mathematics

Article

A New Variant of the Conjugate Descent Method for Solving Unconstrained Optimization Problems and Applications

Aliyu Muhammed Awwal [1,2], Mahmoud Muhammad Yahaya [3], Nuttapol Pakkaranang [4] and Nattawut Pholasa [5,*]

[1] Department of Mathematical Sciences, Faculty of Science, Gombe State University, Gombe 760214, Nigeria; maaliyu@gsu.edu.ng
[2] GSU-Mathematics for Innovative Research Group, Gombe State University, Gombe 760214, Nigeria
[3] KMUTT-Fixed Point Theory and Applications Research Group, Theoretical and Computational Science Center (TaCS), Science Laboratory Building, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand; mahmoudpd@gmail.com
[4] Mathematics and Computing Science Program, Faculty of Science and Technology, Phetchabun Rajabhat University, Phetchabun 67000, Thailand; nuttapol.pak@pcru.ac.th
[5] School of Science, University of Phayao, Phayao 56000, Thailand
* Correspondence: nattawut_math@hotmail.com

Abstract: Unconstrained optimization problems have a long history in computational mathematics and have been identified as being among the crucial problems in the fields of applied sciences, engineering, and management sciences. In this paper, a new variant of the conjugate descent method for solving unconstrained optimization problems is introduced. The proposed algorithm can be seen as a modification of the popular conjugate descent (CD) algorithm of Fletcher. The algorithm of the proposed method is well-defined, and the sequence of the directions of search is shown to be sufficiently descending. The convergence result of the proposed method is discussed under the common standard conditions. The proposed algorithm together with some existing ones in the literature is implemented to solve a collection of benchmark test problems. Numerical experiments conducted show the performance of the proposed method is very encouraging. Furthermore, an additional efficiency evaluation is carried out on problems arising from signal processing and it works well.

Keywords: conjugate descent; conjugate gradient method; unconstrained optimization; line search; signal processing

MSC: 65K05; 90C30; 90C06; 90C56

Citation: Awwal, A.M.; Yahaya, M.M.; Pakkaranang, N.; Pholasa, N. A New Variant of the Conjugate Descent Method for Solving Unconstrained Optimization Problems and Applications. *Mathematics* **2024**, *12*, 2430. https://doi.org/10.3390/math12152430

Academic Editors: Dhananjay Gopal and Carlo Bianca

Received: 30 June 2024
Revised: 2 August 2024
Accepted: 2 August 2024
Published: 5 August 2024

1. Introduction

Consider the Euclidean n-dimensional real space equipped with the Euclidean norm $\|\cdot\|$. Many problems encountered in the sciences, engineering, and management sciences often take the following structure:

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

where the continuously smooth function is real-valued and assumed to be bounded below. Problem (1) is termed an unconstrained optimization problem, and it has attracted considerable attention from researchers in the last few decades due to its practical applications [1–7]. One of the popular methods often used to handle (1) is the conjugate gradient (CG) method, and this method uses the following iterative rule:

$$x^{(k+1)} := x^{(k)} + \alpha^{(k)} d^{(k)}, \quad k \geq 0, \tag{2}$$

where $x^{(k)}$ and $x^{(k+1)}$ are the current and next iterates, respectively. The positive scalar $\alpha^{(k)}$, known as the step length, and the $n \times 1$ nonzero vector $d^{(k)}$, called the search direction, are very crucial components of Formula (2). $\alpha^{(k)}$ is generally ascertained using a suitable line search strategy, which could be exact or inexact. Inexact line search procedures, such as the generalized Wolfe or strong Wolfe line searches, are the most appealing due to their relative ease of usage compared to the exact line search. Some of the frequently used inexact line search conditions have been discussed in Refs. [8,9].

The direction of search, d_k, for the minimizer takes the following structure:

$$d^{(k)} = -g^{(k)} + \psi^{(k)} d^{(k-1)}, \ \ k \geq 0, \tag{3}$$

where $g^{(k)} \equiv \bigtriangledown f(x^{(k)})$. The scalar $\psi^{(k)}$, often known as the CG parameter, usually influences the behaviour of the search direction. If $k = 0$, then $\psi^{(k)} = 0$, otherwise, it is calculated via suitable formulations. Some of these formulations found in the literature are presented below:

$$\psi^{(k)HS} = \frac{g^{(k)T} y^{(k-1)}}{d^{(k-1)T} y^{(k-1)}}, \quad \psi^{(k)PRP} = \frac{g^{(k)T} y^{(k-1)}}{\|g^{(k-1)}\|^2}, \quad \psi^{(k)LS} = \frac{g^{(k)T} y^{(k-1)}}{-g^{(k-1)T} d^{(k-1)}},$$
$$\psi^{(k)FR} = \frac{\|g^{(k)}\|^2}{\|g^{(k-1)}\|^2}, \quad \psi^{(k)CD} = \frac{\|g^{(k)}\|^2}{-d^{(k-1)T} g^{(k-1)}}, \quad \psi^{(k)DY} = \frac{\|g^{(k)}\|^2}{d^{(k-1)T} y^{(k-1)}},$$

where HS, PRP, LU, FR, CD, and DY denote Hestenes–Stiefel [10], Polak–Ribiere–Polyak (PRP) [11,12], Liu–Storey (LS) [13], Fletcher–Reeves (FR) [14], conjugate descent (CD) [15], and Dai–Yuan (DY) [16], respectively. All these parameters have their pros and cons, as discussed by different authors. Interested readers may refer to reference [17]. It was noted in the survey by Hager and Zhang [18] that if the objective function of problem (1) is strongly convex quadratic and the step length is determined using exact line search, then the above-listed CG parameters are equivalent in theory. The CD method has been shown to be sufficiently descending if $\alpha^{(k)}$ satisfies the condition of the strong Wolfe line search strategy, and consequently, global convergence is achieved. However, Hager and Zhang [18] noted that there is an example where the norm square of the search direction, $d^{(k)}$, increases rapidly, which results in the CD method failing to converge for the strong Wolfe line search in general.

This research focuses its attention on the set of CG methods whose parameters contain $\|g^{(k)}\|^2$ in their respective numerators. This set of CG methods is characterized by simplicity in implementation and low storage requirements. However, many authors have raised concerns about their numerical performance as they are affected by jamming phenomena. Hence, some authors proposed different modifications to mitigate the said shortcomings. For instance, some authors considered taking the hybrid of two different parameters to come up with another version that could be numerically efficient. Babaie-Kafaki [17] takes the HS and DY parameters based on the well-known conjugacy condition. The effect of the hybridization is evident, as the method performs better than its counterparts numerically. In addition, the author establishes the global convergence of the hybrid method under some assumptions. Another hybrid CG method found in the literature defines its CG parameter as the convex combination of CD and LS [19]. The author determines the convex combination parameter in such a way that the conjugacy condition is satisfied. Numerical comparisons reveal some superior performance of the hybrid method compared to some existing algorithms. Another form of modification to these methods is incorporating spectral parameters into the search direction by multiplying the first term of the search direction, i.e., $-g^{(k)}$, with a positive parameter that is updated in each iteration. Xue et al. [20] presented a spectral version of the DY CG method which ensured that the objective function is descending as the iteration progresses. The global convergence of their method was established under strong Wolfe conditions. Moreover, the numerical efficiency of the method was also demonstrated in experiments involving impulse noise removal. For more details, readers may refer to the following references [21–29].

Based on the discussions thus far, one sees that research continues to explore ways to improve the theoretical and numerical efficiency of the existing CG methods. Thus, this article presents a modification of the CD CG method. Using the strategy of mathematical induction, the direction of search for the proposed CG method is shown to be descent. Furthermore, the sequence of the search direction is shown to be bounded, independent of any additional condition. The global convergence of the proposed method is established under common assumptions, and numerical experiments on a collection of some benchmark test problems are encouraging. The applicability of the proposed method is demonstrated in signal processing.

The rest of the article is organized by presenting the proposed method and its algorithm as well as its convergence results in the next section. The numerical efficiency of the proposed algorithm is investigated in Section 3, and subsequently, its application in sparse signal reconstruction is demonstrated in Section 4. Finally, the concluding remarks are presented in Section 5.

2. The Proposed Conjugate Descent Variant (CDV) Algorithm and Its Convergence Result

This section begins by stating the following standard assumption.

Assumption 1. *Let $x^{(0)} \in \mathbb{R}$ denote an initial guess. The objective function f at $x \in \mathbb{R}^n$ is bounded below on the level set $\mathcal{Z} = \{x \in \mathbb{R}^n | f(x^{(0)}) \geq f(x)\}$. In addition, throughout some neighborhood of $\mathcal{Z}$, the function f is smooth and its gradient is Lipchitzian.*

Remark 1. *One can quickly draw the following remarks from the above Assumption 1.*

(i) *Given any two different iterates $x^{(k)}$ and $x^{(k-1)}$ in $\widehat{\mathcal{Z}}$, i.e., neighborhood of $\mathcal{Z}$, the gradient of the objective function satisfies the following inequality:*

$$\|g(x^{(k)}) - g(x^{(k-1)})\| \leq L\|x^{(k)} - x^{(k-1)}\|, \quad L > 0. \tag{4}$$

(ii) *Also, the sequences of the gradient $\{g(x^{(k)})\}$ as well as $\{x^{(k)}\}$ are bounded. That is, we can find a constant $r > 0$ such that*

$$\|g(x^{(k)})\| \leq r, \quad \text{and} \quad \|x^{(k)}\| \leq r, \forall\ k. \tag{5}$$

(iii) *Since the objective function is a decreasing function and the sequence of iterates $\{x^{(k)}\}$ generated by Algorithm 1 is contained in a bounded region, then $\{x^{(k)}\}$ converges.*

All is now set to present the proposed conjugate descent variant (CDV) algorithm.

Remark 2. *Since $g^{(k)}$ and $d^{(k-1)}$ are nonzero vectors, then leveraging the fact that there exists an α which satisfies the conditions (7) and (8) in a finite number of iterations gives the conclusion that the CDV Algorithm 1 is well defined.*

The following lemma shows that the proposed method is sufficiently descending.

Algorithm 1: A new Conjugate Descent Variant (CDV) algorithm

Input : $x^{(0)} \in \mathbb{R}^n$, $\sigma, \delta \in (0,\ 1)$ such that $0 < \delta < \sigma < 1$, $tol > 0$, $\alpha^{(0)} = 1$, and set $k := 0$.
Output: $x^{(*)}$
Compute $g^{(0)}$.
Set

$$d^{(0)} \longleftarrow -g^{(0)} \tag{6}$$

while $\|g^{(k)}\| > tol$ **do**

Choose, $\alpha^{(k)}$ that satisfy the following conditions:

$$f(x^{(k)} + \alpha^{(k)} d^{(k)}) \leq f^{(k)} + \delta \alpha^{(k)} g^{(k)T} d^{(k)}, \tag{7}$$

$$g^{(k+1)T} d^{(k)} \geq \sigma g^{(k)T} d^{(k)}, \tag{8}$$

Compute,

$$d^{(k)} := -g^{(k)} + \psi^{(k)} d^{(k-1)} \tag{9}$$

where $\psi^{(k)}$ is defined as

$$\psi^{(k)} := \frac{\delta \|g^{(k)}\|^2}{\max\{\delta d^{(k-1)T} g^{(k)} - g^{(k-1)T} d^{(k-1)},\ \|g^{(k)}\| \|d^{(k-1)}\|\}} \tag{10}$$

Update $x^{(k+1)} \longleftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$.
Set $k \longleftarrow k + 1$.

end

Lemma 1. *Let $\sigma, \delta \in (0,1)$. The sequence of search directions $\{d^{(k)}\}$ generated by Algorithm 1 is sufficiently descending, that is, we can find a positive constant $0 < z = (1 - \delta)$ for which the condition*

$$g^{(k)T} d^{(k)} \leq -z \|g^{(k)}\|^2, \ \ \forall\, k, \tag{11}$$

holds.

Proof. We prove (11) by mathematical induction. Indeed, if $k = 0$, then from step 2 of Algorithm 1, we have $g^{(0)T} d^{(0)} \leq -\|g^{(0)}\|^2 \leq -z\|g^{(0)}\|^2$. That is, (11) holds for $k = 0$. Now, assume the condition (11) holds for $k - 1$, then we have

$$g^{(k-1)T} d^{(k-1)} \leq -z \|g^{(k-1)}\|^2. \tag{12}$$

Next, we show that conclusion (11) holds for k. From the curvature condition (8) and the inequality (12), it is easy to establish that

$$\delta d^{(k-1)T} g^{(k)} - d^{(k-1)T} g^{(k-1)} \geq (\sigma\delta - 1) d^{(k-1)T} g^{(k-1)} > 0. \tag{13}$$

Next, we have two cases.

Case 1: Let $\max\{\delta d^{(k-1)T}g^{(k)} - g^{(k-1)T}d^{(k-1)},\ \|g^{k}\|\|d^{k-1}\|\} = \delta d^{(k-1)T}g^{(k)} - g^{(k-1)T}d^{(k-1)}$, then

$$\begin{aligned}
g^{(k)T}d^{(k)} &= -g^{(k)T}g^{(k)} + \psi^{(k)}g^{(k)T}d^{(k-1)} \\
&\leq -\|g^{(k)}\|^2 + \frac{\delta\|g^{(k)}\|^2}{\delta d^{(k-1)T}g^{(k)} - g^{(k-1)T}d^{(k-1)}}g^{(k)T}d^{(k-1)} \\
&= \|g^{(k)}\|^2\left[-1 + \frac{\delta g^{(k)T}d^{(k-1)}}{\delta d^{(k-1)T}g^{(k)} - g^{(k-1)T}d^{(k-1)}}\right] \\
&= \|g^{(k)}\|^2\left[\frac{g^{(k-1)T}d^{(k-1)}}{\delta d^{(k-1)T}g^{(k)} - g^{(k-1)T}d^{(k-1)}}\right] \\
&\leq \|g^{(k)}\|^2\left[\frac{g^{(k-1)T}d^{(k-1)}}{(\sigma\delta - 1)g^{(k-1)T}d^{(k-1)}}\right] \\
&= -\frac{1}{(1-\sigma\delta)}\|g^{(k)}\|^2 \\
&< -(1-\delta)\|g^{(k)}\|^2.
\end{aligned}$$

The last inequality follows from the fact that if $a, b \in (0,1)$, then $\frac{1}{a} > b$.

Case 2: Let $\max\{\delta d^{(k-1)T}g^{(k)} - g^{(k-1)T}d^{(k-1)},\ \|g^{(k)}\|\|d^{(k-1)}\|\} = \|g^{(k)}\|\|d^{(k-1)}\|$, then

$$\begin{aligned}
g^{(k)T}d^{(k)} &= -g^{(k)T}g^{(k)} + \psi^{(k)}g^{(k)T}d^{(k-1)} \\
&= -\|g^{(k)}\|^2 + \frac{\delta\|g^{(k)}\|^2}{\|g^{(k)}\|\|d^{(k-1)}\|}g^{(k)T}d^{(k-1)} \\
&\leq -\|g^{(k)}\|^2 + \frac{\delta\|g^{(k)}\|^2}{\|g^{(k)}\|\|d^{(k-1)}\|}|g^{(k)T}d^{(k-1)}| \\
&\leq -\|g^{(k)}\|^2 + \frac{\delta\|g^{(k)}\|^2}{\|g^{(k)}\|\|d^{(k-1)}\|}\|g^{(k)}\|\|d^{(k-1)}\| \\
&= -(1-\delta)\|g^{(k)}\|^2.
\end{aligned}$$

Hence, the proof is complete. □

Lemma 2. *Suppose the gradient g is Lipschitz continuous. The sequence of search directions $\{d^{(k)}\}$ generated by Algorithm 1 is bounded by a positive number.*

Proof. From the definition of the CG parameter $\{\psi^{(k)}\}$, it holds that

$$|\psi^{(k)}| \leq \frac{\delta\|g^{(k)}\|^2}{\|g^{(k)}\|\|d^{(k-1)}\|},$$

and so,

$$\begin{aligned}
\|d^{(k)}\| &\leq \|g^{(k)}\| + |\psi^{(k)}|\|d^{(k-1)}\| \\
&\leq \|g^{(k)}\| + \frac{\delta\|g^{(k)}\|^2}{\|g^{(k)}\|\|d^{(k-1)}\|}\|d^{(k-1)T}\| \\
&= (1+\delta)\|g^{(k)}\| \\
&= (1+\delta)r := \widehat{r}.
\end{aligned}$$

Hence, the proof is complete. □

Lemma 3. *Suppose Lemmas 1 and 2 hold and $\alpha^{(k)}$ satisfies the conditions (7) and (8). Then*

$$\sum_{k=0}^{\infty} \frac{(g^{(k)T} d^{(k)})^2}{\|d^{(k)}\|^2} < +\infty. \tag{14}$$

Proof. The proof follows directly from the work of Zoutendijk [30]. □

Now, we prove the convergence result of the proposed method.

Theorem 1. *Suppose Assumption 1 holds. Let g denote the gradient of the objective function f and the sequence of iterates $\{x^{(k)}\}$ be produced by Algorithm 1. Then,*

$$\lim_{k\to\infty} \inf \|g^{(k)}\| = 0. \tag{15}$$

Proof. If (15) does not hold, then there exists some constant $c > 0$ for which

$$\|g^{(k)}\| \geq c, \quad k \geq 0. \tag{16}$$

Furthermore, squaring both sides of (11) gives

$$(g^{(k)T} d^{(k)})^2 \geq z^2 \|g^{(k)}\|^4. \tag{17}$$

If we divide both sides of (17) by $\|d^{(k)}\|^2$ and take the summation, we have

$$\sum_{k=0}^{\infty} \frac{(g^{(k)T} d^{(k)})^2}{\|d^{(k)}\|^2} \geq z^2 \sum_{k=0}^{\infty} \frac{\|g^{(k)}\|^4}{\|d^{(k)}\|^2} \geq z^2 \sum_{k=0}^{\infty} \frac{r^4}{\hat{r}^2} = +\infty.$$

This is a contradiction with (14). Hence, (15) holds. □

3. Comparative Experimentation

In this part, we conduct some comparative experimentation between the proposed CDY algorithm and other sets of algorithms, namely, TTCDDY [31], LSCDCC [32], ARMIL+ [33], and CD [15], for solving large-scale unconstrained optimization problems of the form of (1). All the implementation and experiments are carried out on a computer with a 1.60 GHz Intel Core i5-8265U and 20 GB of RAM on the Ubuntu 22.04.4 LTS operating system.

All the code was written in MATLAB, and then, executed on a personal computer, which has the above-stated specifications. The choices for the parameters in the implementation of the CDV are given, while for the rest of the methods, the selection is based on the reported values of those parameters. In brief, we state those initialized values as follows:

1. CDV algorithm:
   The parameters used are $\sigma = 0.01$, $\delta = 0.0001$, and $tol = 0.000001$.
2. TTCDDY algorithm:
   The parameters are as reported in [31].
3. ARMIL+ algorithm:
   We adopted the initialization of the same values for the parameters as reported in [33].
4. LSCDCC algorithm:
   The initialization of the parameters is as reported in [32].
5. CD algorithm[15]:
   The parameter $\psi^{(k)CD}$ used is defined in the introduction section and a Wolfe line search strategy, with $\sigma = 0.01$, $\delta = 0.0001$, and $tol = 10^{-6}$ is adopted

The benchmark problems used in the experimentation are collected from the CUTEr optimization library [34]. The test problems considered have different starting points and dimensions ranging from $n = 100$ to $n = 500{,}000$. Also, the names of these test problems are presented in Table 1.

Moreover, in implementing these algorithms for all the attempted benchmark test problems, a stopping criterion corresponding to obtaining a solution with $\|g^{(k)}\| < tol * (1 + |f^{(k)}|)$ is used, or when the maximum number of iterations, 2000, is reached.

The performance of any new iterative algorithm is usually compared with some selected existing algorithms found in the literature based on some standard metrics. These metrics include ITR (the number of iterations performed by the algorithm), FVAL (the number of times a function is evaluated before the stopping criteria are attained), and GVAL (the number of gradient evaluations throughout the iteration process). Meanwhile, the time taken by an algorithm to complete a given task is also recorded, and it is denoted by CPU. This information for each algorithm is reported in Table 2.

Looking at the data presented in Table 2, it can be seen that the proposed CDV algorithm recorded no failures, except in 2 cases out of the 58 test problems solved. Interestingly, the proposed CDV algorithm can serve as an alternative to some problems. This is evident from the numerical performance of the CDV for problems 15, 27, and 40. This is because only the CDV algorithm was able to solve these problems within the specified stopping criteria. Considering the above-mentioned metrics of comparison, the CDV method performs better than its competitors in most cases, based on the reported information in the table.

The results obtained from the experimentation are graphically illustrated in Figure 1 based on the performance profile of Dolan and Moré [35]. It is obvious that the proposed CDV massively outperforms its competitors with respect to all the metrics considered.

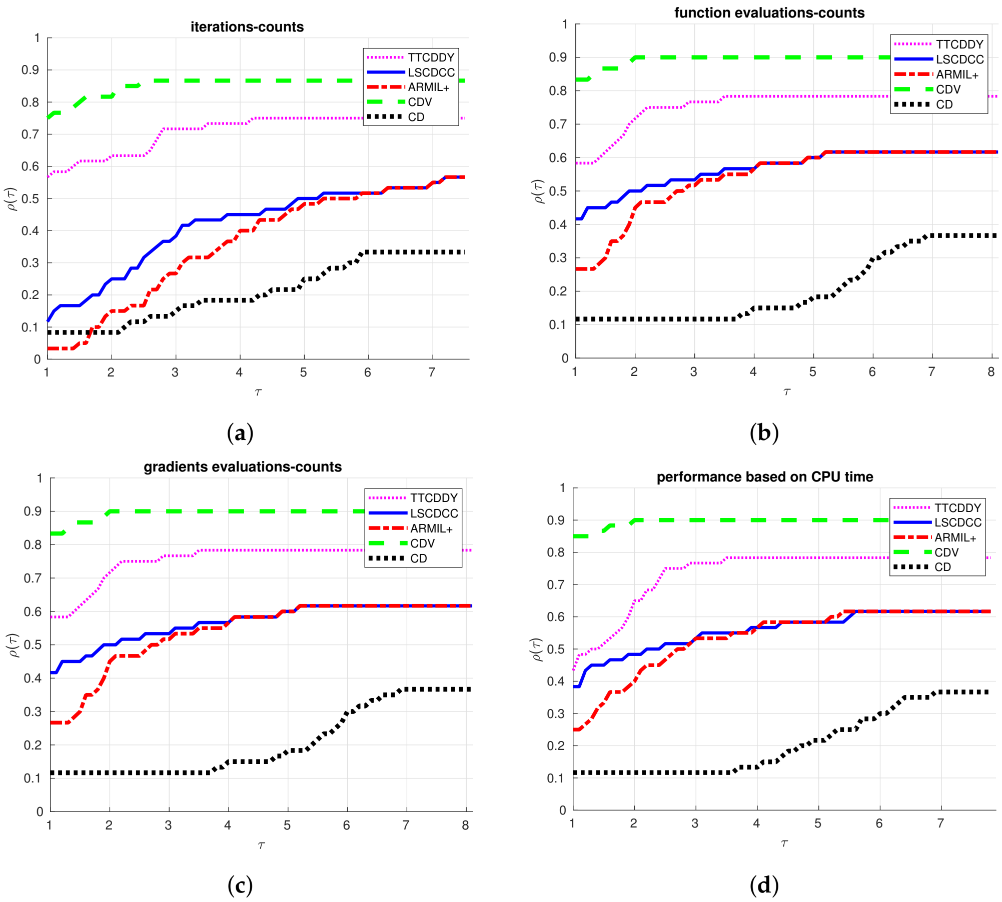

Figure 1. The figures show the performance of the proposed CDV in comparison with the TTCDDY, LSCDCC, ARMIL+ and CD algorithms using four comparative metrics: #ITER, #FVAL, #GVal, and CPU. These performances are indicated on (**a**), (**b**), (**c**), and (**d**) respectively. The y-axis denotes the success rate, which is represented by cumulative probability $\rho(\tau)$, while the x-axis denoted by τ representing a metric data for an algorithm in log_2 scale.

Table 1. List of considered benchmark test problems together with their respective dimensions and starting points written in MATLAB format, where *ones(Dim,1)*:=$[1, 1, \cdots, Dim]$, while *zeros(Dim,1)*:=$[0, 0, \cdots, Dim]$.

No.	Function Name	Dimension	Initial Points
		Dim	$x^{(0)}$
1	COSINE	6000	1.0*ones(Dim,1)
2	COSINE	100,000	1.0*ones(Dim,1)
3	COSINE	500,000	1.0*ones(Dim,1)
4	DIXMAANA	6000	2.0*ones(Dim,1)
5	DIXMAANA	90,000	2.0*ones(Dim,1)
6	DIXMAANB	24,000	2.0*ones(Dim,1)
7	DIXMAANB	48,000	2.0*ones(Dim,1)
8	DIXMAANC	2700	2.0*ones(Dim,1)
9	DIXMAANC	27,000	2.0*ones(Dim,1)
10	DIXMAAND	12,000	2.0*ones(Dim,1)
11	DIXMAAND	90,000	2.0*ones(Dim,1)
12	DIXMAANE	2400	2.0*ones(Dim,1)
13	DQDRTIC	9000	3.0*ones(Dim,1)
14	DQDRTIC	90,000	3.0*ones(Dim,1)
15	DQRTIC	5000	2.0*ones(Dim,1)
16	EDENSCH	7000	zeros(Dim,1)
17	EDENSCH	40,000	zeros(Dim,1)
18	EDENSCH	100,000	zeros(Dim,1)
19	EG2	100	ones(Dim,1)
20	FLETCHCR	1000	zeros(Dim,1)
21	FLETCHCR	50,000	zeros(Dim,1)
22	FLETCHCR	200,000	zeros(Dim,1)
23	GENROSE	10,000	1/(Dim+1)*ones(Dim,1)
24	HIMMELBG	70,000	1.5*ones(Dim,1)
25	PENALTY1	4000	−1.0*ones(Dim,1)
26	PENALTY1	10,000	1.0*ones(Dim,1)
27	QUARTC	4000	2.0*ones(Dim,1)
28	BDEXP	5000	ones(Dim,1)
29	BDEXP	50,000	ones(Dim,1)
30	BDEXP	500,000	ones(Dim,1)
31	EXDENSCHNB	6000	ones(Dim,1)
32	EXDENSCHNB	24,000	ones(Dim,1)
33	GENQUARTIC	9000	ones(Dim,1)
34	GENQUARTIC	90,000	ones(Dim,1)
35	SINE	100,000	ones(Dim,1)
36	SINE	250,000	ones(Dim,1)
37	SINE	500,000	ones(Dim,1)
38	FLETCBV3	100	(1:Dim)/(Dim+1)*ones(Dim,1)
39	NONSCOMP	5000	3.0*ones(Dim,1)
40	NONSCOMP	80,000	3.0*ones(Dim,1)
41	RAYDAN1	500	ones(Dim,1)
42	RAYDAN1	5000	ones(Dim,1)
43	RAYDAN2	2000	ones(Dim,1)
44	RAYDAN2	20,000	ones(Dim,1)
45	RAYDAN2	500,000	ones(Dim,1)
46	DIAGONAL1	800	(1/Dim)*ones(Dim,1)
47	DIAGONAL1	2000	(1/Dim)*ones(Dim,1)
48	DIAGONAL2	8000	(1/(1:Dim))*ones(Dim,1)
49	DIAGONAL3	500	ones(Dim,1)
50	DIAGONAL3	2000	ones(Dim,1)
51	BV	2000	(1:Dim)/(Dim + 1)*((1:Dim)/(Dim + 1) − 1)
52	IE	500	(1:Dim)/(Dim + 1)*((1:Dim)/(Dim + 1) − 1)
53	LIN	100	ones(Dim,1)
54	LIN	1300	ones(Dim,1)
55	OSB2	11	[1.3, 0.65, 0.65, 0.7, 0.6, 3, 5, 7, 2, 4.5, 5.5]
56	PEN2	160	(1/2)*(ones(Dim,1))
57	TRID	500	(−1)*(ones(Dim,1))
58	TRID	8000	(−1)*(ones(Dim,1))

Table 2. The performance of the proposed CDV algorithm in comparison with the TTCDDY, LSCDCC, ARMIL+, and CD algorithms on large-scale problems 1 to 58 evaluated based on the following metrics: #ITER, #FVAL, CPU, and NRM (norm value at an approx. solution). The notation 'NaN' indicates when an algorithm fails to solve a problem within the specified stopping criteria.

No.	TTCDDY	LSCDCC	ARMIL+	CDV	CD
	ITR/FVAL/GVAL/CPU	ITR/FVAL/GVAL/CPU	ITR/FVAL/GVAL/CPU	ITR/FVAL/GVAL/CPU	ITR/FVAL/GVAL/CPU
1	7/73/73/0.459	209/230/230/0.115	222/248/248/0.098	8/83/83/0.053	NaN/NaN/NaN/NaN
2	11/115/115/0.788	834/851/851/5.619	842/861/861/5.284	10/97/97/0.659	NaN/NaN/NaN/NaN
3	7/69/69/2.370	1852/1864/1864/67.264	1852/1864/1864/57.620	7/67/67/2.139	NaN/NaN/NaN/NaN
4	16/177/177/0.475	45/94/94/0.226	77/128/128/0.253	13/143/143/0.274	NaN/NaN/NaN/NaN
5	51/562/562/13.565	140/189/189/4.405	197/253/253/5.941	13/144/144/3.369	NaN/NaN/NaN/NaN
6	10/111/111/0.875	83/135/135/0.956	114/163/163/1.171	10/111/111/0.767	NaN/NaN/NaN/NaN
7	10/111/111/1.498	107/158/158/2.095	168/221/221/2.808	12/128/128/1.641	NaN/NaN/NaN/NaN
8	10/111/111/0.185	36/87/87/0.087	63/115/115/0.112	11/119/119/0.115	NaN/NaN/NaN/NaN
9	11/122/122/0.926	79/129/129/0.971	330/389/389/3.077	12/133/133/1.057	NaN/NaN/NaN/NaN
10	11/122/122/0.542	88/141/141/0.521	109/160/160/0.607	11/122/122/0.467	NaN/NaN/NaN/NaN
11	12/133/133/3.144	133/183/183/4.310	301/360/360/8.511	12/133/133/3.169	NaN/NaN/NaN/NaN
12	240/2618/2618/2.349	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN
13	70/771/771/0.123	NaN/NaN/NaN/NaN	1373/1430/1430/0.227	1090/11951/11951/2.018	NaN/NaN/NaN/NaN
14	40/441/441/0.602	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	1282/14045/14045/18.206	NaN/NaN/NaN/NaN
15	26/287/287/0.446	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	26/287/287/0.420	NaN/NaN/NaN/NaN
16	10/111/111/0.226	42/67/67/0.136	80/105/105/0.283	18/199/199/0.473	821/6151/6151/12.415
17	7/77/77/0.781	83/102/102/1.095	139/159/159/1.655	10/111/111/1.059	351/2737/2737/26.945
18	6/62/62/1.511	123/143/143/3.844	765/785/785/19.804	6/61/61/1.543	521/3944/3944/96.132
19	427/4584/4584/0.167	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN
20	15/146/146/0.019	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	87/867/867/0.029	616/5216/5216/0.164
21	9/95/95/0.085	44/68/68/0.074	NaN/NaN/NaN/NaN	44/483/483/0.544	639/4894/4894/5.359
22	9/99/99/0.397	32/54/54/0.302	361/382/382/1.784	56/612/612/2.298	692/5190/5190/17.995
23	7/78/78/0.062	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	7/78/78/0.016	NaN/NaN/NaN/NaN
24	10/111/111/0.537	373/374/374/1.389	373/374/374/1.302	10/111/111/0.358	2/13/13/0.046
25	12/133/133/1.524	34/95/95/1.107	34/92/92/1.067	12/133/133/1.496	150/1232/1232/13.970
26	14/155/155/10.717	337/399/399/27.857	328/383/383/26.776	14/155/155/10.735	178/1423/1423/97.388
27	25/276/276/0.312	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	25/276/276/0.290	NaN/NaN/NaN/NaN
28	11/122/122/0.177	27/28/28/0.035	27/28/28/0.032	11/122/122/0.101	2/18/18/0.009
29	12/133/133/0.986	78/79/79/0.620	78/79/79/0.633	12/133/133/0.983	2/22/22/0.109
30	13/144/144/10.223	240/241/241/19.090	240/241/241/17.660	13/144/144/10.324	2/15/15/0.906
31	11/121/121/0.035	61/107/107/0.013	97/148/148/0.016	12/129/129/0.013	NaN/NaN/NaN/NaN
32	76/825/825/0.313	105/151/151/0.076	260/312/312/0.134	10/110/110/0.031	NaN/NaN/NaN/NaN
33	11/122/122/0.062	75/127/127/0.034	92/148/148/0.040	16/177/177/0.041	NaN/NaN/NaN/NaN
34	15/166/166/0.278	123/173/173/0.362	281/336/336/0.612	16/177/177/0.266	NaN/NaN/NaN/NaN
35	8/70/70/0.665	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	14/123/123/1.193	NaN/NaN/NaN/NaN
36	39/357/357/8.745	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	30/300/300/7.311	NaN/NaN/NaN/NaN
37	32/110/110/4.180	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	57/135/135/6.346	NaN/NaN/NaN/NaN
38	4/33/33/0.028	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	7/76/76/0.010	284/1870/1870/0.077
39	163/1744/1744/0.524	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	105/1117/1117/0.126	NaN/NaN/NaN/NaN
40	92/950/950/1.201	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	91/953/953/1.294	NaN/NaN/NaN/NaN
41	NaN/NaN/NaN/NaN	309/336/336/0.011	964/987/987/0.031	473/5090/5090/0.144	NaN/NaN/NaN/NaN
42	NaN/NaN/NaN/NaN	187/209/209/0.078	582/599/599/0.115	295/3134/3134/0.462	NaN/NaN/NaN/NaN
43	6/67/67/0.025	28/52/52/0.006	28/52/52/0.005	6/67/67/0.008	339/2037/2037/0.174
44	4/45/45/0.077	63/77/77/0.198	63/77/77/0.136	4/45/45/0.066	329/1979/1979/2.317
45	4/43/43/1.402	302/314/314/13.846	302/314/314/12.031	4/43/43/1.331	4/28/28/0.973
46	NaN/NaN/NaN/NaN	294/315/315/0.029	NaN/NaN/NaN/NaN	134/1294/1294/0.125	1818/14215/14215/1.351
47	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	725/745/745/0.145	133/1151/1151/0.237	1519/11507/11507/2.318
48	311/3176/3176/2.704	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN
49	825/8973/8973/0.605	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	475/5194/5194/0.319	1666/12850/12850/0.749
50	NaN/NaN/NaN/NaN	135/157/157/0.035	987/1020/1020/0.201	234/2544/2544/0.453	1090/8444/8444/1.566
51	104/1132/1132/5.885	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	709/7742/7742/40.712	NaN/NaN/NaN/NaN
52	18/70/70/3.179	12/50/50/2.220	47/89/89/3.926	17/78/78/4.040	NaN/NaN/NaN/NaN
53	1/9/9/0.054	245/269/269/0.194	245/269/269/0.191	1/9/9/0.013	1707/12702/12702/7.174
54	22/102/102/224.403	41/96/96/208.022	41/96/96/193.448	21/99/99/174.088	NaN/NaN/NaN/NaN
55	1815/3663/3663/0.310	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN	NaN/NaN/NaN/NaN
56	NaN/NaN/NaN/NaN	4/8/8/0.004	12/22/22/0.003	4/16/16/0.004	NaN/NaN/NaN/NaN
57	43/123/123/0.065	70/125/125/0.049	541/593/593/0.271	82/151/151/0.071	NaN/NaN/NaN/NaN
58	69/171/171/13.336	871/929/929/84.611	163/222/222/20.396	476/684/684/61.736	NaN/NaN/NaN/NaN

4. Application of the Proposed CDV to Signal Recovery

This segment of the paper pertains to signal recovery, in particular compressing sensing, CS. The problem has received considerable interest among researchers and is formulated as a sum of two terms: an underdetermined linear least-squares formulation and a positive scalar, λ, multiple of a non-smooth function such as the l_1 regularizer. The CS problem essentially aims at reconstructing a signal from a sparsely measured vector. The problem has wide-ranging uses, including file recovery, decoding images, and many more. The problem can be expressed as an unconstrained optimization as follows:

$$\min_{x\in\mathbb{R}^n} \|Kx - l\|_2^2 + \psi g(x) \tag{18}$$

where $K \in \mathbb{R}^{m\times n}, m \ll n$ is often referred to as a sensing matrix, $l \in \mathbb{R}^m$ is a measurement vector, $g(x)$ denotes a regularization function which is non-smooth, while ψ is the regularization constant. The task is to recover a sparse signal $x \in \mathbb{R}^n$. Since, as mentioned above, $g(x)$, for example, $\|\cdot\|_1$, is non-smooth; thus, some smooth approaches that closely approximate the l_1 regularizer were suggested as alternative means for solving problem (18).

There are some recently introduced smooth approximating functions [36] for absolute value functions, such as $g(x) := x \cdot \tanh(x/\lambda)$, in which λ is simply a smoothing constant. It is proved in [36], Theorem 1, that $\|x - g(x)\|_1 < \lambda$. Therefore, we can re-express formulation (18) into its approximate smooth equivalent as

$$\min_{x\in\mathbb{R}^n} \|Kx - l\|_2^2 + \psi \sum_{j=1}^{n} g(x_j), \tag{19}$$

Since the above expression is smooth, the proposed algorithm and other smooth-based algorithms for unconstrained optimization problems can be used to solve it.

Experimental data generation and initialization. In implementing the proposed CDV together with an existing algorithm selected for comparison, we chose the dimension of the signal, $n := 2^{12}$, and set the number of observations, $m = \mu n$, with $\mu = 0.2$. Leveraging the setup put forward in [37], the signal matrix operator K is a Hadamard matrix that is made up of 1 s and -1 s and its columns are orthogonal. The two algorithms were initialized with a *zero* vector, i.e., $\bar{x}^0 = \mathbf{0} \in \mathbb{R}^n$. Moreover, the rest of the parameters associated with solving the CS model (19) are defined as follows:

- The regularized parameter $\psi := \max\{2^{-10},\ \mu_2\|K^T l\|_\infty\}$, where $\mu_2 = 0.001$.
- The positive constant, λ is set to 0.1, as suggested in [36].
- The remaining algorithm-specific parameters remain the same as reported in Section 3, thus, they are unchanged.

The performances of the proposed CDV and TTCDDY algorithms are measured extensively using the relative error metric, which is characterized as the following ratio:

$$\frac{\|\bar{x} - x_{sol}\|}{\|x_{sol}\|} \times 100,$$

where x_{sol} is the approximate solution of the model (19).

We portray the results that were obtained from running the experiment in Figure 2a–d. Figure 2a,b represent the initial or original uncorrupted signal and the corrupted version of the original, respectively. The final outputs or recovered signals obtained by TTCDDY and CDV, as indicated by marked red circles, are depicted by Figure 2c,d. We can observe that the CDV algorithm performs considerably better than TTCDDY with respect to the relative error metric. Thus, agreeing with a similar performance of the CDV algorithm in the numerical section.

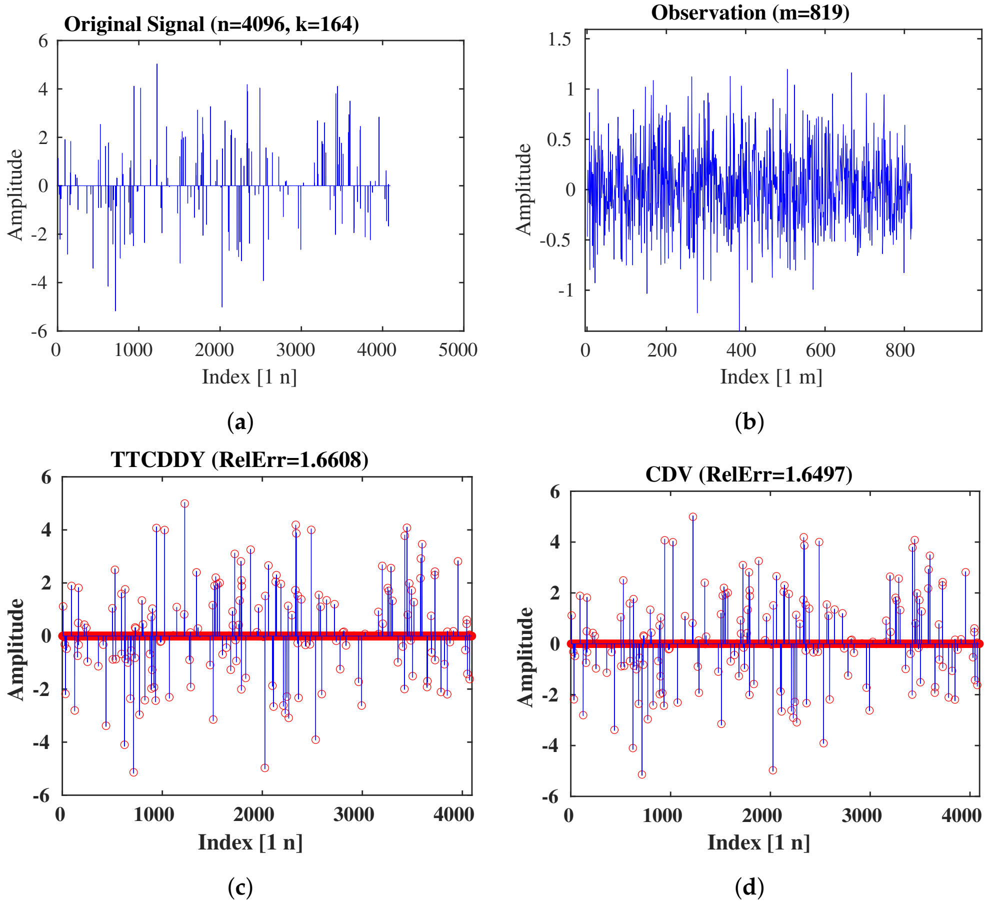

Figure 2. The figures show the performance of the proposed CDV in comparison with the TTCDDY algorithm; the comparison is conducted based on the relative error metric. In which (**a**) is the diagram of the original signal (in blue), (**b**) represent the noisy observation measurement, while the restored signals by both CDV and TTCDDY in red circles versus the original signal in blue peaks is denoted by (**c**) and (**d**) respectively.

5. Conclusions

In this article, we have presented a new conjugate gradient method (named CDV) that is a variant of the popular conjugate descent method (often referred to as CD) [15]. We have extensively discussed the global convergence of the proposed method based on the famous Wolfe line search strategy as well as some stated standard conditions. The CDV method was designed to deal with any problem that can take the structure of general unconstrained optimization problems and has been applied to solve two sets of nonlinear problems, namely, some benchmark test problems and problems arising from compressive sensing. The numerical performance and efficiency of the CDV method are superior compared to some selected CG methods in the literature. Future research should explore how the CDV algorithm could be modified to avoid the differentiability assumption in order to suit problems in the form of nonlinear systems of equations, especially when the solution set is constrained and the underlying function is pseudomonotone [38–41].

Author Contributions: Conceptualization, M.M.Y. and A.M.A.; methodology, A.M.A.; software, A.M.A. and N.P. (Nattawut Pholasa); validation, A.M.A., M.M.Y., N.P. (Nuttapol Pakkaranang) and N.P. (Nattawut Pholasa); formal analysis, A.M.A., M.M.Y. and N.P. (Nattawut Pholasa); investigation, A.M.A. and N.P. (Nuttapol Pakkaranang); resources, M.M.Y. and N.P. (Nattawut Pholasa); data curation, A.M.A. and M.M.Y.; writing—original draft preparation, A.M.A., M.M.Y., N.P. (Nuttapol Pakkaranang) and N.P. (Nattawut Pholasa); writing—review and editing, A.M.A., M.M.Y., N.P. (Nuttapol Pakkaranang) and N.P. (Nattawut Pholasa); visualization, N.P. (Nattawut Pholasa); supervision, A.M.A., N.P. (Nuttapol Pakkaranang) and N.P. (Nattawut Pholasa); project administration, A.M.A.,

N.P. (Nuttapol Pakkaranang) and N.P. (Nattawut Pholasa); funding acquisition, N.P. (Nattawut Pholasa). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the University of Phayao and Thailand Science Research and Innovation Fund (Fundamental Fund 2024) and School of Science, University of Phayao (Grant No. PBTSC66053).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The third author would like to thank Phetchabun Rajabhat University.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, L.; Zhou, B. A Modified Gradient Method for Distributionally Robust Logistic Regression over the Wasserstein Ball. *Mathematics* **2023**, *11*, 2431. [CrossRef]
2. Awwal, A.M.; Sulaiman, I.M.; Malik, M.; Mamat, M.; Kumam, P.; Sitthithakerngkiet, K. A Spectral RMIL+ Conjugate Gradient Method for Unconstrained Optimization With Applications in Portfolio Selection and Motion Control. *IEEE Access* **2021**, *9*, 75398–75414. [CrossRef]
3. Yahaya, M.M.; Kumam, P.; Awwal, A.M.; Aji, S. A structured quasi-Newton algorithm with nonmonotone search strategy for structured NLS problems and its application in robotic motion control. *J. Comput. Appl. Math.* **2021**, *395*, 113582. [CrossRef]
4. Sulaiman, I.M.; Malik, M.; Awwal, A.M.; Kumam, P.; Mamat, M.; Al-Ahmad, S. On three-term conjugate gradient method for optimization problems with applications on COVID-19 model and robotic motion control. *Adv. Contin. Discret. Model.* **2022**, *2022*, 1. [CrossRef]
5. Yahaya, M.M.; Kumam, P.; Chaipunya, P.; Awwal, A.M.; Wang, L. On diagonally structured scheme for nonlinear least squares and data-fitting problems. *Rairo-Oper. Res.* **2024**, *58*, 2887–2905. [CrossRef]
6. Salihu, N.; Babando, H.A.; Arzuka, I.; Salihu, S. A hybrid conjugate gradient method for unconstrained optimization with application. *Bangmod Int. J. Math. Comput. Sci.* **2023**, *9*, 24–44. [CrossRef]
7. Leung, M.F.; Wang, J. A collaborative neurodynamic approach to multiobjective optimization. *IEEE Trans. Neural Networks Learn. Syst.* **2018**, *29*, 5738–5748. [CrossRef] [PubMed]
8. Sun, W.; Yuan, Y.X. *Optimization Theory and Methods: Nonlinear Programming*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 1.
9. Wolfe, P. Convergence conditions for ascent methods. *SIAM Rev.* **1969**, *11*, 226–235; Correction on *SIAM Rev.* **1971**, *13*, 185–188. [CrossRef]
10. Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **1952**, *49*, 409–435. [CrossRef]
11. Polak, E.; Ribiere, G. Note sur la convergence de methodes de directions conjuguees. *USSR Comput. Math. Math. Phys.* **1969**, *9*, 94–112.
12. Polyak, B.T. A general method for solving extremal problems. *Dokl. Akad. Nauk. SSSR* **1967**, *174*, 33–36.
13. Liu, Y.; Storey, C. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.* **1991**, *69*, 129–137. [CrossRef]
14. Fletcher, R.; Powell, M.J.D. A rapidly convergent descent method for minimization. *Comput. J.* **1963**, *6*, 163–168. [CrossRef]
15. Fletcher, R. *Practical Methods of Optimization*; A Wiley Interscience Publication: Hoboken, NJ, USA, 1987.
16. Dai, Y.H.; Yuan, Y.X. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **1999**, *10*, 177–182. [CrossRef]
17. Babaie-Kafaki, S. A hybrid conjugate gradient method based on a quadratic relaxation of the Dai–Yuan hybrid conjugate gradient parameter. *Optimization* **2013**, *62*, 929–941. [CrossRef]
18. Hager, W.W.; Zhang, H. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2006**, *2*, 35–58.
19. Djordjević, S.S. New hybrid conjugate gradient method as a convex combination of LS and CD methods. *Filomat* **2017**, *31*, 1813–1825. [CrossRef]
20. Xue, W.; Ren, J.; Zheng, X.; Liu, Z.; Liang, Y. A new DY conjugate gradient method and applications to image denoising. *IEICE Trans. Inf. Syst.* **2018**, *101*, 2984–2990. [CrossRef]
21. Dai, Z.; Chen, X.; Wen, F. Comments on A hybrid conjugate gradient method based on a quadratic relaxation of the Dai-Yuan hybrid conjugate gradient parameter. *Optimization* **2015**, *64*, 1173–1175. [CrossRef]
22. Djordjević, S.S. New hybrid conjugate gradient method as a convex combination of LS and FR methods. *Acta Math. Sci.* **2019**, *39*, 214–228. [CrossRef]
23. Guefassa, I.; Chaib, Y.; Bechouat, T. Another hybrid conjugate gradient method as a convex combination of WYL and CD methods. *Monte Carlo Methods Appl.* **2024**. [CrossRef]
24. Hallal, A.; Belloufi, M.; Sellami, B. An efficient new hybrid CG-method as convex combination of DY, CD and HS algorithms. *RAIRO-Oper. Res.* **2022**, *56*, 4047–4056. [CrossRef]

25. Jardow, F.N.; Al-Naemi, G.M. A new hybrid conjugate gradient algorithm for unconstrained optimization with inexact line search. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *20*, 939–947. [CrossRef]
26. Zhang, L. Two modified Dai-Yuan nonlinear conjugate gradient methods. *Numer. Algorithms* **2008**, *50*, 1–16. [CrossRef]
27. Tian, Q.; Wang, X.; Pang, L.; Zhang, M.; Meng, F. A new hybrid three-term conjugate gradient algorithm for large-scale unconstrained problems. *Mathematics* **2021**, *9*, 1353. [CrossRef]
28. Jian, J.; Yang, L.; Jiang, X.; Liu, P.; Liu, M. A spectral conjugate gradient method with descent property. *Mathematics* **2020**, *8*, 280. [CrossRef]
29. Shi, Z.J.; Guo, J. A new family of conjugate gradient methods. *J. Comput. Appl. Math.* **2009**, *224*, 444–457. [CrossRef]
30. Zoutendijk, G. *Nonlinear Programming, Computational Methods*; Abadie, J., Ed.; Integer and Nonlinear Programming: Amsterdam, The Netherlands, 1970; pp. 37–86.
31. Deepho, J.; Abubakar, A.B.; Malik, M.; Argyros, I.K. Solving unconstrained optimization problems via hybrid CD-DY conjugate gradient methods with applications. *J. Comput. Appl. Math.* **2022**, *405*, 113823. [CrossRef]
32. Wang, Z.; Li, P.; Li, X.; Pham, H. A Modified Three-Term Type CD Conjugate Gradient Algorithm for Unconstrained Optimization Problems. *Math. Probl. Eng.* **2020**, *2020*, 4381515. [CrossRef]
33. Awwal, A.M.; Wang, L.; Kumam, P.; Sulaiman, M.I.; Salisu, S.; Salihu, N.; Yodjai, P. Generalized RMIL conjugate gradient method under the strong Wolfe line search with application in image processing. *Math. Methods Appl. Sci.* **2023**, *46*, 17544–17556. [CrossRef]
34. Gould, N.I.M.; Orban, D.; Toint, P.L. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Softw. (TOMS)* **2003**, *29*, 373–394. [CrossRef]
35. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *9*, 201–213. [CrossRef]
36. Bagul, Y.J.J. A smooth transcendental approximation to $|x|$. *Int. J. Math. Sci. Eng. Appl. (IJMSEA)* **2017**, *11*, 213–217.
37. Aminifard, Z.; Hosseini, A.; Babaie-Kafaki, S. Modified conjugate gradient method for solving sparse recovery problem with nonconvex penalty. *Signal Process.* **2022**, *193*, 108424. [CrossRef]
38. Yusuf, A.; Manjak, N.H.; Aphane, M. A Modified Three-Term Conjugate Descent Derivative-Free Method for Constrained Nonlinear Monotone Equations and Signal Reconstruction Problems. *Mathematics* **2024**, *12*, 1649. [CrossRef]
39. Awwal, A.M.; Kumam, P.; Mohammad, H.; Watthayu, W.; Abubakar, A.B. A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 regularized problem. *Optimization* **2021**, *70*, 1231–1259. [CrossRef]
40. Awwal, A.M.; Kumam, P.; Sitthithakerngkiet, K.; Muhammad, A.B.; Abubakar, S.H.; Ibrahim, M.S. Derivative-free method based on DFP updating formula for solving convex constrained nonlinear monotone equations and application. *AIMS Math.* **2021**, *6*, 8792–8814. [CrossRef]
41. Awwal, A.M.; Botmart, T. A new sufficiently descent algorithm for pseudomonotone nonlinear operator equations and signal reconstruction. *Numer. Algorithms* **2023**, *94*, 1125–1158. [CrossRef]