Journal of Mathematical Modeling
Vol. -, No. -, 2022, pp. -. Research Article

JMM

A descent family of hybrid conjugate gradient methods with global convergence property for nonconvex functions

Mina Lotfi

Department of Applied Mathematics, Tarbiat Modares University, P.O.Box 14115-175, Tehran, Iran
Email(s): minalotfi@modares.ac.ir

Abstract. In this paper, we present a new hybrid conjugate gradient method for unconstrained optimization that possesses sufficient descent property independent of any line search. In our method, a convex combination of the Hestenes-Stiefel (HS) and the Fletcher-Reeves (FR) methods, is used as the conjugate parameter and the hybridization parameter is determined by minimizing the distance between the hybrid conjugate gradient direction and direction of the three-term HS method proposed by M. Li (*A family of three-term nonlinear conjugate gradient methods close to the memoryless BFGS method,* Optim. Lett. **12** (8) (2018) 1911–1927). Under some standard assumptions, the global convergence property on general functions is established. Numerical results on some test problems in the CUTEst library illustrate the efficiency and robustness of our proposed method in practice.

Keywords: Unconstrained optimization, conjugate gradient method, sufficient descent, least-squares, global convergence.
AMS Subject Classification 2010: 90C30, 65K05.

1 Introduction

We consider the following unconstrained optimization problem :

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

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function and its gradient is denoted by $g(x) = \nabla f(x)$. Conjugate gradient (CG) methods are among the most popular methods for solving (1), especially for large-scale problems [3, 8, 16, 26]. The iterative formula of a CG method is given by

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \quad k \geq 0, \tag{2}$$

where x_k is the current approximation to a solution, and $d_k \in \mathbb{R}^n$ is a search direction defined by

$$d_k = \begin{cases} -g_0, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k \geq 1, \end{cases} \tag{3}$$

in which the β_k (called here CG parameter) is a scalar, which distinguishes a CG method, and the step length $\alpha_k > 0$ is usually determined to satisfy the strong Wolfe line search conditions

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k, \tag{4}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k|, \tag{5}$$

where $0 < \delta < \frac{1}{2}$, $\delta < \sigma < 1$. Several famous formulas for β_k are Fletcher–Reeves (FR) method [11], the Dai–Yuan (DY) method [9], the Hestenes–Stiefel (HS) method [15], and the Polak–Ribière–Polyak (PRP) method [23, 24], with the following parameters, respectively,

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \qquad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k},$$

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \qquad \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$

where $y_k = g_{k+1} - g_k$ and $\|.\|$ stands for Euclidean norm. Although all these methods are equivalent when f is convex quadratic function and the step size α_k obtained by the exact line search, for general functions, they have different performances. It is well known that the FR and DY have strong convergence properties, but they may have modest computational performance. On the other hand, although the PRP and HS methods are computationally efficient, they may fail to converge for non-convex functions [29]. In recent years, much efforts has been made to find the methods having nice convergence properties and efficient numerical performance by hybridizing the CG methods. We refer the interested readers to [1, 2, 5, 6, 19, 21].

In an attempt to develop a modified HS method, Dai and Liao (DL) [7] presented a class of CG methods, in which the CG parameter is given by

$$\beta_k^{DL} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \lambda \frac{g_{k+1}^T s_k}{d_k^T y_k}, \quad \lambda \geq 0. \tag{6}$$

By using the truncation technique of [12], Dai and Liao also established their convergence for general functions. Although the DL method seldom generates uphill search direction in an actual computation, this search direction is not necessarily a descent one in theory. This motivated many researchers to make various modifications on the DL method, in order to achieve some descent properties, see [4, 10, 20, 29]. Note that, if set $\lambda = 0$ then β_k^{DL} reduces to the CG parameter proposed by Hestenes and Stiefel. In [14], Hager and Zhang (HZ) proposed a subclass of DL method that is named CG–DESCENT. In this method the CG parameter is computed by

$$\beta_k^{HZ} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \theta \frac{\|y_k\|^2 g_{k+1}^T d_k}{(d_k^T y_k)^2}, \qquad \theta > \frac{1}{4}. \tag{7}$$

They showed that the search direction in this method satisfies the sufficient descent condition $g_k^T d_k \leq (\frac{1}{4\theta} - 1)\|g_k\|^2$ and also, to guarantee the global convergence, β_k^{HZ} is updated as below

$$\beta_k^{HZ+} = \max\{\eta_k, \beta_k^{HZ}\}, \qquad \eta_k = -\frac{1}{\|d_k\| \min\{\eta, \|g_k\|\}}, \tag{8}$$

where $\eta \in (0,1)$. Numerical results showed that CG-DESCENT method outperforms many existing CG methods.

Recently, Li [18] introduced a descent class of three-term HS method in which the directions (shown by d_{k+1}^{THS}) are defined by

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

where

$$t = \min \left\{ 0.3,\ \max\{0, 1 - \frac{y_k^T s_k}{\|y_k\|^2}\} \right\}.$$

In [18], it is shown that the THS method satisfies the sufficient descent condition and it is also computationally superior to the CG-DESCENT method.

In this paper, motivated by the strong theoretical properties and computational efficiency of the THS method suggested by Li, we propose a new hybrid CG method. We consider the following convex combination of β_k^{HS} and β_k^{FR} methods:

$$\beta_k^{HCG} = (1-\theta_k)\beta_k^{HS} + \theta_k \beta_k^{FR}, \tag{10}$$

where the scalar $\theta_k \in [0,1]$ is called the hybridization parameter. Notice that, if $\theta_k < 0$, then we set $\theta_k = 0$, and if $\theta_k > 1$, then we set $\theta_k = 1$. So, β_k^{HCG} is used as our (CG parameter) in place of β_k in (3) and our proposed parameter θ_k is obtained by minimizing the distance between the CG search direction and search direction proposed by Li [18]. Under standard assumptions, the global convergence of the proposed method are proved for general functions. The new method is implemented in MATLAB environment and tested on some test problems from CUTEst collection. Numerical results illustrate efficiency and robustness of our proposed method in practice. The rest of this paper is organized as follows. In Section 2, we present details of the new CG method and its computational algorithm. Section 3 is devoted to establish the global convergence property of the new algorithm under standard assumptions. Numerical result is obtained on unconstrained optimization problems are reported in Section 4.

2 The new algorithm

This section is devoted to describe the structure of a new hybrid CG method. To this end, let d_{k+1}^{HCG}

$$d_{k+1}^{HCG} = -g_{k+1} + \beta_k^{HCG} d_k, \tag{11}$$

in which

$$\beta_k^{HCG} = (1-\theta_k)\beta_k^{HS} + \theta_k \beta_k^{FR}.$$

Then to derive our new hybridization parameter θ_k, we take advantage of theoretical properties and computational efficiency of the three-term HS (THS) method proposed by Li [18], and similar to the approach of [6], we suggest the following least-squares problem to compute the desired value of θ_k, satisfying

$$\arg\min_{\theta_k} \|d_{k+1}^{HCG} - d_{k+1}^{THS}\|.$$

Obviously, from (11) and (9) for d_{k+1}^{HCG} ; d_{k+1}^{THS} , and the fact that $\gamma = \frac{x^T y}{x^T x}$ is the optimal solution of $\arg\min_{\gamma} \|y - \gamma x\|$, we get the following new hybridization parameter

$$\theta_k^* = \frac{(g_{k+1}^T d_k)\|g_k\|^2 \left(\|y_k\|^2\|d_k\|^2 - t(d_k^T y_k)^2\right)}{(d_k^T y_k)\|d_k\|^2 E},$$

where t is the parameter used in (9) and $E = (g_{k+1}^T y_k)\|g_k\|^2 - \|g_{k+1}\|^2(d_k^T y_k)$. In order to avoid a division by zero, we define θ_k as follows:

$$\theta_k = \begin{cases} \theta_k^*, & \text{if } \theta_k^* \in [0,1], \text{ and } E \neq 0, \\ 1, & \theta_k^* > 1, \\ 0, & \theta_k^* < 0 \text{ or } E = 0. \end{cases} \tag{12}$$

Here, to increase computational efficiency and robustness of our method, we use the truncation technique in [25] and update β_k^{HCG} as below:

$$\beta_k^{HCG+} = (1-\theta_k)\beta_k^{HS+} + \theta_k \beta_k^{FR}. \tag{13}$$

where $\beta_k^{HS+} = \max\{0, \beta_k^{HS}\}$. Then to ensure the new search direction satisfies the descent condition, we employ the idea of the modified HS method of [27], and propose the following search direction

$$d_{k+1}^{THCG+} = -g_{k+1} + \beta_k^{HCG+} d_k - \beta_k^{HCG+} \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2} g_{k+1}, \qquad d_0 = -g_0. \tag{14}$$

An attractive feature of the proposed method is that the search direction d_{k+1}^{THCG+} always satisfies the following sufficient descent condition, which is independent of any line search.

$$g_{k+1}^T d_{k+1}^{THCG+} = -\|g_{k+1}\|^2. \tag{15}$$

Now, we rewrite the structure of our proposed method in the following algorithm:

Algorithm 1. New Hybrid Conjugate Gradient Algorithm

Step 0 : Consider constants $\varepsilon > 0$ and $0 < \delta < \frac{1}{2}, \delta < \sigma < 1$, choose an initial point $x_0 \in \mathbb{R}^n$ and set $k = 0$, $d_0 = -g_0$.

Step 1 : Stop if $\|g_k\|_\infty < \varepsilon$.

Step 2 : Determine the step length α_k such that satisfy the strong Wolfe line search conditions (4) and (5).

Step 3 : Let $x_{k+1} = x_k + \alpha_k d_k$.

Step 4 : Compute θ_k by (12).

Step 5 : Calculate β_k^{HCG+} by (13) and compute d_{k+1} by (14).

Step 6 : Set $k = k+1$, Go to Step 1.

---

3 Convergence analysis

In this section, we investigate the global convergence of the proposed method, under the following assumptions.

Assumption 3.1 The level set $\mathscr{L} = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is bounded, namely, there exists a constant $B > 0$ such that

$$\|x\| \leq B, \quad \forall x \in \mathscr{L}. \tag{16}$$

Assumption 3.2 In some neighborhood $\mathscr{N}$ of $\mathscr{L}$, f is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a positive constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \qquad \forall x, y \in \mathscr{N}. \tag{17}$$

Remark: Assumption 1 implies that there exists a positive constant M such that

$$\|g(x)\| \leq M, \quad \forall x \in \mathscr{L}. \tag{18}$$

To prove the global convergence of Algorithm 1, we need to refer to the following useful Lemma, which is known as Zoutendijk condition [30].

Lemma 1. *Suppose that the Assumptions 3.1 and 3.2 hold. Consider any method of the form* (2) *and* (3), *where* d_k *is a descent direction and* α_k *satisfies the Wolf line search conditions, then*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{19}$$

Lemma 2. *Let Assumptions 3.1 and 3.2 hold. If* $\{x_k\}$ *are generated by Algorithm 1, then we have*

$$\alpha_k \geq \frac{(\sigma - 1) g_k^T d_k}{L\|d_k\|^2}. \tag{20}$$

Proof. From (5), (17) and (2) we have

$$(\sigma - 1) g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq \|g_{k+1} - g_k\| \|d_k\| \leq L\alpha_k \|d_k\|^2,$$

which implies

$$\alpha_k \geq \frac{(\sigma - 1) g_k^T d_k}{L\|d_k\|^2}.$$

This completes the proof of lemma. □

Lemma 3. *Let Assumptions 3.1 and 3.2 hold, and* $\{x_k\}$ *be generated by Algorithm 1. If there exists a constant* $\mu > 0$ *such that that*

$$\|g_k\| \geq \mu, \quad \forall k \geq 0. \tag{21}$$

then there exists positive constant C such that

$$|\beta_k^{HCG+}| \leq C. \tag{22}$$

Proof. We get from the strong Wolfe condition (5) and (15), that

$$d_k^T y_k = d_k^T (g_{k+1} - g_k) \geq -(1-\sigma) d_k^T g_k = (1-\sigma)\|g_k\|^2. \tag{23}$$

From (17), (2), (18), (23), (21), (16) and the fact that $\theta_k^M \in [0,1]$, we conclude

$$\begin{aligned}
|\beta_k^{HCG+}| &= \left|(1-\theta_k)\beta_k^{HS+} + \theta_k \beta_k^{FR}\right|, \\
&= \left|(1-\theta_k)\max\{\frac{g_{k+1}^T y_k}{d_k^T y_k}, 0\} + \theta_k \frac{\|g_{k+1}\|^2}{\|g_k\|^2}\right|, \\
&\leq \frac{\|g_{k+1}\|\|y_k\|}{(1-\sigma)\|g_k\|^2} + \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \\
&\leq \frac{ML\|s_k\|}{(1-\sigma)\mu^2} + \frac{M^2}{\mu^2}, \\
&\leq \frac{ML(\|x_{k+1}\| + \|x_k\|)}{(1-\sigma)\mu^2} + \frac{M^2}{\mu^2}, \\
&\leq \frac{2MLB + (1-\sigma)M^2}{\mu^2(1-\sigma)}.
\end{aligned} \tag{24}$$

By setting $C = \frac{2MLB+(1-\sigma)M^2}{\mu^2(1-\sigma)}$, we have $|\beta_k^{HCG+}| \leq C$. The proof is completed. □

In order to prove the global convergence for general function, we assume that the step lengths are bounded below by a positive constant. The following theorem, which is similar to Theorem 3.2 in [26], establishes global convergence property of the proposed method.

Theorem 1. *Suppose that Assumptions 3.1 and 3.2 hold. If there exists a positive constant α^* such that $\alpha_k \geq \alpha^* > 0$, for all $k \geq 0$, and $\{x_k\}$ are generated by Algorithm 1, then either $\|g_k\| = 0$ for some k or*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{25}$$

Proof. Suppose that (25) does not hold, i.e., there exists a constant $\mu > 0$ such that

$$\|g_k\| \geq \mu, \quad \forall k \geq 0. \tag{26}$$

By (2) and (16), we conclude that

$$\|d_k\| = \left\|\frac{s_k}{\alpha_k}\right\| \leq \frac{\|x_{k+1}\| + \|x_k\|}{\alpha_k} \leq \frac{2B}{\alpha^*} = \gamma, \tag{27}$$

where $\gamma = 2B/\alpha^*$. It follows from (15), (26) and (27) that

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \geq \frac{\mu^4}{\gamma^2} \sum_{k=0}^{\infty} 1 = \infty,$$

which contradicts (19). So, the proof is complete. □

4 Numerical results

This section reports the computational results obtained from the implementation of the Algorithm 1, denoted by THCG+" and some other existing algorithms on some test problems. All algorithms were written in MATLAB and ran on a PC (CPU 2.5 GHZ, RAM 3.8GB) with Linux operating system. The test problems were taken from the CUTEst library [13]. The dimensions of the problems range from at least 50 and up to 10,000. Table 3 lists those problems with their dimensions. For all algorithms, we used the strong Wolfe line search conditions with $\eta = 0.01$, $\sigma = 0.1$ and using Algorithm 3.5 in [22]. Here, for the first iteration we set the initial trial value $\alpha_{0,0} = 1$, and for the subsequent iterations we set it to

$$\alpha_{k+1,0} = \mu \frac{|s_k^T d_{k+1}|}{\|d_{k+1}\|^2} + (1-\mu)\frac{\|s_k\|}{\|d_{k+1}\|},$$

where $\mu = 0.5$, [5]. All algorithms were stopped when the number of iterations exceeded 10000 or $\|g_k\| \le 10^{-6}$.

We compare our results with those of the following CG methods:

- HZ+": Algorithm proposed in [14];
- ADHCG1": Algorithm ADHCG in [19] which the scaling parameter is defined by $\theta_k = \min\{\frac{s_k^T y_k}{\|s_k\|^2}, 1\}$;
- IFD": Algorithm proposed in [17] with $\beta_k = \beta_k^{IFD}$;
- THS+": method proposed in [18];
- TTHS": method proposed in [28];
- HCGB": method proposed in [6].

We utilized the performance profile of Dolan and More [10] (in log_2 scale) to present numerical results of the algorithms. Given a set of problems $\mathscr{P}$ and a set of solvers S, we define $i_{p,s}$ as the number of iterations required to solve problem p by solver s. The performance ratio is given by

$$r_{p,s} = \frac{i_{p,s}}{\min\{i_{p,s} | s \in S\}}.$$

Then, the performance profile is defined by

$$P_s(w) = \frac{\{size \;\; p \in \mathscr{P} | \;\; r_{p,s} \le w\}}{size \;\; \mathscr{P}}.$$

These algorithms are compared based on the number of iterations (n_i), the number of function evaluations (n_f) and the number of gradient evaluations (n_g). In Tables 1 and 2 we present the percentage of the test problems that are solved by each algorithm with the lowest value of n_i, n_f, and n_g.

Figs. 1, 2 and 3 present the performance profiles of THCG+, HZ+, ADHCG1 and TTHS according to the number of iterations, the number of function evaluations, the number of gradient evaluations, respectively. As these figures illustrate, it can be seen that the method THCG+ performs better than the three other methods.

In Fig. 4, we compare the performance profile of THCG+, THS+, HCGB and IFD methods based on number of iterations. From Fig. 5, it is concluded that THCG+ is more efficient than others, with respect to the number of function evaluations. We observed that THCG+ method solves about 71% of the test problems with the least number of function evaluations. Fig. 6 shows that THCG+ method performs slightly superior to THS+, HCGB and IFD methods with respect to the number of gradient evaluations. Numerical results show that MDL method solves about 74% of the test problems with relatively least number gradient evaluations.

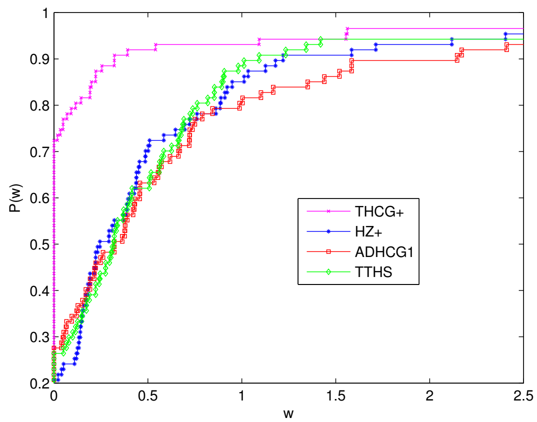

Figure 1: Performance profile of methods in terms of number of iterations.

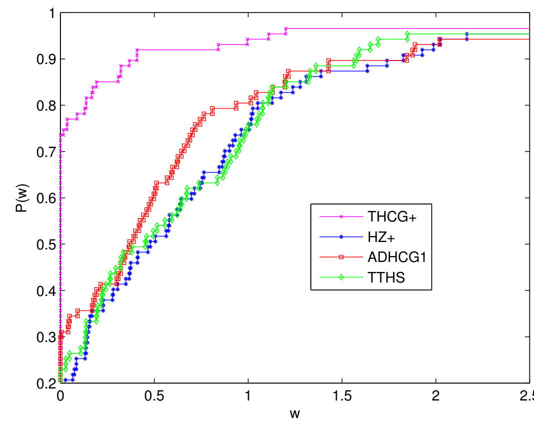

Figure 2: Performance profile of methods in terms of number of function evaluations.

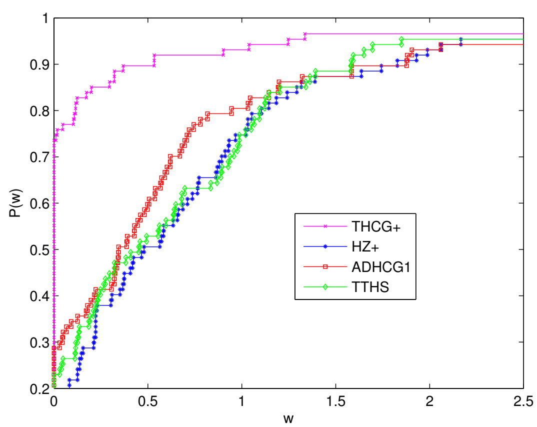

Figure 3: Performance profile of methods in terms of number of gradient evaluations.

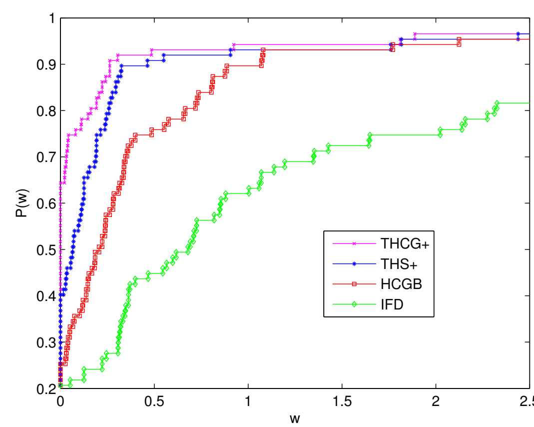

Figure 4: Performance profile of methods in terms of number of iterations.

5 Conclusions

In this paper, we propose a new hybrid conjugate gradient method for unconstrained optimization. As a remarkable feature, the search directions of the proposed method satisfy the sufficient descent condition, independently of line searches. Under some standard assumptions, the global convergence of the proposed method is proved for general functions. Numerical comparisons on some test problems indicate the efficiency and robustness of the proposed method in practice.

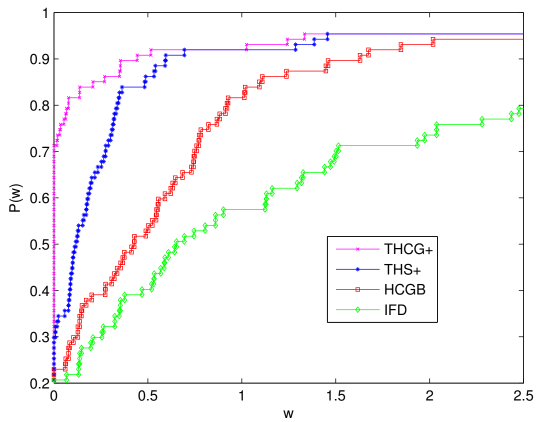

Figure 5: Performance profile of methods in terms of number of function evaluations.

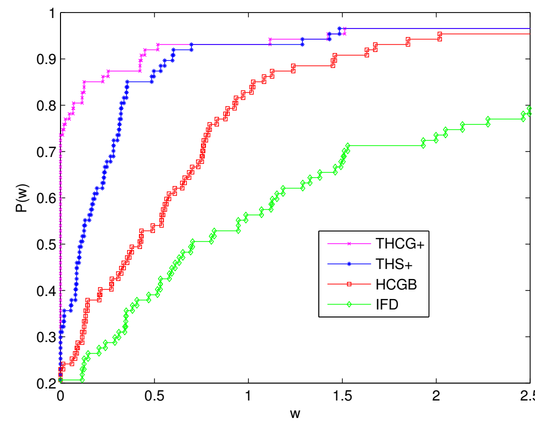

Figure 6: Performance profile of methods in terms of number of gradient evaluations.

Table 1: Percentage of the test problems that each method solves with the lowest value of n_i, n_f and n_g.

	$THCG+(\%)$	$HZ+(\%)$	$ADHCG1(\%)$	$TTHS(\%)$
n_i	72	20	28	26
n_f	74	18	30	23
n_g	74	16	29	23

Table 2: Percentage of the test problems that each method solves with the lowest value of n_i, n_f and n_g.

	$THCG+(\%)$	$THS+(\%)$	$HCGB(\%)$	$IFD(\%)$
n_i	64	40	25	21
n_f	71	30	23	21
n_g	74	31	23	21

References

[1] N. Andrei, *Numerical comparison of conjugate gradient algorithms for unconstrained optimization*, Stud. Inform. Control. **16** (2007) 333–352.

[2] N. Andrei, *Hybrid conjugate gradient algorithm for unconstrained optimization*, J. Optim. Theory Appl. **141** (2009) 249–264.

[3] N. Andrei, *An adaptive conjugate gradient algorithm for large-scale unconstrained optimization*, Comput. Appl. Math. **292** (2016) 83–91.

Table 3: The test problems and their dimensions.

Name	Dim	Name	Dim	Name	Dim
ARGLINA	200	ARGLINA	100	BDEXP	1000
BDEXP	5000	BQPGABIM	50	BIGSB1	100
BIGSB1	1000	BRYBND	50	BRYBND	100
CURLY10	10000	CURLY20	10000	CURLY30	10000
DIXMAANA	3000	DIXMAANA	9000	DIXMAANB	3000
DIXMAANB	9000	DIXMAANC	3000	DIXMAANC	9000
DIXMAAND	3000	DIXMAAND	9000	DIXMAANE	3000
DIXMAANE	9000	DIXMAANF	3000	DIXMAANF	9000
DIXMAANG	3000	DIXMAANG	9000	DIXMAANH	3000
DIXMAANH	9000	DIXMAANI	3000	DIXMAANI	9000
DIXMAANJ	3000	DIXMAANJ	9000	DIXMAANK	3000
DIXMAANK	9000	DIXMAANL	3000	DIXMAANL	9000
DIXMAANM	3000	DIXMAANM	9000	DIXMAANN	3000
DIXMAANN	9000	DIXMAANO	3000	DIXMAANO	9000
DIXMAANP	3000	DIXMAANP	9000	DIXON3DQ	100
DIXON3DQ	1000	DECONVU	63	DEGENGP	50
DQDRTIC	1000	DQDRTIC	5000	EG2	1000
EIGENALS	110	EIGENBLS	110	FMINSRF2	5625
FMINSRF2	10000	FMINSURF	5625	FMINSURF	10000
FLETCHCR	100	FLETCHCR	1000	GILBERT	100
GILBERT	200	LMINSURF	5625	LMINSURF	10000
LISWET1	2002	MANCINO	50	MANCINO	100
MSQRTALS	100	MSQRTBLS	100	NLMSURF	1024
NLMSURF	5625	NONDIA	100	NONDIA	1000
QUARTC	1000	QUARTC	5000	SPARSQUR	5000
SPARSQUR	10000	SPMSRTLS	1000	SPMSRTLS	4999
SROSENBR	5000	SROSENBR	10000	TOINTGSS	5000
TOINTGSS	10000	TRIDIA	5000	TRIDIA	10000
VAREIGVL	100	WOOD	1000	WOOD	10000

[4] Z. Aminifard, S. Babaie-Kafaki, *An optimal parameter choice for the Dai-Liao family of conjugate gradient methods by avoiding a direction of the maximum magnification by the search direction matrix*, 4OR. **17** (2018) 1–14.

[5] S. Babaie-Kafaki, M. Fatemi, N. Mahdavi-Amiri, *Two effective hybrid conjugate gradient algorithms based on modified BFGS updates*, Numer. Algor. **58** (2011) 315–331.

[6] S. Babaie-Kafaki, R. Ghanbari, *A hybridization of the Polak-Ribière-Polyak and Fletcher-Reeves conjugate gradient methods*, Numer. Algor. **68(3)** (2015) 481–495.

[7] Y. Dai, L. Liao, *New conjugate conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim. 43 (2001) 87–101.

[8] Y.H. Dai, J.Y. Han, G.H. Liu, D.F. Sun, H.X. Yin, Y.X. Yuan, *Convergence properties of nonlinear conjugate gradient methods,* SIAM J. Optim. **10(2)** (1999) 348–358.

[9] Y.H. Dai, Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property,* SIAM J. Optim. **10** (1999) 177–182.

[10] E. Dolan, J. Moré, *Benchmarking optimization software with performance profiles,* Math. **91** (2002) 201–213.

[11] R. Fletcher, C.M. Reeves, *Function minimization by conjugate gradients,* Comput. J. **7(2)** (1964) 149–154.

[12] J.C. Gilbert, J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization,* SIAM J. Optim. **2** (1992) 21–42.

[13] N.I.M. Gould, D. Orban, P.L. Toint, *CUTEst : a constrained and unconstrained testing environment with safe threads for mathematical optimization,* Comput. Optim. Appl. **60(3)** (2015) 545–557.

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

[15] M.R. Hestenes, E.L. Stiefel, *Methods of conjugate gradients for solving linear systems,* J. Res. Natl. Bur. Stand. (1952) 409–432.

[16] J. Jian, Q. Chen, X. Jiang, Y. Zeng, J. Yin, *A new spectral conjugate gradient method for large-scale unconstrained optimization,* Optim. Methods Softw. **32** (3) (2017) 503515.

[17] X. Jiang, J. Jian, *Improved Fletcher-Reeves and Dai-Yuan conjugate gradient methods with the strong Wolfe line search,* J. Comput. Appl. Math. **348** (2019) 525–534.

[18] M. Li, *A family of three-term nonlinear conjugate gradient methods close to the memoryless BFGS method,* Optim. Lett. **12 (8)** (2018) 1911–1927.

[19] I.E. Livieris, V. Tampakas, P. Pintelas, *A descent hybrid conjugate gradient method based on the memoryless BFGS update,* Numer. Algor. **79 (4)** (2018) 1169–1185.

[20] M. Lotfi, S.M. Hosseini, *An efficient Dai-Liao type conjugate gradient method by reformulating the CG parameter in the search direction equation,* J. Comput. Appl. Math. (2020) 371:112708

[21] M. Lotfi, S.M. Hosseini, *An efficient hybrid conjugate gradient method with sufficient descent property for unconstrained optimization,* Optim. Methods Softw., 2021, https://doi.org/10.1080/10556788.2021.1977808.

[22] J. Nocedal, S. Wright, *Numerical Optimization*, 2nd edition springer, New York, 2006.

[23] E. Polak, G. Ribière, *Note sur la convergence des mthodes de directions conjuguèes,* Rev. Fr. Inform. Rech. Oper. **16** (1969) 35–43.

[24] E.T. Polyak, *The conjugate gradient method in extreme problems*, USSR Comp. Math. Math. Phys. **9** (1969) 94–112.

[25] M.J.D. Powell, *Restart procedures for the conjugate gradient method,* Math. Program. **12(2)** (1977) 241–254.

[26] G.L. Yuan, *Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems,* Optim. Lett. **3** (2009) 1121.

[27] L. Zhang, W. Zhou, D. Li, *Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search,* Numer. Math. **104** (2006) 561–572.

[28] L. Zhang, W. Zhou, DH. Li, *Some descent three-term conjugate gradient methods and their global convergence.* Optim. Methods Softw. **2(4)**, (2007) 697–711.

[29] Y. Zheng, B. Zheng, *Two new Dai-Liao-type conjugate gradient methods for unconstrained optimization problems,* J. Optim. Theory Appl. **175** (2017) 502–509..

[30] G. Zoutendijk, *Nonlinear programming, computational methods*, In: Abadie, J. (ed.) Integer and Nonlinear Programming, North-Holland, Amsterdam (1970) 37–86.