Hindawi
Journal of Mathematics
Volume 2024, Article ID 5548724, 12 pages
https://doi.org/10.1155/2024/5548724

Research Article

Solving Large-Scale Unconstrained Optimization Problems with an Efficient Conjugate Gradient Class

Sanaz Bojari[1] and Mahmoud Paripour[2]

[1]*Department of Basic Science, Hamedan University of Technology, Hamedan, Iran*
[2]*Department of Computer Engineering and Information Technology, Hamedan University of Technology, Hamedan, Iran*

Correspondence should be addressed to Mahmoud Paripour; m_paripour@yahoo.com

Received 11 September 2023; Revised 20 December 2023; Accepted 19 January 2024; Published 5 February 2024

Academic Editor: Xian-Ming Gu

The main goal of this paper is to introduce an appropriate conjugate gradient class to solve unconstrained optimization problems. The presented class enjoys the benefits of having three free parameters, its directions are descent, and it can fulfill the Dai–Liao conjugacy condition. Global convergence property of the new class is proved under the weak-Wolfe–Powell line search technique. Numerical efficiency of the proposed class is confirmed in three sets of experiments including 210 test problems and 11 disparate conjugate gradient methods.

1. Introduction

In recent years, many iterative methods are developed to solve a large-scale unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f\colon \mathbb{R}^n \longrightarrow \mathbb{R}$ is a smooth function with Lipschitz continuous gradient $g(x)$. The process of an iterative optimization algorithm in iteration point x_k is to find a descent direction d_k and a step length α_k and calculate the next iteration point as follows:

$$x_{k+1} = x_k + \alpha_k d_k. \tag{2}$$

Usually, step lengths α_k are considered if they fulfill the conditions of an inexact line search technique. A well-known example of such inexact line search techniques is the weak-Wolfe–Powell (WWP) technique.

$$\begin{cases} f(x_k + \alpha_k d_k) \leq f_k + \sigma_1 \alpha_k g_k^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k, \end{cases} \tag{3}$$

where $f_k = f(x_k)$, $g_k = g(x_k)$, $0 < \sigma_1 < (1/2)$, and $\sigma_1 < \sigma_2 < 1$ [1]. Different inexact line search techniques are presented in [1] and three improvements of 3 are proposed by Bojari and Eslahchi [2], Yuan et al. [3], and Dai and Kou [4].

On the other hand, descent directions d_k are obtained by the Newton-based methods for small problems, the quasi-Newton-based methods for medium size problems, and the gradient-based methods for large-scale problems [1].

Conjugate gradient (CG) method is one of the most popular gradient-based methods, which combines the negative of the gradient and some other available information to develop the next descent direction. Generally, the CG process can be summarized as follows:

$$\begin{aligned} d_0 &= -g_0, \\ d_{k+1} &= -\theta_{k+1} g_{k+1} + \beta_{k+1} d_k + \gamma_{k+1} p_k, \text{for } k = 0, 1, 2, \ldots, \end{aligned} \tag{4}$$

where θ_{k+1} is the scale parameter, β_{k+1} and γ_{k+1} are the CG parameters, and p_k is an arbitrary vector related to previous iterations.

In classic CG methods such as those of Hestenes–Stiefel (HS) [5], Fletcher–Reeves (FR) [6], Polak–Ribiére–Polyak (PRP) [7, 8], Liu–Storey (LS) [9], and Dai–Yuan (DY) [10], only two parts $-g_{k+1}$ and d_k of direction (4) are considered, and parameter β_{k+1} is defined as follows:

$$\beta_{k+1}^{\mathrm{HS}} = \frac{g_{k+1}^T y_k}{d_k^T y_k},$$

$$\beta_{k+1}^{\mathrm{FR}} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2},$$

$$\beta_{k+1}^{\mathrm{PRP}} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \tag{5}$$

$$\beta_{k+1}^{\mathrm{LS}} = -\frac{g_{k+1}^T y_k}{d_k^T g_k},$$

$$\beta_{k+1}^{\mathrm{DY}} = \frac{\|g_{k+1}\|^2}{d_k^T y_k},$$

where $y_k = g_{k+1} - g_k$ and parameter θ_{k+1} is determined as $\theta_{k+1} = 1$. Note that $\|.\|$ denotes the Euclidean norm of vectors.

Over the years, many researchers developed method (5) and increased their efficiency in theoretical and numerical views. For example, interested readers can see some modifications of the HS method in the study by Faramarzi and Amini [11] and Hu et al. [12], several combinations of the FR method in the work by Abubakar et al. [13] and Sakai and Iiduka [14], various developments of the PRP method in the study by Mishra et al. [15], Wu [16], and Andrei [17], an extended LS method in [18], and variant improvements of the DY method in the study by Deepho et al. [19], Zhu et al. [20], and Jiang and Jian [21]. Furthermore, some researchers used techniques such as quasi-Newton [22, 23], regularization [24–26], a combination of above methods [27, 28], or alternative techniques [29, 30] and introduced appropriate CG methods to solve optimization problems. To discuss the CG methods in more detail, the readers can see [31].

In addition to their original authors, the issue of global convergence of method (5) has also been investigated by some researchers such as Al-Baali [32] and Gilbert and Nocedal [33].

As we mentioned before, one technique to develop a CG method is to think of a $L-$BFGS direction

$$d_{k+1} = -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k}\right)\frac{g_{k+1}^T s_k}{s_k^T y_k}\right] s_k + \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k, \tag{6}$$

where $s_k = \alpha_k d_k = x_{k+1} - x_k$ and $\tau_k \geq 0$ as a three-term CG direction [22, 23]. This point of view usually leads to appropriate behavior in numerical experiments.

Besides, it is known that the PRP method has an excellent global convergence, which means that it generally solves more problems than other classic methods in equation (5). A well-known extension of the PRP method is the three-term CG direction

$$d_{k+1} = -\frac{y_k^T s_k}{\|g_k\|^2} g_{k+1} + \frac{y_k^T g_{k+1}}{\|g_k\|^2} s_k - \frac{g_{k+1}^T s_k}{\|g_k\|^2} y_k, \tag{7}$$

which is introduced by Andrei in [17]. It is established that direction (7) is descent and satisfies Dai and Liao [34] conjugacy condition. Also, the method is globally convergent under the WWP line search technique.

It is known that large-scale optimization problems have wide applications in science, engineering, transport, military, space technology [1, 35], artificial intelligence and image processing [12], risk managing [13, 19], and business and financial management [36, 37]. Furthermore, as we mentioned before, the CG methods are usually the best choices to solve a large-scale optimization problem. For these reasons, and also because of the excellent theoretical and numerical performance of methods (6) and (7), in this paper, we combine them and create a new class of three-term scaled conjugate gradient methods. We indicate that our class inherits all of the superb properties of methods (6) and (7). Furthermore, we illustrate the advantages of using the new class by running multitudinous numerical competitions [38].

The rest of this paper is organized as follows. In the next section, the new class of scaled three-term CG directions is introduced. Then, in Section 3, some properties of the presented class and the global convergence theorems are proved. Finally, the numerical results are presented in Section 4.

2. The Algorithm

In this section, we want to know what will happen if we consider a CG direction with denominators $\|g_k\|^2$, similar to the PRP method and equation (7), and numerators that contained parts $g_{k+1}^T y_k$, $g_{k+1}^T s_k$, and $\|y_k\|^2 g_{k+1}^T s_k$ such as equation (6). Therefore, we first reckon the following direction:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k - \left(1 + \|y_k\|^2\right) g_{k+1}^T s_k}{\|g_k\|^2} s_k - \frac{g_{k+1}^T s_k}{\|g_k\|^2} y_k. \tag{8}$$

Then, as we will show in the following, to confirm that our method satisfies the Dai–Liao conjugacy condition and to prove the global convergence theorems, we were forced to also consider the scaled coefficient $s_k^T y_k / \|g_k\|^2$ of equation (7). So the structure of our directions became

$$d_{k+1} = -\frac{s_k^T y_k}{\|g_k\|^2} g_{k+1} + \frac{g_{k+1}^T y_k - \left(1 + \|y_k\|^2\right) g_{k+1}^T s_k}{\|g_k\|^2} s_k - \frac{g_{k+1}^T s_k}{\|g_k\|^2} y_k, \tag{9}$$

which is actually a modification of equation (7).

In the end, to enjoy the benefits of free parameters, such as the possibility of creating a balance between the components of the direction and the possibility of choosing an appropriate method for different problems, we introduced our new CG class as follows:

$$d_{k+1} = -\tau_1 \frac{s_k^T y_k}{\|g_k\|^2} g_{k+1} + \frac{\tau_1 g_{k+1}^T y_k - \left(\tau_2 + \tau_3 \|y_k\|^2\right) g_{k+1}^T s_k}{\|g_k\|^2} s_k - \tau_1 \frac{g_{k+1}^T s_k}{\|g_k\|^2} y_k, \tag{10}$$

where $\tau_1 > 0$ and $\tau_2, \tau_3 \geq 0$ are three arbitrary constants.

In the rest of this article, for simplicity, we call direction (10) as CG3p class. The process of CG3p class is described in Algorithm 1.

One of the interesting features of the CG3p class is that its members fulfill the conjugacy condition of Dai and Liao [34] whenever $s_k^T y_k > 0$. For example, in Algorithm 1, we use WWP line search technique (3), so the positiveness of $s_k^T y_k$ is guaranteed. Therefore, from the definition of the CG3p class in equation (10) and conditions $\tau_1 > 0$ and $\tau_2, \tau_3 \geq 0$, we have

$$\begin{aligned} d_{k+1}^T y_k &= -\tau_1 \frac{s_k^T y_k}{\|g_k\|^2} g_{k+1}^T y_k + \tau_1 \frac{g_{k+1}^T y_k}{\|g_k\|^2} s_k^T y_k - \frac{\left(\tau_2 + \tau_3 \|y_k\|^2\right) g_{k+1}^T s_k}{\|g_k\|^2} s_k^T y_k \\ -\tau_1 \frac{g_{k+1}^T s_k}{\|g_k\|^2} y_k^T y_k &= -\left[\frac{\left(\tau_2 + \tau_3 \|y_k\|^2\right) s_k^T y_k + \tau_1 \|y_k\|^2}{\|g_k\|^2}\right] g_{k+1}^T s_k \\ &\coloneqq -t_k g_{k+1}^T s_k, \quad t_k \geq 0. \end{aligned} \tag{11}$$

Remark 1. Direction (7) is a member of the CG3p class, which can be established by setting $\tau_1 = 1$ and $\tau_2 = \tau_3 = 0$.

3. Convergence Theorems

To prove the global convergence of the CG3p class, we need Zoutendijk lemma [1] as well as the following common assumption.

Assumption 2

(1) The level set $\Omega = \{x \in \mathbb{R}^n | f(x) < f(x_0)\}$ is bounded

(2) In some neighborhood $\mathcal{N}$ of Ω, function f is continuously differentiable and its gradient function g is Lipschitz continuous

Lemma 3. *(Zoutendijk lemma) Consider an iterative algorithm of form equation (2) and Assumption 2 and define θ_{k+1} as the angles between d_{k+1} and $-g_{k+1}$, i.e.,*

$$\cos(\theta_{k+1}) = -\frac{d_{k+1}^T g_{k+1}}{\|d_{k+1}\| \|g_{k+1}\|}. \tag{12}$$

If directions d_k are descent and step lengths α_k are obtained from WWP condition (3), then

$$\sum_{k\geq 0} \cos^2(\theta_{k+1}) \|g_{k+1}\|^2 < \infty. \tag{13}$$

Proof. See Theorem 3.2 in [1].

Eventually, to have the global convergence of Algorithm 1, we will confirm two issues:

(1) The directions in the CG3p class are descent

(2) $\lim_{k\longrightarrow\infty} \cos^2(\theta_{k+1}) > 0$ □

Theorem 4. *Suppose that Assumption 2 is true. Under the WWP line search technique, the directions of the CG3p class are descent.*

Proof. From the definition of d_{k+1} in equation (8), we have the following equation:

$$\begin{aligned} d_{k+1}^T g_{k+1} = &-\tau_1 \frac{s_k^T y_k}{\|g_k\|^2} \|g_{k+1}\|^2 + \tau_1 \frac{g_{k+1}^T y_k}{\|g_k\|^2} s_k^T g_{k+1} \\ &- \tau_2 \frac{\left(g_{k+1}^T s_k\right)^2}{\|g_k\|^2} - \tau_3 \frac{\|y_k\|^2}{\|g_k\|^2} \left(g_{k+1}^T s_k\right)^2 \\ &- \tau_1 \frac{g_{k+1}^T s_k}{\|g_k\|^2} y_k^T g_{k+1} \leq -\tau_1 \frac{s_k^T y_k}{\|g_k\|^2} \|g_{k+1}\|^2. \end{aligned} \tag{14}$$

Since by the WWP technique, we gain $s_k^T y_k > 0$ and also because $\tau_1 > 0$ and $\tau_2, \tau_3 \geq 0$, the directions of the CG3p class are descent. □

Theorem 5. *Consider Assumption 2. For θ_{k+1} generated by Algorithm 1, we have the following equation:*

$$\lim_{k\longrightarrow\infty} \cos^2(\theta_{k+1}) > 0. \tag{15}$$

Proof. We can rewrite the directions of the CG3p class as follows:

```
Input: An initial point x_0 ∈ ℝ^n and some constants 0 < σ_1 < (1/2), σ_1 < σ_2 < 1, ε > 0, τ_1 > 0, and τ_2, τ_3 ≥ 0.
Set k = 0.
While ‖g_k‖_∞ > ε do
    if k = 0 then
        Set d_k = −g_k.
    else
        Obtain d_k by (10).
    end
    Calculate α_k by (3).
    Set x_{k+1} = x_k + α_k d_k.
    Set k = k + 1.
end
Output: The solution x* of problem (1).
```

ALGORITHM 1: Pseudocode of CG3p class.

$$d_{k+1} = -Q_{k+1} g_{k+1}, \tag{16}$$

with

$$\begin{aligned} Q_{k+1} &\coloneqq \frac{1}{\|g_k\|^2}\left[\tau_1\left(s_k^T y_k\right)I - \tau_1 s_k y_k^T + \left(\tau_2 + \tau_3\|y_k\|^2\right)s_k s_k^T + \tau_1 y_k s_k^T\right] \\ &\coloneqq \frac{1}{\|g_k\|^2}\left[\xi_1 I + \xi_2\left(y_k s_k^T - s_k y_k^T\right) + \xi_3 s_k s_k^T\right], \end{aligned} \tag{17}$$

where I is the $n \times n$ identity matrix. Let us examine the three parts of matrix Q_{k+1} separately:

(1) Part $\xi_1 I$: since $\tau_1 > 0$ and $s_k^T y_k > 0$ (from WWP technique), this part is a positive definite diagonal matrix with eigenvalues far from zero

(2) Part $\xi_2 (y_k s_k^T - s_k y_k^T)$: this part is a skew-symmetric matrix, and therefore, its eigenvalues are purely imaginary or zero. Also, $\xi_2 = \tau_1 > 0$

(3) Part $\xi_3 s_k s_k^T$: here, we have a rank one positive semidefinite matrix with a nonnegative coefficient $\xi_3\,(\tau_2, \tau_3 \geq 0)$

From these three observations, it is obvious that the condition numbers of matrices Q_{k+1}, i.e., κ_{k+1}, are far from zero and their eigenvalues have positive real parts. On the other hand, for all $0 \neq x \in \mathbb{R}^n$, we have the following equation:

$$x^T Q_{k+1} x = \frac{1}{\|g_k\|^2}\left[\xi_1\|x\|^2 + \xi_2\left(\left(x^T y_k\right)\left(s_k^T x\right) - \left(x^T s_k\right)\left(y_k^T x\right)\right) + \xi_3\left\|x^T s_k\right\|^2\right] > 0, \tag{18}$$

which means that the square roots of matrices Q_{k+1} can be defined.

Now, if we set $Z_{k+1} = Q_{k+1}^{1/2} g_{k+1}$, from Kantorovich inequality [1], we obtain the following equation:

$$\begin{aligned} \cos^2\left(\theta_{k+1}\right) &= \frac{d_{k+1}^T g_{k+1}}{\|d_{k+1}\|\,\|g_{k+1}\|} \\ &= \frac{\|Z_{k+1}\|^4}{\left(Z_{k+1}^T Q_{k+1} Z_{k+1}\right)\left(Z_{k+1}^T Q_{k+1}^{(1/2)} Z_{k+1}\right)} \geq \frac{4\kappa_{k+1}}{\left(1+\kappa_{k+1}\right)^2}. \end{aligned} \tag{19}$$

Since κ_{k+1} is far from zero, the proof is complete. □

Theorem 6. *Under Assumption 2, for g_{k+1} obtained from Algorithm 1, we have the following equation:*

$$\lim_{k \longrightarrow \infty} \|g_{k+1}\| = 0. \tag{20}$$

Proof. From Theorem 4 and inequality (9), we gain the following equation:

$$\lim_{k \longrightarrow \infty} \cos^2\left(\theta_{k+1}\right)\|g_{k+1}\|^2 = 0. \tag{21}$$

Therefore from inequality (15) in Theorem 5, we obtain the following equation:

$$\lim_{k\longrightarrow\infty}\|g_{k+1}\|^2 = 0, \quad (22)$$

which leads us to

$$\lim_{k\longrightarrow\infty}\|g_{k+1}\| = 0. \quad (23)$$

□

4. Numerical Results

One important subject for an iterative method is how it performs numerically. To confirm the efficiency of the CG3p class in the structure of Algorithm 1, we create three sets of experiments. In all three sets, we perform the following:

(1) Run our codes in MATLAB 9.5 and a computer (Intel i5-10400F, 2.90 GHz, and 8 GB memory) with Windows 10 operating system.

(2) Terminate the algorithms whenever $\|g_k\| \leq 10^{-5}$ or the number of iterations exceed 4000 or the number of function evaluation exceed 20000. Note that in the last two cases, we say the algorithm is not successful.

(3) We use the WWP line search technique in a bisection form similar to Algorithm 2.5.1 of [35], with $\sigma_1 = 10^{-4}$, $\sigma_2 = 0.8$, and initial step lengths

$$\begin{aligned}\alpha_0^0 &= 1,\\ \alpha_{k+1}^0 &= \alpha_k\frac{\|d_k\|}{\|d_{k+1}\|}, \quad \text{for } k = 0, 1, 2, \cdots\end{aligned} \quad (24)$$

(4) Stop the loops of line search algorithm after 15 tries, to avoid an uphill search direction.

(5) Select 42 test problems from [39], which are shown in Table 1, and consider them in five dimensions [1000, 5000, 10000, 15000, 20000].

(6) Compare the algorithms in four terms:

(i) k: the number of iterations
(ii) nf: the number of function evaluations
(iii) ng: the number of gradient evaluations
(iv) t: the CPU time in seconds

(7) Apply Dolan and Moré method [40] to compare the algorithms. In their method, for a threshold $\tau \geq 1$, the probability function $P_q(\tau)$ represents the percentage of problems that are solved by solver q within a factor τ of the best solution. We call the graph of $P_q(\tau)$ for all solvers q a performance profile.

Moreover, for the first two sets, we test the CG3p class with 25 sets of randomly chosen parameters and consider the best one with $(\tau_1, \tau_2, \tau_3) = (0.7, 0.2, 0.1)$, as our representative in the competitions. This means that although it is not possible to choose the optimal set of parameters for a problem, with a high probability, the users can be sure that any selected set of parameters will solve their problem with appropriate results.

In the first set of experiments, we compare our chosen candidate of the CG3p class, which is attained by setting $(\tau_1, \tau_2, \tau_3) = (0.7, 0.2, 0.1)$, with classic methods (5). $P_q(1)$ of this competition and the percent of problems that are solved by each algorithm are presented in Tables 2 and 3, respectively. In addition, the performance profiles of this competition are displayed in Figures 1 to 4. As we predicted, the PRP method solved more problems than other classic methods, but its results are not good. Hence, in Figures 1 to 4, PRP is usually the worst method at the beginning (for $\tau = 1$) but gradually becomes the best one among the classic methods as the value of τ increases. On the other hand, DY method is the best one among the five classic methods (5). From Tables 2 and 3 and Figures 1 to 4, it is clear that our candidate of the CG3p class is the best method in this competition. Thus, we reach our goal of creating a method with excellent global convergence of PRP and distinguished behavior of L − BFGS-like methods.

Table 1: The test problems.

No.	Name
1	Extended Rosenbrock function
2	Extended withe Holst function
3	Extended penalty function
4	Raydan 2 function
5	Diagonal 2 function
6	Hager function
7	Generalized tridiagonal 1 function
8	Extended tridiagonal 1 function
9	Extended TET function
10	Generalized tridiagonal 2 function
11	Diagonal 5 function
12	Extended Himmelblau function
13	Generalized PSC1 function
14	Extended PSC1 function
15	Extended Powell function
16	Extended BD1 function
17	Extended Maratos function
18	Extended Cliff function
19	Perturbed quadratic diagonal function
20	Extended Wood function
21	Extended quadratic penalty QP2 function
22	Extended quadratic exponential EP1 function
23	Extended tridiagonal 2 function
24	ARGLINB function (CUTE)
25	NONDQUAR function (CUTE)
26	Broyden tridiagonal function
27	LIARWHD function (CUTE)
28	EDENSCH function (CUTE)
29	BDEXP function (CUTE)
30	NONSCOMP function (CUTE)
31	VARDIM function (CUTE)
32	QUARTC function (CUTE)
33	SINQUAD function (CUTE)
34	Extended DENSCHNB function (CUTE)
35	Extended DENSCHNF function (CUTE)
36	LIARWHD function (CUTE)
37	COSINE function (CUTE)
38	Generalized quadratic function
39	Diagonal 7 function
40	Diagonal 8 function
41	Full Hessian FH3 function
42	SINCOS function

Table 2: $P_q(1)$ of all methods in the first set of experiments.

	HS	FR	PRP	LS	DY	CG3p
k	0.2000	0.2667	0.0905	0.1190	0.3238	0.4857
nf	0.1190	0.2286	0.0429	0.1286	0.4524	0.5095
ng	0.1238	0.2333	0.0571	0.1238	0.4143	0.5238
t	0.0905	0.1333	0.0333	0.0095	0.3048	0.3714

Table 3: The percent of problems that are solved by each algorithm in the first set of experiments.

	HS	FR	PRP	LS	DY	CG3p
Percent	67.7619	81.4286	89.0476	75.2381	79.5238	91.4286

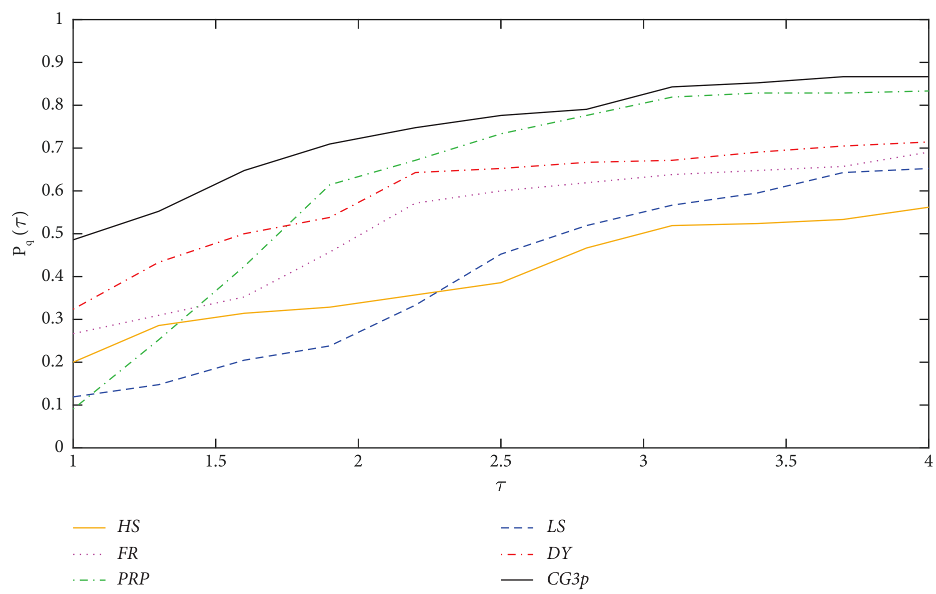

Figure 1: The performance profiles of first competition in term k.

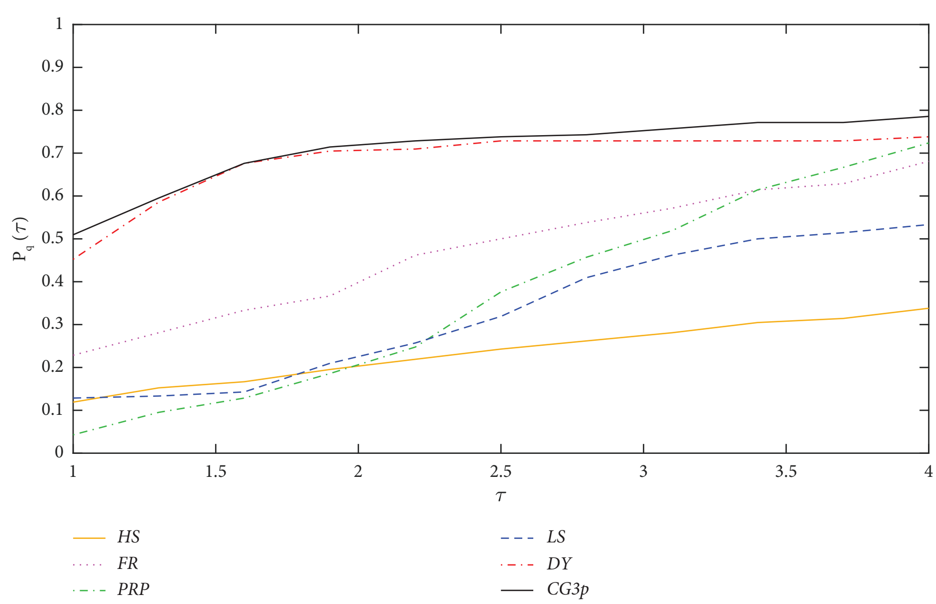

Figure 2: The performance profiles of first competition in term nf.

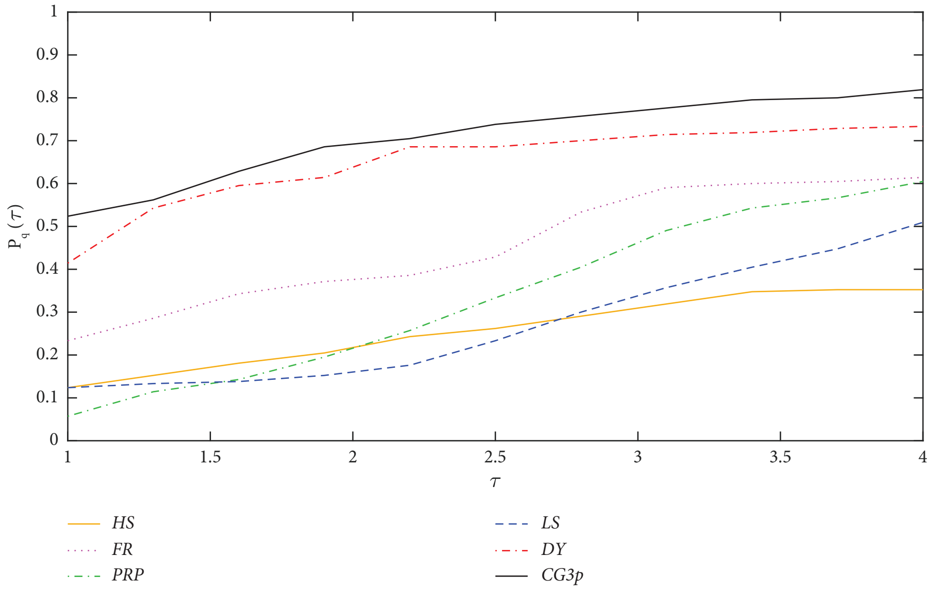

Figure 3: The performance profiles of first competition in term ng.

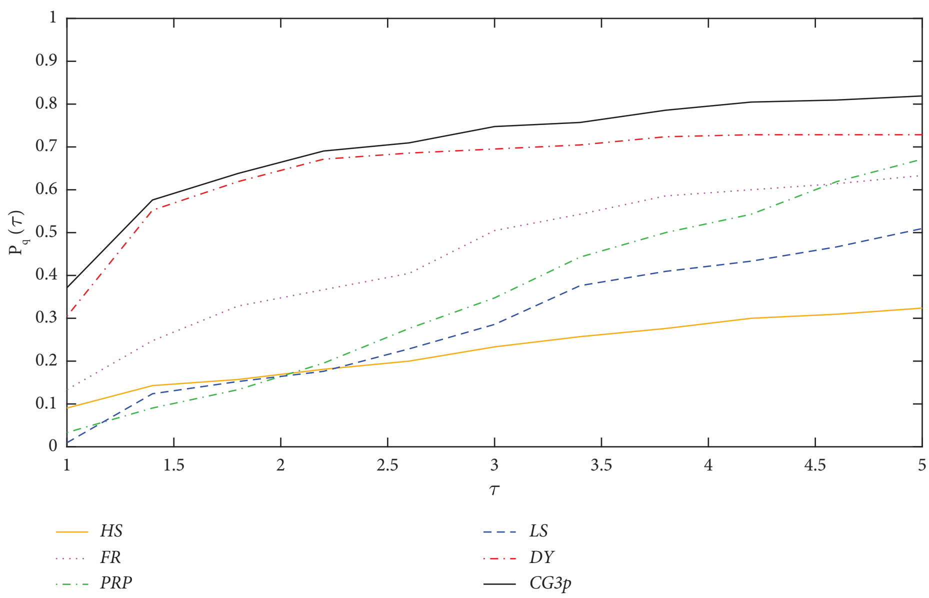

Figure 4: The performance profiles of first competition in term t.

Remark 7. Since the three-term CG directions sometimes are more sensitive to round-off error than the two-term ones, in this set of experiments, we consider $\|g(x^*)\|_\infty$ as a criterion to compare the round-off error of the classic method (5) with our chosen candidate of the CG3p class. The performance profile of the first set of experiments in term $\|g(x^*)\|_\infty$ is presented in Figure 5. From Tables 2 and 3 and Figure 5, it is clear that our method solved more problems with less number of iterations and reached more accurate answers. So it seems that the CG3p class can control the round-off error properly.

For the second set of experiments, we consider seven newly developed CG methods in Table 4.

(1) HZ: a descent two-term member of Dai and Liao family

(2) AABL: a scaled three-term CG method

(3) W: a three-term modification of the PRP method

(4) LFZ: a three-term CG method

(5) ZZW: a two-term modification of the DY method

(6) DAMA: a hybrid two-term modification of the DY method

(7) CG3p: Algorithm 1 with our selected candidate of CG3p class

The results of this competition are demonstrated in Tables 5 and 6 and Figures 6–9.

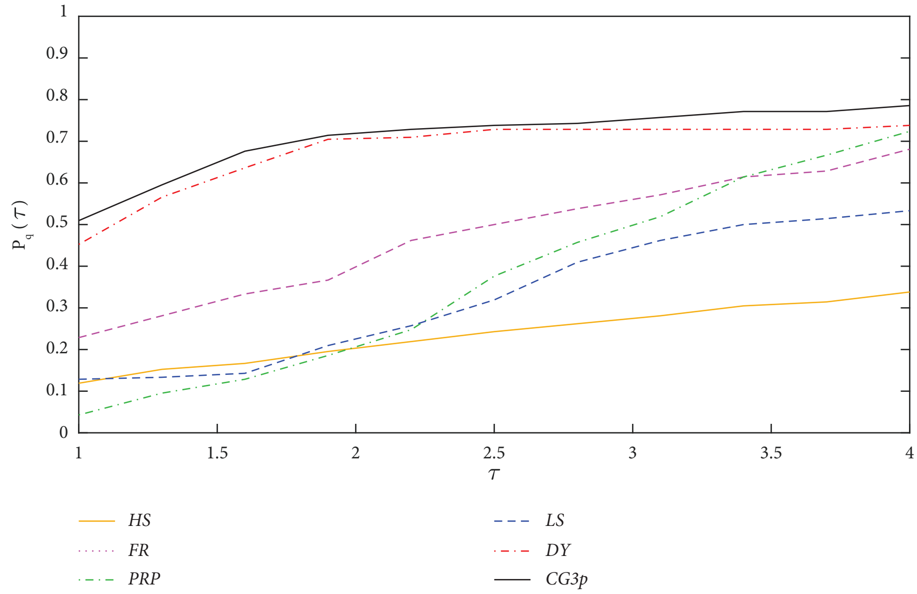

FIGURE 5: The performance profiles of first competition in term $\|g(x^*)\|_\infty$.

TABLE 4: The methods which participate in the second set of experiments.

Name	Direction	Reference
HZ	$d_{k+1} = -g_{k+1} + (g_{k+1}^T y_k - t_k g_{k+1}^T d_k)/(d_k^T y_k) d_k, t_k = 2\|y_k\|^2/d_k^T y_k$	[41]
AABL	$d_{k+1} = -(\|s_k\|^2/s_k^T y_k - \sqrt{(\|s_k\|^2/s_k^T y_k)^2 - \|s_k\|^2/\|y_k\|^2})(g_{k+1} + g_{k+1}^T y_k/\|y_k\|^2 y_k) - (g_{k+1}^T s_k)/(s_k^T y_k) s_k$	[27]
W	$d_{k+1} = -g_{k+1} + (g_{k+1}^T y_k)/2\|g_k\|^2 + 5\|d_k\|\|y_k\| + 3\|d_k\|\|g_k\| d_k - (g_{k+1}^T y_k)/2\|g_k\|^2 + 5\|d_k\|\|y_k\| + 3\|d_k\|\|g_k\| y_k$	[16]
LFZ	$d_{k+1} = -g_{k+1} + g_{k+1}^T y_k/\|d_k\|^2 d_k - g_{k+1}^T d_k/\|d_k\|^2 y_k$	[30]
ZZW	$d_{k+1} = -g_{k+1} + \beta_k d_k,$ $\beta_k = \begin{cases} (g_{k+1}^T (g_{k+1} - g_{k+1}^T d_k/\|d_k\|^2 d_k))/(d_k^T y_k + 1.01 g_{k+1}^T d_k), & g_{k+1}^T d_k \geq 0, \\ 0, & \text{otherwise.} \end{cases}$	[20]
DAMA	$d_{k+1} = -(1 + t_k g_{k+1}^T d_k/w_k) g_{k+1} + (\|g_{k+1}\|^2/w_k - \|g_{k+1}\|^2 g_{k+1}^T d_k/w_k^2) d_k,$ $t_k = \min\{0.02, \max\{0, g_{k+1}^T (y_k - s_k)/\|g_{k+1}\|^2\}\}, w_k = \max\{0.02\|d_k\|\|g_{k+1}\|, -d_k^T g_k, d_k^T y_k\}$	[19]
CG3p	$d_{k+1} = -0.7 s_k^T y_k/\|g_k\|^2 g_{k+1} + 0.7 g_{k+1}^T y_k - (0.2 + 0.1\|y_k\|^2) g_{k+1}^T s_k/\|g_k\|^2 s_k - 0.7 g_{k+1}^T s_k/\|g_k\|^2 y_k$	(8)

Table 5 exhibits that in the structure of Algorithm 1, the representative of the CG3p class has solved 51.43, 41.43, and 45.71 percent of problems with the least number of algorithm iterations, function evaluations, and gradient evaluations, respectively. The method with the shortest CPU time is also CG3p. Figures 6–9 show that the CG3p method has behaved acceptably even in the cases where it was not the best method. Table 6indicates that our chosen member of the CG3p class has solved the largest number of problems (91.4286 percent which is about 31 percent more than the worst result) among the participating methods in this set of experiments. So, all the outcomes of the second set of experiments can easily show the advantages of CG3p class and therefore confirm its superiority.

Although Andrei has shown that finding the best CG method is one of the open problems in optimization [42], in our third set of experiments, we try to numerically investigate the effects of the parameters (τ_2, τ_3), or namely, parts $-g_{k+1}^T s_k$ and $-\|y_k\|^2 g_{k+1}^T s_k$, in local and global convergence of Algorithm 1. To this aim, we consider three members of the CG3p class with $(\tau_1, \tau_2, \tau_3) = (0.1, 0, 0)$,

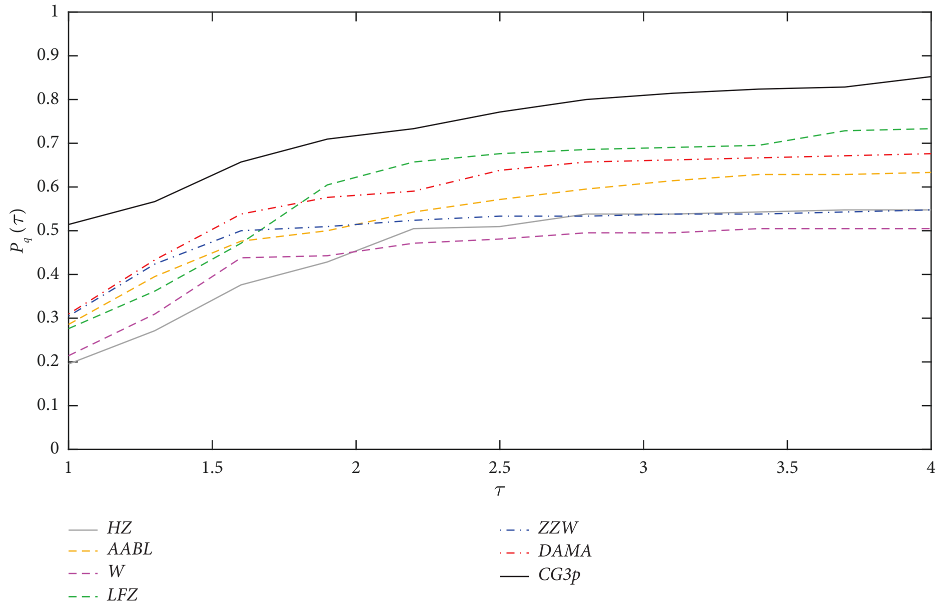

Figure 6: The performance profiles of second competition in term k.

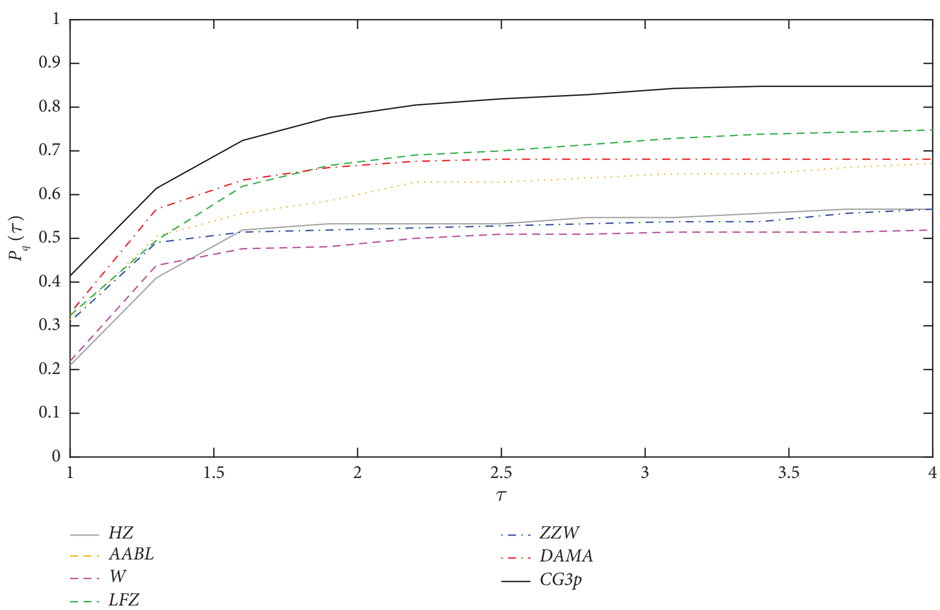

Figure 7: The performance profiles of second competition in term nf.

Table 5: $P_q(1)$ of all methods in the second set of experiments.

	HZ	AABL	W	LFZ	ZZW	DAMA	CG3p
k	0.1952	0.2857	0.2143	0.2762	0.3048	0.3095	0.5143
nf	0.2095	0.3143	0.2190	0.3238	0.3095	0.3286	0.4143
ng	0.2095	0.2905	0.2143	0.3000	0.3095	0.3190	0.4571
t	0.1048	0.0952	0.0333	0.1524	0.1905	0.0667	0.2905

Table 6: The percent of problems that are solved by each algorithm in the second set of experiments.

	HZ	AABL	W	LFZ	ZZW	DAMA	CG3p
Percent	63.3810	69.0476	59.5233	83.3333	63.8095	71.4286	91.4286

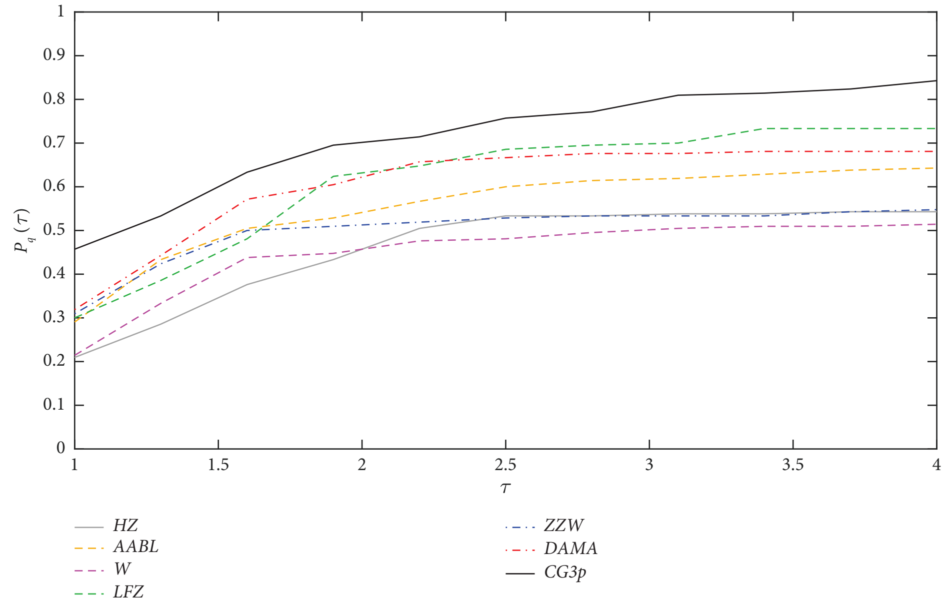

Figure 8: The performance profiles of second competition in term *ng*.

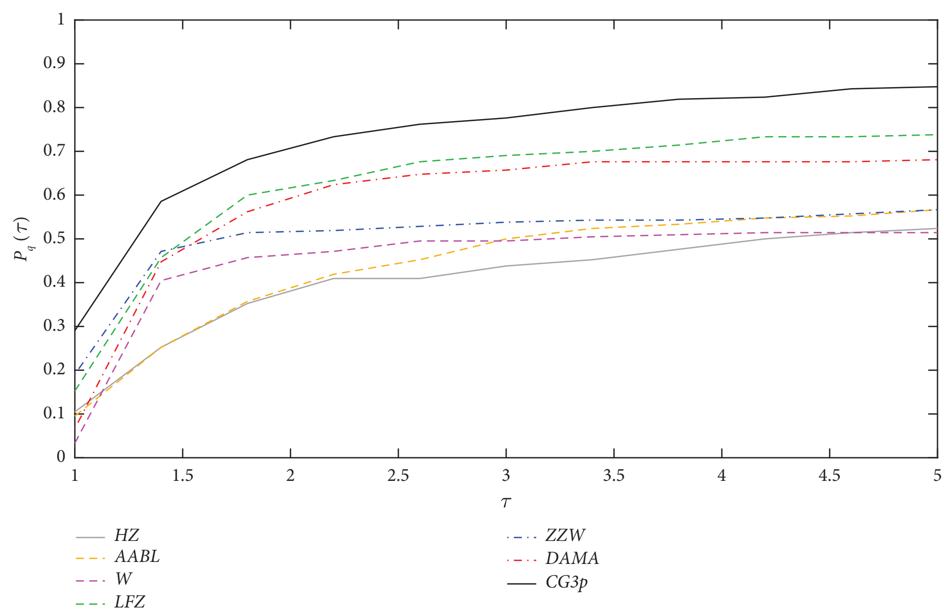

Figure 9: The performance profiles of second competition in term *t*.

Table 7: The results of the third set of experiments.

	CG3p1	CG3p2	CG3p3
$P_q(1)$ in term k	0.3762	0.5381	0.4000
$P_q(1)$ in term nf	0.3571	0.5286	0.4571
$P_q(1)$ in term ng	0.3429	0.5429	0.4286
$P_q(1)$ in term t	0.1571	0.4143	0.3714
The percent of problems that each algorithm solved	88.5714	86.1905	90.0000

$(\tau_1, \tau_2, \tau_3) = (0.1, 0.1, 0)$, and $(\tau_1, \tau_2, \tau_3) = (0.1, 0, 0.1)$ and call them CG3p1, CG3p2, and CG3p3, respectively. Please note that the directions of *CG3p*1 are actually direction (7) multiplied in 0.1.

The results of the third set of experiments, which are displayed in Table 7, suggest that both parts $-g_{k+1}^T s_k$ and $-\|y_k\|^2 g_{k+1}^T s_k$ have improved local convergence and reduced the costs. In addition, it seems that part $-g_{k+1}^T s_k$ had more effect in improving local convergence, and part $-\|y_k\|^2 g_{k+1}^T s_k$ had more effect in improving global convergence.

Remark 8. It seems that the wonderful numerical results of the selected member of CG3p are due to the following reasons:

(1) The coefficient $s_k^T y_k/\|g_k\|^2$ which is known as scaling coefficient. This element controls the first part of a CG direction (part $-g_{k+1}$).
(2) The coefficient $(1+\|y_k\|)g_{k+1}^T s_k$ in the second part of the directions (part s_k). This element is inherited from equation (6), and it is one of the reasons for the appropriate behavior of $L-$ BFGS-like CG directions in numerical experiments.
(3) The denominator $\|g_k\|^2$. This element usually leads to a global convergence for general functions. So, the algorithm presumably solves more problems.
(4) The three free parameters τ_1, τ_2, and τ_3. These parameters create a balance between the components of the direction.

5. Conclusion

In this paper, we developed the new CG class CG3p by considering both PRP and $L-$BFGS methods. In order to encourage the readers to use the CG3p class, we displayed that

(1) The directions of the CG3p class satisfy the Dai–Liao conjugacy condition. So, it is indeed a CG method.
(2) Its directions fulfill inequality $d_{k+1}^T g_{k+1} \le -\tau_1 s_k^T y_k/\|g_k\|^2\|g_{k+1}\|^2$. This means that under any line search technique which can guarantee $s_k^T y_k > 0$, they are descent. In addition, if $\tau_1 s_k^T y_k/\|g_k\|^2 \le 1$, then the directions of the CG3p class could be considered as sufficient descent.
(3) Under WWP line search, the method is globally convergent, without any assumption (such as convexity) on $f(x)$.
(4) Due to the presence of three free parameters, the CG3p class contains an infinite number of directions. Thus, the users can select an appropriate CG method according to their problems.
(5) The method yields amazing results in numerical experiments because of its structure.

Data Availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Disclosure

A preprint has previously been published (Bojari et al. in Research Square (2023)).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All the authors contributed equally to this work.

References

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, NY, USA, 2nd edition, 2006.
[2] S. Bojari and M. R. Eslahchi, "Global convergence of a family of modified BFGS methods under a modified weak-Wolfe-Powell line search for nonconvex functions," *International Journal of Operational Research*, vol. 18, no. 2, pp. 219–244, 2020.
[3] G. Yuan, Z. Wei, and X. Lu, "Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search," *Applied Mathematical Modelling*, vol. 47, no. 1, pp. 811–825, 2017.
[4] Y. H. Dai and C. X. Kou, "A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 296–320, 2013.
[5] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–435, 1952.
[6] R. Fletcher and C. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149–154, 1964.
[7] E. Polak and G. Ribiére, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française D'Informatique Et De Recherche Opérationnelle Série Rouge*, vol. 3, no. 16, pp. 35–43, 1969.
[8] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.
[9] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.
[10] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.
[11] P. Faramarzi and K. Amini, "A spectral three-term Hestenes–Stiefel conjugate gradient method," *International Journal of Operational Research*, vol. 19, no. 1, pp. 71–92, 2021.
[12] W. Hu, J. Wu, and G. Yuan, "Some modified Hestenes–Stiefel conjugate gradient algorithms with application in image restoration," *Applied Numerical Mathematics*, vol. 158, no. 1, pp. 360–376, 2020.
[13] A. B. Abubakar, P. Kumam, M. Malik, P. Chaipunya, and A. H. Ibrahim, "A hybrid FR–DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection," *AIMS Mathematics*, vol. 6, no. 6, pp. 6506–6527, 2021.
[14] H. Sakai and H. Iiduka, "Sufficient descent Riemannian conjugate gradient methods," *Journal of Optimization Theory and Applications*, vol. 190, no. 1, pp. 130–150, 2021.
[15] S. K. Mishra, S. K. Chakraborty, M. E. Samei, and B. Ram, "A q-Polak–Ribière–Polyak conjugate gradient algorithm for unconstrained optimization problems," *Journal of Inequalities and Applications*, vol. 2021, no. 1, pp. 25–29, 2021.
[16] Y. Wu, "A modified three-term PRP conjugate gradient algorithm for optimization models," *Journal of Inequalities and Applications*, vol. 2017, no. 1, p. 97, 2017.

[17] N. Andrei, "A modified Polak-Ribière-Polyak conjugate gradient algorithm for unconstrained optimization," *Optimization*, vol. 60, no. 12, pp. 1457–1471, 2011.
[18] A. Alhawarat, H. Alolaiyan, I. A. Masmali, Z. Salleh, and S. Ismail, "A descent four-term of Liu and Storey conjugate gradient method for large scale unconstrained optimization problems," *European Journal of Pure and Applied Mathematics*, vol. 14, no. 4, pp. 1429–1456, 2021.
[19] J. Deepho, A. B. Abubakar, M. Malik, and I. K. Argyros, "Solving unconstrained optimization problems via hybrid CD–DY conjugate gradient methods with applications," *Journal of Computational and Applied Mathematics*, vol. 405, no. 1, Article ID 113823, 2022.
[20] Z. Zhu, D. Zhang, and S. Wang, "Two modified DY conjugate gradient methods for unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 373, no. 1, Article ID 125004, 2020.
[21] X. Jiang and J. Jian, "Improved Fletcher–Reeves and Dai–Yuan conjugate gradient methods with the strong Wolfe line search," *Journal of Computational and Applied Mathematics*, vol. 348, no. 1, pp. 525–534, 2019.
[22] S. Bojari and M. R. Eslahchi, "Two families of scaled three-term conjugate gradient methods with sufficient descent property for nonconvex optimization," *Numerical Algorithms*, vol. 83, no. 3, pp. 901–933, 2020.
[23] M. R. Eslahchi and S. Bojari, "Global convergence of a new sufficient descent spectral three-term conjugate gradient class for large-scale optimization," *Optimization Methods and Software*, vol. 37, no. 3, pp. 830–843, 2022.
[24] W. W. Hager and H. C. Zhang, "The limited memory conjugate gradient method," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2150–2168, 2013.
[25] T. G. Woldu, H. Zhang, X. Zhang, and Y. H. Fissuh, "A modified nonlinear conjugate gradient algorithm for large-scale nonsmooth convex optimization," *Journal of Optimization Theory and Applications*, vol. 185, no. 1, pp. 223–238, 2020.
[26] T. Zhao, H. Liu, and Z. Liu, "New subspace minimization conjugate gradient methods based on regularization model for unconstrained optimization," *Numerical Algorithms*, vol. 87, no. 4, pp. 1501–1534, 2021.
[27] I. Arzuka, M. A. Bakar, and W. Leong, "A scaled three-term conjugate gradient method for unconstrained optimization," *Journal of Inequalities and Applications*, vol. 2016, no. 325, pp. 1–16, 2016.
[28] W. Sun, H. Liu, and Z. Liu, "A class of accelerated subspace minimization conjugate gradient methods," *Journal of Optimization Theory and Applications*, vol. 190, no. 3, pp. 811–840, 2021.
[29] P. Huang, H. Z. Huang, Y. F. Li, and H. M. Qian, "An efficient and robust structural reliability analysis method with mixed variables based on hybrid conjugate gradient direction," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 8, pp. 1990–2004, 2021.
[30] J. Liu, Y. Feng, and L. Zou, "Some three-term conjugate gradient methods with the inexact line search condition," *Calcolo*, vol. 55, no. 2, p. 16, 2018.
[31] P. S. Stanimirović, B. Ivanov, H. Ma, and D. Mosić, "A survey of gradient methods for solving nonlinear optimization," *Electronic Research Archive*, vol. 28, no. 4, pp. 1573–1624, 2020.
[32] M. Al-Baali, "Descent property and global convergence of the Fletcher-Reeves method with inexact line search," *IMA Journal of Numerical Analysis*, vol. 5, no. 1, pp. 121–124, 1985.
[33] J. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.
[34] Y. -H Dai and L. -Z Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathematics and Optimization*, vol. 43, no. 1, pp. 87–101, 2001.
[35] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer Series in Operations Research, Springer, New York, NY, USA, 2006.
[36] Z. F. Dai, T. Li, and M. Yang, "Forecasting stock return volatility: the role of shrinkage approaches in a data-rich environment," *Journal of Forecasting*, vol. 41, no. 5, pp. 980–996, 2022.
[37] Z. F. Dai, H. Zhu, and X. Zhang, "Dynamic spillover effects and portfolio strategies between crude oil, gold and Chinese stock markets related to new energy vehicle," *Energy Economics*, vol. 109, no. 1, Article ID 105959, 2022.
[38] S. Bojari and M. Paripour, *Solving Large Scale Unconstrained Optimization Problems with an Efficient Conjugate Gradient Class*, Researchsquare, England, UK, 2023.
[39] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.
[40] E. Dolan and J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.
[41] W. W. Hager and H. C. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.
[42] N. Andrei, "Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization," *Bulletin of the Malaysian Mathematical Sciences Society*, vol. 34, no. 2, pp. 319–330, 2011.