak–Ribière–Polyak Method for Large-Scale Nonlinear Bound Constrained Optimization

Wanyou Cheng · Donghui Li

Received: 31 October 2011 / Accepted: 26 May 2012 / Published online: 9 June 2012
© Springer Science+Business Media, LLC 2012

Abstract In this paper, we propose an active set modified Polak–Ribière–Polyak method for solving large-scale optimization with simple bounds on the variables. The active set is guessed by an identification technique at each iteration and the recently developed modified Polak–Ribière–Polyak method is used to update the variables with indices outside of the active set. Under appropriate conditions, we show that the proposed method is globally convergent. Numerical experiments are presented using box constrained problems in the CUTER test problem libraries.

Keywords Bound constrained optimization · Conjugate gradient method · Global convergence

1 Introduction

The bound constrained problem is very important in practical optimization. Many practical problems can be converted into the form of the problem. In addition, it often a subproblem of augmented Lagrangian or penalty schemes for general constrained optimization [1]. In recent works on complementarity and variational inequalities, these problems are reduced to bound constrained minimization problems in an efficient way [2]. Thus, the development of numerical algorithms to efficiently solve it, especially when the dimension is large, is important in both theory and applications.

Communicated by F. Giannessi.

W.Y. Cheng (✉)
College of Computer, Dongguan University of Technology, Dongguan 523808, China
e-mail: chengwanyou421@yahoo.com.cn

D.H. Li
School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China
e-mail: dhli@scnu.edu.cn

The bound constrained optimization problem has received much attention in recent decades. We refer to papers [1, 3] for a review on recent advances in this area.

Active set methods are widely used in solving the bound constrained optimization problem. Early active set methods [4] are quite efficient for small dimensional problems, but are unattractive for large-scale problems [5]. The main reason is that typically at each step of the algorithm, at most one constraint can be added to or dropped from the active set. The potential worst-case may appear, where each of the possible 3^n active sets is visited before discovering the optimal one [1]. Recently, there has been a growing interest in the design of active set methods that are capable of making rapid changes incorrect predictions [1, 6–11].

In [7], Facchinei, Júdice, and Soares proposed an efficient identification technique of the active set at the solution and developed an active set Newton's algorithm for large-scale nonlinear programming with box constraints. The reported numerical results in [7] showed that this identification technique works well. Recently, based on the identification technique, Xiao and Hu [12] proposed an active set subspace Barzilai–Borwein gradient method (SBB). Preliminary numerical results show that the identification technique works well and the SBB method is competitive with the well-known SPG2 method [13].

Conjugate gradient methods are very efficient for solving large-scale unconstrained optimization problems due to their simplicity and low storage [14]. However, the study of conjugate gradient method for nonquadratic bound constrained optimization is relatively rare. In this paper, by the use of the active identification technique in [7] and the recently developed modified Polak–Ribière–Polyak (PRP) method [15], we propose an active set modified PRP method for bound constrained optimization problem. Specifically, at each iteration, the active variables and free variables are defined by the identification technique [7]; we use the method in [7] to update the active variables and use the modified PRP method to update the free variables. The use of the modified PRP method makes the method require lower storage. Hence, the method can be used to solve large-scale problems. In addition, the method possesses the following advantages: all iterates are feasible and rapid changes in the active set are allowed. Under appropriate conditions, we show that the proposed method is globally convergent. The numerical experiments are also given using the bound optimization problems from CUTer [16] collection with more than 500 variables, which shows that the proposed method is competitive with some existing methods such as the SPG2 method [13] and the ASA method [3].

The remainder of the paper is organized as follows. We propose the algorithm in Sect. 2. In Sect. 3, we show that the proposed algorithm is globally convergent. In Sect. 4, we test the performance of the proposed algorithm and compare it with the SPG2 method and the ASA method.

2 Motivation and Properties

Throughout the paper, $\|\cdot\|$ denotes the Euclidean norm of vectors. Let $P_\Omega(x)$ denote the projection of x on the set Ω . If $w = (w_1, w_2, \dots, w_n)^T$ is a n dimension vector and I is an index set such that $I \subset \{1, 2, \dots, n\}$, we denote by w_I the subvector

with components $w_i, i \in I$, and denote by $w_I \geq 0$ the subvector with components $w_i \geq 0, i \in I$.

Consider the solution of the bound constrained nonlinear programming problem

$$\{\min f(x), x \in \Omega := \{x \in \mathcal{R}^n : l \leq x \leq u\}\}, \tag{1}$$

where f is a real-valued, continuously differentiable function in an open set containing Ω , l and u are constant vectors, and the inequalities are valid component wise. Let \bar{x} be a stationary of (1), and consider the associated active sets

$$\bar{L} := \{i : \bar{x}_i = l_i\}, \quad \bar{U} := \{i : \bar{x}_i = u_i\}.$$

Furthermore, let

$$\bar{F} := \{1, \dots, n\} \setminus (\bar{L} \cup \bar{U})$$

be the set of the free variables. By using this notation, a vector \bar{x} is said to be a stationary point for (1) iff it satisfies:

$$\begin{cases} i \in \bar{L} \Rightarrow g_i(\bar{x}) \geq 0 \\ i \in \bar{F} \Rightarrow g_i(\bar{x}) = 0 \\ i \in \bar{U} \Rightarrow g_i(\bar{x}) \leq 0 \end{cases} \tag{2}$$

where $g_i(x)$ is the i th component of the gradient vector of f at x . Facchinei, Júdice, and Soares [7] gave the following approximations $L(x)$, $F(x)$ and $U(x)$ to \bar{L} , \bar{F} , \bar{U} , respectively,

$$\begin{aligned} L(x) &:= \{i : x_i \leq l_i + a_i(x)g_i(x)\}, \\ U(x) &:= \{i : x_i \geq u_i + b_i(x)g_i(x)\}, \\ F(x) &:= \{1, \dots, n\} \setminus (L(x) \cup U(x)), \end{aligned}$$

where $a_i(x)$ and $b_i(x), i = 1, \dots, n$ are nonnegative, continuous, and bounded functions defined on Ω , such that if $x_i = l_i$ or $x_i = u_i$ then $a_i(x) > 0$ or $b_i(x) > 0$, respectively. The following theorem shows that $L(x)$, $F(x)$, and $U(x)$ are indeed good estimates of \bar{L} , \bar{F} , and \bar{U} . For the proof, see Theorem 3.1 in [7].

Theorem 2.1 *For any $x \in \Omega, L(x) \cap U(x) = \mathcal{O}$. Furthermore, if \bar{x} is a stationary point of problem (1) at which the strict complementarity holds, then there exists a neighborhood $N(\bar{x})$ of \bar{x} such that*

$$L(x) = \bar{L}, \quad F(x) = \bar{F}, \quad U(x) = \bar{U}, \quad \forall x \in N(\bar{x}).$$

Following the idea of [7], we are going to develop an active set modified PRP method as follows. We firstly define the search direction. Let $x^k \in \Omega$ be the current point at iteration k . For simplicity, we let $L^k := L(x^k), U^k := U(x^k)$, and $F^k := F(x^k)$. Define the direction $d^k = (d_{L^k}^k, d_{F^k}^k, d_{U^k}^k)^T$ by

$$d_i^k := l_i - x_i^k, \quad i \in L^k \tag{3}$$

and

$$d_i^k := u_i - x_i^k, \quad i \in U^k. \tag{4}$$

In what follows, we are going to define $d_{F^k}^k$. For this, we define the active set indices of f at x^k as follows:

$$A(x^k) := \{i : x_i^k = l_i \text{ or } x_i^k = u_i\}.$$

The active indices are further subdivided into those indices:

$$A_1(x^k) := \{i : x_i^k = l_i, g_i(x^k) \geq 0\}, \quad A_2(x^k) := \{i : x_i^k = l_i, g_i(x^k) < 0\}$$

and

$$A_3(x^k) := \{i : x_i^k = u_i, g_i(x^k) \leq 0\}, \quad A_4(x^k) := \{i : x_i^k = u_i, g_i(x^k) > 0\}.$$

It is clearly that a direction d is a feasible direction of f at x^k if and only if $d_i \geq 0, i \in A_1(x^k) \cup A_2(x^k)$ and $d_i \leq 0, i \in A_3(x^k) \cup A_4(x^k)$. Furthermore, from the definitions of $L(x^k), U(x^k)$ and $A_1(x^k) - A_4(x^k)$ we can observe that the following fact holds:

$$A_1(x^k) \subseteq L(x^k), \quad A_3(x^k) \subseteq U(x^k) \quad \text{and} \\ A_2(x^k) \cap L(x^k) = \emptyset, \quad A_4(x^k) \cap U(x^k) = \emptyset.$$

By (3) and (4), we have $d_i^k \geq 0, i \in A_1(x^k)$ and $d_i^k \leq 0, i \in A_3(x^k)$. However, we can not determine the relation between $A_2(x^k)$ and $U(x^k)$ and the relation between $A_4(x^k)$ and $L(x^k)$. Let $A_2^*(x^k) := A_2(x^k) \cap U(x^k), A_4^*(x^k) := A_4(x^k) \cap L(x^k)$ and $F^*(x^k) := F(x^k) \setminus ((A_4(x^k) - A_4^*(x^k)) \cup (A_2(x^k) - A_2^*(x^k)))$. It follows from (3) and (4) that $d_i^k \geq 0, i \in A_2^*(x^k)$ and $d_i^k \leq 0, i \in A_4^*(x^k)$. Thus, we let

$$d_i^k := -g_i^k, \quad i \in (A_4(x^k) - A_4^*(x^k)) \cup (A_2(x^k) - A_2^*(x^k)) \tag{5}$$

and

$$d_{F^{*k}}^k := \begin{cases} -g_{F^{*k}}^k, & \text{iff } k = 0 \text{ or } \|g_{F^{*k}}^{k-1}\| \leq \epsilon, \\ -g_{F^{*k}}^k + \beta_k^{\text{PRP}} d_{F^{*k}}^{k-1} - \eta_k y_{F^{*k}}^{k-1}, & \text{iff } k > 0 \text{ and } \|g_{F^{*k}}^{k-1}\| > \epsilon, \end{cases} \tag{6}$$

where

$$\beta_k^{\text{PRP}} := \frac{(g_{F^{*k}}^k)^T y_{F^{*k}}^{k-1}}{\|g_{F^{*k}}^{k-1}\|^2}, \quad \eta_k := \frac{(g_{F^{*k}}^k)^T d_{F^{*k}}^{k-1}}{\|g_{F^{*k}}^{k-1}\|^2}, \quad y_{F^{*k}}^{k-1} := g_{F^{*k}}^k - g_{F^{*k}}^{k-1}$$

and $\epsilon > 0$ is given a constant. From the above discussion, we can see that d^k is a feasible direction of f at x^k . Moreover, it is not difficult to see from (5) and (6) that $d_{F^k}^k$ satisfies

$$(g_{F^k}^k)^T d_{F^k}^k = -\|g_{F^k}^k\|^2. \tag{7}$$

The following two theorems show that d^k is a descent direction f at x_k , if x_k is not a stationary point of (1).

Theorem 2.2 *If $\{x^k\} \in \Omega$, then $d^k = 0$ if and only if x^k is a stationary point of (1).*

Proof Let $d^k = 0$. For $i \in L^k$, by (3) and (4) we have

$$0 = d_i^k = l_i - x_i^k \geq -a_i(x^k)g_i(x^k).$$

Since $x_i^k = l_i, a_i(x^k) > 0$, it holds that $g_i(x^k) \geq 0$. For each $i \in U^k$ we have

$$0 = d_i^k = u_i - x_i^k \leq -b_i(x^k)g_i(x^k),$$

This implies that $g_i(x^k) \leq 0$. From the above discussion, we have $A_2(x^k) \cup A_4(x^k) = \emptyset$. Thus, $F^{*k} = F^k$. For $i \in F^k$, by the use of Cauchy–Schwarz inequality to (7), we have

$$\|g_{F^k}^k\| \leq \|d_{F^k}^k\|.$$

Hence, $g_i(x^k) = 0$ for each $i \in F^k$. The above argument has shown that x^k is a stationary point of f on Ω .

Suppose that x^k be a stationary point of f on Ω . Then we have $A_2(x^k) \cup A_4(x^k) = \emptyset$ and $F^{*k} = F^k$. It follows from (2) that

$$L^k = \{i : x_i^k = l_i\}, \quad F^k = \{i : l_i < x_i^k < u_i\}, \quad U^k = \{i : x_i^k = u_i\}.$$

By the definition of d^k , we immediately have $d^k = 0$. The proof is complete. □

Theorem 2.3 *Suppose that x_k be not a stationary point of (1). Then the direction d^k determined by (3)–(6) is a descent direction of f at x^k .*

Proof Following a similar proof as the proof of Theorem 3.2 in [7], it is not difficult to show that for each $i \in L^k \cup U^k$, there exists a positive scalar γ_i such that

$$g_i^k d_i^k \leq -\gamma_i (d_i^k)^2, \tag{8}$$

This together with (7) implies that

$$(g^k)^T d^k \leq -\|g_{F^k}^k\|^2 + \sum_{i \in L^k \cup U^k} -\gamma_i (d_i^k)^2. \tag{9}$$

Since x^k is not a stationary point of (1), by Theorem (2.2), there must exist an index i such that at least one of the two possibilities holds:

- (1) $d_i^k \neq 0, \quad i \in L^k \cup U^k$ or
- (2) $g_i^k \neq 0, \quad i \in F^k$.

Therefore, the right-hand side of (9) must be negative. In other words, d^k is a descent direction of f at x^k . The proof is complete. □

Based on the above discussion, we propose an active set PRP-type method (APRP) for (1) as follows.

Algorithm 2.1 (APRP)

- Step 0. Given initial point $x^0 \in \Omega$ and positive constants ρ, β, ϵ_1 and $\delta \in]0, 1[$. Set $k := 0$.
- Step 1. If $\|P_\Omega(x^k - g^k) - x^k\| \leq \epsilon_1$, then stop.
- Step 2. Compute d^k by (3)–(6).

Step 3. Determine $\alpha^k := \max\{\beta\rho^j, j = 0, 1, \dots\}$ satisfying

$$f(P_\Omega(x^k + \alpha^k d^k)) \leq f(x^k) - \delta(\alpha^k \|d^k\|)^2. \tag{10}$$

Step 4. Let the next iterate be $x^{k+1} := x^k + \alpha^k d^k$.

Step 5. Let $k := k + 1$ and go to Step 1.

Remark 2.1 Since the direction d^k is a feasible descent direction of f at x^k , there exists a positive constant β^k such that $x^k + \alpha d^k \in \Omega$ when $0 < \alpha < \beta^k$. This implies

$$f(P_\Omega(x^k + \alpha^k d^k)) = f(x^k + \alpha^k d^k) \quad \text{for } 0 < \alpha^k < \beta^k.$$

Clearly, Theorem 2.3 implies that the condition (10) necessarily holds after a finite number of reduction of β . Consequently, Algorithm 2.1 is well defined. In addition, it follows from (10) that the function value sequence $\{f(x^k)\}$ is decreasing. We also have from (10) that

$$\sum_{k=0}^{\infty} (\alpha^k)^2 \|d^k\|^2 < \infty,$$

if f is bounded from below. In particular, we have

$$\lim_{k \rightarrow \infty} \alpha^k \|d^k\| = 0. \tag{11}$$

3 Convergence Analysis

In this section, we analyze the convergence of Algorithm 2.1. To this end, we make the following assumptions.

Assumption 3.1

- (i) The level set $D := \{x \in \Omega : f(x) \leq f(x_0)\}$ is bounded.
- (ii) In some neighborhood N of D , f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in N. \tag{12}$$

Since $\{f(x^k)\}$ is decreasing, it is clearly that the sequence $\{x^k\}$ generated by Algorithm 2.1 is contained in D . In addition, from Assumption 3.1, we get that there is a constant $\gamma > 0$ such that

$$\|g(x)\| \leq \gamma, \quad \forall x \in D. \tag{13}$$

The following lemma shows that the sequence $\{d^k\}$ generated by Algorithm 2.1 is bounded, also.

Lemma 3.1 *Let the sequence $\{x^k\}$ be generated by Algorithm 2.1. Then there exists a constant $M > 0$ such that*

$$\|d^k\| \leq M, \quad \forall k. \tag{14}$$

Proof First, by (8), there exists a constant $\bar{\gamma} > \gamma$ such that

$$\|d_{L^k \cup U^k}^k\| \leq \bar{\gamma}.$$

This together with the definition of d^k , (12) and (13) implies that

$$\begin{aligned} \|d^k\| &\leq \|d_{L^k \cup U^k}^k\| + \|d_{F^k}^k\| \\ &\leq \bar{\gamma} + \|g_{F^k}^k\| + 2 \frac{\|g_{F^k}^k\| \|d_{F^k}^{k-1}\| \|y_{F^k}^{k-1}\|}{\|g_{F^k}^{k-1}\|^2} \\ &\leq 2\bar{\gamma} + 2 \frac{\|g^k\| \|d^{k-1}\| \|y^{k-1}\|}{\varepsilon^2} \\ &\leq 2\bar{\gamma} + 2 \frac{\bar{\gamma} L \alpha^{k-1} \|d^{k-1}\|}{\varepsilon^2} \|d^{k-1}\|. \end{aligned}$$

By (11), there exists a constant $r \in]0, 1[$ and an integer k_0 such that the following inequality holds for all $k \geq k_0$,

$$\frac{2L\bar{\gamma}}{\varepsilon^2} \alpha^{k-1} \|d^{k-1}\| \leq r.$$

Hence, we have for all $k \geq k_0$

$$\begin{aligned} \|d^k\| &\leq 2\bar{\gamma} + r \|d^{k-1}\| \\ &\leq 2\bar{\gamma} (1 + r + r^2 + \dots + r^{k-k_0+1}) + r^{k-k_0} \|d^{k_0}\| \\ &\leq 2 \frac{\bar{\gamma}}{1-r} + \|d^{k_0}\|. \end{aligned}$$

Letting $M := \max\{\|d_1\|, \|d_2\|, \dots, \|d^{k_0}\|, 2 \frac{\bar{\gamma}}{1-r} + \|d^{k_0}\|\}$, we get (14). □

Lemma 3.2 *Let the sequence $\{x^k\}$ and $\{d^k\}$ be generated by Algorithm 2.1. If there are subsequences $\{x_k\}_{k \in K} \rightarrow \hat{x}$ and $\{d_k\}_{k \in K} \rightarrow 0$, then \hat{x} is a stationary point of (1).*

Proof Since the number of distinct sets L^k , U^k and F^k is finite, without loss of generality, we can assume that the following relations hold for all $k \in K$

$$L^k = L, \quad F^k = F, \quad U^k = U.$$

Since $d_L^k \rightarrow 0$ and $d_U^k \rightarrow 0$, we obviously have

$$\hat{x}_L = l_L, \quad g_L(\hat{x}) \geq 0, \quad \text{and} \quad \hat{x}_U = u_U, \quad g_U(\hat{x}) \leq 0.$$

On the other hand, by (7), we have

$$\|g_F^k\| \leq \|d_F^k\|.$$

Consequently, we obtain $\|g_F^k(\hat{x})\| = 0$. The proof is complete. □

The following theorem shows that there exists an accumulation point of $\{x^k\}$ that is a stationary point of (1).

Theorem 3.1 *Let $\{x^k\}$ be generated by Algorithm 2.1. Then we have*

$$\liminf_{k \rightarrow \infty} \|d^k\| = 0. \tag{15}$$

Proof For the sake of contradiction, we suppose that (15) is not true. Then there exists a constant $\epsilon > 0$ such that

$$\|d^k\| \geq \epsilon, \quad \forall k.$$

This together with (11) implies that $\lim_{k \rightarrow \infty} \alpha^k = 0$. By the line search rule, $\rho^{-1} \alpha^k \leq \beta^k$ does not satisfy (10) when k is sufficiently large. This means

$$f(x^k + \rho^{-1} \alpha^k d^k) - f(x^k) > -\delta \rho^{-2} (\alpha^k)^2 \|d^k\|^2. \tag{16}$$

By the mean-value theorem and (12), there exists a constant $0 < \theta^k < 1$ such that

$$\begin{aligned} & f(x^k + \rho^{-1} \alpha^k d^k) - f(x^k) \\ &= \rho^{-1} \alpha^k g(x^k + \theta^k \rho^{-1} \alpha^k d^k)^T d^k \\ &= \rho^{-1} \alpha^k (g^k)^T d^k + \rho^{-1} \alpha^k (g(x^k + \theta^k \rho^{-1} \alpha^k d^k) - g^k)^T d^k \\ &\leq \rho^{-1} \alpha^k (g^k)^T d^k + L \rho^{-2} (\alpha^k)^2 \|d^k\|^2. \end{aligned}$$

where L is the Lipschitz constant of g . Substituting the last inequality into (16), we get for all k sufficiently large

$$0 \leq -(g^k)^T d^k \leq (L + \delta) \rho^{-1} \alpha^k \|d^k\|^2.$$

Since $\{d^k\}$ is bounded and $\lim_{k \rightarrow \infty} \alpha^k \|d^k\| = 0$, the last inequality implies

$$\lim_{k \rightarrow \infty} -(g^k)^T d^k = 0.$$

This together with (9) implies that

$$\lim_{k \rightarrow \infty} \|g_{F^k}^k\|^2 = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \sum_{i \in L^k \cup U^k} \gamma_i (d_i^k)^2 = 0. \tag{17}$$

Since $\{x^k\}$ is bounded, there exists an infinite index set K_1 such that $\lim_{k \in K_1} x^k = x^*$. Hence, (17) together with (3)–(4) implies that x^* is a stationary point of (1). By Theorem 2.2, we have $\lim_{k \in K_1} \|d^k\| = 0$. This yields a contradiction. \square

4 Numerical Experiments

In this section, we do some numerical experiments to test the performance of the APRP method and compare it with the SPG2 method [13, 17] and the ASA method [3]. The SPG2 code was obtained from Birgin’s web page at: <http://www.ime.usp.br/~egbirgin/tango/>. The ASA code was obtained from Hager’s web page at: <http://www.math.ufl.edu/~hager/>. The SPG2 code and the APRP code were written in Fortran 77 and the ASA code was written in C. All codes were run on a PC with Pentium IV 3.0 GHz CPU processor and 512 M RAM memory and with a Linux operating system.

The ASA code and the SPG2 code were implemented with default parameters. We implemented the APRP code with the following parameters: $\delta = 10^{-4}$, $\rho = 0.5$, $\epsilon = 10^{-6}$ and

$$\beta := \begin{cases} \min\{1.0, 1.01 \frac{g_{k-1}^T d_{k-1}}{g_k^T d_k}\}, & \text{iff } k > 0 \\ \frac{1}{\|g(x_0)\|_\infty}, & \text{iff } k = 0. \end{cases}$$

Moreover, we also test the APRP method with different parameters $a_i(x)$ and $b_i(x)$ to see that $a_i(x) = b_i(x) = 10^{-6}$ is the best choice.

The test problems consists of 52 box constrained problems in the CUTer library [16] with dimensions between 500 and 15625. For each test problem and all tested methods, the termination condition is

$$\|P_\Omega(x^k - g(x^k)) - x^k\|_\infty \leq 10^{-6}.$$

A limit of 5000 iterations is also imposed. 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. The detailed numerical results concerned with the CPU time in seconds and the number of iterations, function evaluations, and gradient evaluations for each of the tested method are posted at the following Web site: <http://bgchengwy.blog.163.com/>.

First, we note that the SPG2 method fails to satisfy the termination in the problems LINVERSE (999), QRTQUAD (1200–5000), and JNLBRNGB (9375–15625), while the APRP method and the ASA method fails to satisfy the termination in the same problems QRTQUAD (1200–5000). More insight into the numerical results, we consider the statistics “iter,” “fun,” “grad,” “time” and list the number of problems where any method achieves the best value on the number of iteration, function evaluations, gradient evaluations and CPU time in Table 1.

From the above table, we observe that the APRP method works efficiently, which require fewer iterations, fewer function evaluations, fewer gradient evaluations, and less time consuming. Further to show the efficiency of the APRP method, we adopt the performance profiles introduced by Dolan and Moré [18] to evaluate CPU time in Fig. 1. It is not difficult to see from Fig. 1 that the APRP method performs best among all the methods on CPU time. In particular, both the APRP method and the ASA method can solve about 96 % of the test problems, whereas the SPG2 method can solve about 90 % of the test problems. In summary, the results from Table 1 and Fig. 1 show that the APRP method provides an efficient method for solving large-scale nonlinear box constrained problems.

Table 1 Statical data for 52 problems

Method	Iter	Fun	Grad	Time
SPG2	0	7	0	3
ASA	15	11	10	10
APRP	38	32	40	42

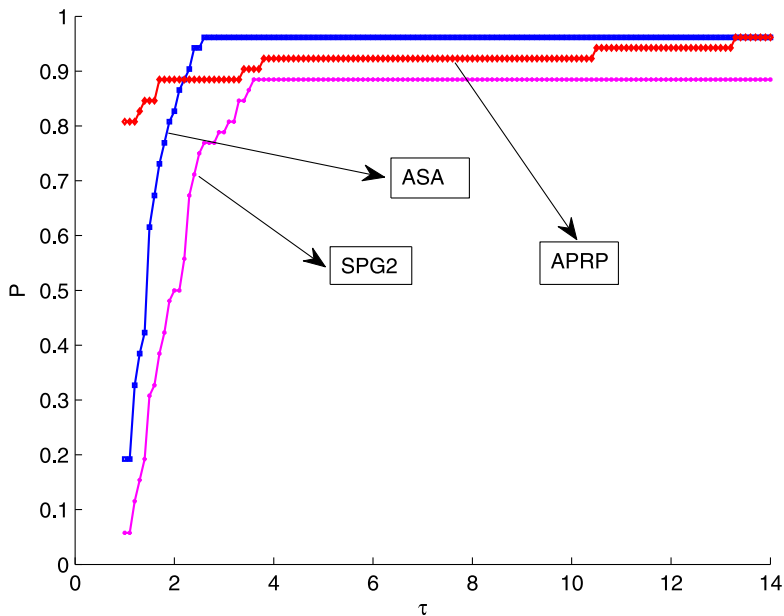


Fig. 1 Performance profiles based on number of CPU time

5 Conclusions

In this paper, we have introduced an active set modified PRP method for large-scale optimization with simple bounds on the variables. Under appropriate conditions, we show that the proposed method is globally convergent. The implementation of the algorithm uses an active set identification technique to estimate the active variables at each iteration, and determine the search direction of free variables by the modified PRP conjugate gradient method. The active set identification strategy allows rapid changes in the active set from one iteration to the next, and the cost of computing the search direction of active variables is reduced. We think that our algorithm is practical and effective for the test problems.

Acknowledgements We would like to thank the referees for their detailed comments and suggestions that have led to a significant improvement of the paper. We also thank the Professors E.G. Birgin, J.M. Martínez, and M. Raydan for their SPG2 code and Professors W.W.Hager and H. Zhang for their ASA code for numerical comparison. W. Cheng's research was supported by the NSF of China via grant (11071087, 11101081) and by Foundation for Distinguished Young Talents in Higher Education of Guangdong, China LYM10127. D. Li's research is supported by the major project of the Ministry of Education of China Grant 309023 and the NSF of China Grant 11071087.

References

1. Gould, N.I.M., Orban, D., Toint, P.L.: Numerical methods for large-scale nonlinear optimization. *Acta Numer.* **14**, 299–361 (2005)
2. Burdakov, O.P., Martínez, J.M., Pilotta, E.A.: A limited-memory multipoint symmetric secant method for bound constrained optimization. *Ann. Oper. Res.* **117**, 51–70 (2002)

3. Hager, W.W., Zhang, H.: A new active set algorithm for box constrained optimization. *SIAM J. Optim.* **17**, 526–557 (2006)
4. Goldfarb, D.: Extension of Davidson’s variable metric method to maximization under linear inequality and constraints. *SIAM J. Appl. Math.* **17**, 739–764 (1969)
5. Dai, Y.H., Fletcher, R.: Projected Barzilai–Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.* **100**, 21–47 (2005)
6. Birgin, E.G., Martínez, J.M.: Large-scale active-set box-constrained optimization method with spectral projected gradients. *Comput. Optim. Appl.* **23**, 101–125 (2002)
7. Facchinei, F., Júdice, J., Soares, J.: An active set newton algorithm for large-scale nonlinear programs with box constraints. *SIAM J. Optim.* **8**, 158–186 (1998)
8. Facchinei, F., Lucidi, S., Palagi, L.: A truncated newton algorithm for large scale box constrained optimization. *SIAM J. Optim.* **12**, 1100–1125 (2002)
9. Lin, C.J., Moré, J.J.: Newton’s method for large bound-constrained optimization problems. *SIAM J. Optim.* **9**, 1100–1127 (1999)
10. Moré, J.J., Toraldo, G.: Algorithms for bound constrained quadratic programming problems. *Numer. Math.* **4**, 377–400 (1989)
11. Ni, Q., Yuan, Y.X.: A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization. *Math. Comput.* **66**, 1509–1520 (1997)
12. Xiao, Y.H., Hu, Q.G.: Subspace Barzilai–Borwein gradient method for large-scale bound constrained optimization. *Appl. Math. Optim.* **58**, 275–290 (2008)
13. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.* **10**, 1121–1196 (2000)
14. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2**, 35–58 (2006)
15. Zhang, L., Zhou, W.J., Li, D.H.: A descent modified Polak–Ribiere–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **26**, 629–640 (2006)
16. Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, P.L.: CUTE: constrained and unconstrained testing environment. *ACM Trans. Math. Softw.* **21**, 123–160 (1995)
17. Birgin, E.G., Martínez, J.M., Raydan, M.: Algorithm 813: SPG-software for convex-constrained optimization. *ACM Trans. Math. Softw.* **27**, 340–349 (2001)
18. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**, 201–213 (2002)