symmetry

MDPI

Article

Fuzzy Adaptive Parameter in the Dai–Liao Optimization Method Based on Neutrosophy

Predrag S. Stanimirović [1,2,*], Branislav D. Ivanov [1], Dragiša Stanujkić [3], Lev A. Kazakovtsev [2], Vladimir N. Krutikov [4] and Darjan Karabašević [5]

[1] Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia; ivanov.branislav@gmail.com

[2] Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia; levk@bk.ru

[3] Technical Faculty in Bor, University of Belgrade, Vojske Jugoslavije 12, 19210 Bor, Serbia; dstanujkic@tfbor.bg.ac.rs

[4] Department of Applied Mathematics, Kemerovo State University, 6 Krasnaya Street, 650043 Kemerovo, Russia; krutikovvn@rambler.ru

[5] Faculty of Applied Management, Economics and Finance, University Business Academy in Novi Sad, Jevrejska 24, 11000 Belgrade, Serbia; darjan.karabasevic@mef.edu.rs

* Correspondence: pecko@pmf.ni.ac.rs

Abstract: The impact of neutrosophy has increased rapidly in many areas of science and technology in recent years. Furthermore, numerous applications of the neutrosophic theory have become more usual. We aim to use neutrosophy to enhance Dai–Liao conjugate gradient (CG) iterative method. In particular, we suggest and explore a new neutrosophic logic system intended to compute the essential parameter t required in Dai–Liao CG iterations. Theoretical examination and numerical experiments signify the effectiveness of the introduced method for controlling t. By incorporation of the neutrosophy in the Dai–Liao conjugate gradient principle, we established novel Dai–Liao CG iterations for solving large-scale unconstrained optimization problems. Global convergence is proved under standard assumptions and with the use of the inexact line search. Finally, computational evidence shows the computational effectiveness of the proposed fuzzy neutrosophic Dai–Liao CG method.

Keywords: neutrosophic logic systems; Dai–Liao conjugate gradient method; backtracking line search; convergence; unconstrained optimization

MSC: 90C70; 90C30; 65K05

check for updates

Citation: Stanimirović, P.S.; Ivanov, B.D.; Stanujkić, D.; Kazakovtsev, L.A.; Krutikov, V.N.; Karabašević, D. Fuzzy Adaptive Parameter in the Dai–Liao Optimization Method Based on Neutrosophy. *Symmetry* **2023**, *15*, 1217. https://doi.org/10.3390/sym15061217

Academic Editors: Quanxin Zhu, Zuowei Cai and Fanchao Kong

Received: 21 April 2023
Revised: 2 June 2023
Accepted: 5 June 2023
Published: 7 June 2023

1. Introduction and Background Results

Numerous iterative methods have been developed for solving the large-scale unconstrained optimization problem

$$\text{minimize} \;\; f(\mathbf{x}), \;\; \mathbf{x} \in \mathbb{R}^n, \tag{1}$$

in which $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function bounded below. Continuing well-established notation, $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)$ stands for the gradient vector of f at the actual iterative point $\mathbf{x}_k$, and further $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$ and $\mathbf{s}_{k-1} = \mathbf{x}_k - x_{k-1}$. Utilizing the extended conjugacy condition

$$\mathbf{d}_k^{\mathrm{T}} \mathbf{y}_{k-1} = -t\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1}, \;\; t > 0, \tag{2}$$

Dai and Liao in [1] suggested the conjugate gradient (CG) iterations

$$x_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{3}$$

in which $\mathbf{x}_k$ is the last calculated iteration, $\mathbf{x}_{k+1}$ is the new iterative point, α_k is a positive step size parameter defined as the output of a proper inexact line search, and $\mathbf{d}_k$ is a descent direction. The search directions $\{\mathbf{d}_k,\ k \geq 0\}$ are generated by the recurrent rule

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_0, & k = 0, \\ -\mathbf{g}_k + \beta_k^{\mathrm{DL}}\, \mathbf{d}_{k-1}, & k \geq 1, \end{cases} \tag{4}$$

where β_k^{DL} is the CG coefficient that describes the type of CG method according to the general rule

$$\beta_k^{\mathrm{DL}} = \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}} - t\, \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}, \tag{5}$$

wherein $t > 0$ is an appropriate scalar. The Dai–Liao (DL) method guarantees global convergence for uniformly convex objective functions. These results have attracted considerable attention, leading to the creation of several methods based on various patterns for defining β_k. Most of these methods are developed by modifying the conjugate gradient parameter β_k^{DL} [2–11]. For more details, see the survey on the DL family of nonlinear CG methods in [12]. One of rules for defining β_k is denoted as β_k^{MHSDL} and defined in [7] by

$$\beta_k^{\mathrm{MHSDL}} = \frac{\mathbf{g}_k^{\mathrm{T}} \widehat{\mathbf{y}_{k-1}}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}} - t\, \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}, \tag{6}$$

such that $\widehat{\mathbf{y}_{k-1}} = \mathbf{g}_k - \frac{\|\mathbf{g}_k\|}{\|\mathbf{g}_{k-1}\|} \mathbf{g}_{k-1}$ and $t > 0$ is as in (5).

Due to the large influence of the quantity t on the numerical results generated by the DL class of CG methods [13], one of the most common issues is the determination of an appropriate value t. We can distinguish two research directions based on the previous results in determining proper values of t in the DL-established CG iterations. The first line of research consists of a group of DL methods that aim to find a suitable constant value for t during iterations [1,2,6–8], while the second line consists of a group of DL methods that propose a suitable control in recalculating t in each iteration. In this research, we prioritize the second research stream: find values of t that change appropriately across iterations. The quantity t determined in the kth iterative step will be denoted by $t_k^{(i)}$, where i is a variant of the algorithm for defining t.

Some of the most important adaptive choices for the DL parameter t_k will be presented in the rest of this section. Hager and Zhang in [14,15] proposed the CG-DESCENT method, which is classified into the group of the DL CG methods (5) defined by $t \equiv t_k^{(1)}$ and

$$t_k^{(1)} = 2 \frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{s}_{k-1}}. \tag{7}$$

Dai and Kou in [16] proposed the DK method, which is based on the CG coefficient β_k^{DK} of the form

$$\beta_k^{\mathrm{DK}} = \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{d}_{k-1}} - \left(\tau_k + \frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{s}_{k-1}} - \frac{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{s}_{k-1}}{\|\mathbf{s}_{k-1}\|^2} \right) \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}. \tag{8}$$

In the equality (8), the parameter τ_k is defined utilizing the self-scaling memoryless BFGS method. It is also obvious from (8) that the DK method is involved in the DL CG class of methods based on the particular value $t \equiv t_k^{(2)}$, which is defined by

$$t_k^{(2)} = \tau_k + \frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{s}_{k-1}} - \frac{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{s}_{k-1}}{\|\mathbf{s}_{k-1}\|^2}. \tag{9}$$

Babaie-Kafaki and Ghanbari in [17] gave the subsequent two rules for computing t in (5):

$$t_k^{(3)} = \frac{\mathbf{s}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}{\|\mathbf{s}_{k-1}\|^2} + \frac{\|\mathbf{y}_{k-1}\|}{\|\mathbf{s}_{k-1}\|} \tag{10}$$

and

$$t_k^{(4)} = \frac{\|\mathbf{y}_{k-1}\|}{\|\mathbf{s}_{k-1}\|}. \tag{11}$$

Andrei in [18] originated a new DL class, denoted DLE, where $t \equiv t_k^{(5)}$ is defined by

$$t_k^{(5)} = \frac{\mathbf{s}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}{\|\mathbf{s}_{k-1}\|^2}. \tag{12}$$

A special place in the DL iterations is occupied by the DL method, which is defined by

$$t_k^{(6)} = v \frac{\|\mathbf{y}_{k-1}\|^2}{\mathbf{s}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}, \tag{13}$$

where $v > \frac{1}{4}$ is a constant that is defined according to the sufficient descent condition

$$\mathbf{g}_k^{\mathrm{T}} \mathbf{d}_k \leq -c \, \|\mathbf{g}_k\|^2, \ \ \forall k \geq 0, \tag{14}$$

such that $c > 0$ is a quantity independent of the cost function convexity and the line search rule (for more details, see [19]).

Lotfi and Hosseini in [20] suggested the subsequent rule

$$t_k^{(7)} = \max\left\{ t_k^{(7^*)}, \, t_k^{(6)} \right\}, \tag{15}$$

where

$$t_k^{(7^*)} = \frac{\left(1 - \hbar_k \|g_{k-1}\|^r\right) \mathbf{s}_{k-1}^{\mathrm{T}} \mathbf{g}_k + \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^{\mathrm{T}} \mathbf{s}_{k-1}} \hbar_k \|\mathbf{g}_{k-1}\|^r \|\mathbf{s}_{k-1}\|^2}{\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1} + \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}} \hbar_k \|\mathbf{g}_{k-1}\|^r \|\mathbf{s}_{k-1}\|^2}, \tag{16}$$

$$\hbar_k = C + \max\left\{ -\frac{\mathbf{s}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}{\|\mathbf{s}_{k-1}\|^2}, 0 \right\} \|\mathbf{g}_{k-1}\|^{-r}, \tag{17}$$

and C, r are positive constants.

Ivanov et al. in [21] proposed a variant of the Dai–Liao CG method (6), known as the Effective Dai–Liao (EDL) CG method, where $t \equiv t_k^{(8)}$ is determined as

$$t_k^{(8)} = \frac{\|\mathbf{g}_k\|^2}{\max\left\{1, \mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{g}_k\right\} + \left(\max\left\{0, \frac{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{g}_k}{\|\mathbf{g}_k\|^2}\right\} + 1\right) \|\mathbf{g}_k\|^2}. \tag{18}$$

Experiments conducted in [21] confirm that EDL iterations outperform many existing CG variants.

Our motivation is to explore principles of neutrosophy to determine the crucial parameter t in the Dai–Liao CG method. Numerical experience clearly shows improvements in the behavior of the DL CG method based on the fuzzy parameter t established in this way. The justification for using the fuzzy values t_k defined on the basis of neutrosophic logic lies in the fact that there is neither a best value for that parameter nor a clear rule for its determination. Our underlying motivation is based on the fact that the neutrosophic logic system is an effective tool to use the behavior of the goal function through iterations and thus avoid divergence or slow convergence. Effective tracking of changes in the objective function value is enabled by properly defined if–then rules and suitable membership functions.

The sections of the paper are arranged as follows. Introduction, basic details of motivation, and a brief review of obtained results are presented in Section 1. Section 2 briefly describes the most important related results about neutrosophy, DL methods, and motivation. A neutrosophic-based control for defining appropriate changeable values t_k is proposed in Section 3. Moreover, we present details of the FDL method in the same section. The global convergence behavior of the FDL method is examined in Section 4. A numerical comparison of the FDL method with the main standard DL methods is presented in Section 5, and a comparison with some known variations of the DL class of methods is also provided. Final conclusions are presented in the concluding section.

2. Related Results about Neutrosophy and Motivation

A fuzzy set theory utilizes a membership function (MF) $\mathrm{T}_\Psi(\lambda) \in [0,1], \lambda \in \Lambda$ in the universe Λ, which defines the degree of membership of λ in $\Psi \subset \Lambda$ [22]. An intuitionistic fuzzy set (IFS) Ψ is based on membership and non-membership functions $T_\Psi(\lambda), F_\Psi(\lambda) \in [0,1], \lambda \in \Lambda$ [23], which satisfy $\mathrm{T}_\Psi(\lambda), \mathrm{F}_\Psi(\lambda) : \Lambda \to [0,1]$ and are jointly correlated by $0 \le \mathrm{T}_\Psi(\lambda) + \mathrm{F}_\Psi(\lambda) \le 1$. The IFS theory was extended in [24,25] by the neutrosophic theory. The key innovation in Neutrosophy is based on the indeterminacy $\mathrm{I}(\lambda)$. As a consequence of such an approach, a phenomenon in a neutrosophic set is determined by three individualistic MFs [24,25]: the truth-MF $\mathrm{T}(\lambda)$, the indeterminacy-MF $\mathrm{I}(\lambda)$, and the falsity-MF $\mathrm{F}(\lambda)$. Because of independence between the three MFs, the neutrosophic logic is established on the symmetry involved in the ordered triple $(\mathrm{T}, \mathrm{I}, \mathrm{F})$ and the inequality $0 \le \mathrm{T} + \mathrm{I} + \mathrm{F} \le 3$. Clearly, T is the symmetric pole to its opposite pole F relative to I, which represents the center of symmetry between T and F [26]. The same observation is valid for refined neutrosophic sets that assume two refined indeterminacies I_1 and I_2 between extremes T and F [27]. The MFs of a neutrosophic set Ψ satisfy $\mathrm{T}_\Psi(\lambda), \mathrm{I}_\Psi(\lambda), \mathrm{F}_\Psi(\lambda) : \Lambda \to [0,1]$, which is based on their independence and implies $0 \le \mathrm{T}_\Psi(\lambda) + \mathrm{I}_\Psi(\lambda) + \mathrm{F}_\Psi(\lambda) \le 3$ and enables a symmetry between them. In [28], the authors originated a neutrosophic-based multiple criteria decision making procedure which is established on the introduced symmetry estimate. A complete overview of neutrosophic theory and applications is available in the exhaustive collections of papers [29,30]. A useful application of neutrosophic logic in natural language processing was discussed in [27].

The advantages of the NL approach over the FL and IFL in improving the gradient descent method are discussed in [31].

It is obvious that there is no single solution for determining the parameter t_k in DL iterations (5). All previous solutions are based on specific rules. Moreover, it is clear that there are infinite possibilities for new rules in defining t_k. Our intention is to improve the behavior of the DL class for solving unconstrained nonlinear optimization problems with the support of an appropriate neutrosophic logic system. The indeterminacy and uncertainty in predicting an optimal value of t_k justify the utilization of fuzzy quantities as well as neutrosophic principles in its determination. A specific convenience in the application of neutrosophy to the determination of the DL parameter arises from the restriction $0 \le t_k \le 1$. Moreover, the neutrosophic logic system is a suitable tool for using the behavior of the objective function in determining required parameters with the help of appropriately defined if–then rules and suitable membership functions.

Application of fuzzy logic controllers in defining some parameters in the development of some recurrent neural network dynamical systems has been a popular topic in recent years [32–36].

We expect that the symmetry principles involved in the basis of neutrosophic logic (see [27,31] for more details) and tracking the prior values of the objective function will lead to better numerical results. Such expectations were confirmed by the results obtained through a series of numerical experiments.

Neutrosophic logic was applied in [31] in regulating proper step sizes for a class of accelerated gradient descent optimization methods. The approach in [31] assumes an additional fuzzy parameter that stabilizes the behavior of an important class of gradient descent family. Motivated by that approach, in this research, we apply neutrosophy to enhance the performances of DL methods. Based on the review and analysis of the class of DL methods, we propose a new method for determining t_k. The proposed method defines t_k as the output generated by an appropriate neutrosophic logic controller (NLC). Our idea is to replace the classical parameter t_k in (5) by an adaptive neutrosophic logic parameter ν_k, determined as the output of the NLC. Since $0 \le t_k \le 1$, we decided to define t_k as the value of ν_k, without additional parameters.

The principal results obtained in this paper are presented as follows.

(1) We examine the application of NL in determining the value t in the Dai–Liao CG method (5).

(2) A theoretical analysis is conducted to confirm the global convergence of the proposed method.

(3) A numerical comparison between the proposed FDL algorithm and other known DL algorithms is provided.

3. Fuzzy Neutrosophic Dai–Liao Conjugate Gradient Method

The fuzzy neutrosophic Dai–Liao CG method is initially conceived as a modification of the Dai–Liao CG method (3), where the search directions $\{\mathbf{d}_k\}$ are calculated by the recurrence rule

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_0, & k=0, \\ -\mathbf{g}_k + \beta_k^{\mathrm{FDL}}\, \mathbf{d}_{k-1}, & k \ge 1, \end{cases} \tag{19}$$

where the CG coefficient β_k^{FDL} is defined by

$$\beta_k^{\mathrm{FDL}} = \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}} - \nu_k \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}, \tag{20}$$

such that ν_k is a proper fuzzy neutrosophic parameter. Our intention is to define ν_k as a function $\nu_k := \nu_k(\Delta_k)$ of $\Delta_k := f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})$. More precisely, $\nu_k(\Delta_k)$ is defined subject to the following constraint

$$0 \le \nu_k(\Delta_k) \le 1. \tag{21}$$

It is know that $\nu_k(\Delta_k) = 0$ reduces (2) into

$$\mathbf{d}_k^{\mathrm{T}}\, \mathbf{y}_{k-1} = 0. \tag{22}$$

Hence, Equation (22) can be considered as a reflection of the conjugacy condition, which, in conjunction with (4), determines the HS parameter [37]

$$\beta_k^{\mathrm{HS}} = \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}} \mathbf{y}_{k-1}}. \tag{23}$$

Alternatively, for $\nu_k(\Delta_k) = 1$, Equation (2) is considered a conjugacy condition that implicitly satisfies the quasi-Newton characteristics. For more details on these cases, see [1,12].

The idea for defining a new parameter t_k in the Dai–Liao CG method (5) comes from the neutrosophic logic. According to this decision, we intend to define $t_k := \nu_k(\Delta_k)$ with values inside the interval $[0,1]$ according to neutrosophic principles.

The generic layout of the fuzzy neutrosophic Dai–Liao CG method is given in the diagram in Figure 1.

The input of the NLC presented in Figure 1 is $\Delta_k := f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})$ and the output is the desired step size ν_k. This means that our basic idea is to define ν_k based on two consecutive values of the objective function f. On the other hand, the backtracking line search is responsible for appropriate step lengths α_k in (3), and then the descent direction $\mathbf{d}_k$ is defined by (19). Using ν_k, it is possible to compute β_k^{FDL} in (20). Finally, (3) generates a new iterative point x_{k+1}.

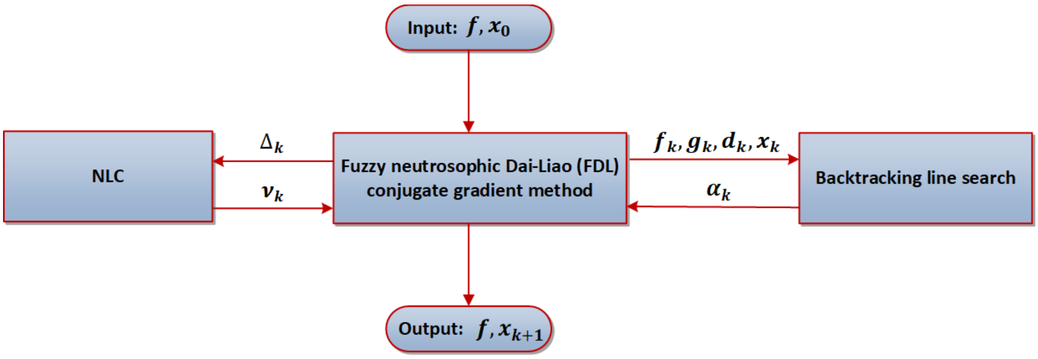


Figure 1. The global structure of the fuzzy neutrosophic Dai–Liao CG method.

To develop the FDL method, it is necessary to plan three global steps: neutrosophication, neutrosophic inference engine, and de-neutrosophication (score function).

(1) *Neutrosophication* maps the input $\Delta_k := f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})$ into neutrosophic ordered triplets $(\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k))$. The MFs are defined to improve the CG iterative rule by exploiting numerical experience. The sigmoid function with the slope defined by ς_1 at the crossover point $\Delta = \varsigma_2$ is a proper choice for T:

$$\mathrm{T}(\Delta) = 1/\left(1 + \mathrm{e}^{-\varsigma_1(\Delta - \varsigma_2)}\right). \tag{24}$$

A proper choice for F is the following sigmoid function:

$$\mathrm{F}(\Delta) = 1/(1 + \mathrm{e}^{\varsigma_1(\Delta - \varsigma_2)}). \tag{25}$$

The Gaussian function with the standard deviation ς_1 and the mean ς_2 defines the indeterminacy I:

$$\mathrm{I}(\Delta) = \mathrm{e}^{-\frac{(\Delta - \varsigma_2)^2}{2\varsigma_1^2}}. \tag{26}$$

Then, the neutrosophication of $\Delta \in \mathbb{R}$ is defined as the transition $\Delta \to \langle \mathrm{T}(\Delta), \mathrm{I}(\Delta), \mathrm{F}(\Delta) \rangle$, where the MFs are determined in (24)–(26).

(2) *Neutrosophic inference* between an input fuzzy set $\Im$ and an output fuzzy set is based on the subsequent "IF–THEN" regulations:

$$\Re_1 : \text{If } \Im = S_P \Longrightarrow \mathfrak{D} = \{\mathrm{T, I, F}\}$$
$$\Re_2 : \text{If } \Im = S_N \Longrightarrow \mathfrak{D} = \{\mathrm{T, I, F}\}.$$

Fuzzy sets S_P and S_N point, respectively, to positive or negative errors. Applying the unification $\Re = \Re_1 \cup \Re_2$, we define $\mathfrak{D}_i = \Im \circ \Re_i$, $i = 1, 2$, where $\circ$ denotes the fuzzy transformation. In addition, for a fuzzy triple $\zeta = \{\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)\}$, it is required $\kappa_{\Im\circ\Re}(\zeta) = \kappa_{\Im\circ\Re_1} \bigvee \kappa_{\Im\circ\Re_2} = \sup(\kappa_{\Im} \bigwedge \kappa_{\mathfrak{D}_i})$, $i = 1, 2$, where $\bigwedge$ and $\bigvee$ denote the $(\min, \max, \max)$ and $(\max, \min, \min)$ operator, respectively. In this research, the centroid defuzzification method is utilized to generate a vector of crisp outputs $\zeta^* \in \mathbb{R}^3$ as follows:

$$\zeta^* = \frac{\int_{\mathfrak{D}} \zeta \, \kappa_{\Im\circ\Re}(\zeta) \mathrm{d}\zeta}{\int_{\mathfrak{D}} \kappa_{\Im\circ\Re}(\zeta) \mathrm{d}\zeta}.$$

(3) *De-neutrosophication* is based on the transformation $\langle \mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k) \rangle \to \nu_k \in \mathbb{R}$, resulting in a crisp value ν_k, proposed as

$$\nu_k = 2 - (\mathrm{T}(\Delta_k) + \mathrm{I}(\Delta_k) + \mathrm{F}(\Delta_k)). \tag{27}$$

The diagram in Figure 2 presents the NLC based on the neutrosophic control.

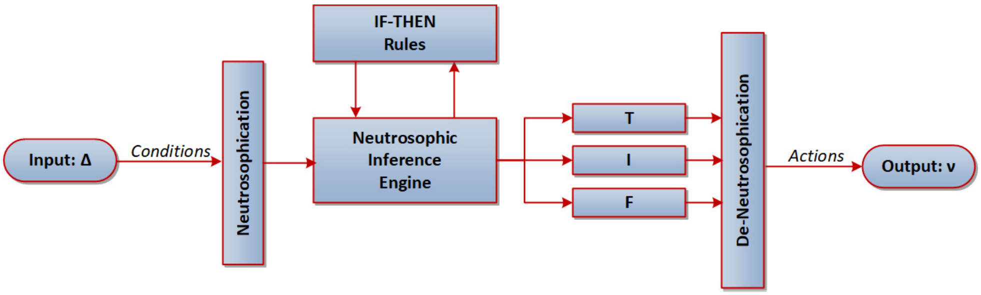


Figure 2. The NLC design based on the neutrosophy.

The settings in the NLC employed in numerical testing are arranged in Table 1.

Table 1. Recommended parameters in NLC.

Set	Membership Function	ς_1	ς_2	Weight
Input	Sigmoid function (24)	1	3	1
	Sigmoid function (25)	1	3	1
	Gaussian function (26)	120	0	1
Output	Score function (27)	-	-	1

Our imperative requirement is $0 \leq \nu_k(\Delta_k) \leq 1$, requested in (21). The fulfillment of those conditions is verified in Lemma 1.

Lemma 1. *The inequality* (21) *holds for the given choice of the score function* (27) *and the parameters given in Table 1.*

Proof. To prove (21), we need to replace the MFs (24)–(26) in (27). After applying the parameters from Table 1, we obtain

$$\mathrm{T}(\Delta_k) = 1/\left(1 + \mathrm{e}^{-(\Delta-3)}\right), \;\; \mathrm{F}(\Delta_k) = 1/\left(1 + \mathrm{e}^{\Delta-3}\right), \;\; \mathrm{I}(\Delta_k) = \mathrm{e}^{-\frac{\Delta_k^2}{2*120^2}}.$$

Elementary calculation provides

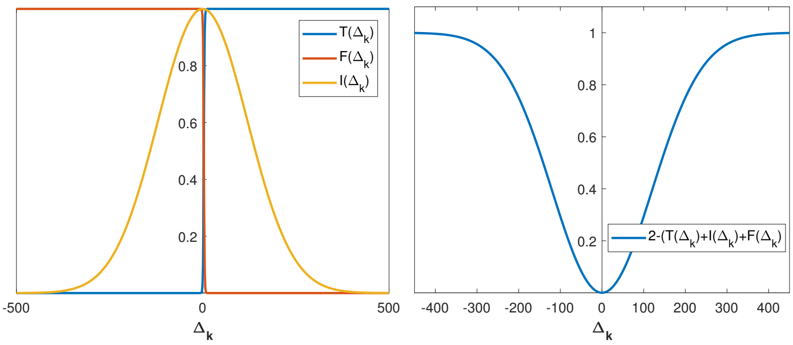
$$\nu_k(\Delta_k) = 2 - (\mathrm{T}(\Delta_k) + \mathrm{I}(\Delta_k) + \mathrm{F}(\Delta_k)) = 1 - \mathrm{e}^{-\frac{\Delta_k^2}{28800}}. \tag{28}$$

A careful analysis of the function (28) inside the interval $\Delta_k \in (-\infty, +\infty)$ discovers $\min \nu_k(\Delta_k) = 0$ and $\max \nu_k(\Delta_k) = 1$, which proves the inequality (21). □

Graphs of $\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)$ are displayed in Figure 3a. The fulfilment of the requirements (21) in the NLC output ν_k generated throughout the described de-neutrosophication is illustrated in Figure 3b.

Remark 1. *The objective function decreases with the flow of iterations and tends to the minimal value, which means* $\lim_{k\to\infty} \Delta_k = 0$*, i.e.,* $\lim_{k\to\infty} \nu_k(\Delta_k) = 0$*. Such behavior leads to* $\nu_k \to 0$ *as the minimum of* f *approaches, so the impact of the proposed neutrosophic strategy decreases and disappears, which agrees with our goal.*

Remark 2. *Obviously, larger values of* Δ_k *lead to increasing values* $\nu_k(\Delta_k)$ *approaching* 1*, which will be denoted as* $\nu_k(\Delta_k) \nearrow 1$*. In addition, based on the limit* $\Delta_k \to 0$*, we anticipate smaller values* $\nu_k(\Delta_k)$ *approaching* 0*, i.e.,* $\nu_k(\Delta_k) \searrow 0$*, in final iterations. As a result,* $\nu_k(\Delta_k)$ *is suitable as an adjustable regulator for the quantity* t *in the Dai–Liao CG method.*



(**a**) MFs in Neutrosophication.

(**b**) Graph of De-neutrosophication.

Figure 3. MFs of Neutrosophication (24)–(26) and de-neutrosophication (27) guided by the quantities in Table 1.

The backtracking line search used in [38] begins at $\alpha = 1$ and generates further step sizes, which ensure a decrease in the goal function in each iteration. Algorithm 1, restated from [39], is used to define the principal step size α_k in (3).

Algorithm 1 The backtracking line search.

```
Require: Objective function $f(\mathbf{x})$, foregoing point $\mathbf{x}_k$, the search direction $\mathbf{d}_k$, a real positive
    constants $0 < \varphi < 1$, and $0 < \omega < 0.5$.
  $\ell = 1$.
  While $f(\mathbf{x}_k + \ell \mathbf{d}_k) > f(\mathbf{x}_k) + \omega \ell \mathbf{g}_k^{\mathrm{T}} \mathbf{d}_k$, do $\ell := \ell\varphi$.
  Output: $\alpha_k = \ell$.
```

Algorithm 2 of the FDL method is described as follows:

Algorithm 2 Fuzzy neutrosophic Dai–Liao (FDL) conjugate gradient method.

```
Require: A starting point $\mathbf{x}_0$ and $0 < \epsilon, \delta < 1$.
  1: Assign $k = 0$ and $\mathbf{d}_0 = -\mathbf{g}_0$.
  2: If
         $\|\mathbf{g}_k\| \leq \epsilon$  and  $\frac{|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)|}{1 + |f(\mathbf{x}_k)|} \leq \delta$,
     STOP;
     else go to Step 3.
  3: (Backtracking line search) Regulate $\alpha_k \in (0, 1]$ utilizing Algorithm 1.
  4: Calculate $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.
  5: Calculate $\mathbf{g}_{k+1}$, $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$,  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$.
  6: Calculate $\Delta_k := f_k - f_{k+1}$.
  7: Calculate $\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k)$, and $\mathrm{F}(\Delta_k)$ as in (24)–(26).
  8: Calculate $\nu_k := \nu_k(\Delta_k)$ using (27).
  9: Calculate $\beta_{k+1}^{\mathrm{FDL}}$ by (20).
 10: Generate $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_{k+1}^{\mathrm{FDL}} \mathbf{d}_k$.
 11: Set $k := k + 1$, and go to Step 2.
```

4. Convergence Examination

The subsequent assumptions are necessary during the theoretical examination of FDL iterations.

Assumption 1. (1) *The level set* $\mho = \{\mathbf{x} \in \mathbb{R}^n | \, f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ *of the iterative process* (3) *is bounded.*

(2) *The objective* f *is continuously differentiable in a neighborhood* $\mathcal{P}$ *of* $\mho$ *with the Lipschitz continuous gradient* $\mathbf{g}$*. Such supposition initiates the existence of a constant* $L > 0$ *which fulfills the inequality*

$$\|\mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{v})\| \leq L\|\mathbf{u} - \mathbf{v}\|, \ \forall \ \mathbf{u}, \mathbf{v} \in \mathcal{P}. \tag{29}$$

Assumption 1 provides the existence of quantities Υ and γ that fulfil

$$\|\mathbf{u} - \mathbf{v}\| \leq \Upsilon, \ \forall \ \mathbf{u}, \mathbf{v} \in \mathcal{P} \tag{30}$$

and

$$\|\mathbf{g}(\mathbf{u})\| \leq \gamma, \ \forall \ \mathbf{u} \in \mathcal{P}. \tag{31}$$

If Assumption 1 holds, given the uniform convexity of f, there exists $\theta > 0$, satisfying

$$(\mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{v}))^{\mathrm{T}}(\mathbf{u} - \mathbf{v}) \geq \theta\|\mathbf{u} - \mathbf{v}\|^2, \ \ \forall \ \mathbf{u}, \mathbf{v} \in \mho, \tag{32}$$

or, equivalently,

$$f(\mathbf{u}) \geq f(\mathbf{v}) + g(\mathbf{v})^{\mathrm{T}}(\mathbf{u} - \mathbf{v}) + \frac{\theta}{2}\|\mathbf{u} - \mathbf{v}\|^2, \ \ \forall \ \mathbf{u}, \mathbf{v} \in \mho. \tag{33}$$

From (32) and (33), it follows

$$\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} \geq \theta\|\mathbf{s}_{k-1}\|^2 \tag{34}$$

and

$$f(\mathbf{x}_{k-1}) - f(\mathbf{x}_k) \geq -\mathbf{g}(\mathbf{x}_k)^{\mathrm{T}}\mathbf{s}_{k-1} + \frac{\theta}{2}\|\mathbf{s}_{k-1}\|^2. \tag{35}$$

By (29) and (34), one concludes

$$\theta\|\mathbf{s}_{k-1}\|^2 \leq \mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} \leq L\|\mathbf{s}_{k-1}\|^2, \tag{36}$$

and, further, $\theta \leq L$.

The inequality (36) implies

$$\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} = \alpha_{k-1}\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} > 0. \tag{37}$$

Taking into account $\alpha_{k-1} > 0$ and the last inequality, we conclude

$$\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} > 0. \tag{38}$$

Lemma 2 ([40,41])**.** *Let the constraints in Assumption 1 hold and the points* $\{\mathbf{x}_k\}$ *be produced by the iterations* (3) *and* (4)*. Then, the following inequality is satisfied:*

$$\sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} < +\infty. \tag{39}$$

Lemma 3. *Observe the suggested fuzzy neutrosophic Dai–Liao CG method defined by* (3), (19), (20)*. If the search procedure enables* (38) *for all* $k \geq 0$*, it follows*

$$\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k \leq -c\|\mathbf{g}_k\|^2 \tag{40}$$

for some, $c \geq 0$.

Proof. In the initial stage, it follows $\mathbf{g}_0^{\mathrm{T}}\mathbf{d}_0 = -\|\mathbf{g}_0\|^2$. Using $c = 1$, it can be concluded that (40) is satisfied in the initial stage $k = 0$. Assume (40) for some $k \geq 1$. Applying the inner product between the left- and right-hand sides of (19) and $\mathbf{g}_k^{\mathrm{T}}$, it is concluded

$$\begin{aligned}
\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k &= -\|\mathbf{g}_k\|^2 + \beta_k^{\mathrm{FDL}}\,\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \\
&= -\|\mathbf{g}_k\|^2 + \left(\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} - \nu_k\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\right)\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \\
&= -\|\mathbf{g}_k\|^2 + \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} - \nu_k\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \\
&= -\|\mathbf{g}_k\|^2 + \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} - \nu_k\frac{\alpha_{k-1}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \\
&= -\|\mathbf{g}_k\|^2 + \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} - \nu_k\frac{\alpha_{k-1}(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})^2}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}.
\end{aligned} \tag{41}$$

Using (21) in conjunction with (38) and $\alpha_{k-1} > 0$, we conclude

$$\nu_k\frac{\alpha_{k-1}(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})^2}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \geq 0. \tag{42}$$

Now, from (41) and (42), it follows

$$\begin{aligned}
\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k &\leq -\|\mathbf{g}_k\|^2 + \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \\
&= \frac{-\|\mathbf{g}_k\|^2(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2 + (\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1})(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})}{(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2}.
\end{aligned} \tag{43}$$

Applying the inequality $P^{\mathrm{T}}Q \leq \frac{1}{2}(\|P\|^2 + \|Q\|^2)$ to the equality (43) with $P = \frac{1}{\sqrt{2}}(\mathbf{d}_{k-1}^{T}\mathbf{y}_{k-1})\mathbf{g}_k$ and $Q = \sqrt{2}(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})\mathbf{y}_{k-1}$, it is obtained

$$\begin{aligned}
\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k &\leq \frac{-\|\mathbf{g}_k\|^2(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2 + \frac{1}{2}\left(\|\frac{1}{\sqrt{2}}(\mathbf{d}_{k-1}^{T}\mathbf{y}_{k-1})\mathbf{g}_k\|^2 + \|\sqrt{2}(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})\mathbf{y}_{k-1}\|^2\right)}{(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2} \\
&= \frac{-\|\mathbf{g}_k\|^2(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2 + \frac{1}{2}\left(\frac{1}{2}(\mathbf{d}_{k-1}^{T}\mathbf{y}_{k-1})^2\|\mathbf{g}_k\|^2 + 2(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})^2\|\mathbf{y}_{k-1}\|^2\right)}{(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2} \\
&= \frac{-\|\mathbf{g}_k\|^2(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2 + \frac{1}{4}(\mathbf{d}_{k-1}^{T}\mathbf{y}_{k-1})^2\|\mathbf{g}_k\|^2 + (\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})^2\|\mathbf{y}_{k-1}\|^2}{(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2} \\
&= -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})^2\|\mathbf{y}_{k-1}\|^2}{(\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2} \\
&\leq -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{\alpha_{k-1}^2(|\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1}|)^2\|\mathbf{y}_{k-1}\|^2}{(\alpha_{k-1}\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2} \\
&\leq -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{\alpha_{k-1}^2\|\mathbf{g}_k\|^2\|\mathbf{d}_{k-1}\|^2\|\mathbf{y}_{k-1}\|^2}{(\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1})^2} \\
&\leq -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{\|\mathbf{g}_k\|^2\|\alpha_{k-1}\mathbf{d}_{k-1}\|^2\|\mathbf{y}_{k-1}\|^2}{\theta^2\|\mathbf{s}_{k-1}\|^4} \\
&\leq -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{\|\mathbf{g}_k\|^2\|\mathbf{s}_{k-1}\|^2\|\mathbf{y}_{k-1}\|^2}{\theta^2\|\mathbf{s}_{k-1}\|^4} \\
&\leq -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{\|\mathbf{g}_k\|^2\|\mathbf{y}_{k-1}\|^2}{\theta^2\|\mathbf{s}_{k-1}\|^2} \\
&\leq -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{L^2\|\mathbf{g}_k\|^2\|\mathbf{s}_{k-1}\|^2}{\theta^2\|\mathbf{s}_{k-1}\|^2} \\
&= -\|\mathbf{g}_k\|^2 + \frac{1}{4}\|\mathbf{g}_k\|^2 + \frac{L^2}{\theta^2}\|\mathbf{g}_k\|^2 \\
&= -\left(1 - \frac{1}{4} - \frac{L^2}{\theta^2}\right)\|\mathbf{g}_k\|^2 = -\left(\frac{3}{4} - \frac{L^2}{\theta^2}\right)\|\mathbf{g}_k\|^2.
\end{aligned} \tag{44}$$

The requirement (40) is satisfied for $c = \left(\frac{3}{4} - \frac{L^2}{\theta^2}\right)$ in (44) and an arbitrary $k \geq 0$. □

Theorem 1 confirms the global convergence of the FDL flow.

Theorem 1. *Let the constraints in Assumption 1 be valid and f be uniformly convex. Under these conditions, iterations $\{\mathbf{x}_k\}$ generated by (3), (19), and (20) fulfill the limit relation*

$$\liminf_{k\to\infty} \|\mathbf{g}_k\| = 0. \tag{45}$$

Proof. Suppose the opposite. Since (45) is not valid, we conclude the existence of $c_1 > 0$, satisfying

$$\|\mathbf{g}_k\| \geq c_1, \quad \text{for all} \;\; k. \tag{46}$$

Squaring both sides of (19), one derives

$$\|\mathbf{d}_k\|^2 = \|\mathbf{g}_k\|^2 - 2\beta_k^{\mathrm{FDL}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} + (\beta_k^{\mathrm{FDL}})^2\|\mathbf{d}_{k-1}\|^2. \tag{47}$$

Taking into account (20), we obtain

$$\begin{aligned} -2\beta_k^{\mathrm{FDL}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} &= -2\left(\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} - \nu_k\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\right)\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \\ &= -2\left(\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} - \nu_k\frac{\alpha_{k-1}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1}\right) \\ &= -2\left(\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} - \nu_k\frac{\alpha_{k-1}(\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1})^2}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\right). \end{aligned} \tag{48}$$

Now, from (42), it follows

$$\begin{aligned} -2\beta_k^{\mathrm{EDL}}\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1} \leq 2\left|\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\right||\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_{k-1}| &\leq 2\frac{\|\mathbf{g}_k\|\|\mathbf{y}_{k-1}\|}{\left|\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}\right|}\|\mathbf{g}_k\|\|\mathbf{d}_{k-1}\| \\ &= 2\frac{\alpha_{k-1}\|\mathbf{g}_k\|^2\|\mathbf{y}_{k-1}\|\|\mathbf{d}_{k-1}\|}{\alpha_{k-1}\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \\ &= 2\frac{\|\mathbf{g}_k\|^2\|\mathbf{y}_{k-1}\|\|\mathbf{s}_{k-1}\|}{\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \\ &\leq 2\frac{\|\mathbf{g}_k\|^2 L\|\mathbf{s}_{k-1}\|\|\mathbf{s}_{k-1}\|}{\theta\|\mathbf{s}_{k-1}\|^2} \\ &= \frac{2L}{\theta}\|\mathbf{g}_k\|^2. \end{aligned} \tag{49}$$

Now, an application of (20) initiates

$$\begin{aligned} \beta_k^{\mathrm{FDL}} &= \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} - \nu_k\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} = \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1} - \nu_k\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \\ &\leq \left|\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1} - \nu_k\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}\right| = \alpha_{k-1}\frac{|\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1} - \nu_k\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}|}{\alpha_{k-1}\mathbf{d}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \\ &= \alpha_{k-1}\frac{|\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1} - \nu_k\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}|}{\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \\ &\leq \alpha_{k-1}\frac{|\mathbf{g}_k^{\mathrm{T}}\mathbf{y}_{k-1} - \nu_k\mathbf{g}_k^{\mathrm{T}}\mathbf{s}_{k-1}|}{\theta\|\mathbf{s}_{k-1}\|^2} = \alpha_{k-1}\frac{|\mathbf{g}_k^{\mathrm{T}}(\mathbf{y}_{k-1} - \nu_k\mathbf{s}_{k-1})|}{\theta\alpha_{k-1}^2\|\mathbf{d}_{k-1}\|^2} \\ &\leq \frac{\|\mathbf{g}_k\|(\|\mathbf{y}_{k-1}\| + \nu_k\|\mathbf{s}_{k-1}\|)}{\theta\alpha_{k-1}\|\mathbf{d}_{k-1}\|^2} \\ &\leq \frac{\|\mathbf{g}_k\|(L\|\mathbf{s}_{k-1}\| + \nu_k)\|\mathbf{s}_{k-1}\|)}{\theta\alpha_{k-1}\|\mathbf{d}_{k-1}\|^2} = \frac{\|\mathbf{g}_k\|(L + \nu_k)\|\mathbf{s}_{k-1}\|}{\theta\alpha_{k-1}\|\mathbf{d}_{k-1}\|^2} \\ &= \frac{\|\mathbf{g}_k\|(L + \nu_k)\alpha_{k-1}\|\mathbf{d}_{k-1}\|}{\theta\alpha_{k-1}\|\mathbf{d}_{k-1}\|^2} \\ &= \frac{(L + \nu_k)\|\mathbf{g}_k\|}{\theta\|\mathbf{d}_{k-1}\|}. \end{aligned} \tag{50}$$

Further, (21) and (50) lead to

$$\beta_k^{\text{FDL}} \leq \frac{(L+1)\|\mathbf{g}_k\|}{\theta\|\mathbf{d}_{k-1}\|}. \tag{51}$$

Using (49) and (51) in (47), we obtain

$$\begin{aligned}
\|\mathbf{d}_k\|^2 &\leq \|\mathbf{g}_k\|^2 + \frac{2L}{\theta}\|\mathbf{g}_k\|^2 + \frac{(L+1)^2\|\mathbf{g}_k\|^2}{\theta^2\|\mathbf{d}_{k-1}\|^2}\|\mathbf{d}_{k-1}\|^2 \\
&= \|\mathbf{g}_k\|^2 + \frac{2L}{\theta}\|\mathbf{g}_k\|^2 + \frac{(L+1)^2\|\mathbf{g}_k\|^2}{\theta^2} \\
&= \left(1 + \frac{2L}{\theta} + \frac{(L+1)^2}{\theta^2}\right)\|\mathbf{g}_k\|^2 \\
&= \left(\frac{\theta+2L}{\theta} + \frac{(L+1)^2}{\theta^2}\right)\|\mathbf{g}_k\|^2 \\
&= \frac{(\theta+2L)\theta + (L+1)^2}{\theta^2}\|\mathbf{g}_k\|^2.
\end{aligned} \tag{52}$$

Dividing both sides of (52) by $\|\mathbf{g}_k\|^4$ and utilizing (46), it can be decided

$$\begin{aligned}
\frac{\|\mathbf{d}_k\|^2}{\|\mathbf{g}_k\|^4} &\leq \frac{(\theta+2L)\theta + (L+1)^2}{\theta^2} \cdot \frac{1}{c_1^2}, \\
\frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} &\geq \frac{\theta^2 \cdot c_1^2}{(\theta+2L)\theta + (L+1)^2}.
\end{aligned} \tag{53}$$

The inequalities in (53) imply

$$\sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} \geq \sum_{k=0}^{\infty} \frac{\theta^2 \cdot c_1^2}{(\theta+2L)\theta + (L+1)^2} = \infty. \tag{54}$$

Therefore, $\|\mathbf{g}_k\| \geq c_1$ causes a contradiction with Lemma 2. □

5. Numerical Results

In current section, the numerical results obtained by the FDL method are analyzed and compared with the numerical results generated by the EDL [21] and DL [1] methods.

All the algorithms were implemented in Matlab R2017a and executed on a 64-bit Lenovo laptop (Intel Core i3 2.0 GHz, RAM 8 GB) with the Windows 10 operating system. The implementation of the FDL method is based on Algorithm 2, while the implementation of EDL and DL is based on algorithms given in [21] and [1], respectively.

The numerical testing is performed on 50 test functions collected in [42,43], with dimensions in the range $[100, 20000]$. All three tested methods start from the same initial point $\mathbf{x}_0$ for each test example. Each case is evaluated 10 times with progressively increased dimensions $n = 10^2, 5 \times 10^2, 10^3, 3 \times 10^3, \times 10^3, 7 \times 10^3, 8 \times 10^3, 10^4, 1.5 \times 10^4$, and 2×10^4.

The uniform terminating criteria for the observed DL, EDL, and FDL algorithms are

$$\|\mathbf{g}_k\| \leq \epsilon \quad \text{and} \quad \frac{|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)|}{1 + |f(\mathbf{x}_k)|} \leq \delta, \;\; \epsilon = 10^{-6},\, \delta = 10^{-16}.$$

We will evaluate the efficiency of the FDL method and compare it with the EDL and DL methods under the backtracking search that uses the values $\omega = 0.0001$ and $\varphi = 0.8$.

Summary numerical results for DL, FDL, and EDL methods, performed on 50 test functions, are shown in Table 2, where 'Test function', 'Nitr', 'Nfe', and 'Tcpu' represent the name of the tested function, total number of iterative steps, total number of function evaluations, and the running time, respectively. The best results between the DL, FDL, and EDL methods in Table 2 are marked by bold text in shaded cells.

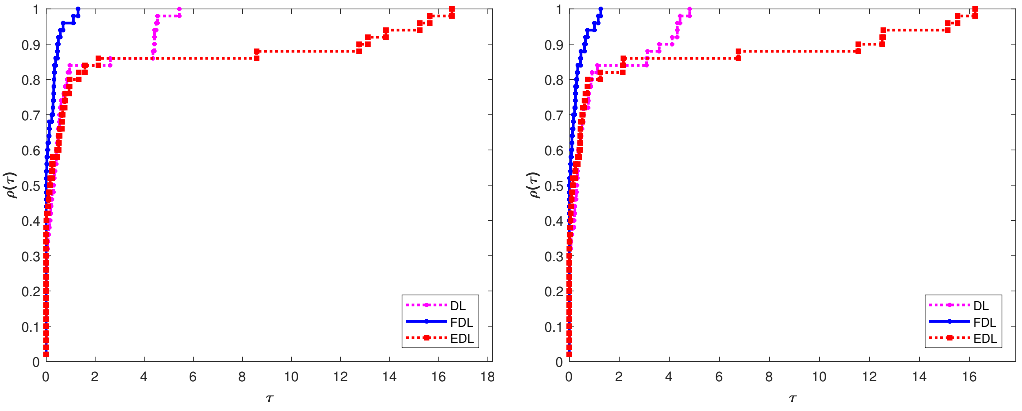
Table 2. Summary of numerical results on unconstrained problems of DL, FDL, and EDL methods for the Nitr, Nfe, and Tcpu.

Test Function	**DL**	**FDL**	**EDL**
	Nitr/Nfe/Tcpu	**Nitr/Nfe/Tcpu**	**Nitr/Nfe/Tcpu**
Extended Penalty	1905/77,578 /32.438	**1610/62,534/24.266**	2304/82,602/39.344
Perturbed Quadratic	14,555/60,6750/379.359	10,800/440213/**206.5**	**10,012**/**408,474**/248.5
Raydan 1	4337/114,595/98.984	5497/122,843/**76.813**	**4194**/**109,164**/96.938
Raydan 2	1427/2864/3.188	**67/144/0.281**	2,572,540/5,145,090/894.453
Diagonal 1	5809/223,750/245.578	5488/212,491/227.781	**4673/178,295/219.109**
Diagonal 3	5247/196,745/423.766	**4531/168,162/307.594**	4596/171636/366.203
Hager	1742/31,516/103.672	**1242/22,799/47.063**	1940/33,206/98.766
Generalized Tridiagonal 1	**2058**/32,313/49.5	2160/**32,033**/**27.5**	2161/33285/44.703
Extended Tridiagonal 1	310/2932/8.391	**182/2501/6.297**	308/4129/12.766
Extended TET	1140/9840/11.031	**619/5808/5.484**	749/6362/5.969
Diagonal 5	1394/2798/6.938	**60/130/0.609**	3,053,907/6,107,824/3124.875
Extended Himmelblau	**50**/2431/1.016	51/2602/**0.813**	**50**/**2413**/0.938
Perturbed quadratic diagonal	1837/69,156/18.453	**1261/36,785/13.875**	2157/86,977/34.797
Quadratic QF1	13,895/526,995/187.313	21,989/846,402/376.156	**10,199/379,554/122.844**
Extended quadratic penalty QP1	**1080/17,440**/9.922	1524/23,840/**8.25**	1157/18043/9.109
Extended quadratic penalty QP2	218/9479/11.047	**112/5513/4.953**	218/9194/8.906
Quadratic QF2	19,211/847,031/348.781	18,861/816,310/**225.891**	**15,555/689,736**/250.891
Extended quadratic exponential EP1	1254/3443/3.172	**56/404/0.516**	21,431/43,829/7.531
Extended Tridiagonal 2	22,468/998,473/549.484	**3668/114,169/87.438**	10989/510713/93.609
TRIDIA (CUTE)	33,278/1,647,913/967.234	40,156/1,977,068/950.547	**29,133/1,428,866/675.422**
ARWHEAD (CUTE)	1624/81,625/44.875	1529/72,379/31.594	**1219/57140/28.672**
Almost Perturbed Quadratic	14,904/621,925/259.797	19,675/829,784/357.359	**13,201/543,372/188.047**
LIARWHD (CUTE)	**30**/**2705**/1.281	**30**/2732/**1.25**	**30**/2739/1.438
POWER (CUTE)	**532,442/44,419,504/16,742.672**	580790/48609979/17435.609	629342/52,431,424/23,630.781
ENGVAL1 (CUTE)	2489/33,103/13.781	2400/32,299/**10.719**	**1975/27,260**/12.922
INDEF (CUTE)	**21/1924/2.125**	26/2238/2.5	30/2610/4.266
Diagonal 6	1583/3197/4.531	**74/185/0.359**	7,052,401/14105032/5037.219
DIXON3DQ (CUTE)	320,921/1775846/1083.281	**229,757/1,368,033/727.172**	257,451/1,517,252/1045.328
COSINE (CUTE)	**20/1600/1.891**	**20**/1697/**1.891**	**20**/1700/2
BIGGSB1 (CUTE)	249,919/1,400,798/832.375	259,475/1,549,293/**810.766**	**236,612/1,389,720**/945.672
Generalized Quartic	**866**/11,273/3.984	1099/**8951**/4.063	959/10,662/**3.125**
Diagonal 7	1453/4564/6.875	**68/162/0.469**	469,477/940,686/140.172
Diagonal 8	1371/3962/5.359	**67/199/0.422**	594,522/1,193,760/195.094
Full Hessian FH3	2237/6202/7.125	**52/513/0.688**	767,988/1,537,759/188.469
Diagonal 9	**3312/138,545**/225.719	5344/217,150/**224.906**	4520/189,307/260.453
HIMMELH (CUTE)	**20/1690**/4.797	**20**/1758/**4.531**	**20**/1760/4.891
FLETCHCR (CUTE)	303,212/10,189,775/5073.688	300,227/10011849/4704.125	**289,670/9,702,961/4411.453**
Extended BD1 (Block Diagonal)	1597/16,783/7.625	1227/15,639/**5.875**	**1200/12,605**/6.625
Extended Maratos	72/3366/1.188	50/2069/**0.719**	**40/1975**/0.75
Extended Cliff	234/**2992/2.078**	**217**/6000/4.891	950/13,187/6.188
Extended Hiebert	**70/7215**/1.938	**70**/7220/**1.828**	**70**/7228/1.859
NONDIA (CUTE)	33/3066/1.375	**30/2829/1.266**	32/3031/1.625
NONDQUAR (CUTE)	58/4652/18.047	**45/3666/17.219**	86/4989/19.016
DQDRTIC (CUTE)	3456/87,105/26.453	**2327/59,047/16.406**	3637/92315/34.953
Extended Freudenstein and Roth	**1376/46,597/10.734**	3390/111,830/28.516	2018/66,654/16.172
Generalized Rosenbrock	282,948/8,410,218/4125.516	**280,440/8,335,396**/4088.547	281,792/8,373,946/**4055.172**
Extended White and Holst	76/5794/9.219	**50/3171/7.281**	59/4022/11.563
Extended Beale	118/6791/14.047	**72/3118/5.906**	181/4748/6.75
EG2 (CUTE)	**507/29,388/47.547**	697/48,512/119.875	811/39,769/122.469
EDENSCH (CUTE)	1694/23,160/89.453	2089/27,821/**83.266**	**1684/22,731**/116.844

To visually compare the performance of the opposed methods, we used the performance profiles technique [44] on the numerical results corresponding to Nitr, Nfe, and Tcpu criteria generated by the DL, FDL, and EDL methods. An upper graph in a performance profile corresponds to the method that shows better performance. The vertical axis of each performance profile in the figures indicates the percentage of test functions for which the considered method is the winner between compared methods, whereby the right-hand side corresponds to the percentage of successfully solved test functions.

Figures 4 and 5 plot the performance profiles for the data in Table 2. The graphs in Figure 4 illustrate the performance profiles Nitr and Nfe for DL, FDL, and EDL iterations based on the data from Table 2. In Figure 4a, it is noticeable that the DL, FDL, and EDL methods can solve all the tested functions. Numerical experience shows that the FDL method produces the best results in 54.0% (27 out of 50) of test functions compared with DL (26.0% (13 out of 50)) and EDL (38.0% (19 out of 50)). From Figure 4a, the FDL graph touches the top first, so FDL is the best relative to the other examined methods concerning the Nitr criterion.

Figure 4b indicates that the FDL graph is the most efficient and successfully solves all test cases. In addition, the obtained numerical results confirm that FDL performs well in most cases. Specifically, FDL is the fastest because it solves about 48.0% (24 out of 50) of the tested functions with the least Nfe compared to the DL and EDL methods. Meanwhile, DL and EDL are superior in solving 22.0% (11 out of 50) and 30.0% (15 out of 50) test functions, respectively. Hence, FDL is superior compared to the DL and EDL methods for the given test functions.



(**a**) Nitr performance profile. (**b**) Nfe performance profile.

Figure 4. Performance profiles for DL, FDL, and EDL methods.

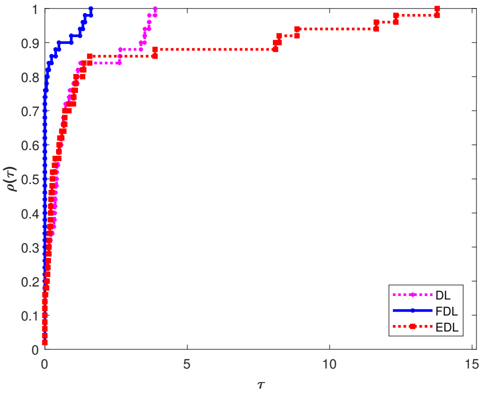


Figure 5. Tcpu performance profiles for DL, FDL, and EDL methods.

Figure 5 shows Tcpu performance profile graphs of the DL, FDL, and EDL methods. It is observable that DL, FDL, and EDL are able to solve all the tested functions. Further examination leads to the conclusion that the FDL method is the best in 74.0% (thirty-seven out of fifty) test cases compared to DL (12.0% (six out of fifty)) and EDL (16.0% (eight out of fifty)). Analyzing the graphs in Figure 5, a clear conclusion is that the FDL graph first achieves the top, which certifies its superiority in terms of Tcpu.

The general conclusion from the results presented in Table 2 is that one of the methods DL, FDL, and EDL is the absolute winner in thirty-two tested functions relative to all three criteria (Nitr, Nfe, and Tcpu). Further analysis of these data leads to the following conclusions:

- the DL method is an absolute winner in five cases;
- the FDL method is an absolute winner in twenty-one cases;
- the EDL method is an absolute winner in six cases.

In addition to these facts, the data contained in Table 2 show that the FDL method is better in six cases than the other two methods in terms of Nitr. The FDL method is better in three cases compared to DL and EDL in terms of Nfe. The FDL method achieves the best results in sixteen test cases in terms of Tcpu.

The comparisons of the DL, FDL, and EDL methods are shown in Figures 4 and 5. As can be noticed, FDL is superior compared to the DL and EDL methods with respect to three criteria (iterations, function evaluations, and processor time). Moreover, the numerical results in Table 2 indicate that the FDL method is the absolute winner. Thus, numerical experiments indicate the effectiveness of the proposed FDL method.

6. Conclusions

In the current research, we propose a novel approach to determining the parameter t in Dai–Liao CG iterations. The new approach is based on finding suitable values for a non-negative parameter t in the DL method using neutrosophic logic. An original strategy in defining the Dai–Liao CG parameter β_k^{FDL} is suggested utilizing $t := \nu_k(\Delta_k)$ in (20), and a novel fuzzy neutrosophic Dai–Liao (FDL) CG method is presented.

Numerical experiments and comparisons with some well-known CG methods and the theoretical convergence analysis demonstrate the effectiveness of the proposed method. The numerical testing and initiated comparison are based on standard performance profiles, such as total number of iterations (Nitr), total number of function evaluations (Nfe), and runtime (Tcpu). Analysis of the obtained numerical results revealed that the FDL method is the most efficient.

We are assured that the proposed methods will serve as motivation for further investigation in defining improvements of DL iterations guided by the neutrosophic logic.

Future scientific research in this area can continue in several directions. Previous research has shown the effectiveness of the neutrosophic principle in scaled gradient descent methods and the DL class of CG methods. The challenge is to apply such a principle to other nonlinear optimization methods. On the other hand, there is a wide range of possibilities for defining further neutrosophication and de-neutrosophication rules, which can be considered in future research. Finally, there is a great opportunity to improve the neutrosophic inference engine used in this research.

Author Contributions: Conceptualization, P.S.S. and B.D.I.; methodology, P.S.S., V.N.K. and L.A.K.; software, B.D.I.; validation, D.S., V.N.K., P.S.S. and L.A.K.; formal analysis, D.S., P.S.S., V.N.K. and D.K.; investigation, P.S.S., D.S., V.N.K. and L.A.K.; resources, B.D.I. and D.K.; data curation, B.D.I. and D.S.; writing—original draft, P.S.S., B.D.I. and D.S.; writing—review and editing, P.S.S., D.K. and D.S.; visualization, B.D.I. and D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data and code will be provided on request to authors.

Acknowledgments: Predrag Stanimirović is supported by the Science Fund of the Republic of Serbia (No. 7750185, Quantitative Automata Models: Fundamental Problems and Applications—QUAM).

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

References

1. Dai, Y.-H.; Liao, L.-Z. New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.* **2001**, *43*, 87–101. [CrossRef]
2. Cheng, Y.; Mou, Q.; Pan, X.; Yao, S. A sufficient descent conjugate gradient method and its global convergence. *Optim. Methods Softw.* **2016**, *31*, 577–590. [CrossRef]
3. Livieris, I.E.; Pintelas, P. A descent Dai-Liao conjugate gradient method based on a modified secant equation and its global convergence. *Isrn Comput. Math.* **2012**, *2012*, 435495. [CrossRef]
4. Peyghami, M.R.; Ahmadzadeh, H.; Fazli, A. A new class of efficient and globally convergent conjugate gradient methods in the Dai-Liao family. *Optim. Methods Softw.* **2015**, *30*, 843–863. [CrossRef]
5. Yabe, H.; Takano, M. Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. *Comput. Optim. Appl.* **2004**, *28*, 203–225. [CrossRef]
6. Yao, S.; Qin, B. A hybrid of DL and WYL nonlinear conjugate gradient methods. *Abstr. Appl. Anal.* **2014**, *2014*, 279891. [CrossRef]
7. Yao, S.; Lu, X.; Wei, Z. A conjugate gradient method with global convergence for large-scale unconstrained optimization problems. *J. Appl. Math.* **2013**, *2013*, 730454. [CrossRef]
8. Zheng, Y.; Zheng, B. Two new Dai-Liao-type conjugate gradient methods for unconstrained optimization problems. *J. Optim. Theory Appl.* **2017**, *175*, 502–509. [CrossRef]
9. Zhou, W.; Zhang, L. A nonlinear conjugate gradient method based on the MBFGS secant condition. *Optim. Methods Softw.* **2006**, *21*, 707–714. [CrossRef]
10. Waziri, M.Y.; Ahmed, K.; Sabiù, J. A Dai–Liao conjugate gradient method via modified secant equation for system of nonlinear equations. *Arab. J. Math.* **2020**, *9*, 443–457. [CrossRef]
11. Khoshsimaye–Bargard, M.; Ashrafi, A. A descent spectral Dai–Liao method based on the quasi–Newton aspects. *Numer. Algor.* **2023**. [CrossRef]
12. Babaie-Kafaki, S. A survey on the Dai-Liao family of nonlinear conjugate gradient methods. *RAIRO-Oper. Res.* **2023**, *57*, 43–58. [CrossRef]
13. Andrei, N. Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization. *Bull. Malays. Math. Sci. Soc.* **2011**, *34*, 319–330.
14. Hager, W.W.; Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **2005**, *16*, 170–192. [CrossRef]
15. Hager, W.W.; Zhang, H. Algorithm 851: CG DESCENT, a conjugate gradient method with guaranteed descent. *Acm Trans. Math. Softw.* **2006**, *32*, 113–137. [CrossRef]
16. Dai, Y.-H.; Kou, C.-X. A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search. *SIAM. J. Optim.* **2013**, *23*, 296–320. [CrossRef]
17. Babaie-Kafaki, S.; Ghanbari, R. The Dai-Liao nonlinear conjugate gradient method with optimal parameter choices. *Europ. J. Oper. Res.* **2014**, *234*, 625–630. [CrossRef]
18. Andrei, N. A Dai-Liao conjugate gradient algorithm with clustering of eigenvalues. *Numer. Algor.* **2018**, *77*, 1273–1282. [CrossRef]
19. Babaie-Kafaki, S. On the sufficient descent condition of the Hager-Zhang conjugate gradient methods. *4OR-Q J. Oper. Res.* **2014**, *12*, 285–292. [CrossRef]
20. Lotfi, M.; Hosseini, S.M. An efficient Dai–Liao type conjugate gradient method by reformulating the CG parameter in the search direction equation. *J. Comput. Appl. Math.* **2020**, *371*, 112708. [CrossRef]
21. Ivanov, B.; Stanimirović, P.S.; Shaini, B.I.; Ahmad, H.; Wang, M.-K. A Novel Value for the Parameter in the Dai-Liao-Type Conjugate Gradient Method. *J. Funct. Spaces* **2021**, *2021*, 6693401. [CrossRef]
22. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
23. Atanassov, K.T. Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **1986**, *20*, 87–96. [CrossRef]
24. Smarandache, F. *A Unifying Field in Logics, Neutrosophy: Neutrosophic Probability, Set and Logic*; American Research Press: Rehoboth, NM, USA, 1999.
25. Wang, H.; Smarandache, F.; Zhang, Y.Q.; Sunderraman, R. Single valued neutrosophic sets. *Multispace Multistruct.* **2010**, *4*, 410–413.
26. Smarandache, F. Special Issue "New types of Neutrosophic Set/Logic/Probability, Neutrosophic Over-/Under-/Off-Set, Neutrosophic Refined Set, and their Extension to Plithogenic Set/Logic/Probability, with Applications". *Symmetry* **2019**. Available online: https://www.mdpi.com/journal/symmetry/special_issues/Neutrosophic_Set_Logic_Probability (accessed on 30 April 2023).
27. Mishra, K.; Kandasamy, I.; Kandasamy, W.B.; Smarandache, F. A novel framework using neutrosophy for integrated speech and text sentiment analysis. *Symmetry* **2020**, *12*, 1715. [CrossRef]
28. Tu, A.; Ye, J.; Wang, B. Symmetry measures of simplified neutrosophic sets for multiple attribute decision-making problems. *Symmetry* **2018**, *10*, 144. [CrossRef]

29. Smarandache, F. (Ed.) *Collected Papers (On Physics, Artificial Intelligence, Health Issues, Decision Making, Economics, Statistics)*; Global Knowledge Publishing House: Miami, FL, USA, 2022; Volume 11. Available online: http://fs.unm.edu/CP11.pdf (accessed on 30 April 2023).
30. Smarandache, F. (Ed.) *Collected Papers (On Various Scientific Topics)*; Global Knowledge Publishing House: Miami, FL, USA, 2022; Volume 13. Available online: http://fs.unm.edu/CP13.pdf (accessed on 30 April 2023).
31. Stanimirović, P.S.; Ivanov, B.; Stanujkić, D.; Katsikis, V.N.; Mourtas, S.D.; Kazakovtsev, L.A.; Edalatpanah, S.A. Improvement of Unconstrained Optimization Methods Based on Symmetry Involved in Neutrosophy. *Symmetry* **2023**, *15*, 250. [CrossRef]
32. Dai, J.; Chen, Y.; Xiao, L.; Jia, L.; He, Y. Design and analysis of a hybrid GNN-ZNN model with a fuzzy adaptive factor for matrix inversion. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2434–2442. [CrossRef]
33. Deng, Y.; Ren, Z.; Kong, Y.; Bao, F.; Dai, Q. A hierarchical fused fuzzy deep neural network for data classification. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1006–1012. [CrossRef]
34. Jia, L.; Xiao, L.; Dai, J.; Cao, Y. A novel fuzzy-power zeroing neural network model for time-variant matrix Moore-Penrose inversion with guaranteed performance. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 2603–2611. [CrossRef]
35. Jia, L.; Xiao, L.; Dai, J.; Qi, Z.; Zhang, Z.; Zhang, Y. Design and Application of an Adaptive Fuzzy Control Strategy to Zeroing Neural Network for Solving Time-Variant QP Problem. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1544–1555. [CrossRef]
36. Katsikis, V.N.; Stanimirović, P.S.; Mourtas, S.D.; Xiao, L.; Karabasević, D.; Stanujkić, D. Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 3426–3435. [CrossRef]
37. Hestenes, M.R.; Stiefel, E.L. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **1952**, *49*, 409–436. [CrossRef]
38. Andrei, N. An acceleration of gradient descent algorithm with backtracking for unconstrained optimization. *Numer. Algorithms* **2006**, *42*, 63–73. [CrossRef]
39. Stanimirović, P.S.; Miladinović, M.B. Accelerated gradient descent methods with line search. *Numer. Algorithms* **2010**, *54*, 503–520. [CrossRef]
40. Cheng, W. A two-term PRP-based descent method. *Numer. Funct. Anal. Optim.* **2007**, *28*, 1217–1230. [CrossRef]
41. Zoutendijk, G. Nonlinear Programming, Computational Methods. In *Integer and Nonlinear Programming*; Abadie, J., Ed.; Springer: Amsterdam, The Netherlands, 1970; pp. 37–86
42. Andrei, N. An unconstrained optimization test functions collection. *Adv. Model. Optim.* **2008**, *10*, 147–161.
43. Bongartz, I.; Conn, A.R.; Gould, N.; Toint, P.L. CUTE: Constrained and unconstrained testing environments. *ACM Trans. Math. Softw.* **1995**, *21*, 123–160. [CrossRef]
44. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.