Scopus Preview

Source details

Journal of Computational and Applied Mathematics

Years currently covered by Scopus: from 1975 to 2026

Publisher: Elsevier

ISSN: 0377-0427

Subject area: Mathematics: Applied Mathematics | Mathematics: Computational Mathematics

Source type: Journal

View all documents > | Set document alert | Save to source list

CiteScore 2024
4.8

SJR 2024
0.688

SNIP 2024
1.374

CiteScore | CiteScore rank & trend | Scopus content coverage

CiteScore 2024

$$4.8 = \frac{10{,}231 \text{ Citations } 2021 - 2024}{2{,}143 \text{ Documents } 2021 - 2024}$$

Calculated on 05 May, 2025

CiteScoreTracker 2025

$$5.1 = \frac{10{,}801 \text{ Citations to date}}{2{,}113 \text{ Documents to date}}$$

Last updated on 05 February, 2026 • Updated monthly

CiteScore rank 2024

Category	Rank	Percentile
Mathematics — Applied Mathematics	#101/665	84th
Mathematics — Computational Mathematics	#43/201	78th

View CiteScore methodology > | CiteScore FAQ > | Add CiteScore to your site

About Scopus

What is Scopus
Content coverage
Scopus blog
Scopus API
Privacy matters

Language

日本語版を表示する
查看简体中文版本
查看繁體中文版本
Просмотр версии на русском языке

Customer Service

Help
Tutorials
Contact us

ELSEVIER

Journal of Computational and Applied Mathematics

Netherlands | Universities and research institutions | Media Ranking

Country
Netherlands

SCIMAGO INSTITUTIONS RANKINGS

SCImago Media Rankings

Subject Area and Category
Mathematics
- Applied Mathematics
- Computational Mathematics

Publisher
Elsevier B.V.

SJR 2024
0.688 Q2

H-Index
149

Publication type
Journals

ISSN
03770427

Coverage
1975-2025

Information
Home
How to publish in this journal
Contact

Scope
The Journal of Computational and Applied Mathematics publishes original papers of high scientific value in all areas of computational and applied mathematics. The main interest of the Journal is in papers that describe and analyze new computational techniques for solving scientific or engineering problems. Also the improved analysis, including the effectiveness and applicability, of existing methods and algorithms is of importance. The computational efficiency (e.g. the convergence, stability, accuracy, ...) should be proved and illustrated by nontrivial numerical examples. Papers describing only variants of existing methods, without adding significant new computational properties are not of interest. The audience consists of: applied mathematicians, numerical analysts, computational scientists and engineers.

Quartiles

Find similar journals

All quartiles | All countries | All subject categories | Clear filters | Download

Only Open Access Journals

1 - Applied Numerical Mathematics

82%

similarity

2 - Numerical Algorithms

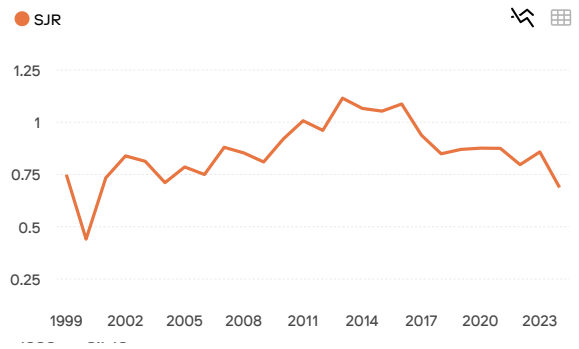
79%

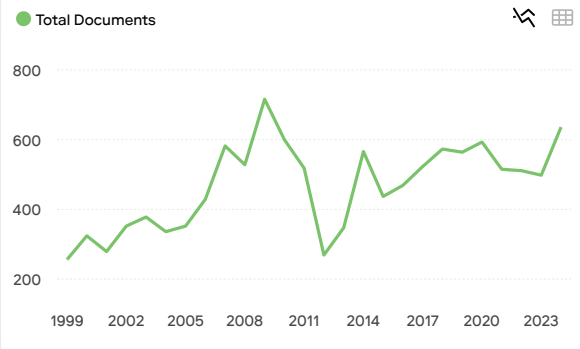
similarity

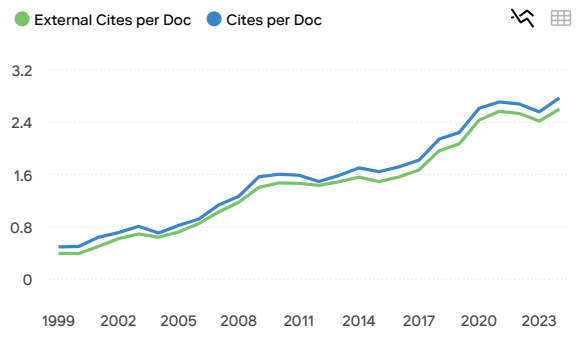
3 - International Journal of Computer Mathematics

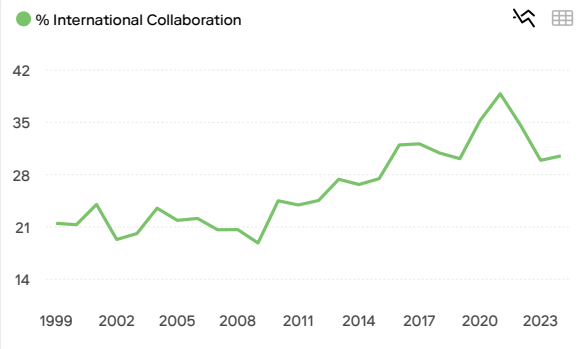
76%

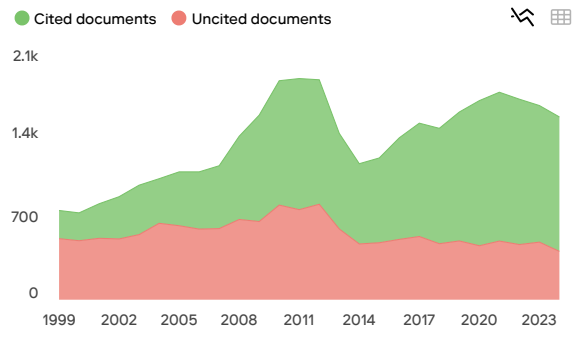
similarity

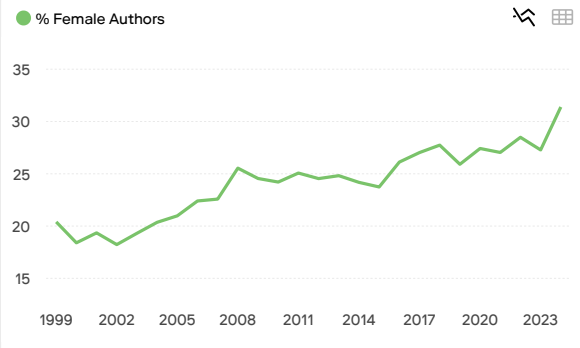


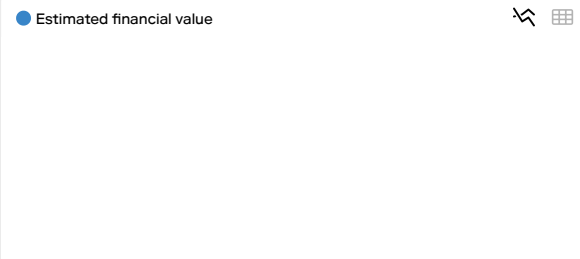


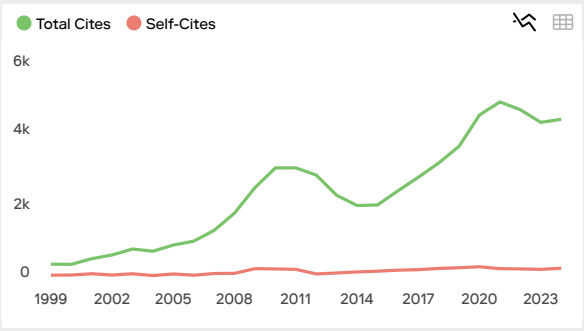


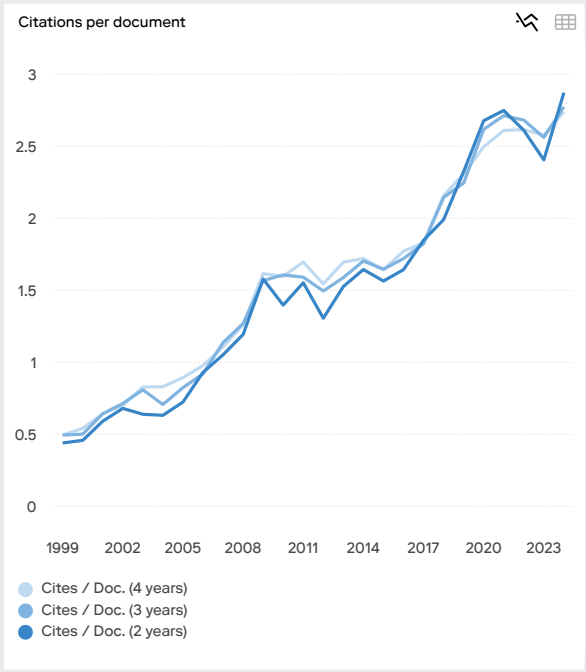


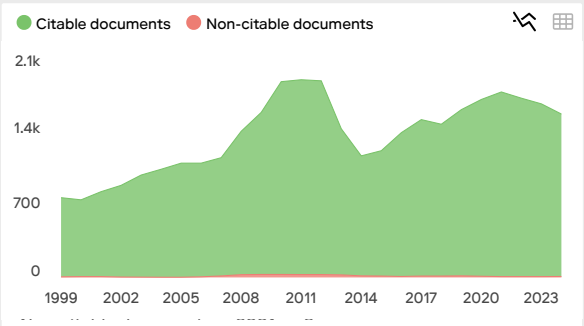


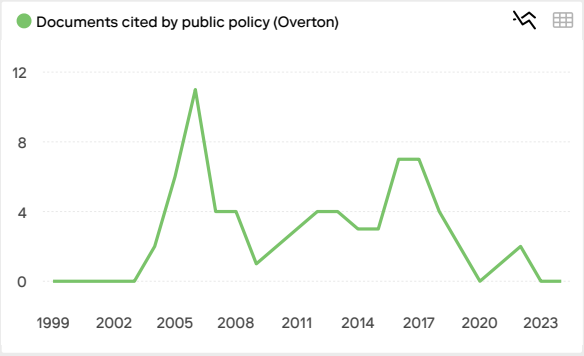


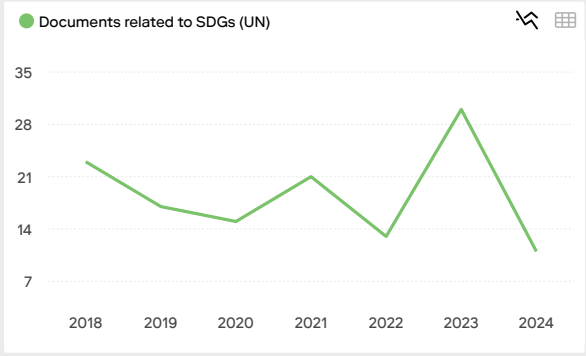


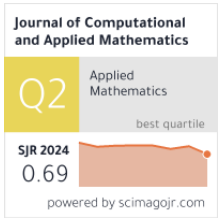








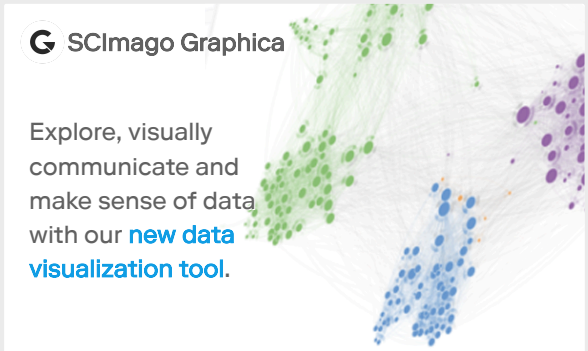




← Show this widget in your own website

Just copy the code below and paste within your html code:

```
<a href="https://www.scimago
```



safaa safouan
1 year ago

Dear SCImago Team,
I would like to bring to your attention concerns regarding the editorial process of Journal of Computational and Applied Mathematics. Our manuscript was rejected on grounds of citation errors and similarity to external sources, but the provided feedback lacked clarity, evidence, and professionalism. The similarity index was 17%, well within acceptable academic thresholds, and citation issues could have been addressed during the review process.

Furthermore, after the revisions were made, the reviewers' feedback was positive, yet the editor still rejected the manuscript, citing vague and unsubstantiated reasons. This raises concerns that the editor may be using their position in a manner inconsistent with professional and ethical standards. Such actions suggest a lack of

Journal of Computational and Applied Mathematics 433 (2023) 115304

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A hybrid HS-LS conjugate gradient algorithm for unconstrained optimization with applications in motion control and image recovery

Poom Kumam [a,b,c], Auwal Bala Abubakar [d,e], Maulana Malik [f], Abdulkarim Hassan Ibrahim [g], Nuttapol Pakkaranang [h,*], Bancha Panyanak [i]

[a] KMUTT Fixed Point Research Laboratory, KMUTT-Fixed Point Theory and Applications Research Group, 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

[b] Center of Excellence in Theoretical and Computational Science (TaCS-CoE), SCL 802 Fixed Point Laboratory, Science Laboratory Building, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand

[c] Research Center for Interneural Computing, China Medical University Hospital, China Medical University, Taichung, 40402, Taiwan

[d] Numerical Optimization Research Group, Department of Mathematical Sciences, Bayero University, Kano, Nigeria

[e] Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa 0204, South Africa

[f] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok 16424, Indonesia

[g] Center for Smart Mobility and Logistics, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

[h] Mathematics and Computing Science Program, Faculty of Sciences and Technology, Phetchabun Rajabhat University, Thailand

[i] Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand

ARTICLE INFO

Article history:
Received 11 August 2022
Received in revised form 22 February 2023

MSC:
65K10
90C52
90C26

Keywords:
Unconstrained optimization
Three-term conjugate gradient method
Line search
Global convergence

ABSTRACT

This article presents a new hybrid conjugate gradient (CG) algorithm for solving unconstrained optimization problem. The search direction is defined as a combination of Hestenes–Stiefel (HS) and the Liu–Storey (LS) CG parameters and is close to the direction of the memoryless Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton direction. In addition, the search direction is descent and bounded. The global convergence of the algorithm is obtained under the Wolfe-type and Armijo-type line searches. Numerical experiments on some benchmark test problems is carried out to depict the efficiency and robustness of the hybrid algorithm. Furthermore, a practical application of the algorithm in motion control of robot manipulator and image restoration is provided.

1. Introduction

Consider the unconstrained optimization problem:

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

* Corresponding author.
E-mail address: nuttapol.pak@pcru.ac.th (N. Pakkaranang).

https://doi.org/10.1016/j.cam.2023.115304
0377-0427/

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and its gradient $g(x) = \nabla f(x)$ is Lipschitz continuous. There are several methods for solving (1) such as the Newton method, the quasi-Newton method, and some of their variants [1–3]. However, these methods are not suitable for large-scale problems as they require computing and storing of the Hessian matrix at every iteration. In addition, the methods fail when the Hessian matrix is singular. As a result of these drawbacks, the conjugate gradient (CG) methods were introduced.

CG methods are among the recently interesting iterative methods for solving large-scale problems of the form (1) because of their simplicity in implementation, Hessian free and less storage requirements [4]. A CG method for solving (1) produces a sequence $\{x_k\}$ using the formula

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

where d_k is the search direction given by

$$d_k := \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k > 0. \end{cases} \tag{3}$$

The scalar $\alpha_k > 0$ is the stepsize obtained through a line search procedure, and β_k is the CG parameter. However, for a descent method, d_k is required to satisfy the inequality

$$g_k^T d_k \leq -c\|g_k\|^2, \quad c > 0. \tag{4}$$

Some of the well-known nonlinear conjugate gradient parameters are the Fetcher–Reeves (FR) [5], Polak–Ribiére–Polyak (PRP) [6,7], Hestenes–Stiefel (HS) [8], conjugate descent (CD) [9], Liu–Storey (LS) [10] and Dai–Yuan (DY) [11]. These parameters are defined as follows:

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \qquad \beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}}, \qquad \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \tag{5}$$

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \qquad \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \qquad \beta_k^{LS} = \frac{g_k^T y_{k-1}}{-g_{k-1}^T d_{k-1}}. \tag{6}$$

where $g_k = g(x_k)$, $\quad y_{k-1} = g_k - g_{k-1}$ and $\|\cdot\|$ is the Euclidean norm.

As a way of improving the classical two-term direction above, hybrid and three-term CG methods were developed. For example, in [12], Liu and Li proposed a hybrid CG method for solving (1). The method is a combination of the LS and DY methods. The search direction satisfies the Newton direction as well as the Dai–Liao (DL) conjugacy condition. Global convergence was achieved under strong Wolfe line search. Xu and Kong [13] proposed two hybrid methods, where the first method is a combination of DY and HS methods while the second is that of FR and PRP. Dong et al. [14] proposed a modified HS method that is close to the Newton direction and satisfies the descent condition. The hybridization parameter was obtained using some conjugacy condition and the global convergence for general functions was proved under certain assumptions. Li [15] proposed a three-term PRP 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. The method satisfies the descent condition. The global convergence of the method was established using an appropriate line search. Numerical results show that the method is efficient for the standard unconstrained problems in the CUTEr library. Again, Li [16] proposed a nonlinear CG method which generates a search direction that is close to that of the memoryless BFGS quasi-Newton method. Moreover, the search direction satisfies the descent condition. Global convergence for strongly convex functions and nonconvex functions was established under the strong Wolfe line search. For more on CG methods, readers are referred to [17–28].

Motivated by the ideas given in [15,16], we present a hybrid CG method for solving (1). The direction combines the three-term PRP and LS directions. Furthermore, the direction is close to that of the memoryless BFGS quasi-Newton method and has the descent and trusts region properties. The global convergence under both the Wolfe and Armijo line searches is proved. Numerical results reported show the superiority of the hybrid method over those proposed in [15,16]. The advantage and novelty of the propose method is that; it possess the nice properties of both PRP and LS directions. Lastly, application of the method in motion control of robot manipulator is shown. The remainder of this paper is organized as follows. In the next section, the new method is derived together with its convergence. In Section 3, we provide some numerical experimental results.

2. Algorithm and theoretical results

Recall a three-term CG method for solving (1). From an initial guess x_0, the method compute the search direction as follows:

$$d_0 := -g_0, \qquad d_k := -g_k + \beta_k d_{k-1} + \gamma_k y_{k-1}, \quad k \geq 1, \tag{7}$$

where β_k, γ_k are parameters. Distinct choices of the parameters β_k and γ_k correspond to distinct three-term CG methods. It is clear that, the three-term CG methods collapses to the classical ones when $\gamma_k = 0$.

Next, we will recall the memoryless BFGS method proposed by Nocedal [29] and Shanno [30], where the search direction can be written as

$$d_k^{BFGS} := -Q_k g_k,$$

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

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

$$d_k^{BFGS} := -g_k + \left(\beta_k^{HS} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}\right) d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}(y_{k-1} - s_{k-1}). \tag{8}$$

Based on the above, Li [31] proposed a three-term HS CG method with direction defined as

$$d_k^{THS} := -g_k + \left(\beta_k^{LS} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}\right) d_{k-1} + t_k \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} y_{k-1}. \tag{9}$$

Replacing β_k^{HS} with β_k^{LS} and $\frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}$ with $\frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(-d_{k-1}^T g_{k-1})^2}$ in (8), we define another three-term search direction as

$$d_k^{TLS} := -g_k + \left(\beta_k^{LS} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(-d_{k-1}^T g_{k-1})^2}\right) d_{k-1} + t_k \frac{g_k^T d_{k-1}}{-d_{k-1}^T g_{k-1}}. \tag{10}$$

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

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

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

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

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

$$E_k E_k^T = t^2 y_{k-1}y_{k-1}^T - t[A_k y_{k-1}^T + y_{k-1}A_k^T] + A_k A_k^T$$

and

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

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

$$2t\|y_{k-1}\|^2 - 2y_{k-1}^T A_k = 0,$$

which implies

$$t = \frac{y_{k-1}^T(y_{k-1} - s_{k-1})}{\|y_{k-1}\|^2}. \tag{11}$$

Therefore, we choose t_k as

$$t_k := \min\left\{\bar{t}, \max\left\{0, \frac{y_{k-1}^T(y_{k-1} - s_{k-1})}{\|y_{k-1}\|^2}\right\}\right\}, \tag{12}$$

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

Motivated by the three-term CG directions defined in (9) and (10), we propose a hybrid three-term CG based algorithm for solving (1), where the search direction is defined as

$$d_0 := -g_0, \qquad d_k := -g_k + \beta_k d_{k-1} + \gamma_k y_{k-1}, \quad k \ge 1, \tag{13}$$

where

$$\beta_k := \frac{g_k^T y_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{w_k^2}, \qquad \gamma_k := t_k \frac{g_k^T d_{k-1}}{w_k} \tag{14}$$

and

$$w_k := \max\{\mu\|d_{k-1}\|\|y_{k-1}\|, d_{k-1}^T y_{k-1}, -d_{k-1}^T g_{k-1}\}, \quad \mu > 0. \tag{15}$$

Remark 2.1. Observe that the parameter β_k is a hybrid of $\beta_k^{LS} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{(-d_{k-1}^T g_{k-1})^2}$ and $\beta_k^{PRP} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{\|g_{k-1}\|^4}$ in (9) and (10), respectively. Furthermore, the direction defined by (13) is close to that of the memoryless BFGS method when $t_k = \frac{y_{k-1}^T(y_{k-1}-s_{k-1})}{\|y_{k-1}\|^2}$.

Remark 2.2. Note that, we carefully choose the first component of Eq. (15) so that the direction possess a descent property (see Lemma 2.7) independent of the line search.

Lemma 2.3. *The search direction d_k defined by (13) satisfy (4) with $c = \left(1 - \frac{(1+\bar{t})^2}{4}\right)$.*

Proof. Multiplying each side of (13) with g_k^T, we get

$$\begin{aligned}
g_k^T d_k &= -\|g_k\|^2 + \frac{g_k^T y_{k-1}}{w_k} g_k^T d_{k-1} - \frac{\|y_{k-1}\|^2}{w_k^2}(g_k^T d_{k-1})^2 + t_k \frac{g_k^T y_{k-1}}{w_k} g_k^T d_{k-1} \\
&\leq -\|g_k\|^2 + (1+t_k)\frac{g_k^T y_{k-1}}{w_k} g_k^T d_{k-1} - \frac{\|y_{k-1}\|^2}{w_k^2}(g_k^T d_{k-1})^2 \\
&= -\|g_k\|^2 + 2\left(\frac{(1+t_k)}{2} g_k^T\right)\frac{y_{k-1}}{w_k} g_k^T d_{k-1} - \frac{\|y_{k-1}\|^2}{w_k^2}(g_k^T d_{k-1})^2 \\
&\leq -\|g_k\|^2 + \frac{(1+t_k)^2}{4}\|g_k\|^2 + \frac{\|y_{k-1}\|^2}{w_k^2}(g_k^T d_{k-1})^2 - \frac{\|y_{k-1}\|^2}{w_k^2}(g_k^T d_{k-1})^2 \\
&= -\|g_k\|^2 + \frac{(1+t_k)^2}{4}\|g_k\|^2 \\
&\leq -\left(1 - \frac{(1+\bar{t})^2}{4}\right)\|g_k\|^2. \quad \blacksquare
\end{aligned}$$

Next, we will attempt to establish the convergence of the proposed scheme by first considering the standard Wolfe line search conditions [32],

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \vartheta \alpha_k g_k^T d_k, \tag{16}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \tag{17}$$

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

Assumption 2.4. The level set $\mathcal{J} = \{x : f(x) \leq f(x_0)\}$ is bounded.

Assumption 2.5. Suppose $\mathcal{J}$ is some neighborhood of $\mathcal{K}$, then f is continuously differentiable and its gradient Lipschitz continuous on $\mathcal{J}$. That is, we can find $L > 0$ such that for all x,

$$\|g(x) - g(\bar{x})\| \leq L\|x - \bar{x}\|, \quad \bar{x} \in \mathcal{J}. \tag{18}$$

From Assumptions 2.4 and 2.5, we can deduce that for all $x \in \mathcal{K}$ there exist $a_1,\ a_2 > 0$ for which

- $\|x\| \leq a_1$.
- $\|g(x)\| \leq a_2$.

Furthermore, the sequence $\{x_k\} \in \mathcal{K}$ because $\{f(x_k)\}$ is decreasing. Henceforth, we will suppose that Assumptions 2.4–2.5 hold and that the objective function is bounded below. We will now prove the convergence result.

Theorem 2.6. *Let conditions (16) and (17) be fulfilled. If*

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} = \ +\infty. \tag{19}$$

Then

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{20}$$

Proof. Suppose by contradiction that Eq. (20) does not hold, then there exists a nonnegative scalar ϵ such that

$$\|g_k\| \geq \epsilon \quad \text{for all} \quad k \in \mathbb{N}. \tag{21}$$

From Lemma 2.3 and (16),

$$f(x_{k+1}) \leq f(x_k) + \vartheta \alpha_k g_k^T d_k \leq f(x_k) - \vartheta \alpha_k \|g_k\|^2 \leq f(x_k) \leq f(x_{k-1}) \leq \cdots \leq f(x_0).$$

Likewise, by Lemma 2.3, condition (17) and Assumption 2.5, we have

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

Combining the above inequality with (16), we obtain

$$\frac{\vartheta(1-\sigma)}{L} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \leq f(x_k) - f(x_{k+1})$$

and

$$\frac{\vartheta(1-\sigma)}{L} \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \leq (f(x_0) - f(x_1)) + (f(x_1) - f(x_2)) + \cdots \leq f(x_0) < +\infty,$$

since $\{f(x_k)\}$ is bounded. The above implies that

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{22}$$

Now, inequality (21) with (4) implies that

$$\begin{aligned} g_k^T d_k &\leq -\left(1 - \frac{(1+\bar{t})^2}{4}\right) \|g_k\|^2 \\ &\leq -\left(1 - \frac{(1+\bar{t})^2}{4}\right) \epsilon^2. \end{aligned} \tag{23}$$

Squaring both sides and dividing by $\|d_k\|^2 \neq 0$ of (23), yields

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \left(1 - \frac{(1+\bar{t})^2}{4}\right)^2 \sum_{k=0}^{\infty} \frac{\epsilon^4}{\|d_k\|^2} = +\infty. \tag{24}$$

This contradicts (22). Therefore, the conclusion of the theorem holds. ■

Next, we will establish the convergence of the proposed method under the Armijo-type backtracking line search procedure. The procedure was first introduced by Grippo and Lucidi [33], where the step size α_k is determined as follows: $\rho,\ \vartheta \in (0, 1)$, $\alpha_k = \rho^i$, where i is the smallest nonnegative integer for which the relation

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \vartheta \alpha_k^2 \|d_k\|^2 \tag{25}$$

hold.

From (25) and the fact that $\{f(x_k)\}$ is decreasing, we can deduce that

$$\sum_{k=0}^{\infty} \alpha_k^2 \|d_k\|^2 < +\infty,$$

which further implies that

$$\lim_{k \to \infty} \alpha_k \|d_k\| = 0. \tag{26}$$

Lemma 2.7. *If $\{d_k\}$ is defined by (13), then there is $N_3 > 0$ for which $\|d_k\| \leq N_3$.*

Proof. From (14),

$$\begin{aligned} |\beta_k| &= \left| \frac{g_k^T y_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 g_k^T d_{k-1}}{w_k^2} \right| \\ &\leq \frac{\|g_k\| \|y_{k-1}\|}{\mu \|d_{k-1}\| \|y_{k-1}\|} - \frac{\|y_{k-1}\|^2 \|g_k\| \|d_{k-1}\|}{(\mu \|d_{k-1}\| \|y_{k-1}\|)^2} \\ &= \left(\frac{1}{\mu} + \frac{1}{\mu^2}\right) \frac{\|g_k\|}{\|d_{k-1}\|}. \end{aligned} \tag{27}$$

Also,

$$\begin{aligned}|\gamma_k| &= \left|t_k \frac{g_k^T d_{k-1}}{w_k}\right| \\ &= t_k \left|\frac{g_k^T d_{k-1}}{w_k}\right| \\ &\leq \bar{t}\frac{\|g_k\|\|d_{k-1}\|}{w_k} \\ &\leq \bar{t}\frac{\|g_k\|\|d_{k-1}\|}{\mu\|d_{k-1}\|\|y_{k-1}\|} \\ &= \frac{\bar{t}}{\mu}\frac{\|g_k\|}{\|y_{k-1}\|}. \end{aligned} \tag{28}$$

Therefore, from (13), (27) and (28), we have

$$\begin{aligned}\|d_k\| &= \|-g_k + \beta_k d_{k-1} + \gamma_k y_{k-1}\| \\ &\leq \|g_k\| + |\beta_k|\|d_{k-1}\| + |\gamma_k|\|y_{k-1}\| \\ &\leq \|g_k\| + \left(\frac{1}{\mu} + \frac{1}{\mu^2}\right)\frac{\|g_k\|}{\|d_{k-1}\|}\|d_{k-1}\| + \frac{\bar{t}}{\mu}\frac{\|g_k\|}{\|y_{k-1}\|}\|y_{k-1}\| \qquad (29) \\ &= \left(1 + \frac{1+\bar{t}}{\mu} + \frac{1}{\mu^2}\right)\|g_k\| \\ &= \left(1 + \frac{1+\bar{t}}{\mu} + \frac{1}{\mu^2}\right)a_2. \qquad (30)\end{aligned}$$

Letting $N_3 = \left(1 + \frac{1+\bar{t}}{\lambda} + \frac{1}{\lambda^2}\right)a_2$, we have

$$\|d_k\| \leq N_3. \quad \blacksquare \tag{31}$$

Theorem 2.8. *If the step size α_k is obtained via relation (25), then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{32}$$

Proof. Suppose by contradiction Eq. (32) is not true. Then for all k, we can find an $r_2 > 0$ for which

$$\|g_k\| \geq r_2. \tag{33}$$

Let $\alpha_k = \rho^{-1}\alpha_k$, then α_k does not satisfy (25). That is

$$f(x_k + \rho^{-1}\alpha_k d_k) > f(x_k) - \vartheta\rho^{-2}\alpha_k^2\|d_k\|^2. \tag{34}$$

Applying the mean value theorem together with Lemma 2.7, (4) and (18), there exist an $l_k \in (0, 1)$ such that

$$\begin{aligned}f(x_k + \rho^{-1}\alpha_k d_k) - f(x_k) &= \rho^{-1}\alpha_k g(x_k + l_k\rho^{-1}\alpha_k d_k) \\ &= \rho^{-1}\alpha_k g_k^T d_k + \rho^{-1}\alpha_k(g(x_k + l_k\rho^{-1}\alpha_k d_k) - g_k)^T d_k \\ &\leq \rho^{-1}\alpha_k g_k^T d_k + L\rho^{-2}\alpha_k^2\|d_k\|^2.\end{aligned}$$

Inserting the above relation in (34), together with (31) and (33) we have

$$\alpha_k \geq \frac{\rho\|g_k\|^2}{(L+\vartheta)\|d_k\|^2} \geq \frac{\rho r_2^2}{(L+\vartheta)N_2^2} > 0.$$

This and (26) give

$$\lim_{k\to\infty} \|d_k\| = 0. \tag{35}$$

On the other hand, applying backward Cauchy–Schwarz inequality on (4) gives

$$\|d_k\| \geq \left(1 - \frac{(1+\bar{t})^2}{4}\right)\|g_k\|.$$

Thus, we have $\lim_{k\to\infty}\|g_k\| = 0$ which leads to a contradiction and therefore (32) holds. $\blacksquare$

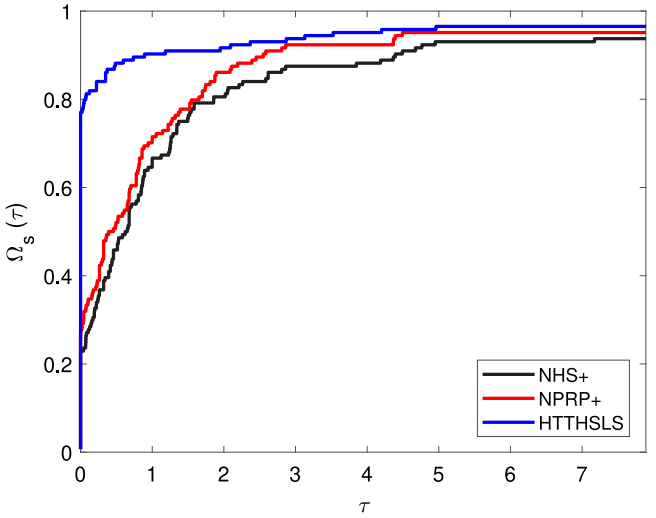


Fig. 1. Performance profiles on NOI.

3. Numerical experiments

This section we evaluate the performance of our proposed method by using 144 problems with range dimensions from 2 to 1,000,000. The idea of this numerical experiment was carried out to see the computational performance of the proposed method. The test function and initial points are presented in Table 1 and considered by Andrei [34]. Our proposed method is symbolized as HTTHSLS and will be compared with NHS+ [16] and NPRP+ [15] methods. The comparison all the methods were implemented in Matlab and run using Matlab R2019a on a personal laptop with specification; Intel Core i7 processor, 16 GB RAM, 64 bit Windows 10 Pro operating system. For each function, we use five different dimensions to see the ability of methods in solving the problems. Comparison of computations is carried out objectively by using a standard Wolfe line search that is using the parameter values that correspond in the papers [15,16]: $\vartheta = 0.1,\ \sigma = 0.9$, and $\bar{t} = 0.3$ for NHS+ and NPRP+ methods. Specifically, we take the parameter $\vartheta = 0.0001,\ \sigma = 0.009,\ \bar{t} = 0.3$, and $\mu = 0.01$ for HTTHSLS method. The all methods were stopped when $\|g_k\| \leq 10^{-6}$, and will be failed if the number of iterations more than 10,000 or never be reach the optimal value.

The numerical results of all methods are listed in Table 2, which consists of the number of iterations (NOI), the number of function evaluations, and the central of processing unit (CPU) time. The methods that fail to solve the problem are marked with "–". From Table 2, the NHS+ method was successful in solving the problem by 93%, the NPRP+ method 95%, and HTTHSLS method 97%.

The computational performances of each method were compared based on NOI, NOF, and CPU time and presented profile curve in Figs. 1–3 by using performance profile suggested by Dolan and Moré [35]. Generally, in performance profile, $\Omega_s(\tau)$ is the fraction problem with a ratio performance τ. So, the higher the $\Omega_s(\tau)$ value, it can be said that the method is superior. The formula of $\Omega_s(\tau)$ is defined as follows:

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

where P, S, n_s, n_p are set of the test problems, set of the methods, the number of methods, and the number of test problems, respectively. The performance profile $\Omega : \mathbb{R} \to [0, 1]$ is defined that $c_{p,s} > 0$ is NOI (or NOF or CPU time) required to solve problems p by method s for each $s \in S$, $p \in P$, $\tau > 0$, $size\{p \in P : \log_2 r_{p,s} \leq \tau\}$ is the number of the elements in the set $\{p \in P : \log_2 r_{p,s} \leq \tau\}$, and $r_{p,s}$ is performance ratio formulated as $r_{p,s} = c_{p,s}/\min\{c_{p,s}\}$.

From the properties of the performance profiles curve, that the higher the curve in the figures, then the method is efficient. Likewise, according to numerical reports in Table 2 and Figs. 1–3, the proposed HTTHSLS method shows some advantages, i.e HTTHSLS is effective for the 144 tested problems and the most efficient of the NHS+ and NPRP+ methods. Moreover, the numerical performance of HTTHSLS method is relatively stable due to the right choice (14) and (15) as the parameter. In other words, the three methods have all proven to be practically effective, at least for these particular sets of numerical experiments, according to the numerical results of the three comparisons and their corresponding performance profiles. Additionally, it is important to note that the HTTHSLS functions very robustly, particularly when tackling challenging problems.

Table 1
List of test functions and initial points.

Problem	Functions	Dimension	Initial Points
1	Extended White & Holst	50,000	(1.1, ...,1.1)
2	Extended White & Holst	100,000	(1.1, ...,1.1)
3	Extended White & Holst	1,000,000	(1.1, ...,1.1)
4	Extended Rosenbrock	50,000	(0.1,1, ...,0.1,1)
5	Extended Rosenbrock	100,000	(0.1,1, ...,0.1,1)
6	Extended Rosenbrock	1,000,000	(0.1,1, ...,0.1,1)
7	Extended Freudenstein & Roth	1,000	(0.5,−2, ...,0.5,−2)
8	Extended Freudenstein & Roth	50,000	(0.5,−2, ...,0.5,−2)
9	Extended Freudenstein & Roth	100,000	(0.5,−2, ...,0.5,−2)
10	Extended Beale	1,000	(1,0.8, ...,1,0.8)
11	Extended Beale	50,000	(1,0.8, ...,1,0.8)
12	Extended Beale	100,000	(1,0.8, ...,1,0.8)
13	Raydan 1	10	(1.08, ...,1.08)
14	Raydan 1	50	(1.08, ...,1.08)
15	Raydan 1	100	(1.08, ...,1.08)
16	Extended Tridiagonal 1	10	(−2.1, ...,−2.1)
17	Extended Tridiagonal 1	50	(−2.1, ...,−2.1)
18	Extended Tridiagonal 1	100	(−2.1, ...,−2.1)
19	Diagonal 4	1,000	(0.1, ...,0.1)
20	Diagonal 4	5,000	(0.1, ...,0.1)
21	Diagonal 4	50,000	(0.1, ...,0.1)
22	Extended Himmelblau	1,000	(5, ...,5)
23	Extended Himmelblau	50,000	(5, ...,5)
24	Extended Himmelblau	100,000	(5, ...,5)
25	FLETCHCR	100	(−5, ...,−5)
26	FLETCHCR	5,000	(−5, ...,−5)
27	FLETCHCR	50,000	(−5, ...,−5)
28	Extended Powel	100	(8, ...,8)
29	Extended Powel	1,000	(8, ...,8)
30	NONSCOMP	2	(10, 10)
31	NONSCOMP	4	(10, ...,10)
32	NONSCOMP	10	(10, ...,10)
33	Extended DENSCHNB	1,000	(1, ...,1)
34	Extended DENSCHNB	50,000	(1, ...,1)
35	Extended DENSCHNB	100,000	(1, ...,1)
36	Extended Penalty Function U52	5	(1, ...,5)
37	Extended Penalty Function U52	10	(1, 2, ...,10)
38	Extended Penalty Function U52	50	(1, 2, ...,50)
39	Hager	5	(1, ...,1)
40	Hager	10	(1, ...,1)
41	Hager	50	(1, ...,1)
42	Extended Maratos	10	(−0.5, ...,−0.5)
43	Extended Maratos	50	(−0.5, ...,−0.5)
44	Extended Maratos	100	(−0.5, ...,−0.5)
45	Six Hump Camel	2	(−1.5, −2)
46	Six Hump Camel	2	(−5, −10)
47	Three Hump Camel	2	(−1.5, −2)
48	Three Hump Camel	2	(−5, −10)
49	Booth	2	(5, 5)
50	Booth	2	(10, 10)
51	Trecanni	2	(−1, 0.5)
52	Trecanni	2	(−5, 10)
53	Zettl	2	(0, 0)
54	Zettl	2	(10, 10)
55	Shallow	1,000	(1.001, ...,1.001)
56	Shallow	50,000	(1.001, ...,1.001)
57	Shallow	100,000	(1.001, ...,1.001)
58	Generalized Quartic	100	(1.001, ...,1.001)
59	Generalized Quartic	5,000	(1.001, ...,1.001)
60	Generalized Quartic	10,000	(1.001, ...,1.001)
61	Quadratic QF2	10	(0.5, ...,0.5)
62	Quadratic QF2	100	(0.5, ...,0.5)
63	Quadratic QF2	1000	(0.5, ...,0.5)
64	Leon	2	(−2, −2)
65	Leon	2	(−8, −8)

(*continued on next page*)

Table 1 (*continued*).

Problem	Functions	Dimension	Initial Points
66	Generalized Tridiagonal 1	5	(15, ...,15)
67	Generalized Tridiagonal 1	10	(15, ...,15)
68	Generalized Tridiagonal 1	100	(15, ...,15)
69	Generalized Tridiagonal 2	10	(4, ...,4)
70	Generalized Tridiagonal 2	50	(4, ...,4)
71	Generalized Tridiagonal 2	500	(4, ...,4)
72	POWER	10	(3, ...,3)
73	POWER	50	(3, ...,3)
74	POWER	500	(3, ...,3)
75	Quadratic QF1	100	(1, ...,1)
76	Quadratic QF1	1,000	(1, ...,1)
77	Quadratic QF1	10,000	(1, ...,1)
78	Extended Quadratic Penalty QP2	5	(1, ...,1)
79	Extended Quadratic Penalty QP2	50	(1, ...,1)
80	Extended Quadratic Penalty QP2	500	(1, ...,1)
81	Extended Quadratic Penalty QP1	5	(2, ...,2)
82	Extended Quadratic Penalty QP1	10	(2, ...,2)
83	Extended Quadratic Penalty QP1	100	(2, ...,2)
84	Quartic	4	(0.5, ...,0.5)
85	Quartic	4	(0.001, ...,0.001)
86	Matyas	2	(1, 1)
87	Matyas	2	(20, 20)
88	Colville	4	(1.2, 1.2, 1.2, 1.2)
89	Colville	4	(−0.5, ...,−0.5)
90	Dixon and Price	1,000	(0.5, ...,0.5)
91	Dixon and Price	10,000	(0.5, ...,0.5)
92	Dixon and Price	100,000	(0.5, ...,0.5)
93	Sphere	1,000	(1, ...,1)
94	Sphere	10,000	(1, ...,1)
95	Sphere	100,000	(1, ...,1)
96	Sum Squares	1,000	(0.1, ...,0.1)
97	Sum Squares	10,000	(0.1, ...,0.1)
98	Sum Squares	50,000	(0.1, ...,0.1)
99	DENSCHNA	1,000	(−1, ...,−1)
100	DENSCHNA	10,000	(−1, ...,−1)
101	DENSCHNA	100,000	(−1, ...,−1)
102	DENSCHNC	100	(1.5, ...,1.5)
103	DENSCHNC	5,000	(1.5, ...,1.5)
104	DENSCHNC	50,000	(1.5, ...,1.5)
105	Extended Block-Diagonal BD1	100	(1.02, ...,1.02)
106	Extended Block-Diagonal BD1	5,000	(1.02, ...,1.02)
107	Extended Block-Diagonal BD1	50,000	(1.02, ...,1.02)
108	HIMMELBH	10	(0.8, ...,0.8)
109	HIMMELBH	50	(0.8, ...,0.8)
110	HIMMELBH	100	(0.8, ...,0.8)
111	Extended Hiebert	1,000	(5, ...,5)
112	Extended Hiebert	10,000	(5, ...,5)
113	Extended Hiebert	100,000	(5, ...,5)
114	ENGVAL1	5	(1, ...,1)
115	ENGVAL1	10	(1, ...,1)
116	ENGVAL1	50	(1, ...,1)
117	ENGVAL8	5	(0.5, ...,0.5)
118	ENGVAL8	10	(0.5, ...,0.5)
119	ENGVAL8	50	(0.5, ...,0.5)
120	Linear Perturbed	100	(0.1, ...,0.1)
121	Linear Perturbed	5,000	(0.1, ...,0.1)
122	Linear Perturbed	50,000	(0.1, ...,0.1)
123	QUARTICM	1,000	(4, ...,4)
124	QUARTICM	50,000	(4, ...,4)
125	QUARTICM	100,000	(4, ...,4)
126	Price 4	2	(−2, 3)
127	Price 4	2	(3, −2)
128	Zirilli or Aluffi-Pentini's	2	(1, 1)
129	Zirilli or Aluffi-Pentini's	2	(−1, −1)
130	Diagonal Double Border Arrow Up	10	(0.4, ...,0.4)
131	Diagonal Double Border Arrow Up	500	(0.4, ...,0.4)
132	Diagonal Double Border Arrow Up	5,000	(0.4, ...,0.4)

(*continued on next page*)

Table 1 (*continued*).

Problem	Functions	Dimension	Initial Points
133	HARKERP	1,000	(1, ...,1000)
134	HARKERP	50,000	(1,2, ...,50000)
135	HARKERP	100,000	(1,2, ...,100000)
136	Extended Quadratic Penalty QP3	5	(1, ...,1)
137	Extended Quadratic Penalty QP3	10	(1, ...,1)
138	Extended Quadratic Penalty QP3	100	(1, ...,1)
139	DIAG-AUP1	10	(−1, ...,−1)
140	DIAG-AUP1	1,000	(−1, ...,−1)
141	DIAG-AUP1	10,000	(−1, ...,−1)
142	ARWHEAD	10	(1, ...,1)
143	ARWHEAD	50	(1, ...,1)
144	ARWHEAD	100	(1, ...,1)

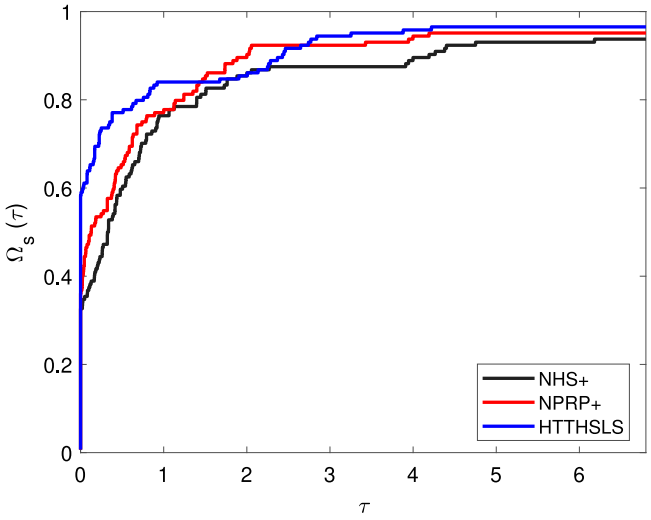


Fig. 2. Performance profiles on NOF.

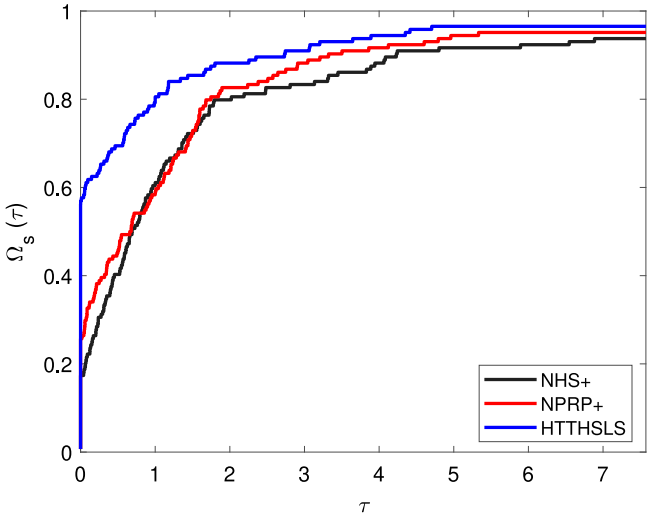


Fig. 3. Performance profiles on CPU.

Table 2
The numerical results of all methods using standard Wolfe line search.

Problem	NHS+			NPRP+			HTTHSLS		
	NOI	NOF	CPU	NOI	NOF	CPU	NOI	NOF	CPU
1	29	108	2.0158	26	97	2.0029	7	38	0.6411
2	29	108	3.9452	26	97	4.0478	8	41	1.3829
3	29	108	41.705	26	97	36.7233	8	41	13.5392
4	31	134	1.0043	24	99	1.1229	13	74	0.5209
5	33	140	2.0673	27	107	2.0159	13	74	1.0018
6	33	140	22.4099	23	97	15.3982	13	74	10.5875
7	–	–	–	17	76	0.0168	7	41	0.0333
8	–	–	–	–	–	–	–	–	–
9	–	–	–	–	–	–	–	–	–
10	31	93	0.073	15	54	0.0248	12	57	0.0294
11	34	101	2.0634	15	54	1.3434	12	57	1.0554
12	35	103	4.2888	15	54	2.3733	12	57	2.067
13	96	192	0.0158	105	210	0.0054	20	73	0.0018
14	59	154	0.0041	56	141	0.0089	61	178	0.007
15	72	217	0.0114	73	220	0.0221	68	332	0.0157
16	42	126	0.0258	22	66	0.0028	7	38	9.25E−04
17	43	129	0.0062	42	155	0.0168	7	38	0.0025
18	43	129	0.0167	41	140	0.0198	7	38	0.0055
19	6	16	0.0175	6	16	0.0145	7	18	0.0101
20	6	16	0.0207	6	16	0.0293	7	18	0.0213
21	6	16	0.1491	6	16	0.1598	7	18	0.0213
22	12	49	0.0339	12	48	0.022	7	30	0.0157
23	13	52	0.4244	12	48	0.4211	7	30	0.2562
24	13	52	0.8046	12	48	0.8413	7	30	0.4516
25	36	162	0.0208	38	171	0.0141	38	180	0.0233
26	41	172	0.1608	39	174	0.1807	40	182	0.1671
27	54	239	1.773	33	145	1.2283	34	157	1.1791
28	3060	9236	0.597	99	343	0.0403	–	–	–
29	4670	14066	6.2855	183	676	0.1057	2107	6446	2.7555
30	22	88	0.011	19	75	0.0011	12	62	0.0056
31	204	652	0.005	49	177	0.0045	209	711	0.0093
32	1284	4081	0.0376	2017	6130	0.079	9388	29233	0.2516
33	8	24	0.0255	6	19	0.0098	5	19	0.0265
34	8	24	0.213	6	19	0.1804	5	19	0.1556
35	8	24	0.4034	6	19	0.3652	5	19	0.2805
36	26	89	0.0032	18	61	0.0014	30	103	0.0093
37	21	84	0.0015	22	82	0.0021	16	62	0.0013
38	22	95	0.017	31	127	0.0063	85	303	0.0094
39	13	30	0.0022	11	28	9.93E−04	25	50	0.0011
40	19	48	9.32E−04	15	40	0.0086	21	51	0.0012
41	20	61	0.0041	20	61	0.0057	19	60	0.0039
42	22	89	0.0028	29	137	0.004	16	104	0.0015
43	22	89	0.0039	29	137	0.0222	16	104	0.0052
44	22	89	0.0044	29	137	0.0132	16	104	0.0073
45	8	29	9.24E−04	8	30	0.0087	5	24	2.80E−04
46	8	40	7.14E−04	11	49	9.96E−04	7	37	5.42E−04
47	17	50	0.0251	20	65	0.0088	22	341	0.0055
48	–	–	–	–	–	–	19	301	0.0028
49	2	6	0.0871	2	6	0.0075	2	6	4.60E−03
50	2	6	2.10E−04	2	6	2.42E−04	2	6	1.34E−04
51	1	3	1.88E−04	1	3	0.0052	1	3	4.00E−03
52	9	32	7.22E−04	8	31	0.0017	5	21	2.72E−04
53	2	6	1.86E−04	2	6	0.0075	1	4	3.80E−03
54	20	64	9.04E−04	25	78	0.0014	13	57	8.68E−04
55	6	16	2.31E−02	7	19	0.0078	3	11	1.56E−02
56	9	23	0.2059	11	29	0.2974	3	11	0.1198
57	9	23	0.3979	11	29	0.5328	3	11	0.1832
58	–	–	–	–	–	–	16	151	0.0148
59	25	145	0.1264	38	317	0.2441	13	89	0.0728
60	–	–	–	29	295	0.5282	19	110	0.1949
61	28	89	0.8985	29	93	0.0185	26	91	0.0076
62	113	369	0.0305	97	321	0.0271	102	359	0.0273
63	398	1316	0.2932	399	1301	0.2838	353	1272	0.2388
64	45	265	0.0041	42	214	0.0017	19	138	0.0016
65	65	454	0.0057	63	471	0.0115	36	336	0.0038

(*continued on next page*)

Table 2 (*continued*).

Problem	NHS+			NPRP+			HTTHSLS		
	NOI	NOF	CPU	NOI	NOF	CPU	NOI	NOF	CPU
66	21	80	0.0041	23	90	0.0023	18	80	0.0016
67	32	125	0.0033	28	111	0.0028	29	125	0.0042
68	34	133	0.0173	34	133	0.0286	34	158	0.0102
69	42	125	0.0027	–	–	–	23	77	0.0094
70	–	–	–	–	–	–	50	151	0.0054
71	–	–	–	–	–	–	48	231	0.0403
72	10	30	0.002	10	30	8.18E−04	10	30	6.06E−04
73	65	195	0.0095	66	198	0.0104	66	198	0.0052
74	770	2310	0.3748	773	2319	0.3232	3975	11925	1.5423
75	56	168	0.015	56	168	0.0149	56	168	0.0093
76	187	561	0.1561	187	561	0.1324	187	561	0.1325
77	606	1818	3.6258	606	1818	3.6506	606	1818	3.5241
78	18	110	0.0034	15	104	9.85E−04	15	121	0.0022
79	36	267	0.0139	35	299	0.0233	25	252	0.0088
80	48	411	0.0896	44	362	0.0924	61	667	0.1113
81	11	42	0.0024	9	35	9.99E−04	7	32	1.25E−02
82	10	37	1.00E−03	12	44	0.001	7	33	0.0012
83	12	56	0.0054	13	60	0.0147	8	42	0.0046
84	48	206	0.0039	204	686	0.0184	420	1377	0.0328
85	46	139	0.0027	69	209	0.015	16	51	1.50E−03
86	63	252	0.0042	63	252	0.0021	9	63	6.70E−03
87	80	320	0.0067	80	320	0.0099	11	77	1.20E−03
88	23	82	0.0012	57	184	0.0041	17	65	7.88E−04
89	109	395	0.0043	43	181	0.0029	65	258	0.0031
90	62	189	0.0403	64	195	0.0517	79	246	0.0616
91	62	189	0.4356	64	195	0.3995	79	246	0.5104
92	62	189	4.0203	64	195	3.0299	79	246	5.4423
93	1	3	0.0032	1	3	9.66E−04	1	3	1.80E−03
94	1	3	0.017	1	3	0.0156	1	3	0.0137
95	1	3	0.0725	1	3	0.0866	1	3	0.0693
96	178	534	0.1469	178	534	0.1275	178	534	0.1333
97	578	1734	3.4678	578	1734	3.2786	578	1734	3.1566
98	1310	3930	37.7475	1310	3930	31.1912	1310	3930	32.8052
99	14	39	0.024	21	56	0.0406	8	28	0.0215
100	14	39	0.1755	23	61	0.2629	8	28	0.1136
101	16	44	1.7127	23	61	2.0306	8	28	0.8927
102	61	189	0.023	46	145	0.0216	26	637	0.0546
103	69	213	0.5006	51	160	0.3985	29	760	1.3348
104	71	219	6.4551	56	175	3.5353	39	1096	29.5603
105	22	67	0.0084	21	66	0.0055	9	125	0.0087
106	24	73	0.0929	24	75	0.101	10	128	0.1398
107	24	73	0.9687	25	78	1.2307	10	130	1.2586
108	5	15	0.0016	5	15	5.04E−04	4	12	3.95E−04
109	5	15	0.0012	5	15	0.0013	4	12	1.40E−03
110	5	15	0.0023	5	15	0.0127	4	12	0.0017
111	79	332	0.0801	42	201	0.0585	28	217	0.0555
112	45	244	0.4976	45	211	0.425	28	217	0.4007
113	40	221	5.28	153	478	12.5747	28	217	3.9269
114	21	66	0.0024	23	72	0.0019	16	247	0.0043
115	21	66	0.0016	22	69	0.0017	19	360	0.0032
116	21	66	0.0042	24	75	0.0156	20	364	0.0095
117	36	112	0.0038	41	127	0.0026	35	603	0.0049
118	39	121	0.0032	42	130	0.0191	35	580	0.0072
119	39	121	0.0084	42	130	0.0064	34	589	0.0201
120	54	162	0.016	54	162	0.018	54	162	0.0076
121	406	1218	1.0576	406	1218	1.0436	406	1218	1.0019
122	1310	3930	29.4132	1310	3930	68.1057	1310	3930	30.5202
123	90	658	0.3314	90	658	0.3384	4	36	0.0233
124	103	847	16.7984	103	847	17.4984	5	55	1.0036
125	105	879	34.7643	105	879	49.8536	5	55	2.051
126	7	39	7.42E−04	32	130	0.0021	218	727	0.011
127	38	165	0.0023	28	107	0.0084	515	1576	2.12E−02
128	12	24	5.99E−04	30	60	0.0013	9	18	4.70E−04
129	7	17	5.89E−04	7	17	4.39E−04	13	27	6.80E−04
130	1874	29130	1.94E−01	178	1785	2.41E−02	69	1379	0.0126

(*continued on next page*)

Table 2 (*continued*).

Problem	NHS+			NPRP+			HTTHSLS		
	NOI	NOF	CPU	NOI	NOF	CPU	NOI	NOF	CPU
131	2149	31230	3.9207	413	5743	0.6204	118	2077	0.2089
132	8946	125214	105.6371	1282	18556	12.8012	62	1725	1.13
133	12	85	0.0309	11	74	0.0294	9	71	0.0125
134	19	266	1.6273	19	222	1.5759	–	–	–
135	19	236	2.9363	16	202	2.9738	–	–	–
136	8	27	0.0025	8	27	7.09E−04	5	20	4.49E−04
137	12	47	0.0012	10	40	0.0024	10	47	0.0011
138	24	112	0.011	22	106	0.01	11	73	0.007
139	1067	16397	0.1343	97	852	0.0123	74	1318	0.0182
140	5151	74429	24.8758	1199	15445	2.8983	1113	15504	2.5074
141	–	–	–	4801	62584	132.214	2764	37546	64.1872
142	710	9171	0.0769	196	3528	0.0271	175	3223	0.0224
143	1816	29942	0.1568	1041	18481	0.1006	1013	18019	0.0888
144	3102	50617	0.2709	1984	35138	0.168	1957	34624	0.2024

3.1. Motion control of robot manipulator

This subsection will examine the efficiency of the hybrid method in solving motion control of robot manipulator problem. For the experiment, we use the Armijo line search (25) with $\rho = 0.6$, $\vartheta = 0.018$ and the direction given by (13)–(15) with $\bar{t} = 0.3$ and $\mu = 0.01$. These parameters were chosen because they provide the best results out of so many options. Consider the motion control of two-joint planar robot manipulator [36,37]. Moreover, we can extend the motion control of robot manipulator with more joints as follows.

Suppose $r_{dk} \in \mathbb{R}^2$ is the desired path vector at time instant $t_k \in [0, t_f]$, then the discrete-time kinematics equation of two-joint planar robot manipulator at the position level is equivalent to finding the joint angle vector $\upsilon_k \in \mathbb{R}^2$ such that

$$f(\upsilon_k) = r_{dk}, \tag{36}$$

where $f(\cdot)$ is the kinematics function defined as

$$f(\upsilon) = \begin{bmatrix} l_1 a_1 + l_2 a_2 \\ l_1 b_1 + l_2 b_2 \end{bmatrix},$$

where $a_1 = \cos(\upsilon_1)$, $b_1 = \sin(\upsilon_1)$, $a_2 = \cos(\upsilon_1 + \upsilon_2)$, $b_2 = \sin(\upsilon_1 + \upsilon_2)$, and l_i is the length of the ith rod ($i = 1, 2$). However, problem (36) can be written as an optimization problem as follows

$$\min_{\upsilon_k \in \mathbb{R}^2} \frac{1}{2} \|f(\upsilon_k) - r_{dk}\|^2.$$

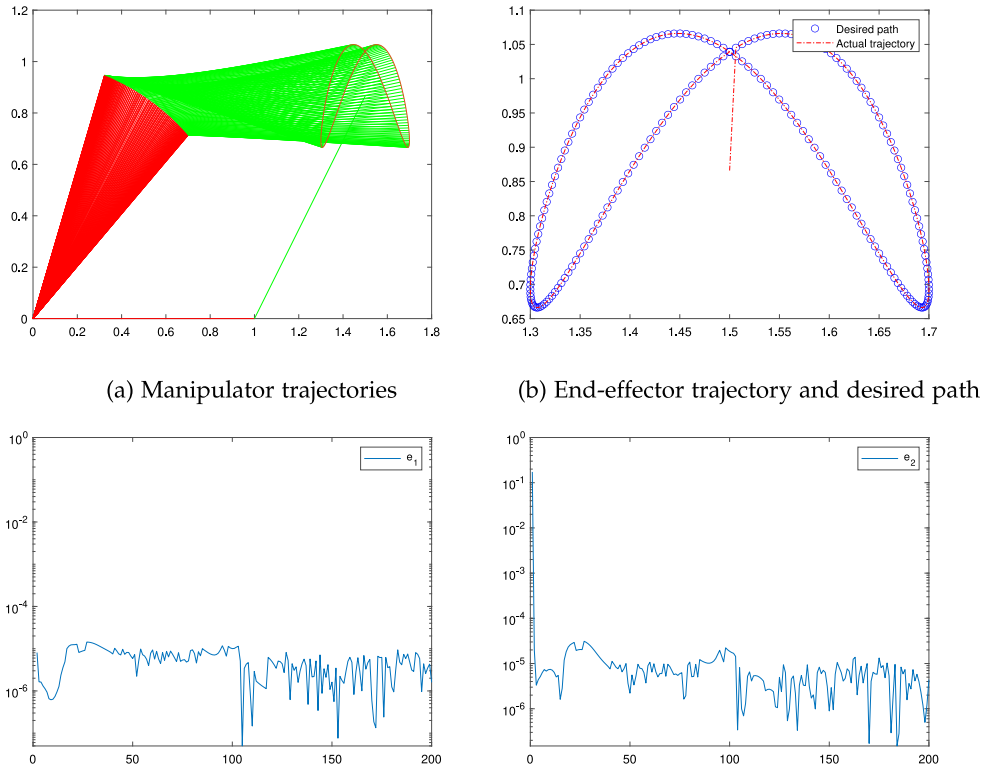
To make things easy, we choose $l_i = 1 \quad (i = 1, 2)$ and the end-effector is controlled to track the following Lissajous curve [36]

$$r_{dk} = \begin{bmatrix} 1.5 + 0.2 \sin(\pi t_k/5) \\ \sqrt{3}/2 + 0.2 \sin(2\pi t_k/5 + \pi/3) \end{bmatrix}.$$

The starting point υ_0 is carefully selected as $\upsilon_0 = [0, \pi/3]^\top$, and the starting point of the ith optimization problem is set as the approximate optimal solution of the $i-1$-th optimization problem ($i = 2, 3, \ldots$). Also, the experiment duration is [0, 10] and divided into 200 equal parts. The results of the experiment is given in Fig. 4, which illustrates that the proposed method successfully completes the motion control. In particular, the robot trajectories synthesized by the proposed method is drawn in Fig. 4(a), and the end-effector trajectory and desired path is plotted in Fig. 4(b). Figs. 4(c) and 4(d) show the error generated by the proposed method on x-axis and y-axis, respectively. In conclusion, Figs. 4(c) and 4(d) show that the generated errors are below 10^{-5} throughout the experiment duration.

3.2. Image recovery

In this section, we evaluate the performance of HTTHSLS algorithm in image restoration. It is a common knowledge that images get corrupted by impulse noise as a result of noisy sensors or channels of communication. Impulse noise is among common noise models where only a portion of the pixels gets contaminated. Several image related applications in most cases require some good techniques of noise suppression in order to produce reliable results to restore the original image. This problems are mostly taken as tedious problems in optimization due to their non-smoothness. In recent times, researchers are focused on developing algorithms for restoring corrupted images, see [38,39] and references therein. For CG algorithms for image restoration, see [40–44] and references therein.



(a) Manipulator trajectories (b) End-effector trajectory and desired path

(c) Tracking errors on the horizontal x-axis (d) Tracking errors on the vertical y-axis

Fig. 4. Numerical results generated by HTTHSLS.

In particular, a two phase algorithm for removing salt and pepper noise was proposed by Chan et al. [45]. The first phase used an adaptive median filter to recognize noisy pixels and in phase two, the noisy pixels are restored by minimizing the function below:

$$\Phi(v) = \sum_{(i,j)\in\mathcal{N}} |v_{ij} - \eta_{mn}| + \lambda \sum_{(i,j)\in\mathcal{N}} \sum_{(mn)\in U_{ij}\setminus\mathcal{N}} \Phi_\alpha(v_{ij} - \eta_{mn}) + \frac{\lambda}{2} \sum_{(i,j)\in\mathcal{N}} \sum_{(mn)\in U_{ij}\cap\mathcal{N}} \Phi_\alpha(v_{ij} - \eta_{mn}), \tag{37}$$

where λ is the regularization parameter. This method is capable of preserving edges during denoising but may not work well in detecting noisy pieces. As such, the non-smooth term $|v_{ij} - \eta_{mn}|$ in (37) is removed and the following smooth image restoration problem is obtained as follows:

$$\min \Phi(v), \tag{38}$$

where

$$\Phi(v) = \sum_{(i,j)\in\mathcal{N}} \sum_{(mn)\in U_{ij}\setminus\mathcal{N}} \Phi_\alpha(v_{ij} - \eta_{mn}) + \frac{1}{2} \sum_{(i,j)\in\mathcal{N}} \sum_{(mn)\in U_{ij}\cap\mathcal{N}} \Phi_\alpha(v_{ij} - \eta_{mn}),$$

$$\mathcal{N} = \{(i,j) \in B : \bar{\eta_{ij}} \neq \eta_{ij}, \eta_{ij} = s_{min} \text{ or } S_{max}\},$$

is the set for the noisy pixel, X is the original image of $M \times N$ pixels, X_{ij} denotes the gray level of X at the pixel (i,j), for all $(i,j) \in B = \{1,2,\ldots,M\} \times \{1,2,\ldots,N\}$, and

$$U_{ij} = \{(i,j-1),(i,j+1),(i-1,j),(i+1,j)\},$$

is a neighborhood of (i,j), η is the observed noisy image of X distorted by salt and pepper, $\bar{\eta}$ is gotten by using an adaptive median filter to the noisy image η, s_{min} and S_{max} denote the minimum and maximum of a noisy pixel, respectively and Φ_α is an edge preserving functional taken as $\Phi_\alpha(a) = \sqrt{a^2 + \alpha}$.

Table 3
Image restoration outputs based on Iter, CPUt and PSNR.

Image	Noise ratio	FHTTCGM_N			HTTHSLS			HZ		
		Iter	CPUt	PSNR	Iter	CPUt	PSNR	Iter	CPUt	PSNR
Lena	30%	12	4.28	37.00	14	4.46	36.97	20	4.92	36.93
Lena	50%	19	7.53	34.43	14	6.58	34.34	20	7.64	34.36
Lena	70%	18	9.26	31.02	19	9.52	31.05	31	12.65	31.15
Lena	90%	28	15.67	26.13	28	15.23	26.19	43	22.45	26.24
Hill	30%	14	4.61	34.95	18	5.28	34.98	14	4.53	34.93
Hill	50%	19	7.77	32.63	16	8.59	32.59	20	7.97	32.58
Hill	70%	20	10.30	29.72	19	9.66	29.67	29	12.23	29.77
Hill	90%	33	16.75	25.60	33	17.11	25.58	46	22.35	25.52
Man	30%	16	4.49	31.54	14	4.23	31.54	20	4.96	31.54
Man	50%	19	7.88	29.14	19	7.83	29.15	23	8.19	29.11
Man	70%	27	11.58	26.27	22	10.15	26.25	32	13.73	26.25
Man	90%	34	19.25	22.48	31	15.94	22.44	50	22.12	22.43
Boat	30%	16	4.71	33.66	16	4.59	33.66	18	4.79	33.64
Boat	50%	23	8.75	31.15	20	7.89	31.16	23	9.21	31.11
Boat	70%	23	10.74	28.22	21	10.02	28.23	26	11.08	28.26
Boat	90%	35	18.09	24.10	31	16.20	24.06	42	21.49	24.02
Thai fabric pattern in Phetchabun	30%	16	3.96	35.50	14	3.86	35.52	18	4.45	35.49
Thai fabric pattern in Phetchabun	50%	17	7.06	32.59	17	7.02	32.56	17	7.07	32.54
Thai fabric pattern in Phetchabun	70%	22	9.83	29.21	20	9.22	29.18	31	12.57	29.24
Thai fabric pattern in Phetchabun	90%	32	17.46	25.09	26	14.11	24.95	38	23.04	25.07

Fig. 5. Original images (Lena, Hill, Man, Boat and Thai fabric pattern in Phetchabun).

In this experiments, we consider Lena, Hill, Man and Boat as the test images with 512×512 gray level. We then apply the HTTHSLS algorithm to solve (38) and compare it with FHTTCGM-N proposed by Jiang et al. [46] and HZ algorithm, respectively. The weak Wolfe line search was considered for the experiments and the used the following as the stopping criterion:

$$Iter > 300 \quad \text{or} \quad \frac{|\Phi_\alpha(v_k) - \Phi_\alpha(v_{k-1})|}{|\Phi_\alpha(v_k)|} \leq 10^{-4}.$$

In addition, to assess the quality of restoration, the peak signal to noise ratio (PSNR) defined by:

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{M \times N} \|X^* - X\|_F^2},$$

where X^* denotes the restored image and $\|\|_F$ is the Frobenius norm.

Table 3 has the reports of number of iterations (Iter), the computer time (CPUt) and the PSNR values of each algorithm. Fig. 5 has the original images, while Figs. 6 and 7 has the blurred and restored image by each algorithm with 70% and 90% noise ratio, respectively. From Fig. 5, it can be seen that, HTTHSLS has the least Iter and CPUt in most cases compared with FHTTCGM-N and HZ. Furthermore, HTTHSLS has the highest PSNR in most cases. In this regard, we can say that HTTHSLS is superior to the other two algorithms.

4. Conclusion

In this article, we present a hybrid CG algorithm that combined the HS and LS CG parameters. The search direction of the algorithm is sufficiently descent and bounded independent of the line search. Moreover, the step size was obtained via Wolfe and Armijo line searches. Convergence of the algorithm was established under suitable assumptions. To support the convergence results, three kind of numerical experiments were reported; the first on solving benchmark unconstrained optimization problems, the second is on motion control of robotic arm and finally on image restoration. In all three experiments, the proposed algorithm called HTTHSLS can be seen as a better alternative compared with other algorithms at it proves to be more efficient and robust. However, we observed some challenges and limitations of the proposed algorithm; (a) the algorithm has high computational cost, (b) finding an optimal control parameters that will give the best results are among the challenges. In case (a), it is likely due to the hybridization which involves so many computations.

Fig. 6. Blurred images with 70% ratio (first row), rsetoration by FHTTCGM-N (second row), rsetoration by HTTHSLS (third row), rsetoration by HZ (bottom row).

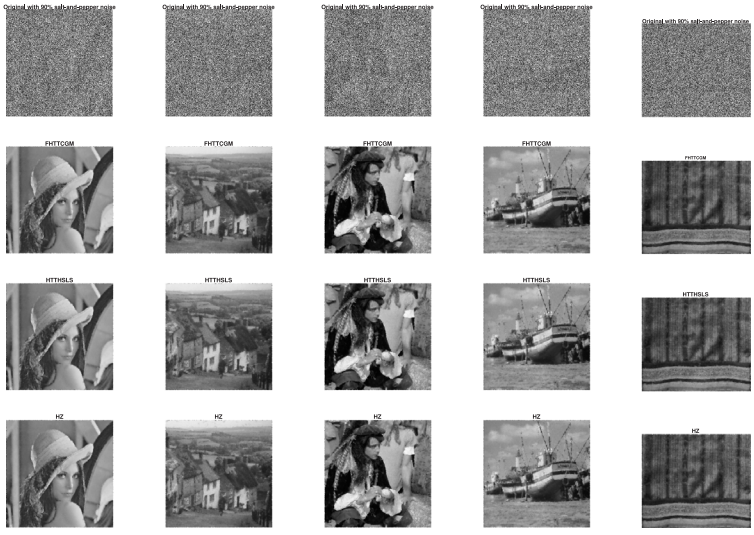


Fig. 7. Blurred images with 90% ratio (first row), rsetoration by FHTTCGM-N (second row), rsetoration by HTTHSLS (third row), rsetoration by HZ (bottom row).

Data availability

Data will be made available on request.

Acknowledgments

The second author acknowledge with thanks, the Department of Mathematics and Applied Mathematics at the Sefako Makgatho Health Sciences University, Thailand. The fifth author was partially supported by Phetchabun Rajabhat University, Thailand and Thailand Science Research and Innovation (grant number 182093). The sixth author was partially supported by Chiang Mai University, Thailand and the NSRF via the Program Management Unit for Human Resources and Institutional Development, Research and Innovation, Thailand (grant number B05F640183). Moreover, this project is funded by National Research Council of Thailand (NRCT) under Research Grants for Talented Mid-Career Researchers (Contract no. N41A640089).

References

[1] J.E. Dennis, J.J. Moré, A characterization of superlinear convergence and its application to quasi-Newton methods, Math. Comp. 28 (126) (1974) 549–560.
[2] J.E. Dennis, J.J. Moré, Quasi-Newton methods, motivation and theory, SIAM Rev. 19 (1) (1977) 46–89.
[3] P.E. Gill, W. Murray, Newton-type methods for unconstrained and linearly constrained optimization, Math. Program. 7 (1) (1974) 311–350.
[4] W.W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pac. J. Optim. 2 (1) (2006) 35–58.
[5] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (2) (1964) 149–154.
[6] E. Polak, G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, Rev. Fr. Inform. Rech. Oper. Ser. Rouge 3 (16) (1969) 35–43.
[7] B.T. Polyak, The conjugate gradient method in extremal problems, USSR Comput. Math. Math. Phys. 9 (4) (1969) 94–112.
[8] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (6) (1952) 409–436.
[9] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, 2013.
[10] Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms, part 1: Theory, J. Optim. Theory Appl. 69 (1) (1991) 129–137.
[11] Y.H. Dai, Y.X. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim. 10 (1) (1999) 177–182.
[12] J.K. Liu, S.J. Li, New hybrid conjugate gradient method for unconstrained optimization, Appl. Math. Comput. 245 (2014) 36–43.
[13] X. Xu, F. Kong, New hybrid conjugate gradient methods with the generalized wolfe line search, SpringerPlus 5 (1) (2016) 1–10.
[14] X.L. Dong, D.R. Han, R. Ghanbari, X.L. Li, Z.F. Dai, Some new three-term Hestenes–Stiefel conjugate gradient methods with affine combination, Optimization 66 (5) (2017) 759–776.
[15] M. Li, A three term Polak–Ribière–Polyak conjugate gradient method close to the memoryless BFGS quasi-Newton method, J. Ind. Manag. Optim. 16 (1) (2020) 245.
[16] M. Li, A modified Hestense–Stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method, Optim. Methods Softw. 33 (2) (2018) 336–353.
[17] G. Yuan, J. Lu, Z. Wang, The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems, Appl. Numer. Math. 152 (2020) 1–11.
[18] G. Yuan, J. Lu, Z. Wang, The modified PRP conjugate gradient algorithm under a non-descent line search and its application in the muskingum model and image restoration problems, Soft Comput. 25 (2021) 5867–5879.
[19] A.B. Abubakar, M. Malik, P. Kumam, H. Mohammad, M. Sun, A.H. Ibrahim, A.I. Kiri, A liu-storey-type conjugate gradient method for unconstrained minimization problem with application in motion control, J. King Saud Univ., Sci. 34 (4) (2022) 101923.
[20] N. Andrei, Accelerated adaptive perry conjugate gradient algorithms based on the self-scaling memoryless BFGS update, J. Comput. Appl. Math. 325 (2017) 149–164.
[21] Z. Khoshgam, A. Ashrafi, A new hybrid conjugate gradient method for large-scale unconstrained optimization problem with non-convex objective function, Comput. Appl. Math. 38 (4) (2019) 186.
[22] L. Zhang, W. Zhou, D.H. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, IMA J. Numer. Anal. 26 (4) (2006) 629–640.
[23] J.C. Gilbert, J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, SIAM J. Optim. 2 (1) (1992) 21–42.
[24] M. Al-Baali, Descent property and global convergence of the Fletcher–Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1) (1985) 121–124.
[25] W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM J. Optim. 16 (1) (2005) 170–192.
[26] J. Deepho, A.B. Abubakar, M. Malik, I.K. Argyros, Solving unconstrained optimization problems via hybrid CD-dy conjugate gradient methods with applications, J. Comput. Appl. Math. 405 (2022) 113823.
[27] A.H. Ibrahim, P. Kumam, A. Kamandi, A.B. Abubakar, An efficient hybrid conjugate gradient method for unconstrained optimization, Optim. Methods Softw. 37 (4) (2022) 1370–1383.
[28] A.B. Abubakar, P. Kumam, M. Malik, A.H. Ibrahim, A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems, Math. Comput. Simulation 201 (2022) 640–657.
[29] J. Nocedal, Updating quasi-Newton matrices with limited storage, Math. Comp. 35 (151) (1980) 773–782.
[30] D.F. Shanno, Conjugate gradient methods with inexact searches, Math. Oper. Res. 3 (3) (1978) 244–256.
[31] M. Li, A modified hestense–stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method, Optim. Methods Softw. 33 (2) (2018) 336–353.
[32] W. Sun, Y.X. Yuan, Optimization Theory and Methods: Nonlinear Programming, Springer, 2006.
[33] L. Grippo, S. Lucidi, A globally convergent version of the polak-ribière conjugate gradient method, Math. Program. 78 (3) (1997) 375–391.
[34] N. Andrei, Nonlinear Conjugate Gradient Methods for Unconstrained Optimization, Springer, 2020.
[35] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2) (2002) 201–213.
[36] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, Y. Shi, General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization, J. Comput. Appl. Math. 347 (2019) 314–329.
[37] M. Sun, J. Liu, Y. Wang, Two improved conjugate gradient methods with application in compressive sensing and motion control, Math. Probl. Eng. 2020 (2020).
[38] K. Kankam, W. Cholamjiak, P. Cholamjiak, Convergence analysis of a modified forward–backward splitting algorithm for minimization and application to image recovery, Comput. Math. Methods 2022 (2022).
[39] S. Kesornprom, P. Cholamjiak, C. Park, New proximal type algorithms for convex minimization and its application to image deblurring, Comput. Appl. Math. 41 (7) (2022) 333.
[40] J. Cao, J. Wu, A conjugate gradient algorithm and its applications in image restoration, Appl. Numer. Math. 152 (2020) 243–252.
[41] G. Yu, J. Huang, Y. Zhou, A descent spectral conjugate gradient method for impulse noise removal, Appl. Math. Lett. 23 (5) (2010) 555–560.
[42] G. Yuan, T. Li, W. Hu, A conjugate gradient algorithm and its application in large-scale optimization problems and image restoration, J. Inequal. Appl. 2019 (1) (2019) 247.
[43] G. Yuan, T. Li, W. Hu, A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems, Appl. Numer. Math. 147 (2020) 129–141.
[44] S. Babaie-Kafaki, N. Mirhoseini, Z. Aminifard, A descent extension of a modified Polak–Ribière–Polyak method with application in image restoration problem, Optim. Lett. (2022) 1–17.
[45] R.H Chan, C. Ho, M. Nikolova, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, IEEE Trans. Image Process. 14 (10) (2005) 1479–1485.
[46] X. Jiang, W. Liao, J. Yin, J. Jian, A new family of hybrid three-term conjugate gradient methods with applications in image restoration, Numer. Algorithms (2022) 1–31.