Research Article

Extension of Modified Polak-Ribière-Polyak Conjugate Gradient Method to Linear Equality Constraints Minimization Problems

Zhifeng Dai¹,²

¹ College of Business, Central South University, Hunan 410083, China
² College of Mathematics and Computational Science, Changsha University of Science and Technology, Hunan 410114, China

Correspondence should be addressed to Zhifeng Dai; zhifengdai823@163.com

Received 19 March 2014; Revised 1 July 2014; Accepted 21 July 2014; Published 6 August 2014

Abstract and Applied Analysis

Copyright © 2014 Zhifeng Dai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Combining the Rosen gradient projection method with the two-term Polak-Ribière-Polyak (PRP) conjugate gradient method, we propose a two-term Polak-Ribière-Polyak (PRP) conjugate gradient projection method for solving linear equality constraints optimization problems. The proposed method possesses some attractive properties: (1) search direction generated by the proposed method is a feasible descent direction; consequently the generated iterates are feasible points; (2) the sequences of function values are decreasing. Under some mild conditions, we show that it is globally convergent with Armijio-type line search. Preliminary numerical results show that the proposed method is promising.

1. Introduction

In this paper, we consider solving the following linear equality constraints optimization problem:

\[ \min \ f(x) \]

s.t. \ Ax = b, \quad (1)

where \( f(x) : \mathbb{R}^n \to \mathbb{R} \) is a smooth function and \( A \) is a \( m \times n \) matrix of rank \( m (m \leq n) \). In this paper, the feasible region \( D \) and the feasible direction set \( S \) are defined, respectively, as follows:

\[ D = \{ Ax = b \}, \quad S = \{ Ad = 0 \}. \quad (2) \]

Taking the negative gradient for a search direction \( d = -\nabla f(x_k) \) is a natural way of solving unconstrained optimization problems. However, this approach does not work for constrained problems, since the gradient may not be a feasible direction. A basic technique to overcome this difficulty was initiated by Rosen [1] in 1960. To obtain a feasible search direction, Rosen projected the gradient into the feasible region; that is,

\[ d = -P \nabla f(x_k), \quad P = I - A^T(AA^T)^{-1} A. \quad (3) \]

The convergence of Rosen's gradient projection method can be proved by Du; see [2–4].

In fact, Rosen's gradient projection method is an extension of the steepest-descent method. It is well known that the drawback of the steepest-descent method is easy to suffer from zig-zagging specially when the graph of \( f(x) \) has an "elongated" form. To overcome the zig-zagging, we want to use the conjugate gradient method to modify the projection direction.

It is well known that nonlinear conjugate gradient methods such as the Polak-Ribière-Polyak (PRP) method [5, 6] are very efficient for large-scale unconstrained optimization problems due to their simplicity and low storage. However, it does not necessarily satisfy the descent conditions

\[ g_k^T d_k \leq -c \| g_k \|^2, \quad c > 0. \]

Recently, Cheng [7] proposed a two-term modified PRP method (called TMPRP), in which the direction \( d_k \) is given by

\[ d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{PRP} \left( I - \frac{g_k g_k^T}{\| g_k \|^2} \right) d_{k-1}, & \text{if } k \geq 1. \end{cases} \quad (4) \]
An attractive property of the TMPRP method is that the direction generated by the method satisfies
\[ g_k^T d_k = -\|g_k\|^2, \] (5)
which is independent of any line search. The presented numerical results show some potential advantage of the TMPRP method in Cheng [7]. In fact, we can easily rewrite the above direction \( d_k \) (4) as a three-term form:
\[ d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{PRP} d_{k-1} - \beta_k^{PRP} \frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k, & \text{if } k \geq 1. \end{cases} \] (6)

In the past few years, researchers have paid increasing attention to the conjugate gradient methods and their applications. Among others, we mention here the following works, for example, [8–29].

In the past few years, some researchers also paid attention to equality constrained problems. Martinez et al. [30] proposed a spectral gradient method for linearly constrained optimization by the following way to obtain the search direction: \( d_k \in \mathbb{R}^n \) is the unique solution of
\[
\min \frac{1}{2} d^T B_k d + \nabla f(x)^T d \\
\text{s.t.} \quad Ad = 0. \tag{7}
\]

In this algorithm, \( B_k \) can be computed by quasi-Newton method in which the approximate Hessians satisfy a weak secant equation. The spectral choice of steplength is embedded into the Hessian approximation and the whole process is combined with a nonmonotone line search strategy.

C. Li and D. H. Li [31] proposed a feasible Fletcher-Reeves conjugate gradient method for solving linear equality constrained optimization problem with exact line search. Their idea is to use original Fletcher-Reeves conjugate gradient method to modify the Zoutendijk direction. The Zoutendijk direction is the feasible steepest descent direction. It is a solution of the following problem:
\[
\min \nabla f(x)^T d \\
\text{s.t.} \quad Ad = 0, \|d\| \leq 1. \tag{8}
\]

Li et al. [32] also extended the modified Fletcher-Reeves conjugate gradient method in Zhang et al. [33] to solve linear equality constraints optimization problems which combined with the Zoutendijk feasible direction method. Under some mild conditions, Li et al. [32] showed that the proposed method with Armijo-type line search is globally convergent.

In this paper, we will extend the two-term Polak-Ribière-Polyak (PRP) conjugate gradient method in Cheng [7] to solve linear equality constraints optimization problems (I), which combines with the Rosen gradient projection method in Rosen [1]. Under some mild conditions, we show that it is globally convergent with Armijo-type line search.

The rest of this paper is organized as follows. In Section 2, we firstly propose the algorithm and prove the feasible descent direction. In Section 3, we prove the global convergence of the proposed method. In Section 4, we give some improvement for the algorithm. In Section 5, we report some numerical results to test the proposed method.

2. Algorithm and the Feasible Descent Direction

In this section, we propose a two-term Polak-Ribière-Polyak conjugate gradient projection method for solving the linear equality constraints optimization problem (I). The proposed method is a combination of the well-known Rose gradient projection method and the two-term Polak-Ribière-Polyak (PRP) conjugate gradient method in Cheng [7].

The iterative process of the proposed method is given by
\[ x_{k+1} = x_k + \alpha_k d_k, \tag{9} \]
and the search direction \( d_k \) is defined by
\[ d_k = \begin{cases} -Pg_k, & \text{if } k = 0, \\ -Pg_k + \beta_k^{PRP} d_{k-1} - \beta_k^{PRP} \frac{g_k^T d_{k-1}}{\|Pg_k\|^2} Pg_k, & \text{if } k \geq 1, \end{cases} \] (10)

where
\[ P = I - A^T (AA^T)^{-1} A, \quad g_k = \nabla f(x_k), \]
\[ y_{k-1} = g_k - g_{k-1}, \quad \beta_k^{PRP} = \frac{(Pg_k)^T y_{k-1}}{\|Pg_{k-1}\|^2}, \tag{11} \]
and \( \alpha_k > 0 \) is a steplength obtained by a line search.

For convenience, we call the method (9) and (10) as EMPRP method. Now we prove that the direction \( d_k \) defined by (10) and (11) is a feasible descent direction of \( f \) at \( x_k \).

**Theorem 1.** Suppose that \( x_k \in D, P = I - A^T (AA^T)^{-1} A \) is defined by (10). If \( d_k \neq 0 \), then \( d_k \) is a feasible descent direction of \( f \) at \( x_k \).

**Proof.** From (9), (10), and the definition of \( P \), we have
\[
g_k^T d_k = g_k^T \left( -Pg_k + \beta_k^{PRP} d_{k-1} - \beta_k^{PRP} \frac{g_k^T d_{k-1}}{\|Pg_k\|^2} Pg_k \right) \\
= -\|Pg_k\|^2 + \beta_k^{PRP} g_k^T d_{k-1} - \beta_k^{PRP} \frac{g_k^T d_{k-1}}{\|Pg_k\|^2} \|Pg_k\|^2 \\
= -\|Pg_k\|^2. \tag{12} \]

This implies that \( d_k \) provides a descent direction of \( f \) at \( x_k \).

In what follows, we show that \( d_k \) is a feasible descent direction of \( f \) at \( x_k \). From (9), we have that
\[ A(Pg_k) = A(\left( I - A^T (AA^T)^{-1} A \right) g_k) = 0. \tag{13} \]
Abstract and Applied Analysis

It follows from (10) and (13) that
\[ A d_k = A \left( -P g_k + \beta_k^\text{PRP} P_{d_{k-1}} - \beta_k^\text{PRP} g_k^T d_{k-1} - P g_k \right) \]
\[ = \beta_k^\text{PRP} A d_{k-1}. \]

When \( k = 0 \), we have
\[ A d_0 = A P g_0 = 0. \]

It is easy to get from (14) and (15) that, for all \( k \), \( A d_k = 0 \) is satisfied. That is, \( d_k \) is feasible direction.

In the remainder of this paper, we always assume that \( f(x) \) satisfies the following assumptions.

**Assumption A.**

(i) The level set
\[ \Omega = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \} \] is bounded.

(ii) Function \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable and bounded from below. Its gradient is Lipschitz continuous, on an open ball \( \mathcal{N} \) containing \( \Omega \); that is, there is a constant \( L > 0 \) such that
\[ \| g(x) - g(y) \| \leq L \| x - y \|, \quad \forall x, y \in \mathcal{N}. \]

Since \( f(x_k) \) is decreasing, it is clear that the sequence \( \{x_k\} \) generated by Algorithm 2 is contained in \( \Omega \). In addition, we get from Assumption A that there is a constant \( \gamma > 0 \), such that
\[ \| g(x) \| \leq \gamma, \quad \forall x \in \Omega. \]

Since the matrix \( P \) is a projection matrix, it is reasonable to assume that there is a constant \( C > 0 \), such that
\[ \| P g(x) \| \leq C, \quad \forall x \in \Omega. \]

We state the steps of the algorithm as follows.

**Algorithm 2** (EMPRP method with Armijo-type line search).

**Step 0.** Choose an initial point \( x_0 \in D, \varepsilon > 0 \). Let \( k := 0 \).

**Step 1.** Compute \( d_k \) by (10), where \( \beta_k \) is computed by (11).

**Step 2.** If \( \| P g_k \| \leq \varepsilon \) stop, else go to Step 3.

**Step 3.** Given \( \delta \in (0, 1/2), \rho \in (0, 1) \). Determine a stepsize \( \alpha_k = \max(\rho^j \mid j = 0, 1, 2, \ldots) \) satisfying
\[ f(x_k + \alpha_k d_k) \leq f(x_k) - \delta \alpha_k^2 \| d_k \|^2. \]

**Step 4.** Let \( x_{k+1} = x_k + \alpha_k d_k \), and \( k := k + 1 \). Go to Step 1.

### 3. Global Convergence

In what follows, we establish the global convergence theorem of the EMPRP method for general nonlinear objective functions. We firstly give some important lemmas of the EMPRP method.

**Lemma 3.** Suppose that \( x_k \in D, P = I - A^T (AA^T)^{-1} A \) is defined by (10) and (11). Then we have \( (P g_k)^T d_k = g_k^T d_k \), \( (P g_{k+1})^T d_k = g_{k+1}^T d_k \), \( (P g_k)^T d_k = g_k^T d_k = -\| P g_k \|^2 \).

**Proof.** By the definition of \( d_k \), we have
\[ (P g_k)^T d_k = g_k^T \left( I - A^T (AA^T)^{-1} A \right) d_k \]
\[ = g_k^T d_k - g_k^T \left( A^T (AA^T)^{-1} A \right) d_k = g_k^T d_k. \]

It follows from (12) and (21) that
\[ (P g_k)^T d_k = g_k^T d_k = -\| P g_k \|^2. \]

On the other hand, we also have
\[ (P g_{k+1})^T d_k = g_{k+1}^T d_k \]
\[ = g_{k+1}^T d_k - g_k^T \left( A^T (AA^T)^{-1} A \right) d_k = g_{k+1}^T d_k. \]

**Lemma 4.** Suppose that Assumption A holds. \( \{x_k\} \) is generated by Algorithm 2. If there exists a constant \( \varepsilon > 0 \) such that
\[ \| P g_k \| \geq \varepsilon, \quad \forall k; \]
then there exists a constant \( M > 0 \) such that
\[ \| d_k \| \leq M, \quad \forall k. \]

**Proof.** It follows from (20) that the function value sequence \( \{f(x_k)\} \) is decreasing. We also have from (20) that
\[ \sum_1^{\infty} \delta \alpha_k^2 \| d_k \|^2 < +\infty, \]
as \( f \) is bounded from below. In particular, we have
\[ \lim_{k \to \infty} \alpha_k \| d_k \| = 0. \]

By the definition of \( d_k \), we get from (17), (19), and (24) that
\[ \| d_k \| \leq \| P g_k \| + \| P g_k \| \gamma_k \| d_{k-1} \| \| d_{k-1} \| \]
\[ + \| P g_k \| \gamma_k \| d_{k-1} \| \| g_k \| \| d_{k-1} \| \| P g_k \| \]
\[ \leq C + \frac{CL \alpha_k \| d_{k-1} \| \| d_{k-1} \| \gamma \| d_{k-1} \| \| d_{k-1} \| \}
\[ \leq C + \left( \frac{CL}{\varepsilon^2} + \frac{\gamma L}{\varepsilon^2} \right) \alpha_k \| d_{k-1} \| \]
Hence, we have, for any \(k > k_0\),
\[
\|d_k\| \leq C + r \|d_{k-1}\| \leq C\left(1 + r \left(1 + r + r^2 + \cdots + r^{k-k_0-1}\right)\right)
\]  
\[+ r^{k-k_0} \|d_{k_0}\| \leq \frac{C}{1-r} + \|d_{k_0}\|.
\]  
(30)

Letting \(M = \max\{\|d_1\|, \|d_1\|, \ldots, \|d_{k_0}\|, C/(1-r) + \|d_{k_0}\|\}\), we can get (25).

We now establish the global convergence theorem of the EMPRP method for general nonlinear objective functions.

**Theorem 5.** Suppose that Assumption A holds. \(\{x_k\}\) is generated by Algorithm 2. Then we have
\[
\lim \inf_{k \to \infty} \|Pg_k\| = 0.
\]  
(31)

*Proof.* Suppose that \(\lim \inf_{k \to \infty} \|Pg_k\| \neq 0\) for all \(k\). Then there exists a constant \(\epsilon > 0\) such that
\[
\|Pg_k\| > \epsilon, \quad \forall k \geq 0.
\]  
(32)

We now prove (31) by considering the following two cases.

**Case (i).** \(\lim \inf_{k \to \infty} \alpha_k > 0\). We get from (22) and (27) that \(\lim \inf_{k \to \infty} \|Pg_k\| = 0\). This contradicts assumption (32).

**Case (ii).** \(\lim \inf_{k \to \infty} \alpha_k = 0\). That is, there is an infinite index set \(K\) such that
\[
\lim_{k \in K, k \to \infty} \alpha_k = 0.
\]  
(33)

When \(k \in K\) is sufficiently large, by the line search condition, \(\rho^{-1} \alpha_k\) does not satisfy inequality (20). This means
\[
f\left(x_k + \alpha_k d_k\right) - f\left(x_k\right) > -\delta \rho^{-2} \alpha_k^2 \|d_k\|^2.
\]  
(34)

By the mean-value theorem and inequality (17), there is a \(t_k \in (0, 1)\) such that \(x_k + t_k \rho^{-1} \alpha_k d_k \in N\) and
\[
f\left(x_k + \rho^{-1} \alpha_k d_k\right) - f\left(x_k\right)
\]  
\[= \rho^{-1} \alpha_k g\left(x_k + t_k \rho^{-1} \alpha_k d_k\right)^T d_k
\]  
\[= \rho^{-1} \alpha_k g_k^T d_k + \rho^{-1} \alpha_k \left(g\left(x_k + t_k \rho^{-1} \alpha_k d_k\right) - g_k\right)^T d_k
\]  
\[\leq \rho^{-1} \alpha_k g_k^T d_k + \rho^{-1} \alpha_k \left\|g\left(x_k + t_k \rho^{-1} \alpha_k d_k\right) - g_k\right\| \|d_k\|
\]  
\[\leq \rho^{-1} \alpha_k g_k^T d_k + L \rho^{-2} \alpha_k^2 \|d_k\|^2.
\]  
(35)

Substituting the last inequality into (34), for all \(k \in K\) sufficiently large, we have from (12) that
\[
\|Pg_k\|^2 = -g_k^T d_k \leq \rho \rho^{-1} (L + \delta) \alpha_k \|d_k\|^2.
\]  
(36)

Since \(\{d_k\}\) is bounded and \(\lim_{k \to \infty} \alpha_k = 0\), the last inequality implies
\[
\lim_{k \in K, k \to \infty} \|Pg_k\| = 0.
\]  
(37)

This also yields a contradiction. The proof is then complete.

### 4. Improvement for Algorithm 2

In this section, we propose techniques for improving the efficiency of Algorithm 2 in practical computation which is about computing projections and the stepsize of the Armijo-type line search.

**4.1. Computing Projection.** In this paper, as in Gould et al. [34], instead of computing a basis for null space of matrix \(A\), we choose to work directly with the matrix of constraint gradients, computing projections by normal equations. As the computation of the projection is a key step in the proposed method, following Gould et al. [34], this projection can be computed in an alternative way.

Let
\[
g^+ = -Pg_k, \quad P = I - AA^T\left(AA^T\right)^{-1} A.
\]  
(38)

We can express this as
\[
g^+ = -g + AA^T v,
\]  
(39)

where \(v\) is the solution of
\[
AA^T v = Ag.
\]  
(40)

Noting that (40) is the normal equation. Since \(A\) is a \(m \times n\) matrix of rank \(m(m \leq n)\), \(AA^T\) is symmetric positive definite matrix. We use the Doolittle (called LU) factorization of \(AA^T\) to solve (40).

For the matrix \(A\) is constant, the factorization of \(AA^T\) needs to be carried out only once at the beginning of the iterative process. Using a Doolittle factorization of \(AA^T\), (40) can be computed in the following form:
\[
L(U^Tv) = Ag \iff \left\{\begin{array}{l}
Ly = Ag \\
U^Tv = y,
\end{array}\right.
\]  
(41)

where \(L, U\) is the Doolittle factor of \(AA^T\).

**4.2. Computing Stepsize.** The drawback of the Armijo line search is how to choose the initial stepsize \(\alpha_k\). If \(\alpha_k\) is too large then the procedure needs to call much more function evaluations. If \(\alpha_k\) is too small then the efficiency of related algorithm will be decreased. Therefore, we should choose an adequate initial stepsize \(\alpha_k\) at each iteration. In what follows, we propose a way to generate the initial stepsize.

We first estimate the stepsize determined by the exact line search. Support at the moment that \(f\) is twice continuously differentiable. We denote by \(G(x)\) the Hessian of \(f\) at \(x\) and abbreviate \(G(x_k)\) as \(G_k\). Notice that the exact line search stepsize \(\delta_k\) satisfies
\[
\delta_k = \left(g\left(x_k + \alpha_k d_k\right) - g_k\right)^T d_k = \alpha_k d_k^T G_k d_k.
\]  
(42)

This shows that scalar \(\delta_k \equiv -g_k^T d_k / d_k^T G_k d_k\) is an estimation to \(\delta_k\). To avoid the computation of the second derivative, we further estimate \(\delta_k\) by letting
\[
\eta_k = \frac{-\epsilon_k g_k^T d_k}{d_k^T (g\left(x_k + \epsilon_k d_k\right) - g_k)},
\]  
(43)

\(\epsilon_k\) is chosen to control the number of function evaluations.
where positive sequence satisfies \( \{ \epsilon_k \} \to 0 \) as \( k \to \infty \). Let the initial stepsize of the Armijo line search be an approximation to
\[
\delta_k \equiv \frac{-g_k^T d_k}{\| d_k \|} = \frac{-\epsilon_k g_k^T d_k}{\| g_k \|_2 (g(x_k + \epsilon_k d_k) - g_k)} \equiv y_k. \tag{44}
\]
It is not difficult to see that if \( \epsilon_k \) and \( \| d_k \| \) are sufficiently small, then \( \delta_k \) and \( y_k \) are good estimation to \( \alpha \).

So to improve the efficiency of EMPRP method in practical computation, we utilize the following line search process,

\[
\text{Line Search Process. If inequality,}
\]
\[
f(x_k + |y_k| d_k) \leq f(x_k) + |\delta| y_k^T \| d_k \|^2, \tag{45}
\]
holds, then we let \( \alpha_k = |y_k| \). Otherwise we let \( \alpha_k \) be the largest scalar in the set \( \{|y_k| \rho^i, i = 0, 1, \ldots \} \) such that inequality (45) is satisfied.

5. Numerical Experiments

This section reports some numerical experiments. Firstly, we test the EMPRP method and compare it with the Rose gradient projection method in [1] on low dimensional problems. Secondly, we test the EMPRP method and compare it with the spectral gradient method in Martínez et al. [30] and the feasible Fletcher-Reeves conjugate gradient method in Li et al. [32] on large dimensional problems. In the line search process, we set \( \epsilon_k = 10^{-5} \), \( \rho = 0.3 \), \( \delta = 0.02 \).

The methods in the tables have the following meanings:

(i) "EMPRP" stands for the EMPRP method with the Armijo-type line search (20).
(ii) "ROSE" stands for Rose gradient projection method in [1] with the Armijo-type line search (20). That is, in Algorithm 2, the direction \( d_k = -\nabla g_k \).
(iii) "SPG" stands for the spectral gradient method with the nonmonotone line search in Martínez et al. [30], where \( M = 10, P = 2 \).
(iv) "FFR" stands for the feasible modified Fletcher-Reeves conjugate gradient method with the Armijo-type line search in Li et al. [32].

We stop the iteration if the condition \( \| P g_k \| \leq \epsilon \) is satisfied, where \( \epsilon = 10^{-3} \). If the iteration number exceeds \( 10^5 \), we also stop the iteration. Then we call it failure. All of the algorithms are coded in Matlab 7.0 and run on a personal computer with a 2.0 GHZ CPU processor.

5.1. Numerical Comparison of EMPRP and ROSE.

We test the performance of EMPRP and ROSE methods on the following test problems with given initial points. The results are listed in Table 1. \( n \) stands for the dimension of tested problem and \( n_c \) stands for the number of constraints. We will report the following results: the CPU time \( \text{Time} \) (in seconds), the number of iterations \( \text{Iter} \), the number of gradient evaluations \( \text{Geval} \), and the number of function evaluations \( \text{Feval} \).

Problem 1 (HS28 [35]). The function HS28 in [35] is defined as follows:
\[
f(x) = (x_1 + x_3)^2 + (x_2 + x_3)^2, \tag{46}
\]
\[
s.t. \ x_1 + 2x_2 + 3x_3 - 1 = 0, \]
with the initial point \( x_0 = (-4, 1, 1) \). The optimal solution \( x^* = (0.5, -0.5, 0.5) \) and optimal function value \( f(x^*) = 0 \).

Problem 2 (HS48 [35]). The function HS48 in [35] is defined as follows:
\[
f(x) = (x_1 - 1)^2 + (x_2 - x_3)^2 + (x_4 - x_3)^2, \tag{47}
\]
\[
s.t. \ x_1 + x_2 + x_3 + x_4 + x_5 - 5 = 0, \]
\[
x_1 + 2(x_3 + x_5) + 3 = 0, \tag{48}
\]
with the initial point \( x_0 = (3, 5, -3, 2, -2) \). The optimal solution \( x^* = (1, 1, 1, 1, 1) \) and optimal function value \( f(x^*) = 0 \).

Problem 3 (HS49 [35]). The function HS49 in [35] is defined as follows:
\[
f(x) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_4 - 1)^4 + (x_5 - 1)^6, \tag{49}
\]
\[
s.t. \ x_1 + 2x_2 + 3x_3 - 4x_4 = 7 = 0, \]
\[
x_3 + 5x_5 - 6 = 0, \tag{47}
\]
with the initial point \( x_0 = (10, 7, -2, -3, 0.8) \). The optimal solution \( x^* = (1, 1, 1, 1, 1) \) and optimal function value \( f(x^*) = 0 \).

Problem 4 (HS50 [35]). The function HS50 in [35] is defined as follows:
\[
f(x) = (x_1 - x_3)^2 + (x_2 - x_3)^2 + (x_4 - x_5)^2, \tag{47}
\]
\[
s.t. \ x_1 + 2x_2 + 3x_3 - 6 = 0, \]
\[
x_3 + 2x_3 + 3x_4 - 6 = 0, \tag{47}
\]
with the initial point \( x_0 = (35, -31, 11, 5, -5) \). The optimal solution \( x^* = (1, 1, 1, 1, 1) \) and optimal function value \( f(x^*) = 0 \). Moreover, we extend the dimension of function HS51 [35] to 10, 20 with the initial point \( x_0 = (35, -31, 11, \ldots) \). The optimal solution \( x^* = (1, 1, \ldots, 1) \) and optimal function value \( f(x^*) = 0 \).

Problem 5 (HS51 [35]). The function HS51 in [35] is defined as follows:
\[
f(x) = (x_1 - x_2)^2 + (x_2 - x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2, \tag{47}
\]
\[
s.t. \ x_1 + 3x_2 - 4 = 0, \]
\[
x_3 + x_4 - 2x_5 = 0, \tag{47}
\]
\[
x_2 - x_5 = 0, \tag{47}
\]
Table 1: Test results for Problems 1–5 with given initial points.

<table>
<thead>
<tr>
<th>Name</th>
<th>𝑛</th>
<th>𝑛𝑖</th>
<th>Time</th>
<th>Iter</th>
<th>Geval</th>
<th>Feval</th>
<th>Time</th>
<th>Iter</th>
<th>Geval</th>
<th>Feval</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPRP</td>
<td>ROSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS28</td>
<td>3</td>
<td>1</td>
<td>0.0156</td>
<td>20</td>
<td>22</td>
<td>21</td>
<td>0.0468</td>
<td>71</td>
<td>72</td>
<td>143</td>
</tr>
<tr>
<td>HS48</td>
<td>5</td>
<td>2</td>
<td>0.0156</td>
<td>26</td>
<td>28</td>
<td>27</td>
<td>0.0468</td>
<td>65</td>
<td>66</td>
<td>122</td>
</tr>
<tr>
<td>HS49</td>
<td>5</td>
<td>2</td>
<td>0.0156</td>
<td>29</td>
<td>30</td>
<td>30</td>
<td>0.0780</td>
<td>193</td>
<td>194</td>
<td>336</td>
</tr>
<tr>
<td>HS50</td>
<td>5</td>
<td>3</td>
<td>0.0156</td>
<td>22</td>
<td>23</td>
<td>23</td>
<td>0.0624</td>
<td>76</td>
<td>77</td>
<td>139</td>
</tr>
<tr>
<td>HS51</td>
<td>5</td>
<td>3</td>
<td>0.0156</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>0.0312</td>
<td>70</td>
<td>71</td>
<td>148</td>
</tr>
<tr>
<td>HS52</td>
<td>5</td>
<td>3</td>
<td>0.0156</td>
<td>30</td>
<td>31</td>
<td>31</td>
<td>0.0624</td>
<td>70</td>
<td>71</td>
<td>141</td>
</tr>
<tr>
<td>HS50E1</td>
<td>10</td>
<td>8</td>
<td>0.0156</td>
<td>16</td>
<td>17</td>
<td>17</td>
<td>0.0312</td>
<td>63</td>
<td>64</td>
<td>127</td>
</tr>
<tr>
<td>HS50E2</td>
<td>20</td>
<td>18</td>
<td>0.0156</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>0.0936</td>
<td>63</td>
<td>64</td>
<td>127</td>
</tr>
</tbody>
</table>

Table 2: Test results for Problem 6 with given initial points.

<table>
<thead>
<tr>
<th>𝑘</th>
<th>𝑛</th>
<th>𝑛𝑖</th>
<th>Time</th>
<th>Iter</th>
<th>Time</th>
<th>Iter</th>
<th>Time</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPRP</td>
<td>FFR</td>
<td>SPCG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>99</td>
<td>49</td>
<td>0.0568</td>
<td>73</td>
<td>0.0660</td>
<td>80</td>
<td>0.0256</td>
<td>70</td>
</tr>
<tr>
<td>100</td>
<td>199</td>
<td>99</td>
<td>0.1048</td>
<td>103</td>
<td>0.1060</td>
<td>120</td>
<td>0.0756</td>
<td>90</td>
</tr>
<tr>
<td>200</td>
<td>399</td>
<td>199</td>
<td>0.0868</td>
<td>108</td>
<td>0.2660</td>
<td>123</td>
<td>0.0936</td>
<td>97</td>
</tr>
<tr>
<td>300</td>
<td>599</td>
<td>299</td>
<td>0.2252</td>
<td>121</td>
<td>0.4260</td>
<td>126</td>
<td>0.7504</td>
<td>123</td>
</tr>
<tr>
<td>400</td>
<td>799</td>
<td>399</td>
<td>0.3142</td>
<td>121</td>
<td>0.5150</td>
<td>126</td>
<td>0.9804</td>
<td>153</td>
</tr>
<tr>
<td>500</td>
<td>999</td>
<td>499</td>
<td>0.4045</td>
<td>138</td>
<td>0.7030</td>
<td>146</td>
<td>1.4320</td>
<td>176</td>
</tr>
<tr>
<td>1000</td>
<td>1999</td>
<td>999</td>
<td>2.6043</td>
<td>138</td>
<td>2.1560</td>
<td>146</td>
<td>3.3468</td>
<td>198</td>
</tr>
<tr>
<td>3000</td>
<td>5999</td>
<td>2999</td>
<td>22.4075</td>
<td>138</td>
<td>15.1720</td>
<td>146</td>
<td>28.2177</td>
<td>276</td>
</tr>
<tr>
<td>4000</td>
<td>7999</td>
<td>3999</td>
<td>40.1268</td>
<td>121</td>
<td>35.0460</td>
<td>126</td>
<td>56.0604</td>
<td>324</td>
</tr>
<tr>
<td>5000</td>
<td>9999</td>
<td>4999</td>
<td>105.6252</td>
<td>138</td>
<td>85.0460</td>
<td>146</td>
<td>128.1504</td>
<td>476</td>
</tr>
</tbody>
</table>

with the initial point $x_0 = (2.5, 0.5, 2, -1, 0.5)$. The optimal solution $x^* = (1, 1, 1, 1, 1)$ and optimal function value $f(x^*) = 0$.

From Table 1, we can see that the EMPRP method performs better than the Rosen gradient projection method in [1], which implies that the EMPRP method can improve the computational efficiency of the Rosen gradient projection method for solving linear equality constrained optimization problems.

5.2. Numerical Comparison of EMPRP, FFR, and SPCG. In this subsection, we test the EMPRP method and compare it with the spectral gradient method (called SPCG) in Martínez et al. [30] and the feasible Fletcher-Reeves conjugate gradient method (called FFR) in Li et al. [32] on the following large dimensional problems with given initial points. The results are listed in Tables 2, 3, 4, 5, and 6. We will report the following results: the CPU time $\text{Time}$ (in seconds), the number of iterations $\text{Iter}$.

**Problem 6.** Given a positive integer $k$, the function is defined as follows:

$$f(x) = \frac{1}{2} \sum_{i=1}^{k-2} (x_{k+i+1} - x_{k+i})^2,$$

s.t. $x_{k+i} - x_{i+1} + x_i = i, \quad i = 1, \ldots, k-1,$

with the initial point $x_0 = (1, 2, \ldots, k, 2, 3, \ldots, k)$.

**Problem 7.** Given a positive integer $n$, the function is defined as follows:

$$f(x) = \sum_{i=1}^{n} \cos \left( 2\pi x_i \sin \left( \frac{\pi}{20} \right) \right),$$

s.t. $x_i - x_{i+1} = 0.4, \quad i = 1, \ldots, n-1,$

with the initial point $x_0 = (1, 0.6, 0.2, \ldots, 1 - 0.4n^{-1})$. This problem comes from Asaadi [36] and is called MAD6.

**Problem 8.** Given a positive integer $k$, the function is defined as follows:

$$f(x) = \frac{1}{2} \sum_{i=1}^{k-2} (x_{k+i+1} - x_{k+i})^4,$$

s.t. $x_{k+i} - x_{i+1} + x_i = i, \quad i = 1, \ldots, k-1,$

with the initial point $x_0 = (1, 2, \ldots, k, 2, 3, \ldots, k)$.

The optimization function value $f(x^*) = 0$. This problem comes from Martinez et al. [30].
### Table 3: Test results for Problem 7 with given initial points.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( n_c )</th>
<th>EMPRP</th>
<th>Time</th>
<th>Iter</th>
<th>FFR</th>
<th>Time</th>
<th>Iter</th>
<th>SPCG</th>
<th>Time</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>99</td>
<td>0.1468</td>
<td>83</td>
<td></td>
<td>0.1679</td>
<td>94</td>
<td></td>
<td>0.1286</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>199</td>
<td>0.2608</td>
<td>94</td>
<td></td>
<td>0.2860</td>
<td>103</td>
<td></td>
<td>0.2266</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>299</td>
<td>0.6252</td>
<td>121</td>
<td></td>
<td>0.8422</td>
<td>132</td>
<td></td>
<td>0.7504</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>399</td>
<td>1.4075</td>
<td>123</td>
<td></td>
<td>1.6510</td>
<td>138</td>
<td></td>
<td>1.4177</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>499</td>
<td>3.280</td>
<td>141</td>
<td></td>
<td>2.7190</td>
<td>146</td>
<td></td>
<td>2.8790</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>999</td>
<td>8.4045</td>
<td>189</td>
<td></td>
<td>4.6202</td>
<td>202</td>
<td></td>
<td>5.1280</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>1999</td>
<td>17.2608</td>
<td>254</td>
<td></td>
<td>13.4255</td>
<td>288</td>
<td></td>
<td>19.1266</td>
<td>316</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>2499</td>
<td>20.6252</td>
<td>321</td>
<td></td>
<td>19.0940</td>
<td>330</td>
<td></td>
<td>29.1504</td>
<td>383</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4: Test results for Problem 8 with given initial points.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( n )</th>
<th>( n_c )</th>
<th>EMPRP</th>
<th>Time</th>
<th>Iter</th>
<th>FFR</th>
<th>Time</th>
<th>Iter</th>
<th>SPCG</th>
<th>Time</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>99</td>
<td>49</td>
<td>0.0668</td>
<td>78</td>
<td></td>
<td>0.0750</td>
<td>83</td>
<td></td>
<td>0.0286</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>199</td>
<td>99</td>
<td>0.0988</td>
<td>113</td>
<td></td>
<td>0.1462</td>
<td>130</td>
<td></td>
<td>0.0826</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>399</td>
<td>199</td>
<td>0.1978</td>
<td>123</td>
<td></td>
<td>0.2960</td>
<td>133</td>
<td></td>
<td>0.1864</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>599</td>
<td>299</td>
<td>0.3436</td>
<td>128</td>
<td></td>
<td>0.5260</td>
<td>136</td>
<td></td>
<td>0.3504</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>799</td>
<td>399</td>
<td>0.4142</td>
<td>134</td>
<td></td>
<td>0.7143</td>
<td>136</td>
<td></td>
<td>0.5804</td>
<td>186</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>999</td>
<td>499</td>
<td>0.6039</td>
<td>152</td>
<td></td>
<td>0.9030</td>
<td>158</td>
<td></td>
<td>0.9331</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1999</td>
<td>999</td>
<td>2.9065</td>
<td>152</td>
<td></td>
<td>2.9570</td>
<td>158</td>
<td></td>
<td>3.1260</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>3999</td>
<td>1999</td>
<td>8.3408</td>
<td>152</td>
<td></td>
<td>7.8280</td>
<td>158</td>
<td></td>
<td>8.2266</td>
<td>298</td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>5999</td>
<td>2999</td>
<td>19.6086</td>
<td>152</td>
<td></td>
<td>17.4350</td>
<td>158</td>
<td></td>
<td>19.8672</td>
<td>330</td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>7999</td>
<td>3999</td>
<td>46.3268</td>
<td>152</td>
<td></td>
<td>38.1460</td>
<td>136</td>
<td></td>
<td>56.1256</td>
<td>368</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>9999</td>
<td>4999</td>
<td>75.3972</td>
<td>152</td>
<td></td>
<td>58.3867</td>
<td>158</td>
<td></td>
<td>125.7680</td>
<td>398</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5: Test results for Problem 9 with given initial points.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( n )</th>
<th>( n_c )</th>
<th>EMPRP</th>
<th>Time</th>
<th>Iter</th>
<th>FFR</th>
<th>Time</th>
<th>Iter</th>
<th>SPCG</th>
<th>Time</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>99</td>
<td>49</td>
<td>0.0568</td>
<td>65</td>
<td></td>
<td>0.0640</td>
<td>70</td>
<td></td>
<td>0.0266</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>199</td>
<td>99</td>
<td>0.0868</td>
<td>109</td>
<td></td>
<td>0.1362</td>
<td>125</td>
<td></td>
<td>0.0726</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>399</td>
<td>199</td>
<td>0.2110</td>
<td>120</td>
<td></td>
<td>0.2862</td>
<td>128</td>
<td></td>
<td>0.2064</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>599</td>
<td>299</td>
<td>0.3250</td>
<td>127</td>
<td></td>
<td>0.5062</td>
<td>132</td>
<td></td>
<td>0.4524</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>799</td>
<td>399</td>
<td>0.4543</td>
<td>138</td>
<td></td>
<td>0.7456</td>
<td>139</td>
<td></td>
<td>0.6804</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>999</td>
<td>499</td>
<td>0.6246</td>
<td>156</td>
<td></td>
<td>0.9268</td>
<td>155</td>
<td></td>
<td>1.0876</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1999</td>
<td>999</td>
<td>3.9245</td>
<td>156</td>
<td></td>
<td>3.4268</td>
<td>162</td>
<td></td>
<td>3.4560</td>
<td>279</td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>5999</td>
<td>2999</td>
<td>20.6125</td>
<td>156</td>
<td></td>
<td>19.6548</td>
<td>162</td>
<td></td>
<td>20.8656</td>
<td>358</td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>7999</td>
<td>3999</td>
<td>48.2890</td>
<td>156</td>
<td></td>
<td>44.4330</td>
<td>162</td>
<td></td>
<td>57.7680</td>
<td>386</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>9999</td>
<td>4999</td>
<td>95.4680</td>
<td>156</td>
<td></td>
<td>83.8650</td>
<td>162</td>
<td></td>
<td>128.8760</td>
<td>420</td>
<td></td>
</tr>
</tbody>
</table>

**Problem 9.** Given a positive integer \( k \), the function is defined as follows:

\[
f(x) = \sum_{i=1}^{k-2} 100(x_{k+i+1} - x_{k+i})^2 + (1 - x_{k+i})^2. \tag{54}
\]

subject to \( x_{k+i} - x_{i+1} + x_i = i, \quad i = 1, \ldots, k-1 \),

with the initial point \( x_0 = (1, 2, \ldots, k, 2, 3, \ldots, k) \). The optimization function value \( f(x^*) = 0 \).

**Problem 10.** Given a positive integer \( n \), the function is defined as follows:

\[
f(x) = \sum_{i=1}^{n} \cos^2 \left( 2\pi x_i \cdot \sin \left( \frac{\pi}{20} \right) \right), \tag{55}
\]

subject to \( x_i - x_{i+1} = 0.4 \quad i = 1, \ldots, n-1 \),

with the initial point \( x_0 = (1, 0.6, 0.2, \ldots, 1 - 0.4 \ast (n-1)) \).
From Tables 2–6, we can see the EMPRP method and the FFR method in [32] perform better than the SPCG method in Martínez et al. [30] for solving large-scale linear equality constrained optimization problems, as the EMPRP method and the FFR method all are first order methods. But the SPCG method in Martínez et al. [30] needs to compute $B_k$ with quasi-Newton method in which the approximate Hessians satisfy a weak secant equation. However, as the EMPRP method also needs to compute projection, the FFR method in [32] performs better than the EMPRP method when the test problem becomes large.

6. Conclusions

In this paper, we propose a new conjugate gradient projection method for solving linear equality constrained optimization problem (1), which combines the two-term modified Polak-Ribière-Polyak (PRP) conjugate gradient method in Cheng [7] with the Rosen projection method. The proposed method also can be regarded as an extension of the recently developed two-term modified Polak-Ribière-Polyak (PRP) conjugate gradient method in Cheng [7]. Under some mild conditions, we show that it is globally convergent with the Armijio-type line search.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author thanks the anonymous referees for their valuable comments and suggestions that improved the presentation. This work is supported by the NSF of China Grants 11301041, 11371154, 71371065, and 71371195 and the Ministry of Education Humanities and Social Sciences Project Grants 12YJC790027, Project funded by China Postdoctoral Science Foundation, Natural Science Foundation of Hunan Province.

References


