

An Efficient FR-CD-Like Algorithm for Unconstrained Optimization

Jitsupa Deepho^a, Auwal Bala Abubakar^{b,c,f*}, Maulana Malik^d, Abdulkarim Hassan Ibrahim^e, Aliyu Ibrahim Kiri^b

^aFaculty of Science, Energy and Environment, King Mongkut's University of Technology North Bangkok, 19 Moo 11, Nonglalek, Bankhai, Rayong 21120 Thailand

^bNumerical optimization research group, Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano. Kano, Nigeria

^cDepartment of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa-0204, South Africa

^dDepartment of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok 16424, Indonesia

^eKMUTTFixed Point Research Laboratory, Room SCL 802 Fixed Point Laboratory, Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand

^fFaculty of Education and Arts, Sohar University, Sohar 311, Oman [2mm]

E-mail: jitsupa.d@sciee.kmutnb.ac.th

E-mail: ababubakar.mth@buk.edu.ng

E-mail: m.malik@sci.ui.ac.id

E-mail: ibrahimkarym@gmail.com

E-mail: sssaly2014@gmail.com

ABSTRACT. A hybrid conjugate gradient (CG) method is proposed for solving unconstrained optimization problems. The direction of the method

*Corresponding Author

Received 20 May 2021; Accepted 18 October 2022
©2026 Academic Center for Education, Culture and Research TMU

is a combination of a three-term conjugate descent (CD) and Fletcher-Reeves (FR) CG directions. Also, it is close to the direction of the memoryless Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. In addition, under the Wolfe-type line search, the global convergence of the method is established. Numerical experiments are conducted on some benchmark test problems and the results are reported to show the efficiency of the propose method compared with some existing methods.

Keywords: Unconstrained optimization, Gonjugate gradient method, Global convergence.

2000 Mathematics subject classification: 65K10, 90C52, 90C26.

1. INTRODUCTION

In this work, we consider solving the following problem:

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and its gradient $\vartheta(x) = \nabla f(x)$ is Lipschitz continuous. This problem is of interest as it has vast applications in many area of research.

Though nonlinear conjugate gradient (CG) methods are not the fastest optimization methods but recently are the most popular among mathematicians and engineers for solving (1.1). This is because they are Hessian free, easy to implement and have low storage [14]. Based on these, a significant number of CG methods were proposed, see [24, 11, 3, 12, 13, 6, 1, 2, 7] and references therein.

The nonlinear CG method for solving (1.1) has the following procedure: Given an initial guess $\{x_k\}$, the next point is obtained as

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots, \quad (1.2)$$

where d_k is defined as

$$d_k := \begin{cases} -\vartheta_k, & \text{if } k = 0, \\ -\vartheta_k + \beta_k d_{k-1}, & \text{if } k > 0, \end{cases} \quad (1.3)$$

and $\vartheta_k := \vartheta(x_k)$. $\alpha_k > 0$ is called the step size and β_k the CG parameter. To ensure a decrease in the function value at each iteration, d_k is required to satisfy

$$\vartheta_k^T d_k \leq -c \|\vartheta_k\|^2, \quad c > 0. \quad (1.4)$$

Among the classical CG parameters are the Fletcher-Reeves (FR) [10], Hestenes-Stiefel (HS)[15], conjugate descent (CD) [9]. These parameters are defined as

$$\beta_k^{FR} = \frac{\|\vartheta_k\|^2}{\|\vartheta_{k-1}\|^2}, \quad \beta_k^{HS} = \frac{\vartheta_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{CD} = \frac{\|\vartheta_k\|^2}{-d_{k-1}^T \vartheta_{k-1}}, \quad (1.5)$$

where $y_{k-1} = \vartheta_k - \vartheta_{k-1}$ and $\|\cdot\|$ is the Euclidean norm.

In order to build upon the successes recorded by the classical CG methods, three-term and hybrid CG-type methods were introduced. For example, Andrei [5] proposed a three-term CG method for solving (1.1). The method is a modified version of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula. In addition, the convergence of the method was established under some assumptions. Liu and Li [20] proposed a hybrid CG method for solving (1.1). The method is a combination of the Liu-Storey (LS) and Dai-Yuan (DY) CG methods. Under the Strong Wolfe line search, the global convergence of the hybrid method was established. Min Li [18] proposed a three-term Polak-Ribière-Polyak (PRP) CG method with the search direction close to the direction of the memoryless BFGS quasi-Newton method. The method reduces to the standard PRP method when an exact line search is used. Numerical results show that the method is efficient and competitive. Also, Li [17] proposed a CG method with a search direction close to that of the memoryless BFGS method. Global convergence for strongly convex functions and nonconvex functions was established under the strong Wolfe line search.

Motivated by the works in [17, 18], we present in this paper, a hybrid method for solving (1.1). The direction is a combination of the three-term CD and FR directions. Moreso, the hybrid direction is close to the direction of the memoryless BFGS method. It is worth mentioning that the method proposed in [18] is a special case of the hybrid method. The direction satisfies the descent condition (1.4) and has the trust region property. The global convergence is proved under the Wolfe condition. Numerical results are also reported to depict the efficiency of the hybrid method over some existing methods. The rest of the paper is organized as follows. In the next section, the hybrid method is derived together with the convergence analysis of the algorithm. In Section 3, we provide some numerical results to strengthen the theoretical results.

2. ALGORITHM AND THEORETICAL RESULTS

A three-term CG method for solving (1.1) compute the search direction as follows:

$$d_0 := -\vartheta_0, \quad d_k := -\vartheta_k + \beta_k d_{k-1} + \gamma_k \vartheta_k, \quad k \geq 1, \quad (2.1)$$

where β_k, γ_k are parameters. It can be seen that, the three-term CG methods reduces to the classical ones when $\gamma_k = 0$.

Next, we will recall the memoryless BFGS method proposed by Nocedal [21] and Shanno [22], where the search direction can be written as

$$d_k^{BFGS} := -Q_k \vartheta_k,$$

$$Q_k = - \left(I - \frac{s_{k-1} y_{k-1}^T}{s_{k-1}^T y_{k-1}} - \frac{y_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}} + \frac{s_{k-1} y_{k-1}^T y_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}} + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}} \right),$$

$s_{k-1} = x_k - x_{k-1} = \alpha_{k-1} d_{k-1}$ and I is the identity matrix. After simplification, d_k^{BFGS} can be rewritten as

$$d_k^{BFGS} := -\vartheta_k + \left(\beta_k^{HS} - \frac{\|y_{k-1}\|^2 \vartheta_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2} \right) d_{k-1} + \frac{\vartheta_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} (y_{k-1} - s_{k-1}). \quad (2.2)$$

Replacing β_k^{HS} with β_k^{CD} and $\frac{\|y_{k-1}\|^2 \vartheta_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}$ with $\frac{\|\vartheta_k\|^2 \vartheta_k^T d_{k-1}}{(-\vartheta_{k-1}^T d_{k-1})^2}$ in (2.2), we define a three-term search direction as

$$d_k^{TTCD} := -\vartheta_k + \left(\beta_k^{CD} - \frac{\|\vartheta_k\|^2 \vartheta_k^T d_{k-1}}{(-\vartheta_{k-1}^T d_{k-1})^2} \right) d_{k-1} - t_k \frac{\vartheta_k^T d_{k-1}}{-\vartheta_{k-1}^T d_{k-1}} \vartheta_k. \quad (2.3)$$

Again, replacing β_k^{HS} with β_k^{FR} and $\frac{\|y_{k-1}\|^2 \vartheta_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}$ with $\frac{\|\vartheta_k\|^2 \vartheta_k^T d_{k-1}}{\|\vartheta_{k-1}\|^4}$ in (2.2), we define a three-term search direction as

$$d_k^{TFR} := -\vartheta_k + \left(\beta_k^{FR} - \frac{\|\vartheta_k\|^2 \vartheta_k^T d_{k-1}}{\|\vartheta_{k-1}\|^4} \right) d_{k-1} - t_k \frac{\vartheta_k^T d_{k-1}}{\|\vartheta_{k-1}\|^2} \vartheta_k. \quad (2.4)$$

To find the parameter t_k , we require the solution of the univariate minimal problem

$$\min_{t \in \mathbf{R}} \|(y_{k-1} - s_{k-1}) - t \vartheta_k\|^2.$$

Let $A_k = (y_{k-1} - s_{k-1}) - t \vartheta_k$, then

$$\begin{aligned} A_k A_k^T &= [(y_{k-1} - s_{k-1}) - t \vartheta_k][(y_{k-1} - s_{k-1}) - t \vartheta_k]^T \\ &= [(y_{k-1} - s_{k-1}) - t \vartheta_k][(y_{k-1} - s_{k-1})^T - t \vartheta_k^T] \\ &= t^2 \vartheta_k \vartheta_k^T - t[(y_{k-1} - s_{k-1}) \vartheta_k^T + \vartheta_k (y_{k-1} - s_{k-1})^T] + (y_{k-1} - s_{k-1})(y_{k-1} - s_{k-1})^T. \end{aligned}$$

Letting $E_k = y_{k-1} - s_{k-1}$, then

$$A_k A_k^T = t^2 \vartheta_k \vartheta_k^T - t[E_k \vartheta_k^T + \vartheta_k E_k^T] + E_k E_k^T$$

and

$$\begin{aligned} \text{tr}(A_k A_k^T) &= t^2 \|\vartheta_k\|^2 - t[\text{tr}(E_k \vartheta_k^T) + \text{tr}(\vartheta_k E_k^T)] + \|E_k\|^2 \\ &= t^2 \|\vartheta_k\|^2 - 2t \vartheta_k^T E_k + \|E_k\|^2. \end{aligned}$$

Differentiating the above with respect to t and equating to zero, we have

$$2t \|\vartheta_k\|^2 - 2 \vartheta_k^T E_k = 0,$$

which implies

$$t = \frac{\vartheta_k^T (y_{k-1} - s_{k-1})}{\|\vartheta_k\|^2}. \quad (2.5)$$

Hence, we select t_k as

$$t_k := \min \left\{ \bar{t}, \max \left\{ 0, \frac{\vartheta_k^T (y_{k-1} - s_{k-1})}{\|\vartheta_k\|^2} \right\} \right\}, \quad (2.6)$$

which implies $0 \leq t_k \leq \bar{t} < 1$.

Motivated by the three-term CG directions defined in (2.3) and (2.4), we propose a hybrid three-term CG based algorithm for solving (1.1), where the search direction is defined as

$$d_0 := -\vartheta_0, \quad d_k := -\vartheta_k + \beta_k^{TTH} d_{k-1} + \gamma_k \vartheta_k, \quad k \geq 1, \quad (2.7)$$

where

$$\beta_k^{TTH} := \frac{\|\vartheta_k\|^2}{w_k} - \frac{\|\vartheta_k\|^2 \vartheta_k^T d_{k-1}}{w_k^2}, \quad \gamma_k := -t_k \frac{\vartheta_k^T d_{k-1}}{w_k} \quad (2.8)$$

and

$$w_k := \max \{ \lambda \|d_{k-1}\| \|\vartheta_k\|, -d_{k-1}^T \vartheta_{k-1}, \|\vartheta_{k-1}\|^2 \}, \quad \lambda > 0. \quad (2.9)$$

Remark 2.1. Note that, relation (2.9) is carefully defined such that the search direction becomes a three-term hybrid of FR and CD directions and to also possess the sufficient descent condition (see Lemma 2.2) independent of the line search.

Lemma 2.2. *The search direction d_k defined by (2.7) satisfy (1.4) with $c = \frac{3}{4}$.*

Proof. Multiplying both sides of (2.7) with ϑ_k^T , we have

$$\begin{aligned} \vartheta_k^T d_k &= -\|\vartheta_k\|^2 + \frac{\|\vartheta_k\|^2}{w_k} \vartheta_k^T d_{k-1} - \frac{\|\vartheta_k\|^2}{w_k^2} (\vartheta_k^T d_{k-1})^2 - t_k \frac{\|\vartheta_k\|^2}{w_k} \vartheta_k^T d_{k-1} \\ &\leq -\|\vartheta_k\|^2 + (1 - t_k) \frac{\|\vartheta_k\|^2}{w_k} \vartheta_k^T d_{k-1} - \frac{\|\vartheta_k\|^2}{w_k^2} (\vartheta_k^T d_{k-1})^2 \\ &= -\|\vartheta_k\|^2 + 2 \left(\frac{(1 - t_k)}{2} \vartheta_k^T \right) \frac{\vartheta_k}{w_k} \vartheta_k^T d_{k-1} - \frac{\|\vartheta_k\|^2}{w_k^2} (\vartheta_k^T d_{k-1})^2 \\ &\leq -\|\vartheta_k\|^2 + \frac{(1 - t_k)^2}{4} \|\vartheta_k\|^2 + \frac{\|\vartheta_k\|^2}{w_k^2} (\vartheta_k^T d_{k-1})^2 - \frac{\|\vartheta_k\|^2}{w_k^2} (\vartheta_k^T d_{k-1})^2 \\ &= -\|\vartheta_k\|^2 + \frac{(1 - t_k)^2}{4} \|\vartheta_k\|^2 \\ &= - \left(1 - \frac{(1 - t_k)^2}{4} \right) \|\vartheta_k\|^2 \\ &\leq -\frac{3}{4} \|\vartheta_k\|^2. \end{aligned}$$

□

Next, we will turn our attention to establishing the convergence of the proposed scheme by first considering the standard Wolfe line search conditions [23],

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \theta \alpha_k \vartheta_k^T d_k, \quad (2.10)$$

$$\vartheta(x_k + \alpha_k d_k)^T d_k \geq \sigma \vartheta_k^T d_k \quad (2.11)$$

where $0 < \theta < \sigma < 1$. In addition, we will assume that

Assumption 2.3. *The level set $\mathcal{H} = \{x : f(x) \leq f(x_0)\}$ is bounded.*

Assumption 2.4. *Suppose $\mathcal{H}$ is some neighborhood of $\mathcal{L}$, then f is continuously differentiable and its gradient Lipschitz continuous on $\mathcal{H}$. That is, we can find $L > 0$ such that for all x*

$$\|\vartheta(x) - \vartheta(\bar{x})\| \leq L\|x - \bar{x}\|, \quad \bar{x} \in \mathcal{H}. \quad (2.12)$$

We will now prove the convergence result, however, we require the following result.

Lemma 2.5. *Let $\{d_k\}$ be defined by (2.7)-(2.9), then there is $\kappa > 0$ such that $\|d_k\| \leq \kappa \|\vartheta_k\|$.*

Proof. From (2.8),

$$\begin{aligned} |\beta_k^{TTH}| &= \left| \frac{\|\vartheta_k\|^2}{w_k} - \frac{\|\vartheta_k\|^2 \vartheta_k^T d_{k-1}}{w_k^2} \right| \\ &\leq \frac{\|\vartheta_k\|^2}{\lambda \|d_{k-1}\| \|\vartheta_k\|} + \frac{\|\vartheta_k\|^3 \|d_{k-1}\|}{(\lambda \|d_{k-1}\| \|\vartheta_k\|)^2} \\ &= \left(\frac{1}{\lambda} + \frac{1}{\lambda^2} \right) \frac{\|\vartheta_k\|}{\|d_{k-1}\|}. \end{aligned} \quad (2.13)$$

Also,

$$\begin{aligned} |\gamma_k| &= \left| -t_k \frac{\vartheta_k^T d_{k-1}}{w_k} \right| \\ &= t_k \left| \frac{\vartheta_k^T d_{k-1}}{w_k} \right| \\ &\leq \bar{t} \frac{\|\vartheta_k\| \|d_{k-1}\|}{w_k} \\ &\leq \bar{t} \frac{\|\vartheta_k\| \|d_{k-1}\|}{\lambda \|d_{k-1}\| \|\vartheta_k\|} \\ &= \frac{\bar{t}}{\lambda}. \end{aligned} \quad (2.14)$$

Therefore, from (2.7), (2.8), (2.9), (2.13) and (2.14), we have

$$\begin{aligned} \|d_k\| &= \left\| -\vartheta_k + \beta_k^{TTH} d_{k-1} + \gamma_k \vartheta_k \right\| \\ &\leq \|\vartheta_k\| + |\beta_k| \|d_{k-1}\| + |\gamma_k| \|\vartheta_k\| \\ &\leq \|\vartheta_k\| + \left(\frac{1}{\lambda} + \frac{1}{\lambda^2} \right) \frac{\|\vartheta_k\|}{\|d_{k-1}\|} \|d_{k-1}\| + \frac{\bar{t}}{\lambda} \|\vartheta_k\| \\ &= \left(1 + \frac{1 + \bar{t}}{\lambda} + \frac{1}{\lambda^2} \right) \|\vartheta_k\|. \end{aligned} \quad (2.15)$$

Letting $\kappa = \left(1 + \frac{1+\bar{\ell}}{\lambda} + \frac{1}{\lambda^2}\right)$, we have

$$\|d_k\| \leq \kappa \|\vartheta_k\|. \quad (2.16)$$

Theorem 2.6. *Let Assumption 2.3-2.4 hold and the sequences $\{x_k\}$ and $\{d_k\}$ are defined by (1.2) and (2.7), respectively. If $\{\alpha_k\}$ satisfies (2.10)-(2.11), then*

$$\lim_{k \rightarrow \infty} \|\vartheta_k\| = 0. \quad (2.17)$$

Proof. Using (1.4) and (2.10), we have

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \theta \alpha_k \vartheta_k^T d_k \\ &\leq f(x_k) - \frac{3}{4} \theta \alpha_k \|\vartheta_k\|^2, \end{aligned}$$

which implies

$$\frac{3}{4} \theta \alpha_k \|\vartheta_k\|^2 \leq f(x_k) - f(x_k + \alpha_k d_k).$$

Taking the infinite sum of the above inequality together with Assumption 2.3, we get

$$\frac{3}{4} \theta \sum_{k=0}^{\infty} \alpha_k \|\vartheta_k\|^2 < \infty,$$

which implies

$$\alpha_k \|\vartheta_k\|^2 \rightarrow 0, k \rightarrow \infty. \quad (2.18)$$

Also from (1.4), (2.11) and Assumption 2.4, we have

$$\begin{aligned} L \|\alpha_k d_k\| \|d_k\| &\geq \|\vartheta(x_k + \alpha_k d_k) - \vartheta_k\| \|d_k\| \\ &\geq [h(x_k + \alpha_k d_k) - \vartheta_k]^T d_k \\ &\geq -(1 - \sigma) \vartheta_k^T d_k \\ &\geq \frac{3}{4} (1 - \sigma) \|\vartheta_k\|^2. \end{aligned}$$

So, we have from (2.16) and the above inequality that

$$\alpha_k \geq \frac{3(1 - \sigma) \|\vartheta_k\|^2}{4L \|d_k\|^2} \geq \frac{3(1 - \sigma)}{4L \kappa^2}. \quad (2.19)$$

Combining (2.18) and (2.19), we get the desired result (2.17). $\square$

3. NUMERICAL EXAMPLES

In this section, we compare the computational performance of the proposed algorithm TTFRCD for solving unconstrained optimization problems with five existing methods. Two methods are chosen from classical methods, namely the Fletcher-Reeves (FR) proposed by Fletcher and Reeves [10] and conjugate descent algorithm (CD) by Fletcher [9] with the search direction in the form (1.3). The parameter β_k^{FR} is for FR method and β_k^{CD} for CD method. Meanwhile, the other methods are from three-term type method as follows:

- TTRMIL: The three-term Rivaie-Mustafa-Ismail-Leong method proposed by in [19] in the form $d_0 = -\vartheta_0$ and for $k > 0$,

$$d_k = -\vartheta_k + \frac{\vartheta_k^T y_{k-1}}{\|d_{k-1}\|^2} d_{k-1} - \frac{\vartheta_k^T d_{k-1}}{\|d_{k-1}\|^2} y_{k-1};$$

- NHS+: The three-term Hager-Zhang-Dai-Kou method proposed by Min Li [17] in the form $d_0 = -\vartheta_0$ and for $k > 0$,

$$d_k := \begin{cases} -\vartheta_k + \beta_k d_{k-1} + \lambda_k y_{k-1}, & \text{if } \beta_k > 0, \\ -\vartheta_k, & \text{otherwise,} \end{cases}$$

where

$$\beta_k = \beta_k^{HZDK} = \frac{\vartheta_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{\|y_{k-1}\|^2 \vartheta_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2},$$

$$\lambda_k = \lambda_k^{HZDK} = t_k \frac{\vartheta_k^T d_{k-1}}{d_{k-1}^T y_{k-1}},$$

$0 \leq t_k \leq \bar{t} < 1$, and $\bar{t}$ is a constant.

- NPRP+: The three-term modified PRP method proposed by Min Li [18] in the same form with NHS+ method where

$$\beta_k = \beta_k^{DPRP1} = \frac{\vartheta_k^T y_{k-1}}{\|\vartheta_{k-1}\|^2} - \frac{\|y_{k-1}\|^2 \vartheta_k^T d_{k-1}}{\|\vartheta_{k-1}\|^4},$$

and

$$\lambda_k = \lambda_k^{DPRP1} = t_k \frac{\vartheta_k^T d_{k-1}}{\|\vartheta_{k-1}\|^2}.$$

All algorithms were coded in MATLAB R2019b and run using a personal laptop; Intel Core i7 processor, 16 GB RAM, 64 bit Windows 10 Pro operating system. The source code for the TTRMIL, NHS+, and NPRP+ methods can be found in the link <https://drive.google.com/drive/folders/1Tjj9A3m82yeoM6EH06syR2Hhn-bNqg86?usp=sharing>. The iteration process is stopped when $\|\vartheta_k\| \leq 10^{-6}$.

To see the efficiency and robustness of the proposed algorithm, we used 52 test functions with 144 problems as in Table 1-5. The test functions are taken from Andrei [4] and Jamil-Yang [16]. We choose the following parameters for the implementation of each method:

- FR and CD methods: $\theta = 0.0001$, $\sigma = 0.009$.
- TTRMIL method: all parameters are maintained as in [19].
- NHS+ and NPRP+ methods: all parameters are maintained as in [17] and [18], respectively.
- TTFRCD method: $\bar{t} = 0.03$, $\lambda = 0.01$, $\theta = 0.008$, $\sigma = 0.02$.

Most of initial points for the test problems were suggested by Andrei [4] and we consider using small, medium and high dimensions ranging from 2 to 1,000,000. When the number of iterations exceeds 10,000 or $\|\vartheta_k\| > 10^{-6}$,

we use the symbol "F" to indicate a failure. **The numerical results of the experiments can be found in the link <https://drive.google.com/drive/folders/1Tjj9A3m82yeoM6EH06syR2Hhn-bNQg86?usp=sharing>**, where ITER, FVAL and TIME denote the number of iterations, function evaluations, and CPU time (in seconds), respectively. We also use the performance profile introduced by Dolan and Moré in [8] to draw Figures 1,2 and3. According to [8], the performance profiles are described in terms of a performance measure $t_{p,s} > 0$ obtained for each problem $p \in P$ and solver $s \in S$. The performance ratio $r_{p,s}$ is defined as follows, for any pair (p, s) of problem p and solver s :

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

Meanwhile, the performance profile of a solver $s \in S$ is defined as

$$\rho(\tau) = \frac{1}{n_p} \text{size} \{p \in P | \log_2 r_{p,s} \leq \tau\},$$

where n_p denotes the number of problems and $\tau \in \mathbb{R}^+$. Generally, when τ takes certain value, the solver with high value of $\rho(\tau)$ represents the best solver. Hence, the top curve represents the most efficient method.

Furthermore, we can observe the numerical results, that the NHS+ method solves 93% (135 out of 144) of the test problems, the TTRMIL and CD methods solves 84% (122 out of 144), the FR method solves 85% (123 out of 144), the NPRP+ methods solves 95% (137 out of 144) and our proposed TTFRCD method solves 98% (142 out of 144). In this perspective, it can be easily seen that TTFRCD method is better in comparison to NHS+, TTRMIL, CD, FR and NPRP+ methods.

Figures 1, 2 and 3 illustrate the performance profile of these methods based on number of iterations, function evaluations and CPU time, respectively. Thus, TTFRCD method performs better than other methods in term of number of iterations, function evaluations and CPU time.

<i>Problem</i>	<i>Functions</i>	<i>Dimensions</i>	<i>InitialPoints</i>
1	<i>POWER</i>	10	(3, 3, ..., 3)
2	<i>POWER</i>	50	(3, 3, ..., 3)
3	<i>POWER</i>	500	(3, 3, ..., 3)
4	<i>QuadraticQF1</i>	100	(1, 1, ..., 1)
5	<i>QuadraticQF1</i>	1, 000	(1, 1, ..., 1)
6	<i>QuadraticQF1</i>	10, 000	(1, 1, ..., 1)
7	<i>ExtendedQuadraticPenaltyQP2</i>	5	(1, 1, ..., 1)
8	<i>ExtendedQuadraticPenaltyQP2</i>	50	(1, 1, ..., 1)
9	<i>ExtendedQuadraticPenaltyQP2</i>	500	(1, 1, ..., 1)
10	<i>ExtendedQuadraticPenaltyQP1</i>	5	(2, 2, ..., 2)
11	<i>ExtendedQuadraticPenaltyQP1</i>	10	(2, 2, ..., 2)
12	<i>ExtendedQuadraticPenaltyQP1</i>	100	(2, 2, ..., 2)
13	<i>Quartic</i>	4	(0.5, 0.5, ..., 0.5)
14	<i>Quartic</i>	4	(0.001, 0.001, ..., 0.001)
15	<i>Matyas</i>	2	(1, 1)
16	<i>Matyas</i>	2	(20, 20)
17	<i>Colville</i>	4	(1.2, 1.2, 1.2, 1.2)
18	<i>Colville</i>	4	(-0.5, -0.5, ..., -0.5)
19	<i>DixonandPrice</i>	1, 000	(0.5, 0.5, ..., 0.5)
20	<i>DixonandPrice</i>	10, 000	(0.5, 0.5, ..., 0.5)
21	<i>DixonandPrice</i>	100, 000	(0.5, 0.5, ..., 0.5)
22	<i>Sphere</i>	1, 000	(1, 1, ..., 1)
23	<i>Sphere</i>	10, 000	(1, 1, ..., 1)
24	<i>Sphere</i>	100, 000	(1, 1, ..., 1)
25	<i>SumSquares</i>	1, 000	(0.1, 0.1, ..., 0.1)
26	<i>SumSquares</i>	10, 000	(0.1, 0.1, ..., 0.1)
27	<i>SumSquares</i>	50, 000	(0.1, 0.1, ..., 0.1)
28	<i>DENSCHNA</i>	1, 000	(-1, -1, ..., -1)
29	<i>DENSCHNA</i>	10, 000	(-1, -1, ..., -1)
30	<i>DENSCHNA</i>	100, 000	(-1, -1, ..., -1)

TABLE 1. List of Test Problems (1).

<i>Problem</i>	<i>Functions</i>	<i>Dimensions</i>	<i>InitialPoints</i>
31	<i>DENSCHNC</i>	100	(1.5, 1.5, ..., 1.5)
32	<i>DENSCHNC</i>	5,000	(1.5, 1.5, ..., 1.5)
33	<i>DENSCHNC</i>	50,000	(1.5, 1.5, ..., 1.5)
34	<i>ExtendedBlock – DiagonalBD1</i>	100	(1.02, 1.02, ..., 1.02)
35	<i>ExtendedBlock – DiagonalBD1</i>	5,000	(1.02, 1.02, ..., 1.02)
36	<i>ExtendedBlock – DiagonalBD1</i>	50,000	(1.02, 1.02, ..., 1.02)
37	<i>HIMMELBH</i>	10	(0.8, 0.8, ..., 0.8)
38	<i>HIMMELBH</i>	50	(0.8, 0.8, ..., 0.8)
39	<i>HIMMELBH</i>	100	(0.8, 0.8, ..., 0.8)
40	<i>ExtendedHiebert</i>	1,000	(5, 5, ..., 5)
41	<i>ExtendedHiebert</i>	10,000	(5, 5, ..., 5)
42	<i>ExtendedHiebert</i>	100,000	(5, 5, ..., 5)
43	<i>ENGVAL1</i>	5	(1, 1, ..., 1)
44	<i>ENGVAL1</i>	10	(1, 1, ..., 1)
45	<i>ENGVAL1</i>	50	(1, 1, ..., 1)
46	<i>ENGVAL8</i>	5	(0.5, 0.5, ..., 0.5)
47	<i>ENGVAL8</i>	10	(0.5, 0.5, ..., 0.5)
48	<i>ENGVAL8</i>	50	(0.5, 0.5, ..., 0.5)
49	<i>LinearPerturbed</i>	100	(0.1, 0.1, ..., 0.1)
50	<i>LinearPerturbed</i>	5,000	(0.1, 0.1, ..., 0.1)
51	<i>LinearPerturbed</i>	50,000	(0.1, 0.1, ..., 0.1)
52	<i>QUARTICM</i>	1,000	(4, 4, ..., 4)
53	<i>QUARTICM</i>	50,000	(4, 4, ..., 4)
54	<i>QUARTICM</i>	100,000	(4, 4, ..., 4)
55	<i>Price4</i>	2	(-2, 3)
56	<i>Price4</i>	2	(3, -2)
57	<i>ZirilliorAluffi – Pentini's</i>	2	(1, 1)
58	<i>ZirilliorAluffi – Pentini's</i>	2	(-1, -1)
59	<i>DiagonalDoubleBorderArrowUp</i>	10	(0.4, 0.4, ..., 0.4)
60	<i>DiagonalDoubleBorderArrowUp</i>	500	(0.4, 0.4, ..., 0.4)

TABLE 2. List of Test Problems (2).

<i>Problem</i>	<i>Functions</i>	<i>Dimensions</i>	<i>InitialPoints</i>
61	<i>DiagonalDoubleBorderArrowUp</i>	5,000	(0.4, 0.4, ..., 0.4)
62	<i>HARKERP</i>	1,000	(1, ..., 1000)
63	<i>HARKERP</i>	50,000	(1, 2, ..., 50000)
64	<i>HARKERP</i>	100,000	(1, 2, ..., 100000)
65	<i>ExtendedQuadraticPenaltyQP3</i>	5	(1, 1, ..., 1)
66	<i>ExtendedQuadraticPenaltyQP3</i>	10	(1, 1, ..., 1)
67	<i>ExtendedQuadraticPenaltyQP3</i>	100	(1, 1, ..., 1)
68	<i>DIAG – AUP1</i>	10	(-1, -1, ..., -1)
69	<i>DIAG – AUP1</i>	1,000	(-1, -1, ..., -1)
70	<i>DIAG – AUP1</i>	10,000	(-1, -1, ..., -1)
71	<i>ARWHEAD</i>	10	(1, 1, ..., 1)
72	<i>ARWHEAD</i>	50	(1, 1, ..., 1)
73	<i>ARWHEAD</i>	100	(1, 1, ..., 1)
74	<i>ExtendedWhite&Holst</i>	50,000	(1.1, 1.1, ..., 1.1)
75	<i>ExtendedWhite&Holst</i>	100,000	(1.1, 1.1, ..., 1.1)
76	<i>ExtendedWhite&Holst</i>	1,000,000	(1.1, 1.1, ..., 1.1)
77	<i>ExtendedRosenbrock</i>	50,000	(0.1, 1, ..., 0.1, 1)
78	<i>ExtendedRosenbrock</i>	100,000	(0.1, 1, ..., 0.1, 1)
79	<i>ExtendedRosenbrock</i>	1,000,000	(0.1, 1, ..., 0.1, 1)
80	<i>ExtendedFreudenstein&Roth</i>	1,000	(0.5, -2, ..., 0.5, -2)
81	<i>ExtendedFreudenstein&Roth</i>	50,000	(0.5, -2, ..., 0.5, -2)
82	<i>ExtendedFreudenstein&Roth</i>	100,000	(0.5, -2, ..., 0.5, -2)
83	<i>ExtendedBeale</i>	1,000	(1, 0.8, ..., 1, 0.8)
84	<i>ExtendedBeale</i>	50,000	(1, 0.8, ..., 1, 0.8)
85	<i>ExtendedBeale</i>	100,000	(1, 0.8, ..., 1, 0.8)
86	<i>Raydan1</i>	10	(1.08, 1.08, ..., 1.08)
87	<i>Raydan1</i>	50	(1.08, 1.08, ..., 1.08)
88	<i>Raydan1</i>	100	(1.08, 1.08, ..., 1.08)
89	<i>ExtendedTridiagonal1</i>	10	(-2.1, -2.1, ..., -2.1)
90	<i>ExtendedTridiagonal1</i>	50	(-2.1, -2.1, ..., -2.1)

TABLE 3. List of Test Problems (3).

<i>Problem</i>	<i>Functions</i>	<i>Dimensions</i>	<i>InitialPoints</i>
91	<i>ExtendedTridiagonal1</i>	100	$(-2.1, -2.1, \dots, -2.1)$
92	<i>Diagonal4</i>	1,000	$(0.1, 0.1, \dots, 0.1)$
93	<i>Diagonal4</i>	5,000	$(0.1, 0.1, \dots, 0.1)$
94	<i>Diagonal4</i>	50,000	$(0.1, 0.1, \dots, 0.1)$
95	<i>ExtendedHimmelblau</i>	1,000	$(5, 5, \dots, 5)$
96	<i>ExtendedHimmelblau</i>	50,000	$(5, 5, \dots, 5)$
97	<i>ExtendedHimmelblau</i>	100,000	$(5, 5, \dots, 5)$
98	<i>FLETCHCR</i>	100	$(-5, -5, \dots, -5)$
99	<i>FLETCHCR</i>	5,000	$(-5, -5, \dots, -5)$
100	<i>FLETCHCR</i>	50,000	$(-5, -5, \dots, -5)$
101	<i>ExtendedPowel</i>	100	$(8, 8, \dots, 8)$
102	<i>ExtendedPowel</i>	1,000	$(8, 8, \dots, 8)$
103	<i>NONSCOMP</i>	2	$(10, 10)$
104	<i>NONSCOMP</i>	4	$(10, 10, \dots, 10)$
105	<i>NONSCOMP</i>	10	$(10, 10, \dots, 10)$
106	<i>ExtendedDENSCHNB</i>	1,000	$(1, 1, \dots, 1)$
107	<i>ExtendedDENSCHNB</i>	50,000	$(1, 1, \dots, 1)$
108	<i>ExtendedDENSCHNB</i>	100,000	$(1, 1, \dots, 1)$
109	<i>ExtendedPenaltyFunctionU52</i>	5	$(1, 2, \dots, 5)$
110	<i>ExtendedPenaltyFunctionU52</i>	10	$(1, 2, \dots, 10)$
111	<i>ExtendedPenaltyFunctionU52</i>	50	$(1, 2, \dots, 50)$
112	<i>Hager</i>	5	$(1, 1, \dots, 1)$
113	<i>Hager</i>	10	$(1, 1, \dots, 1)$
114	<i>Hager</i>	50	$(1, 1, \dots, 1)$
115	<i>ExtendedMaratos</i>	10	$(-0.5, -0.5, \dots, -0.5)$
116	<i>ExtendedMaratos</i>	50	$(-0.5, -0.5, \dots, -0.5)$
117	<i>ExtendedMaratos</i>	100	$(-0.5, -0.5, \dots, -0.5)$
118	<i>SixHumpCamel</i>	2	$(-1.5, -2)$
119	<i>SixHumpCamel</i>	2	$(-5, -10)$
120	<i>ThreeHumpCamel</i>	2	$(-1.5, -2)$

TABLE 4. List of Test Problems (4).

<i>Problem</i>	<i>Functions</i>	<i>Dimensions</i>	<i>InitialPoints</i>
121	<i>ThreeHumpCamel</i>	2	$(-5, -10)$
122	<i>Booth</i>	2	$(5, 5)$
123	<i>Booth</i>	2	$(10, 10)$
124	<i>Trecanni</i>	2	$(-1, 0.5)$
125	<i>Trecanni</i>	2	$(-5, 10)$
126	<i>Zetttl</i>	2	$(0, 0)$
127	<i>Zetttl</i>	2	$(10, 10)$
128	<i>Shallow</i>	1,000	$(1.001, 1.001, \dots, 1.001)$
129	<i>Shallow</i>	50,000	$(1.001, 1.001, \dots, 1.001)$
130	<i>Shallow</i>	100,000	$(1.001, 1.001, \dots, 1.001)$
131	<i>GeneralizedQuartic</i>	100	$(1.001, 1.001, \dots, 1.001)$
132	<i>GeneralizedQuartic</i>	5,000	$(1.001, 1.001, \dots, 1.001)$
133	<i>GeneralizedQuartic</i>	10,000	$(1.001, 1.001, \dots, 1.001)$
134	<i>QuadraticQF2</i>	10	$(0.5, 0.5, \dots, 0.5)$
135	<i>QuadraticQF2</i>	100	$(0.5, 0.5, \dots, 0.5)$
136	<i>QuadraticQF2</i>	1,000	$(0.5, 0.5, \dots, 0.5)$
137	<i>Leon</i>	2	$(-2, -2)$
138	<i>Leon</i>	2	$(-8, -8)$
139	<i>GeneralizedTridiagonal1</i>	5	$(15, 15, \dots, 15)$
140	<i>GeneralizedTridiagonal1</i>	10	$(15, 15, \dots, 15)$
141	<i>GeneralizedTridiagonal1</i>	100	$(15, 15, \dots, 15)$
142	<i>GeneralizedTridiagonal2</i>	10	$(4, 4, \dots, 4)$
143	<i>GeneralizedTridiagonal2</i>	50	$(4, 4, \dots, 4)$
144	<i>GeneralizedTridiagonal2</i>	500	$(4, 4, \dots, 4)$

TABLE 5. List of Test Problems (5).

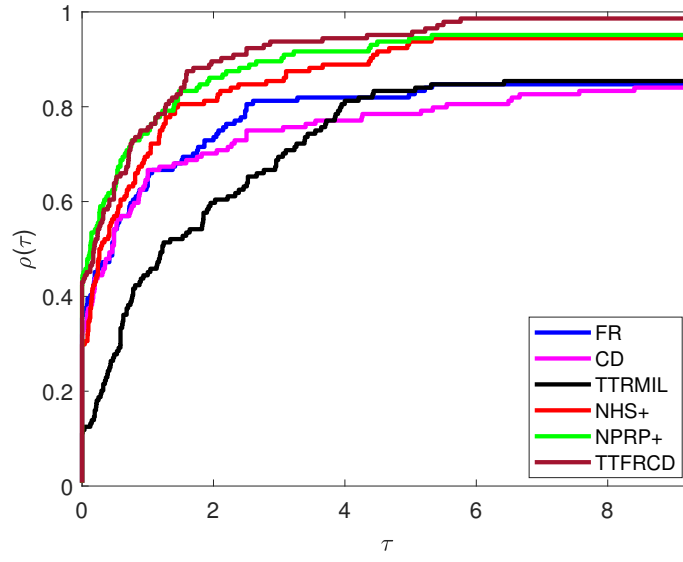


FIGURE 1. Performance profile with respect to number of iterations.

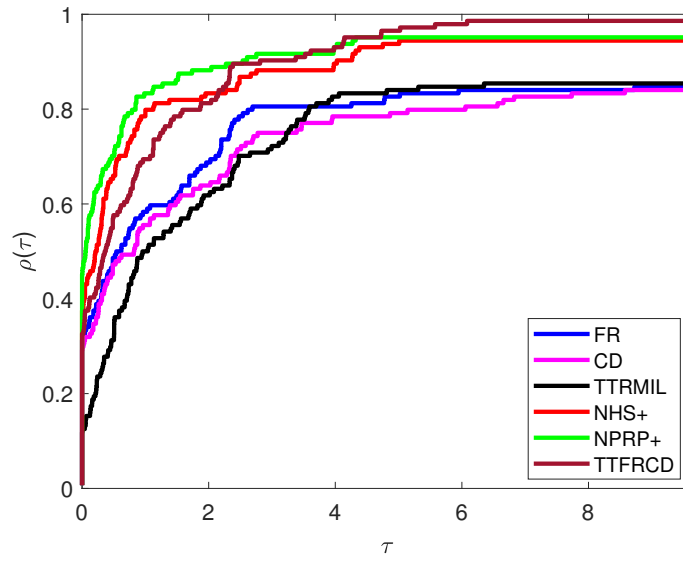


FIGURE 2. Performance profile with respect to number of function evaluations.

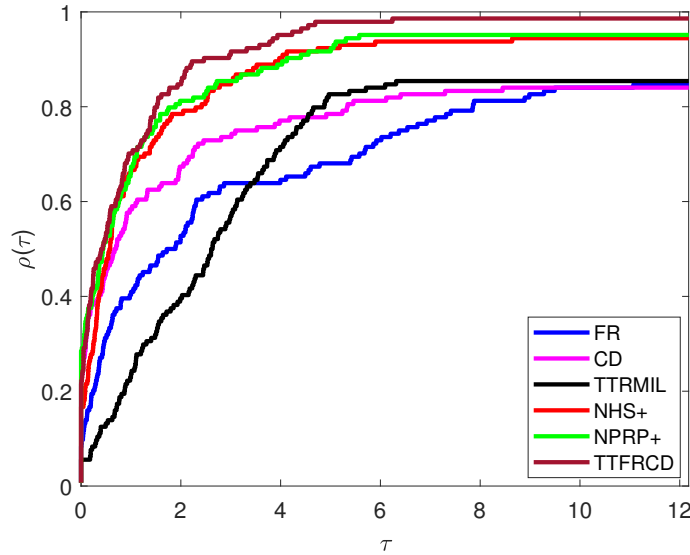


FIGURE 3. Performance profile with respect to CPU time.

4. CONCLUSIONS

In this work, a hybrid CG method for solving (1.1) was proposed. The method is a hybrid three-term of two existing classical methods, namely; CD and FR methods. The direction has the sufficient descent and boundedness properties. Under the Wolfe-type conditions, the global convergence of the method was proved. Based on the reported numerical results as well as the figures presented, the hybrid method proves to be more efficient and robust than some existing methods.

ACKNOWLEDGMENTS

This research was funded by National Science, Research and Innovation Fund (NSRF), and King Mongkut's University of Technology North Bangkok with Contract no. KMUTNB-FF-65-60. The second author acknowledges with thanks, the Department of Mathematics and Applied Mathematics at the Se-fako Makgatho Health Sciences University.

REFERENCES

1. A. B. Abubakar, P. Kumam, M. Malik, P. Chaipunya, A. H. Ibrahim, A Hybrid FR-DY Conjugate Gradient Algorithm for Unconstrained Optimization with Application in Portfolio Selection, *AIMS Mathematics*, **6**(6), (2021), 6506-6527.

2. A. B. Abubakar, P. Kumam, M. Malik, and A. H. Ibrahim, A Hybrid Conjugate Gradient Based Approach for Solving Unconstrained Optimization and Motion Control Problems, *Mathematics and Computers in Simulation*, **201**, (2022), 640-657.
3. M. Al-Baali, Descent Property and Global Convergence of the Fletcher-Reeves Method with Inexact Line Search, *IMA Journal of Numerical Analysis*, **5**(1), (1985), 121-124.
4. N. Andrei, *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*, Springer, 2020.
5. N. Andrei, A Simple Three-Term Conjugate Gradient Algorithm for Unconstrained Optimization, *Journal of Computational and Applied Mathematics*, **241**, (2013), 19-29.
6. Y. H. Dai, J. Han, G. Liu, D. Sun, H. Yin, Y. X. Yuan, Convergence Properties of Nonlinear Conjugate Gradient Methods, *SIAM Journal on Optimization*, **10**(2), (2000), 345-358.
7. J. Deepho, A. B. Abubakar, M. Malik, I. K. Argyros, Solving Unconstrained Optimization Problems via Hybrid CD-DY Conjugate Gradient Methods with Applications, *Journal of Computational and Applied Mathematics*, **405**, (2022), 113823.
8. E. D. Dolan, J. J. Moré, Benchmarking Optimization Software with Performance Profiles, *Mathematical Programming*, **91**(2), (2002), 201-213.
9. R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 2013.
10. R. Fletcher, C. M. Reeves, Function Minimization by Conjugate Gradients, *The Computer Journal*, **7**(2), 1964, 149-154.
11. J. C. Gilbert, J. Nocedal, Global Convergence Properties of Conjugate Gradient Methods for Optimization, *SIAM Journal on Optimization*, **2**(1), 1992, 21-42.
12. W. Hager, H. Zhang, A New Conjugate Gradient Method with Guaranteed Descent and an Efficient Line Search, *SIAM Journal on Optimization*, **16**(1), (2005), 170-192.
13. W. Hager, H. Zhang, Algorithm 851: CG-Descent, a Conjugate Gradient Method with Guaranteed Descent, *ACM Transactions on Mathematical Software (TOMS)*, **32**(1), (2006), 113-137.
14. W. Hager, H. Zhang, A Survey of Nonlinear Conjugate Gradient Methods, *Pacific journal of Optimization*, **2**(1), (2006), 35-58.
15. M. R. Hestenes, E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, *Journal of Research of the National Bureau of Standards*, **49**(6), (1952), 409-436.
16. M. Jamil, X. S. Yang, A Literature Survey of Benchmark Functions for Global Optimisation problems, *International Journal of Mathematical Modelling and Numerical Optimisation*, **4**(2), (2013), 150-194.
17. M. Li, A Modified Hestense–Stiefel Conjugate Gradient Method Close to the Memoryless BFGS quasi-Newton Method, *Optimization Methods and Software*, **33**(2), (2018), 336-353.
18. M. Li, A Three Term Polak-Ribière-Polyak Conjugate Gradient Method Close to the Memoryless BFGS quasi-Newton Method, *Journal of Industrial & Management Optimization*, **16**(1), (2020), 245.
19. J. Liu, Y. M. Feng, L. M. Zou, Some Three-Term Conjugate Gradient Methods with the Inexact Line Search Condition, *Calcolo*, **55**(2), (2018), 1-16.
20. J. Liu, S. J. Li, New Hybrid Conjugate Gradient Method for Unconstrained Optimization, *Applied Mathematics and Computation*, **245**, (2014), 36-43.
21. J. Nocedal, Updating quasi-Newton Matrices with Limited Storage, *Mathematics of Computation*, **35**(151), (1980), 773-782.
22. D. F. Shanno, Conjugate Gradient Methods with Inexact Searches, *Mathematics of Operations Research*, **3**(3), (1978), 244-256.
23. W. Sun, Y. X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, 2006.

24. L. Zhang, W. Zhou, D. H. Li, A Descent Modified Polak–Ribière–Polyak Conjugate Gradient Method and its Global Convergence, *IMA Journal of Numerical Analysis*, **26**(4), (2006), 629-640.