symmetry

MDPI

Article

Improvement of Unconstrained Optimization Methods Based on Symmetry Involved in Neutrosophy

Predrag S. Stanimirović [1,2,*], Branislav Ivanov [3], Dragiša Stanujkić [3], Vasilios N. Katsikis [4], Spyridon D. Mourtas [2,4], Lev A. Kazakovtsev [2,5], Seyyed Ahmad Edalatpanah [6]

1 Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia
2 Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, Krasnoyarsk 660041, Russia; spirmour@econ.uoa.gr (S.D.M.); levk@bk.ru (L.A.K.)
3 Technical Faculty in Bor, University of Belgrade, Vojske Jugoslavije 12, 19210 Bor, Serbia; bivanov@tfbor.bg.ac.rs (B.I.); dstanujkic@tfbor.bg.ac.rs (D.S.)
4 Department of Economics, Division of Mathematics and Informatics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece; vaskatsikis@econ.uoa.gr
5 Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology, Prosp. Krasnoyarskiy Rabochiy 31, Krasnoyarsk 660037, Russia
6 Department of Applied Mathematics, Ayandegan Institute of Higher Education, Tonekabon P.O. Box 46818-53617, Mazandaran, Iran; s.a.edalatpanah@aihe.ac.ir
* Correspondence: pecko@pmf.ni.ac.rs

Abstract: The influence of neutrosophy on many fields of science and technology, as well as its numerous applications, are evident. Our motivation is to apply neutrosophy for the first time in order to improve methods for solving unconstrained optimization. Particularly, in this research, we propose and investigate an improvement of line search methods for solving unconstrained nonlinear optimization models. The improvement is based on the application of symmetry involved in neutrosophic logic in determining appropriate step size for the class of descent direction methods. Theoretical analysis is performed to show the convergence of proposed iterations under the same conditions as for the related standard iterations. Mutual comparison and analysis of generated numerical results reveal better behavior of the suggested iterations compared with analogous available iterations considering the Dolan and Moré performance profiles and statistical ranking. Statistical comparison also reveals advantages of the neutrosophic improvements of the considered line search optimization methods.

Keywords: unconstrained optimization; neutrosophic logic systems; gradient descent methods; convergence

MSC: 90C70; 90C30; 65K05

check for updates

Citation: 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. https://doi.org/10.3390/sym15010250

Academic Editors: Alexander Zaslavski and Wei-Shih Du

Received: 1 December 2022
Revised: 7 January 2023
Accepted: 12 January 2023
Published: 16 January 2023

1. Introduction, Preliminaries, and Motivation

We investigate applications of neutrosophic logic in determining an additional step size in gradient descent methods for solving the multivariate unconstrained optimization problem

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

in which the objective $f : \mathbb{R}^n \to \mathbb{R}$ is uniformly convex and twice continuously differentiable.

The most general iteration aimed to solve (1) is the descent direction (DD) method

$$x_{k+1} = x_k + t_k d_k, \tag{2}$$

such that x_{k+1} is the actual approximation, x_k is the former approximation, $t_k > 0$ is a step size, and d_k is an appropriate search direction that satisfies the descent condition

$g_k^{\mathrm{T}} d_k < 0$, in which $g_k = \nabla f(x_k)$ stands for the gradient vector of the objective f. The most common choice is the antigradient direction $d_k = -g_k$, leading to the gradient descent (GD) iterations

$$x_{k+1} = x_k - t_k g_k, \tag{3}$$

in which the learning rate t_k is typically determined by an inexact line search procedure. The iterative rule of the general quasi-Newton (QN) class of iterations with line search

$$x_{k+1} = x_k - t_k H_k\, g_k \tag{4}$$

utilizes an appropriate symmetric and positive definite estimation B_k of the Hessian $G_k = \bigtriangledown^2 f(x_k)$ and $H_k = B_k^{-1}$ [1]. The upgrade B_{k+1} from B_k is established based on the QN characteristic

$$B_{k+1}\varsigma_k = \xi_k, \quad \text{such that} \quad \varsigma_k = x_{k+1} - x_k,\ \ \xi_k = g_{k+1} - g_k. \tag{5}$$

Computation of the Hessian or its approximations that include matrix operations is time-consuming and prohibitive. Following the goal to make optimization methods efficient in solving large-scale problems, we use the simplest scalar Hessian's approximation [2,3]:

$$B_k = \gamma_k I,\ \ \gamma_k > 0. \tag{6}$$

In this paper, we are interested in the following iterative scheme

$$x_{k+1} = x_k - \gamma_k^{-1} t_k g_k. \tag{7}$$

Iterations (7) are introduced as *improved gradient descent* (IGD) methods. The roles of the additional step γ_k and the basic step length t_k are clearly separated and complement each other. The quantity t_k is defined as the output of an inexact line search methodology, while γ_k is calculated based on Taylor series of $f(x)$.

Diverse forms and improvements of the IGD iterative scheme (7) were suggested in [4–8]. The SM method proposed in [6] corresponds to the iteration

$$x_{k+1} = x_k - t_k(\gamma_k^{SM})^{-1} g_k, \tag{8}$$

where $\gamma_k^{SM} > 0$ is the gain parameter determined utilizing the Taylor approximation of $f(x_k - t_k(\gamma_k^{SM})^{-1} g_k)$, which results

$$\gamma_{k+1}^{SM} = \mho\left(2\gamma_k^{SM} \frac{\gamma_k^{SM}\Delta_k + t_k\|g_k\|^2}{t_k^2\|g_k\|^2}\right),$$

such that $f_p := f(x_p)$, $\Delta_k := f_{k+1} - f_k$ and

$$\mho(x) = \begin{cases} x, & x > 0 \\ 1, & x \leq 0. \end{cases}$$

The modification of the SM method was defined as the transformation $MSM = \mathcal{M}(SM)$ [9]

$$x_{k+1} = \mathcal{M}(SM)(x_k) = x_k - t_k\tau_k(\gamma_k^{MSM})^{-1} g_k, \tag{9}$$

where $t_k \in (0,1)$ is defined by the backtracking search, $\tau_k = 1 + t_k - t_k^2$, and

$$\gamma_{k+1}^{MSM} = \mho\left(2\gamma_k^{MSM} \frac{\gamma_k^{MSM}\Delta_k + t_k\tau_k\|g_k\|^2}{(t_k\tau_k)^2\|g_k\|^2}\right). \tag{10}$$

We propose improvements of line search iterative rules for solving (1). The main idea is based on the application of neutrosophic logic in determining appropriate step length for various gradient descent rules. This idea is based on the hybridization principle proposed in [5,9,10], where an appropriate correction parameter α_k with a fixed value is used. A hybridization of the SM iterations (termed HSM) was introduced in [5] as the iterative rule

$$x_{k+1} = H(SM)(x_k) = x_k - (\eta_k + 1)(\gamma_k^{HSM})^{-1} t_k g_k, \tag{11}$$

such that η_k is the correction quantity and γ_k^{HSM} is the gain value defined as

$$\gamma_{k+1}^{HSM} = \mho\left(2\gamma_k^{HSM} \frac{\gamma_k^{HSM}\Delta_k + (\eta_k + 1)t_k\|g_k\|^2}{(\eta_k + 1)^2 t_k^2 \|g_k\|^2} \right).$$

The hybridizations of several IGD methods, including the MSM method, were proposed and investigated in [9,10]. An overview of methods derived by the hybridization of IGD iterations with the Picard–Mann, Ishikawa, and Khan iterative processes [11–13] was given in [14]. Some common fixed point results for fuzzy mappings were derived in [15]. A detailed numerical comparison between hybrid and nonhybrid IGD methodswas performed in [14]. Four gradient descent algorithms with adaptive step size were proposed and investigated in [16].

Our goal in this paper is to use an adaptive neutrosophic logic parameter ν_k instead of the fixed correction parameter $\eta_k + 1$ in determining appropriate step sizes for various gradient descent methods. The parameter ν_k in each iteration will be determined on the basis of the neutrosophic logic controller (NLC).

Consider the universe $\mathcal{U}$. The fuzzy set theory relies on a membership function $\mathrm{T}(u) \in [0,1], u \in \mathcal{U}$ [17]. In addition, a fuzzy set $\mathcal{N}$ over $\mathcal{U}$ is a set of ordered pairs $\mathcal{N} = \{\langle u, \mathrm{T}(u)\rangle \mid u \in \mathcal{U}\}$.

The intuitionistic fuzzy set (IFS) was established based on the nonmembership function $F(u) \in [0,1], u \in \mathcal{U}$ [18]. Following the philosophy of using two opposing membership functions, an IFS $\mathcal{N}$ in $\mathcal{U}$ is defined as the set of ordered triples

$$\mathcal{N} = \{\langle u, \mathrm{T}(u), \mathrm{F}(u)\rangle \mid u \in \mathcal{U}\},$$

which are based on the independence of the members, that is $\mathrm{T}(u), \mathrm{F}(u) : \mathcal{U} \to [0,1]$ and $0 \le \mathrm{T}(u) + \mathrm{F}(u) \le 1$.

The IFS theory was extended by Smarandache in [19] and Wang et al. [20]. The novelty is the introduction of the indeterminacy-membership function $\mathrm{I}(u)$, which symbolizes hesitation in a decision-making process. As a result, elements of a set in the neutrosophic theory are defined by three individualistic membership functions [19,20] defined by the rules of symmetry: the truth-membership function $\mathrm{T}(u)$, the indeterminacy-membership function $\mathrm{I}(u)$, and the falsity-membership $\mathrm{F}(u)$ function. A single-valued neutrosophic set (SVNS) $\mathcal{N}$ over $\mathcal{U}$ is the set of neutrosophic numbers of the form $\mathcal{N} = \{\langle u, \mathrm{T}(u), \mathrm{I}(u), \mathrm{F}(u)\rangle \mid u \in \mathcal{U}\}$. Values of the membership functions independently take values from $[0,1]$, which initiates $\mathrm{T}(u), \mathrm{I}(u), \mathrm{F}(u) : \mathcal{U} \to [0,1]$ and $0 \le \mathrm{T}(u) + \mathrm{I}(u) + \mathrm{F}(u) \le 3$.

A neutrosophic set is symmetric in nature since the indeterminacy I appears in the middle between the Truth T and False F [21,22]. Furthermore, a refined neutrosophic set with two indeterminacies I_1 and I_2 in the middle between T and F also includes a kind of symmetry [22]. In [23], the authors firstly introduced a normalized and a weighted symmetry measure of simplified neutrosophic sets and then proposed a neutrosophic multiple criteria decision-making method based on the introduced symmetry estimate.

Fuzzy logic (FL), intuitionistic fuzzy logic (IFL), and neutrosophic logic (NL) appear as efficient tools to handle mathematical models with uncertainty, fuzziness, ambiguity, inaccuracy, incomplete certainty, incompleteness, inconsistency, and redundancy. NL can be considered as one of the new theories based on the fundamental principles of neutrosophy, which actually belongs to the group of many-valued logics and actually represents an extension of FL. NL can also be considered as a new branch of logic that deals with the shortcomings of FL and classical logic, as well as IFL. Some of the disadvantages of FL, such as the failure to handle inconsistent information, are significantly reduced by applying NL. Truth and falsity in NL are independent, while in IFL they are dependent. Neutrosophic logic can manipulate both incomplete and inconsistent data. Thus, there is a need to explore the use of NL in various domains from medical treatment to the role of recommendation systems using new advanced computational intelligence techniques. An NL is a better choice than the FL and IFL in the representation of real-world data and their executions, because of the following reasons:

(a) FL and IFL systems neglect the importance of indeterminacy. A fuzzy logic controller (FLC) is based on membership and nonmembership of a particular element to a particular set and take into account the indeterminate nature of generated data.
(b) An FL or IFL system is further constrained by the fact that the sum of membership and nonmembership values is limited to 1. More details are available in [24].
(c) NL reasoning clearly distinguishes concepts of absolute truth and relative truth, assuming the existence of the absolute truth with assigned value 1^+.
(d) NL is applicable in the situation of overlapping regions of the fuzzy systems [25].

Neutrosophic sets (NS) have important applications for denoising, clustering, segmentation, and classification in numerous medical image-processing applications. A utilization of neutrosophic theory in denoising medical images and their segmentation was proposed in [26], such that a neutrosophic image is characterized by three membership sets. Several applications of neutrosophic systems were described in [27]. An application of neutrosophy in natural language processing and sentiment analysis was investigated in [22].

Our goal in the present paper is to improve some of the main gradient descent methods for solving unconstrained nonlinear optimization problems utilizing the advantages of neutrosophic systems. Principal results of the current investigation are emphasized as follows.

(1) We investigate applications of neutrosophic logic in determining an additional step size in line search methods for solving the unconstrained optimization problem.
(2) Applications of neutrosophic logic in multiple step-size methods for solving unconstrained optimization problems are described and investigated.
(3) Rigorous theoretical analysis is performed to show convergence of the proposed iterations under the same conditions as for the corresponding original methods.
(4) Numerical comparison between suggested algorithms given the corresponding available iterations considering the Dolan and Moré benchmarking and the statistical ranking is presented.

The remaining sections are developed according to the following arrangement. Optimization methods based on additional neutrosophic parameters are presented in Section 2. Convergence analysis is investigated in Section 3. Section 4 gives numerical experiments and comparisons. Section 4 gives numerical experiments and compares the MSM, SM, and GD methods with the neutrosophic extensions FMSM, FSM, and FGD methods, equipped with neutrosophic control. Moreover, the application of the new methods in regression analysis is given within this section. Some closing remarks and a vision of future investigation are presented in Section 5.

2. Fuzzy Optimization Methods

Fuzzy descent direction (FDD) iterations are defined as a modification of the DD iterations (2), as follows:

$$x_{k+1} = \Phi(DD)(x_k) = x_k + \nu_k t_k d_k, \tag{12}$$

where $\nu_k > 0$ is an appropriately defined fuzzy parameter. In general, ν_k should satisfy

$$\nu_k \begin{cases} < 1, & \text{if } \Delta_k > 0, \\ = 1, & \text{if } \Delta_k = 0, \\ > 1, & \text{if } \Delta_k < 0. \end{cases} \tag{13}$$

The main idea used in (13) is to decrease the composite step size $\nu_k t_k$ of iterations (12) in the case where f increases and increase $\nu_k t_k$ in the case when f decreases.

We define the general fuzzy QN (FQN) iterative scheme with the line search as

$$x_{k+1} = \Phi(QN)(x_k) = \Phi(x_k - H_k g_k) = x_k - \nu_k H_k g_k, \tag{14}$$

The fuzzy GD method (FGD) is defined by

$$x_{k+1} = \Phi(GD)(x_k) = \Phi(x_k - t_k g_k) = x_k - \nu_k t_k g_k. \tag{15}$$

The fuzzy SM method (FSM) is defined as

$$x_{k+1} = \Phi(SM)(x_k) = x_k - \nu_k t_k (\gamma_k^{FSM})^{-1} g_k, \tag{16}$$

where

$$\gamma_{k+1}^{FSM} = \mho\left(2\gamma_k^{FSM} \frac{\gamma_k^{FSM}\Delta_k + \nu_k t_k \|g_k\|^2}{(\nu_k t_k)^2 \|g_k\|^2}\right). \tag{17}$$

Starting from (9) and (14), we define the fuzzy MSM method ($FMSM$) by

$$x_{k+1} = \Phi(MSM)(x_k) = x_k - \nu_k t_k \tau_k (\gamma_k^{FMSM})^{-1} g_k, \tag{18}$$

where

$$\gamma_{k+1}^{FMSM} = \mho\left(2\gamma_k^{FMSM} \frac{\gamma_k^{FMSM}\Delta_k + \nu_k t_k \tau_k \|g_k\|^2}{(\nu_k t_k \tau_k)^2 \|g_k\|^2}\right). \tag{19}$$

Table 1 summarizes different steps utilized in the iterations utilized in this paper, in which the strike means absence of a suitable parameter.

Table 1. Parameters in gradient descent methods and neutrosophic modifications.

Method	Step Sizes		
	First	Second	Third
GD	t_k	-	-
FGD	ν_k	t_k	-
SM	t_k	$(\gamma_k^{SM})^{-1}$	-
FSM	ν_k	t_k	$(\gamma_k^{SM})^{-1}$
MSM	τ_k	$(\gamma_k^{MSM})^{-1}$	-
$FMSM$	ν_k	τ_k	$(\gamma_k^{MSM})^{-1}$

Algorithm 1, restated from [6,28], is exploited to determine the step length t_k.

Algorithm 1 The backtracking inexact line search.

```
Input: Goal function $f(x)$, a vector $d_k$ at $x_k$ and real quantities $0 < \sigma < 0.5$, $\beta \in (0,1)$.
1: $t = 1$.
2: While $f(x_k + td_k) > f(x_k) + \sigma t g_k^{\mathrm{T}} d_k$, perform $t := t\beta$.
3: Output: $t_k = t$.
```

Algorithm 2 describes the general framework of the FDD class of methods.

Algorithm 2 Framework of *FDD* methods.

```
Input: Objective $f(x)$ and an initial point $x_0 \in \text{dom}(f)$.
1: Put $k = 0$, $\nu_0 = 1$, calculate $f(x_0)$, $g_0 = \nabla f(x_0)$, and generate a descent direction $d_0$.
2: If stopping indicators are fulfilled, then stop; otherwise, go to the subsequent step.
3: (Backtracking) Determine $t_k \in (0,1]$ applying Algorithm 1.
4: Compute $x_{k+1}$ using (12).
5: Compute $f(x_{k+1})$ and generate descent vector $d_{k+1}$.
6: (Score function) Compute $\Delta_k := f_{k+1} - f_k$.
7: (Neutrosophistication) Compute $\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)$ using appropriate membership
   functions.
8: Define neutrosophic inference engine.
9: (De-neutrosophistication) Compute $\nu_k(\Delta_k)$ using de-neutrosophication rule.
10: $k := k + 1$ and go to step 2.
11: Output: $\{x_{k+1}, f(x_{k+1})\}$.
```

It is worth mentioning that the general structure of fuzzy neutrosophic optimization methods follows the philosophy described in the diagram of Figure 1.

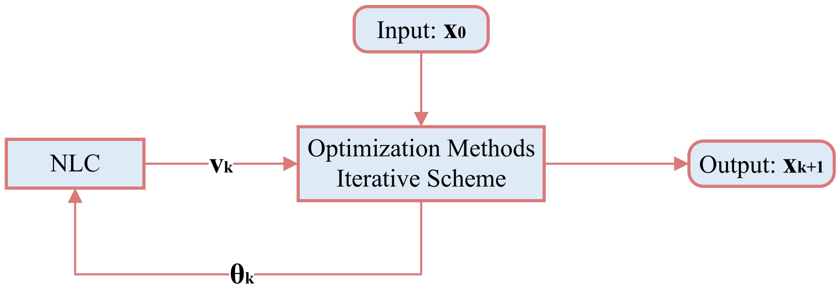

Input: $\mathbf{x_0}$
NLC
$\mathbf{v_k}$
Optimization Methods Iterative Scheme
Output: $\mathbf{x_{k+1}}$
$\boldsymbol{\theta_k}$

Figure 1. The general structure of the fuzzy optimization methods.

FMSM Method

To define the FMSM method, we need to define the steps Score function, neutrosophistication and de-neutrosophistication in Algorithm 2.

(1) *Neutrosophication*. Using three membership functions, neutrosophic logic maps the input $\vartheta := f(x_k) - f(x_{k+1})$ into neutrosophic triplets $(\mathrm{T}(\vartheta), \mathrm{I}(\vartheta), \mathrm{F}(\vartheta))$.
The truth-membership function is defined as the sigmoid function:

$$\mathrm{T}(\vartheta) = 1/(1 + \mathrm{e}^{-c_1(\vartheta - c_2)}). \tag{20}$$

The parameter c_1 is responsible for its slope at the crossover point $\vartheta = c_2$. The falsity-membership function is the sigmoid function:

$$\mathrm{F}(\vartheta) = 1/(1 + \mathrm{e}^{c_1(\vartheta - c_2)}). \tag{21}$$

The indeterminacy-membership function is the Gaussian function:

$$\mathrm{I}(\vartheta) = \mathrm{e}^{-\frac{(\vartheta - c_2)^2}{2c_1^2}}, \tag{22}$$

where the parameter c_1 stands for the standard deviation, and the parameter c_2 is the mean. The neutrosophication of the crisp value $\vartheta \in \mathbb{R}$ used in the implementation is the transformation of ϑ into $\langle \vartheta : \mathrm{T}(\vartheta), \mathrm{I}(\vartheta), \mathrm{F}(\vartheta) \rangle$, where the membership functions are defined in (20)–(22).
Since the final goal is to minimize $f(x)$, it is reasonable to use Δ_k as a measure in the developed NLC. So, we consider the dynamic neutrosophic set (DNS) defined by $\mathfrak{D} := \{\langle \mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k) \rangle; \Delta_k \in \mathbb{R}\}$.

(2) *Neutrosophic inference engine*: The neutrosophic rule between the fuzzy input set $\mathfrak{I}$ and the fuzzy output set under the neutrosophic format $\mathfrak{O} = \{\mathrm{T}, \mathrm{I}, \mathrm{F}\}$ is described by the following "IF–THEN" rules:

$$R_1 : \text{If } \mathfrak{I} = P \text{ then } \mathfrak{O} = \{\mathrm{T}, \mathrm{I}, \mathrm{F}\}$$
$$R_2 : \text{If } \mathfrak{I} = N \text{ then } \mathfrak{O} = \{\mathrm{T}, \mathrm{I}, \mathrm{F}\}.$$

The notations P and N stand for fuzzy sets and exactly indicate a positive and negative error, respectively. Using the unification $R = R_1 \cup R_2$, we obtain $\mathfrak{O}_i = \mathfrak{I} \circ R_i$, $i = 1, 2$, where $\circ$ symbolizes the fuzzy transformation. Furthermore, it follows that $\kappa_{\mathfrak{I}\circ R}(\zeta) = \kappa_{\mathfrak{I}\circ R_1} \bigvee \kappa_{\mathfrak{I}\circ R_2}$, $\kappa_{\mathfrak{I}\circ R}(\zeta) = \sup(\kappa_{\mathfrak{I}} \bigwedge \kappa_{\mathfrak{O}_i})$, and $i = 1, 2$, where $\bigwedge$ (resp. $\bigvee$) denotes the $(\min, \max, \max)$ operator, (resp. $(\max, \min, \min)$ operator). The process of turning the fuzzy outputs into a single, crisp output value is known as defuzzification. There are various defuzzification methods that can be used to perform this procedure. The centroid method, the weighted average method, and the max or mean–max membership principles are some popular defuzzification methods. In this study, the following defuzzification method, called centroid, is employed to obtain a vector of crisp outputs $\zeta^* = [\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)] \in \mathbb{R}^3$ of the fuzzy vector $\zeta = \{\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)\}$:

$$\zeta^* = \frac{\int_{\mathfrak{O}} \zeta\, \kappa_{\mathfrak{I}\circ R}(\zeta) \mathrm{d}\zeta}{\int_{\mathfrak{O}} \kappa_{\mathfrak{I}\circ R}(\zeta) \mathrm{d}\zeta}. \tag{23}$$

(3) *De-neutrosophication*. This step assumes conversion $\langle \mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)\rangle \rightarrow \nu_k(\Delta_k) \in \mathbb{R}$ resulting in a single (crisp) value $\nu_k(\Delta_k)$.
The following *de-neutrosophication rule* is proposed to obtain the parameter $\nu_k(\Delta_k)$ using the rule (24), which follows the constraints stated in (13):

$$\nu_k(\Delta_k) = \begin{cases} 1 - (\mathrm{T}(\Delta_k) + \mathrm{I}(\Delta_k) + \mathrm{F}(\Delta_k))/c_1, & \Delta_k > 0 \\ 1, & \Delta_k = 0 \\ 3 - (\mathrm{T}(\Delta_k) + \mathrm{I}(\Delta_k) + \mathrm{F}(\Delta_k)), & \Delta_k < 0. \end{cases} \tag{24}$$

The parameter $c_1 \geq 3$ maintains the lower limit $\nu_k(\Delta_k) < 1$ of $\nu_k(\Delta_k)$ in the case $\Delta_k > 0$. Moreover, definition (24) assumes that the membership functions must satisfy $\mathrm{T}(\Delta_k) + \mathrm{I}(\Delta_k) + \mathrm{F}(\Delta_k) < 2$ in the case $\Delta_k > 0$.

For better understanding, the NLC structure decomposed by the neutrosophic rules is presented in the diagram of Figure 2. It is crucial to remember that the NLC controller structure was built specifically to solve the issues discussed in this paper, including the membership functions chosen, the number of fuzzy rules chosen, the defuzzification method chosen, and the de-neutrosophication method chosen. As a result, the NLC controller structure is heuristic, and different structures can be required for various applications.

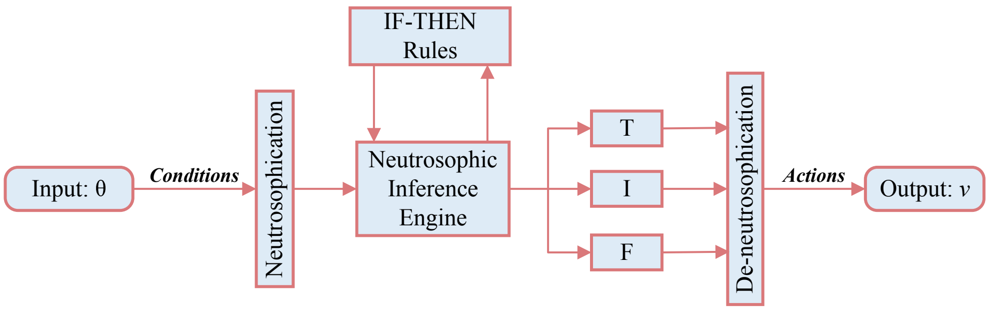


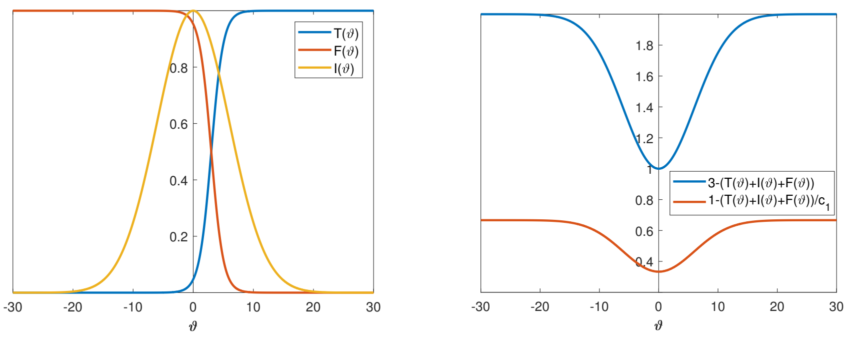
Figure 2. The NLC structure decomposed by the neutrosophic rules.

The utilized settings for the NLC employed in all numerical experiments and graphs of this paper are presented in Table 2.

Table 2. Recommended parameters in NLC.

Set	**Membership Function**		c_1	c_2	**Weight**
Input	Truth	Sigmoid	1	3	1
	Falsity	Sigmoid	1	3	1
	Indeterminacy	Gaussian	6	0	1
Output	(24)		3	-	1

Our imperative requirement is $\nu_k(\Delta_k) \geq 0$. The fulfillment of this requirement immediately follows from the membership values $\mathrm{T}(\Delta_k), \mathrm{F}(\Delta_k), \mathrm{I}(\Delta_k)$ during the neutrosophication process, which are presented in Figure 3a. The NLC output value, $\nu_k(\Delta_k)$, during the de-neutrosophication process is presented in Figure 3b.



(**a**) (**b**)

Figure 3. Neutrosophication (20)–(22) and de-neutrosophication (24) under the parameters in Table 2. (**a**) Neutrosophication. (**b**) De-neutrosophication.

Figure 3 clearly shows that (24) satisfies basic requirements imposed in (13). More precisely, graphs in Figure 3 show $1-(\mathrm{T}(\Delta_k)+\mathrm{I}(\Delta_k)+\mathrm{F}(\Delta_k))/c_1 < 1$ in the case $\Delta_k > 0$, and $3-(\mathrm{T}(\Delta_k)+\mathrm{I}(\Delta_k)+\mathrm{F}(\Delta_k)) \geq 1$ in the case $\Delta_k < 0$.

Remark 1. *During the iterations, the function decreases and tends to the minimum, so* $\lim_{k\to\infty} \Delta_k = 0$, *that is,* $\lim_{k\to\infty} \nu_k(\Delta_k) = 1$. *This observation leads to the conclusion that the parameter* $\nu_k \to 1$ *decreases as we approach the minimum of the function, and thus the influence of neutrosophy on the gradient methods decreases. Such desirable behavior of* $\nu_k(\Delta_k)$ *was our intention.*

Algorithm 3 is the algorithmic framework of the FMSM method.

Algorithm 3 Framework of *FMSM* method.

```
Input: Objective $f(x)$ and appropriate initialization $x_0 \in \mathrm{dom}(f)$.
 1: Put $k = 0$ and compute $f(x_0)$, $g_0 = \nabla f(x_0)$ and take $\gamma_0 = 1$, $\nu_0 = 1$.
 2: If stopping criteria are satisfied, then stop; otherwise, go to the subsequent step.
 3: (Backtracking) Find the step size $t_k \in (0,1]$ using Algorithm 1 utilizing the search
    direction $d_k = -\nu_k \tau_k (\gamma_k^{FMSM})^{-1} g_k$.
 4: Compute $x_{k+1}$ using (18).
 5: Calculate $f(x_{k+1})$ and $g_{k+1} = \nabla f(x_{k+1})$.
 6: Compute $\gamma_{k+1}^{FMSM}$ applying (19).
 7: Compute $\Delta_k := f_{k+1} - f_k$.
 8: Compute $\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)$ using (20)–(22), respectively.
 9: Compute $\zeta^* = [\mathrm{T}(\Delta_k), \mathrm{I}(\Delta_k), \mathrm{F}(\Delta_k)]$ using (23).
10: Compute $\nu_k := \nu_k(\Delta_k)$ using (24).
11: Put $k := k + 1$, and go to Step 2.
12: Return $\{x_{k+1}, f(x_{k+1})\}$.
```

3. Convergence Analysis

The following assumptions are necessary, and the following auxiliary results are useful.

Assumption 1. (1) *The level set $\mathcal{M} = \{x \in \mathbb{R}^n | \ f(x) \leq f(x_0)\}$, defined around the initial iterate x_0 of (2), is bounded.*

(2) *The objective f is continuous and differentiable in a neighborhood $\mathcal{P}$ of $\mathcal{M}$, and its gradient g is Lipschitz continuous, i.e., there exists $L > 0$, which satisfies*

$$\|g(v) - g(w)\| \leq L\|v - w\|, \ \forall \ v, w \in \mathcal{P}. \tag{25}$$

Several useful results from [28–30] and [31,32] are restated for completeness. Let d_k be chosen as a descent direction, and let the gradient $g(x)$ fulfill the Lipschitz requirement (25). The step length t_k derived in the backtracking Algorithm 1 satisfies

$$t_k \geq \min\left\{1, -\frac{\beta(1-\sigma)}{L}\frac{g_k^{\mathrm{T}} d_k}{\|d_k\|^2}\right\}. \tag{26}$$

The notation $f \in \Re^n$ (resp. $f \in \Im^n$) is used to indicate that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and uniformly convex (resp. uniformly convex) on $\mathbb{R}^n$. From [31,32], it follows that Assumption 1 is satisfied if $f \in \Re^n$.

Lemma 1 ([31,32])**.** *Assumption $f \in \Re^n$ implies the existence of real numbers m, M, such that*

$$0 < m \leq 1 \leq M. \tag{27}$$

Moreover, $f(p)$ possesses a unique minimum p^*, such that

$$m\|q\|^2 \leq q^{\mathrm{T}}\nabla^2 f(p)\, q \leq M\|q\|^2, \quad \forall \ p, q \in \mathbb{R}^n; \tag{28}$$

$$\frac{1}{2}m\|p - p^*\|^2 \leq f(p) - f(p^*) \leq \frac{1}{2}M\|p - p^*\|^2, \quad \forall \ p \in \mathbb{R}^n; \tag{29}$$

$$m\|p - q\|^2 \leq (g(p) - g(q))^{\mathrm{T}}(p - q) \leq M\|p - q\|^2, \quad \forall \ p, q \in \mathbb{R}^n. \tag{30}$$

For simplicity, denote the SM and MSM iterations as

$$x_{k+1}^{(M)SM} = x_k^{(M)SM} - t_k\omega_k(\gamma_k^{(M)SM})^{-1}g_k,$$

where $x_k^{(M)SM}$ denotes x_k^{SM} (resp. x_k^{MSM}) in the case of the SM (resp. MSM) method and $\omega_k = 1$ (resp. $\omega_k = \tau_k := 1 + t_k - t_k^2$) in the case of the SM (resp. MSM) method. Similarly, the FSM and $FMSM$ iterations are denoted by the common notation

$$x_{k+1}^{F(M)SM} = x_k^{F(M)SM} - \nu_k t_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k,$$

where $x_k^{F(M)SM}$ denotes x_k^{FSM} (resp. x_k^{FMSM}) in the case of the FSM (resp. $FMSM$) method and $\omega_k = 1$ (resp. $\omega_k = \tau_k$) in the case of the FSM (resp. $FMSM$) method. Since the scalar matrix approximation of the Hessian enables to assume that f is twice continuously differentiable, instead of (28) and (27), we assume only the following bounds for $\gamma_k^{F(M)SM}$:

$$m \leq \gamma_k^{F(M)SM} \leq M, \ \ 0 < m \leq 1 \leq M, \ \ m, M \in \mathbb{R}. \tag{31}$$

In addition, $f \in \Re^n$ reduces to $f \in \Im^n$.

Lemma 2 estimates the iterative decreasing of f ensured by SM and MSM iterations.

Lemma 2 ([6,9])**.** *Let $f \in \Im^n$ and (31) be valid. Then, the SM sequence $\{x_k\}$ produced by (8), and the MSM sequence $\{x_k\}$ produced by (9), satisfy*

$$f(x_k^{(M)SM}) - f(x_{k+1}^{(M)SM}) \geq \mu \|g_k\|^2, \tag{32}$$

such that

$$\mu = \min\left\{\frac{\sigma}{M}, \frac{\sigma(1-\sigma)}{L}\beta\right\}. \tag{33}$$

Theorem 1 investigates the convergence of the $FMSM$ and FSM iterative sequences.

Theorem 1. *Let $f \in \Im^n$ and (31) be valid. Under these conditions, the FSM sequence induced by (16), and the FMSM sequence induced by (18), satisfy*

$$f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) \geq \mu_{\nu_k} \|g_k\|^2, \tag{34}$$

such that

$$\mu_{\nu_k} = \min\left\{\frac{\sigma\nu_k}{M}, \frac{\sigma(1-\sigma)}{L}\beta\right\}. \tag{35}$$

Proof. The FSM and $FMSM$ iterations $x_{k+1}^{F(M)SM} = x_k^{F(M)SM} - t_k\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k$ are of the general DD pattern $x_{k+1} = x_k + t_kd_k$ in the case $d_k = -\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k$. According to the stopping condition used in Algorithm 1, it follows

$$f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) \geq -\sigma t_k g_k^{\mathrm{T}} d_k, \quad \forall\, k \in \mathbb{N}. \tag{36}$$

In the occurrence $t_k < 1$, using (36) with $d_k = -\nu_k\,\omega_k(\gamma_k^{F(M)SM})^{-1}g_k$, one obtains

$$f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) \geq -\sigma t_k g_k^{\mathrm{T}} d_k = -\sigma t_k g_k^{\mathrm{T}}\left(-\nu_k\,\omega_k(\gamma_k^{F(M)SM})^{-1}g_k\right). \tag{37}$$

Now, (26) implies

$$\begin{aligned} t_k \geq -\frac{\beta(1-\sigma)}{L}\cdot\frac{g_k^{\mathrm{T}}d_k}{\|d_k\|^2} &= -\frac{\beta(1-\sigma)}{L}\cdot\frac{g_k^{\mathrm{T}}\left(-\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k\right)}{\left\|-\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k\right\|^2} \\ &= \frac{\beta(1-\sigma)}{L}\cdot\frac{\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}\|g_k\|^2}{\nu_k^2\omega_k^2(\gamma_k^{F(M)SM})^{-2}\|g_k\|^2} \\ &= \frac{\beta(1-\sigma)}{L}\cdot\frac{\gamma_k^{F(M)SM}}{\nu_k\omega_k}. \end{aligned}$$

Now, (37), in conjunction with the last inequality, initiates

$$\begin{aligned} f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) &\geq \sigma t_k\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k^{\mathrm{T}}g_k \\ &\geq \sigma\frac{\beta(1-\sigma)}{L}\cdot\frac{\gamma_k^{F(M)SM}}{\nu_k\omega_k}\nu_k\omega_k(\gamma_k^{F(M)SM})^{-1}g_k^{\mathrm{T}}g_k \\ &\geq \sigma\frac{(1-\sigma)\beta}{L}\|g_k\|^2. \end{aligned}$$

According to (31), in the occurrence $t_k = 1$, we conclude

$$f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) \geq -\sigma g_k^{\mathrm{T}} d_k = -\sigma g_k^{\mathrm{T}}(-\nu_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k)$$
$$= \frac{\sigma \nu_k}{\gamma_k^{F(M)SM}} \|g_k\|^2$$
$$\geq \frac{\sigma \nu_k}{M} \|g_k\|^2.$$

Starting from the above two inequalities, we obtain (34) in both possible situations, $t_k < 1$ and $t_k = 1$, which completes the statement. □

Remark 2. *Based on (32) and (34), respectively, it follows* $f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) \in [\mu_{\nu_k}\|g_k\|^2, +\infty)$ *and* $f(x_k^{(M)SM}) - f(x_{k+1}^{(M)SM}) \in [\mu\|g_k\|^2, +\infty)$. *According to (13), it follows* $\mu_{\nu_k} \geq \mu$ *if* $f(x_{k+1}^{F(M)SM}) < f(x_k^{F(M)SM})$. *So,* $f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM}) \in [\mu_{\nu_k}\|g_k\|^2, +\infty) \subseteq [\mu\|g_k\|^2, +\infty)$. *This means that values* $f(x_k^{F(M)SM}) - f(x_{k+1}^{F(M)SM})$ *belong to the interval with values greater than or equal to the interval which includes values* $f(x_k^{(M)SM}) - f(x_{k+1}^{(M)SM})$. *Furthermore, it means that possibilities for the reduction of* $f(x_{k+1}^{F(M)SM})$ *compared with* $f(x_k^{F(M)SM})$ *are greater than or equal to possibilities for the reduction of* $f(x_{k+1}^{(M)SM})$ *compared with* $f(x_k^{(M)SM})$.

Theorem 2 confirms a linear convergence rate of the $F(M)SM$ method for uniformly convex functions.

Theorem 2. *Let* $f \in \Im^n$ *and (31) be valid. If the iterates* $\{x_k\}$ *are generated by Algorithm 3, it follows that*

$$\lim_{k\to\infty} \|g_k^{F(M)SM}\| = 0, \tag{38}$$

and $\{x_k\}$ *converges to* x^* *with the least linear convergence rate.*

Proof. The proof is analogous to [6] (Theorem 4.1). □

In Lemma 3, we investigate the convergence of the $F(M)SM$ method on the class of quadratic strictly convex functions

$$f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x, \tag{39}$$

wherein A is a real $n \times n$ symmetric positive definite and $b \in \mathbb{R}^n$. Denote by $\lambda_1 \leq \cdots \leq \lambda_n$ the sorted eigenvalues of A. The gradient of (39) is given as

$$g_k = Ax_k - b. \tag{40}$$

Lemma 3. *The eigenvalues of* $f \in \Im^n$ *defined in (39) by a positive definite symmetric matrix* $A \in \mathbb{R}^n$ *satisfy*

$$\lambda_1 \leq \frac{\gamma_{k+1}^{F(M)SM}}{t_{k+1}} \leq \frac{2\lambda_n}{\sigma}, \quad k \in \mathbb{N}, \tag{41}$$

such that $\gamma_k^{F(M)SM}$ *is determined by (17) and (19), and* t_k *is defined in Algorithm 1.*

Proof. Simple calculation leads to

$$f(x_{k+1}^{F(M)SM}) - f(x_k)^{F(M)SM} = \frac{1}{2}x_{k+1}^{\mathrm{T}}Ax_{k+1} - b^{\mathrm{T}}x_{k+1} - \frac{1}{2}x_k^{\mathrm{T}}Ax_k + b^{\mathrm{T}}x_k. \tag{42}$$

The replacement of (18) in (42) leads to

$$\begin{aligned}
f(x_{k+1}^{F(M)SM}) - f(x_k^{F(M)SM}) &= \frac{1}{2}\Big[x_k - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k\Big]^T A[x_k - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k\Big] \\
&\quad - b^{\mathrm{T}}[x_k - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k] - \frac{1}{2} x_k^{\mathrm{T}} A x_k + b^{\mathrm{T}} x_k \\
&= -\frac{1}{2}\nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} x_k^{\mathrm{T}} A g_k - \frac{1}{2}\nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k^{\mathrm{T}} A x_k \\
&\quad + \frac{1}{2}(\nu_k t_k \omega_k)^2 (\gamma_k^{F(M)SM})^{-2} g_k^{\mathrm{T}} A g_k + \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} b^{\mathrm{T}} g_k.
\end{aligned}$$

Applying (40) in the previous equation, we conclude

$$\begin{aligned}
f(x_{k+1}^{F(M)SM}) - f(x_k^{F(M)SM}) &= \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} [b^{\mathrm{T}} g_k - x_k^{\mathrm{T}} A g_k] + \frac{1}{2}(\nu_k t_k \omega_k)^2 (\gamma_k^{F(M)SM})^{-2} g_k^{\mathrm{T}} A g_k \\
&= \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} [b^{\mathrm{T}} - x_k^{\mathrm{T}} A] g_k + \frac{1}{2}(\nu_k t_k \omega_k)^2 (\gamma_k^{F(M)SM})^{-2} g_k^{\mathrm{T}} A g_k \\
&= -\nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k^{\mathrm{T}} g_k + \frac{1}{2}(\nu_k t_k \omega_k)^2 (\gamma_k^{F(M)SM})^{-2} g_k^{\mathrm{T}} A g_k.
\end{aligned} \tag{43}$$

After replacing (43) into (19), the parameter $\gamma_{k+1}^{F(M)SM}$ becomes

$$\begin{aligned}
\gamma_{k+1}^{F(M)SM} &= 2\gamma_k^{F(M)SM} \frac{\gamma_k^{F(M)SM}(f_{k+1} - f_k) + \nu_k t_k \omega_k \|g_k\|^2}{(\nu_k t_k \omega_k)^2 \|g_k\|^2} \\
&= 2\gamma_k^{F(M)SM} \frac{-\nu_k t_k \omega_k \|g_k\|^2 + \frac{1}{2}(\nu_k t_k \omega_k)^2 (\gamma_k^{F(M)SM})^{-1} g_k^{\mathrm{T}} A g_k + \nu_k t_k \omega_k \|g_k\|^2}{(\nu_k (t_k \omega_k))^2 \|g_k\|^2} \\
&= 2\gamma_k^{F(M)SM} \frac{\frac{1}{2}(\nu_k t_k \omega_k)^2 (\gamma_k^{F(M)SM})^{-1} g_k^{\mathrm{T}} A g_k}{(\nu_k t_k \omega_k)^2 \|g_k\|^2} \\
&= \frac{g_k^{\mathrm{T}} A g_k}{\|g_k\|^2}.
\end{aligned}$$

The last identity implies that $\gamma_{k+1}^{F(M)SM}$ is the Rayleigh quotient of the real symmetric matrix A at g_k. So,

$$\lambda_1 \le \gamma_{k+1}^{F(M)SM} \le \lambda_n, \ \ k \in \mathbb{N}. \tag{44}$$

The left inequality in (41) is implied by (44), due to $t_{k+1} \in (0,1]$. To verify the right inequality from (41), we use the upper limit imposed by the line search

$$t_k \ge \frac{\beta(1-\sigma)\gamma_k}{L},$$

which implies

$$\frac{\gamma_{k+1}^{F(M)SM}}{t_{k+1}} < \frac{L}{\beta(1-\sigma)}. \tag{45}$$

Taking into account (40), and the symmetricity of A, we derive

$$\|g(x) - g(y)\| = \|Ax - b - (Ay - b)\| = \|Ax - Ay\| \le \|A\|\|x - y\| = \lambda_n \|x - y\|.$$

Based on the last inequality, it is concluded that the constant L in (45) can be defined as the largest eigenvalue λ_n of A. Considering the backtracking parameters $\sigma \in (0, 0.5)$, $\beta \in (\sigma, 1)$, it is obtained that

$$\frac{\gamma_{k+1}^{F(M)SM}}{t_{k+1}} < \frac{L}{\beta(1-\sigma)} = \frac{\lambda_n}{\beta(1-\sigma)} < \frac{2\lambda_n}{\sigma}. \tag{46}$$

Therefore, the right-hand side inequality in (41) is proved, and the proof is finished. □

In Theorem 3, we consider the convergence of the FSM and $FMSM$ iterations under the supplemental presumption $\lambda_n < 2\lambda_1$.

Theorem 3. *Let f be a strictly convex quadratic in (39). If the eigenvalues of A satisfy $\lambda_n < 2\lambda_1$, FSM iterations (16) and $FMSM$ iterations (18) fulfill*

$$(d_i^{k+1})^2 \leq \delta^2 (d_i^k)^2, \tag{47}$$

wherein

$$\delta = \max\left\{1 - \frac{\sigma\lambda_1}{2\lambda_n}, \frac{\lambda_n}{\lambda_1} - 1\right\}, \tag{48}$$

and

$$\lim_{k\to\infty} \|g_k^{F(M)SM}\| = 0. \tag{49}$$

Proof. Let $\{x_k\}$ be the output of Algorithm 3 and $\{v_1, \dots, v_n\}$ be orthonormal eigenvectors of A. In this case, for a random vector x_k in (40), there exist real constants $d_1^k, d_2^k, \dots, d_n^k$ such that

$$g_k = \sum_{i=1}^{n} d_i^k v_i. \tag{50}$$

Now, using (18), we have

$$\begin{aligned} g_{k+1} &= Ax_{k+1} - b \\ &= A\Big(x_k - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} g_k\Big) - b \\ &= Ax_k - b - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} A g_k \\ &= g_k - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} A g_k \\ &= \Big(I - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} A\Big) g_k. \end{aligned}$$

Next, using the (50), we obtain

$$g_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i = \sum_{i=1}^{n} \Big(1 - \nu_k t_k \omega_k (\gamma_k^{F(M)SM})^{-1} \lambda_i\Big) d_i^k v_i. \tag{51}$$

To prove (47), it is enough to show that $\left|1 - \frac{\lambda_i}{(\nu_k t_k \omega_k)^{-1}\gamma_k^{F(M)SM}}\right| \leq \delta$. Two cases are observable. Firstly, if $\lambda_i \leq \frac{\gamma_k^{F(M)SM}}{\nu_k t_k \omega_k}$ using (41), we deduce

$$1 > \frac{\lambda_i}{\nu_k t_k \omega_k)^{-1}\gamma_k^{F(M)SM}} \geq \frac{\sigma\lambda_1}{2\lambda_n} \Longrightarrow 1 - \frac{\lambda_i}{(\nu_k t_k \omega_k)^{-1}\gamma_k^{F(M)SM}} \leq 1 - \frac{\sigma\lambda_1}{2\lambda_n} \leq \delta. \tag{52}$$

Now, let us examine another case $\frac{\gamma_k^{F(M)SM}}{\nu_k t_k \omega_k} < \lambda_i$. Since

$$1 < \frac{\lambda_i}{(\nu_k t_k \omega_k)^{-1}\gamma_k^{F(M)SM}} \leq \frac{\lambda_n}{\lambda_1}, \tag{53}$$

it follows that

$$\left|1 - \frac{\lambda_i}{(\nu_k t_k \omega_k)^{-1}\gamma_k^{F(M)SM}}\right| \leq \frac{\lambda_n}{\lambda_1} - 1 \leq \delta. \tag{54}$$

Now, we use the orthonormality of the eigenvectors $\{v_1, \ldots, v_n\}$ and (50) and obtain

$$\|g_k\|^2 = \sum_{i=1}^{n} (d_i^k)^2. \tag{55}$$

Since (47) holds and $0 < \delta < 1$, based on (55), it follows that (50) holds, which completes the proof. □

4. Numerical Experiments

In this section, we prove the numerical efficiency of the gradient methods based on a dynamic neutrosophic set (DNS). We consider six methods, of which three are $FMSM$, FSM, and FGD based on DNS, while the other three methods, MSM, SM, and GD, are well-known in the literature. To this aim, we perform competitions on standard test functions with given initial points from [33,34]. We compare the MSM, SM, GD, $FMSM$, FSM, and FGD methods on three criteria:

- The CPU time in seconds—CPUts.
- The number of iterative steps—NI.
- The number of function evaluations—NFE.

The methods which participate in the competition are presented in Section 2 (Table 1). Test problems in ten dimensions [100, 500, 1000, 3000, 5000, 7000, 8000, 10,000, 15,000, 20,000] are considered. The codes are tested in MATLAB R2017a and an LAP (Intel(R) Core(TM) i3-6006U, up to 2.0 GHz, 8 GB Memory) with the Windows 10 Pro operating system.

Algorithms MSM, SM, GD, FSM, FGD, and $FMSM$ are compared using the backtracking line search with parameters $\sigma = 0.0001$, $\beta = 0.8$ and the stopping criterion

$$\|g_k\| \leq \epsilon \quad \text{and} \quad \frac{|\Delta_k|}{1 + |f_k|} \leq \delta,$$

where $\epsilon = 10^{-6}$ and $\delta = 10^{-16}$. Specific parameters used only in the FSM, FGD, and $FMSM$ methods are given in Table 2.

In the following, we give a double analysis of the obtained numerical results. One analysis of the numerical results is based on the Dolan–Moré performance profile, and the other on the ranking of the optimization methods.

4.1. Comparison Based on the Dolan–Moré Performance Profile

In this subsection, we give numerical results for the FSM, FGD, and $FMSM$ methods and then compare them with the numerical results obtained for the MSM, SM, and GD methods.

Summarized numerical results for the competition (between MSM, SM, GD, FSM, FGD, and $FMSM$ methods), obtained by testing 30 test functions (300 tests), are given in Tables 3–5. Tables 3–5 include numerical results obtained by monitoring the criteria NI, NFE, and CPUts.

Table 3. Summary of NI results for MSM, SM, GD, FSM, FGD, and $FMSM$.

Test Function	**No. of Iterations**					
	MSM	**FMSM**	**SM**	**FSM**	**GD**	**FGD**
Extended Penalty Function	651	377	549	372	1255	1250
Perturbed Quadratic function	44,419	75,431	77,458	74,473	372,356	369,992
Raydan 1 function	12,965	12,437	15,913	11,035	58,743	58,594
Raydan 2 function	90	87	90	94	67	129
Diagonal 1 function	52,527	11,571	8955	12,189	41,208	42,290
Diagonal 2 function	26,215	24,866	30,912	29,957	543,249	543,054
Diagonal 3 function	7545	12,586	13,892	13,050	62,128	61,072
Hager function	28,073	800	839	817	3104	2956

Table 3. *Cont.*

Test Function	No. of Iterations					
	MSM	**FMSM**	**SM**	**FSM**	**GD**	**FGD**
Generalized Tridiagonal 1 function	290	440	270	376	656	665
Extended TET function	130	248	130	225	1974	1856
Extended quadratic penalty QP1 function	328	189	246	177	563	549
Extended quadratic penalty QP2 function	1538	2105	3302	3564	134,401	122,926
Quadratic QF2 function	44,911	14,203	83,957	11,488	409,859	411,364
Extended quadratic exponential EP1 function	87	100	64	109	496	528
Extended tridiagonal 2 function	568	421	419	415	1145	1099
Almost perturbed quadratic function	44,029	78,452	80,559	79,793	374,841	375,518
ENGVAL1 function (CUTE)	363	298	302	291	573	557
QUARTC function (CUTE)	185	216	246	211	524,612	524,612
Diagonal 6 function	90	87	90	95	67	129
Generalized quartic function	150	150	157	238	1453	1751
Diagonal 7 function	124	113	90	136	543	570
Diagonal 8 function	100	86	103	89	583	573
Diagonal 9 function	16,920	17,221	11,487	17,752	195,362	195,155
HIMMELH function (CUTE)	100	90	100	90	90	90
Extended Rosenbrock	50	50	50	50	50	50
Extended BD1 function (block diagonal)	189	204	191	223	650	682
NONDQUAR function (CUTE)	42	39	42	35	33	30
DQDRTIC function (CUTE)	827	635	1263	497	15,320	15,398
Extended Beale function	480	980	639	831	12,834	12,826
EDENSCH function (CUTE)	337	314	275	275	663	705

Table 4. Summary of NFE results for MSM, SM, GD, FSM, FGD, and $FMSM$.

Test Function	No. of Funct. Evaluation					
	MSM	**FMSM**	**SM**	**FSM**	**GD**	**FGD**
Extended Penalty Function	3527	2585	2394	2388	47,378	48,057
Perturbed quadratic function	257,063	438,335	439,924	423,195	16,171,466	16,069,927
Raydan 1 function	89,508	69,791	87,508	61,595	1,667,238	1,658,647
Raydan 2 function	190	233	190	235	144	291
Diagonal 1 function	526,958	56,914	47,874	58,155	1,615,828	1,664,760
Diagonal 2 function	158,515	144,005	171,300	166,567	1,086,508	1,086,118
Diagonal 3 function	41,528	71,024	76,336	70,540	2,407,025	2,364,254
Hager function	271,940	3402	3308	3165	56,824	54,818
Generalized tridiagonal 1 function	1012	1587	931	1445	10,867	11,432
Extended TET function	440	681	440	601	19,800	18,859
Extended quadratic penalty QP1 function	1918	1992	2507	1842	10,771	11,268
Extended quadratic penalty QP2 function	10,731	14,285	24,234	26,528	3,875,768	3,545,317
Quadratic QF2 function	245,407	102,882	465,615	80,626	19,072,367	19,141,623
Extended quadratic exponential EP1 function	807	604	587	830	13,643	14,852
Extended tridiagonal 2 function	2550	2123	2285	2111	9570	9464
Almost perturbed quadratic function	259,487	452,388	452,360	445,028	16,285,621	16,309,931
ENGVAL1 function (CUTE)	1974	2700	2098	2315	8787	8593
QUARTC function (CUTE)	420	492	542	472	1,049,274	1,049,304
Diagonal 6 function	229	335	229	263	158	332
Generalized quartic function	409	470	423	781	19,062	25,071
Diagonal 7 function	458	547	293	1094	3348	4286
Diagonal 8 function	326	462	980	612	3921	4078
Diagonal 9 function	141,781	90,948	71,353	89,023	8,449,946	8,455,412
HIMMELH Function (CUTE)	210	190	210	190	190	190
Extended Rosenbrock	110	110	110	110	110	110
Extended BD1 function (Block Diagonal)	558	696	598	691	7660	8452
NONDQUAR function (CUTE)	2084	2085	2057	2060	2500	2501
DQDRTIC function (CUTE)	4090	2805	6518	2542	395,014	400,147
Extended Beale function	2200	4720	3277	3416	207,852	208,551
EDENSCH function (CUTE)	1198	1213	956	872	9403	10,615

Table 5. Summary of CPUts results for MSM, SM, GD, FSM, FGD, and $FMSM$.

Test Function	**CPU Time**					
	MSM	**FMSM**	**SM**	**FSM**	**GD**	**FGD**
Extended penalty function	3.734	1.969	1.969	1.844	17.672	19.078
Perturbed quadratic function	167.063	323.266	298.813	317.250	10,163.688	9771.406
Raydan 1 function	46.813	35.141	50.953	30.234	727.281	667.094
Raydan 2 function	0.453	0.281	0.281	0.344	0.250	0.531
Diagonal 1 function	522.703	86.500	59.297	99.953	1836.766	2091.281
Diagonal 2 function	236.531	228.188	271.094	276.281	2105.219	2158.156
Diagonal 3 function	75.484	172.250	139.859	157.594	3842.625	4025.688
Hager function	384.438	9.594	9.453	9.250	116.922	118.609
Generalized tridiagonal 1 function	2.656	3.188	2.000	3.797	11.641	14.875
Extended TET function	0.953	1.313	0.906	1.359	15.922	16.281
Extended quadratic penalty QP1 function	1.688	1.625	1.875	1.578	4.203	4.391
Extended quadratic penalty QP2 function	5.844	9.891	7.203	10.516	746.328	770.500
Quadratic QF2 function	124.344	47.875	243.688	35.359	7611.656	8436.359
Extended quadratic exponential EP1 function	0.969	0.594	0.469	1.109	5.281	7.297
Extended tridiagonal 2 function	1.906	1.313	1.609	1.266	3.359	3.766
Almost perturbed quadratic function	135.484	314.953	238.625	267.750	9271.016	13,902.047
ENGVAL1 function (CUTE)	2.031	1.797	1.844	1.828	4.125	4.422
QUARTC function (CUTE)	2.813	2.984	3.250	3.219	6253.828	8032.547
Diagonal 6 function	0.328	0.219	0.344	0.484	0.203	0.438
Generalized quartic function	0.344	0.266	0.438	0.625	6.766	11.922
Diagonal 7 function	0.953	0.797	0.531	1.813	3.672	4.406
Diagonal 8 function	0.781	0.922	1.797	1.047	5.578	4.469
Diagonal 9 function	249.875	74.484	53.234	77.219	2478.422	2705.781
HIMMELH function (CUTE)	0.797	0.594	0.781	0.797	0.609	0.641
Extended Rosenbrock	0.203	0.094	0.156	0.203	0.219	0.141
Extended BD1 function (block diagonal)	0.766	0.766	0.859	0.969	4.984	4.469
NONDQUAR function (CUTE)	7.266	8.891	7.797	9.047	9.406	10.406
DQDRTIC function (CUTE)	2.516	1.500	2.906	1.500	118.250	127.844
Extended Beale function	7.219	18.734	9.766	16.016	488.328	546.359
EDENSCH function (CUTE)	6.141	6.422	4.016	5.063	24.672	36.766

The performance profiles given in [35] are applied to compare numerical results for the criteria CPUts, NI, and NFE, generated by considered methods. The method that achieves the best results generates the upper performance profile curve.

In Figure 4 (resp. Figure 5), we compare the performance profiles NI (resp. NFE) for the MSM, SM, GD, FSM, FGD, and $FMSM$ methods based on numerical values included in Table 3 (resp. Table 4). A careful analysis reveals that the FMSM method solves 20.00% of the test problems, with the least NI compared with MSM (33.33%), SM (26.67%), FSM (33.33%), GD (13.33%), and FGD (10.00%). From Figure 4, it is perceptible that the $FMSM$ graph attains the top level first, which indicates that $FMSM$ outperforms other methods with respect to NI.

From Figure 5, we see that the $FMSM$ and FSM methods are more efficient than the MSM, SM, GD, and FGD methods, with respect to NFE, since they solve $FMSM$ (10.00%) and FMS (33.33%) of the test problems with the least NFE compared with MSM (40.00%), SM (26.67%), GD (13.33%), and FGD (6.67%). From Figure 5, it can be observed that the $FMSM$ and FSM graphs first come to the top, so that $FMSM$ and FSM are the winners relative to NFE. On the other hand, the slowest iterations are GD and FGD.

Figure 6 shows the performance profile of the considered methods based on the CPUts for the numerical values included in Table 5. The $FMSM$ method solves 23.33% of the test problems with the least CPUts compared with MSM (30.00%), SM (23.33%), FSM (23.33%), GD (6.67%), and FGD (0%). According to Figure 6, the $FMSM$ and FSM graphs achieve the upper limit level 1 first, which verifies their dominance considering CPUts. Moreover, GD and FGD are the slowest methods.

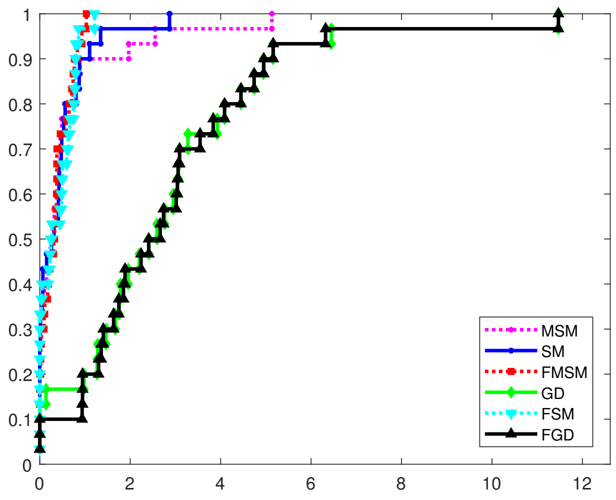


Figure 4. NI performance profiles for the *MSM*, *SM*, *GD*, *FSM*, *FGD*, and *FMSM* methods.

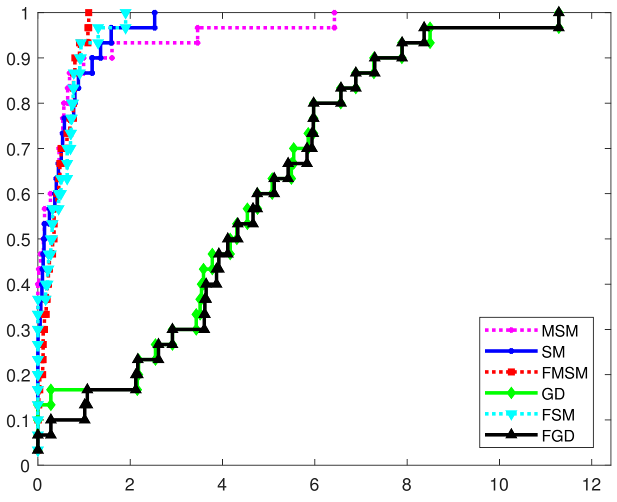


Figure 5. NFE performance profiles for the *MSM*, *SM*, *GD*, *FSM*, *FGD*, and *FMSM* methods.

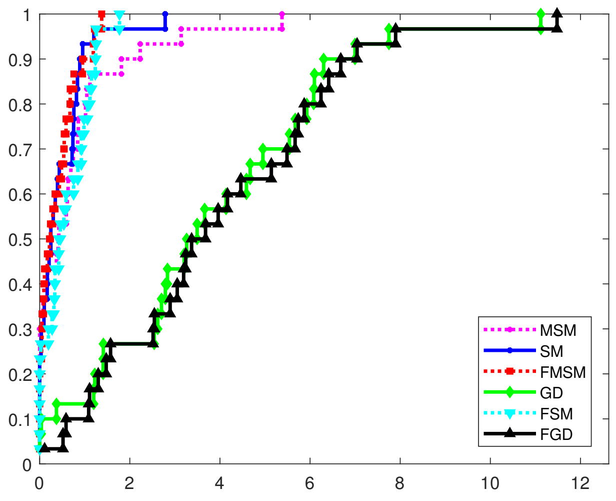


Figure 6. CPUts performance profiles for the *MSM*, *SM*, *GD*, *FSM*, *FGD*, and *FMSM* methods.

Based on the data involved in Tables 3–5 and graphs in Figures 4–6, it is noticed that the *FMSM* and *FSM* methods achieved the best results compared with the *MSM*, *SM*, *GD*, and *FGD* methods, with respect to three basic criteria: NI, NFE, and CPUts.

Table 6 contains the average CPU time, average number of iterations, and the average number of function evaluations for all 300 numerical experiments. Minimal values are marked in bold.

Table 6. Average numerical outcomes for 30 test functions tested on 10 numerical experiments.

Average Performances	MSM	FMSM	SM	FSM	GD	FGD
Average no. of iterations	9477.43	**8493.20**	11,086.33	8631.57	91,962.60	91,565.67
Average no. of funct. evaluation	67,587.60	49,020.13	62,247.90	**48,309.73**	2,416,934.77	2,406,242.00
Average CPU time (s)	66.44	45.21	47.19	**44.51**	1529.30	1783.27

The average results in Table 6 confirm that the average results for *FMSM* and *FSM* are smaller with respect to the corresponding values for *MSM* and *SM* relative to NI, NFE, and CPUts. Such observation leads us to conclude that the use of a dynamic neutrosophic set (DNS) in gradient methods enables an improvement in the numerical results.

4.2. Closer Examination of the Optimization Methods

A closer examination of the optimization methods is presented in this subsection. The optimization methods *GD*, *SM*, *MSM*, *FGD*, *FSM*, and *FMSM* are used to solve two test functions from Tables 3–5 under different initial conditions (ICs). These functions are the Extended Penalty and the Diagonal 6, while the ICs were set to IC1: $1.5 \cdot \mathbf{1}_{100}$, IC2: $-\mathbf{1}_{100}$, and IC3: $4.5 \cdot \mathbf{1}_{100}$ for the former and IC1: $1.5 \cdot \mathbf{1}_{100}$, IC2: $2.5 \cdot \mathbf{1}_{100}$, and IC3: $3.5 \cdot \mathbf{1}_{100}$ for the latter. It is important to note that $\mathbf{1}_{100}$ denotes a vector of ones with dimensions 100×1. The results of the optimization methods are depicted in Figure 7.

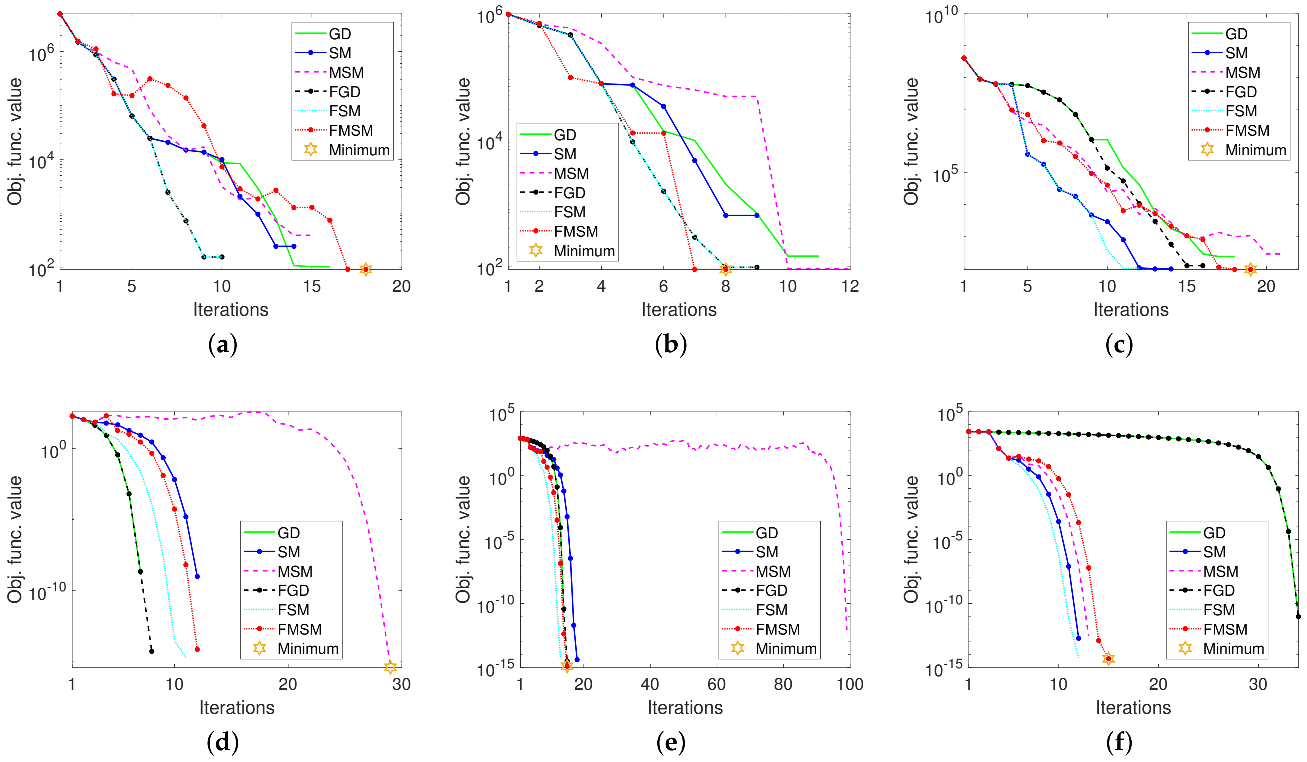


Figure 7. Convergence of the optimization methods under different ICs. (**a**) Extended Penalty function with IC1. (**b**) Extended Penalty function with IC2. (**c**) Extended Penalty function with IC3. (**d**) Diagonal 6 function with IC1. (**e**) Diagonal 6 function with IC2. (**f**) Diagonal 6 function with IC3.

In the case of the Extended Penalty function, Figure 7a–c show, respectively, the convergence of the optimization methods with IC1, IC2 and IC3. Therein, the convergence of FGD and FSM are identical in the cases of IC1 and IC2, whereas the convergence of FGD is slightly faster than GD's, and the convergence of FSM is slightly faster than SM's in the case of IC3. The convergence of $FMSM$ is faster than MSM's in the cases of IC2 and IC3, but it is slower than the convergence of FGD and FSM in the case of IC1. Additionally, $FMSM$ finds the function's minimum point for all ICs with greater accuracy than the other methods.

In the case of the Diagonal 6 function, Figure 7d–f show, respectively, the convergence of the optimization methods with IC1, IC2, and IC3. Therein, the convergence of GD and FGD are identical for all ICs, whereas the convergence of FSM is faster than SM's for all ICs. The convergence of $FMSM$ is faster than MSM's in the cases of IC1 and IC2 and slower in the case of IC3. However, $FMSM$ finds the function's minimum point in the cases of IC2 and IC3 with greater accuracy than the other methods, while MSM finds the function's minimum point in the case of IC1 with greater accuracy than the other methods. Additionally, GD and FGD have the fastest convergence in the case of IC1, while FSM has the fastest convergence in the cases of IC2 and IC3.

In general, all the optimization methods presented here were able to find the minimum of the Extended Penalty and the Diagonal 6 functions. The ICs have a significant impact on the optimization methods' accuracy and speed of convergence. However, FGD, FSM, and $FMSM$ have faster convergence than GD, SM, and MSM, respectively, in most cases.

4.3. Ranking the Optimization Methods

In this subsection, the performances of the optimization methods GD, SM, MSM, FGD, FSM, and $FMSM$ on solving the 30 test functions included in Table 3–5 are ranked from best to worst, i.e., rank 1 to rank 6, respectively.After determining the rank for each test function for each method, it is necessary to calculate the final rank of the methods. The final rank of the methods is based on the average of the ranks obtained for each method in relation to the observed test functions. The method with the lowest average has the highest rank, i.e., rank 1, while the method with the highest average has the lowest rank, i.e., rank 6. We denote by n_m (resp. n_{tf}) the number of methods (resp. the number of test functions). Given a set of methods M and a set of functions F, the rank of the method x on the function y is defined by $r_{x,y}$. In our case, $r_{x,y}$ stands rank method x for the observed test function y and can have rank 1 to rank 6. The average rank of method $x \in M$ is calculated in the following way:

$$AR_x = \frac{\sum_{y \in F} r_{x,y}}{n_{tf}},$$

where AR_x represents the average of all ranks of the observed method x. The final average rank in our case is obtained when all average ranks are ranked from best to worst, i.e., rank 1 to rank 6, respectively.

Figure 8 shows the iterations' performance rank of the optimization methods on 30 functions and their average iterations' rank. Note that a method is regarded as rank 1 if it requires the fewest iterations out of all the considered methods. If a method has the second-fewest iterations compared with all the compared methods, it would be considered rank 2, and so on. Particularly, Figure 8a displays the number of functions in which each method is ranked as rank 1, rank 2, etc., while Figure 8b displays the final rank of the methods based on the average of the results presented in Figure 8a.

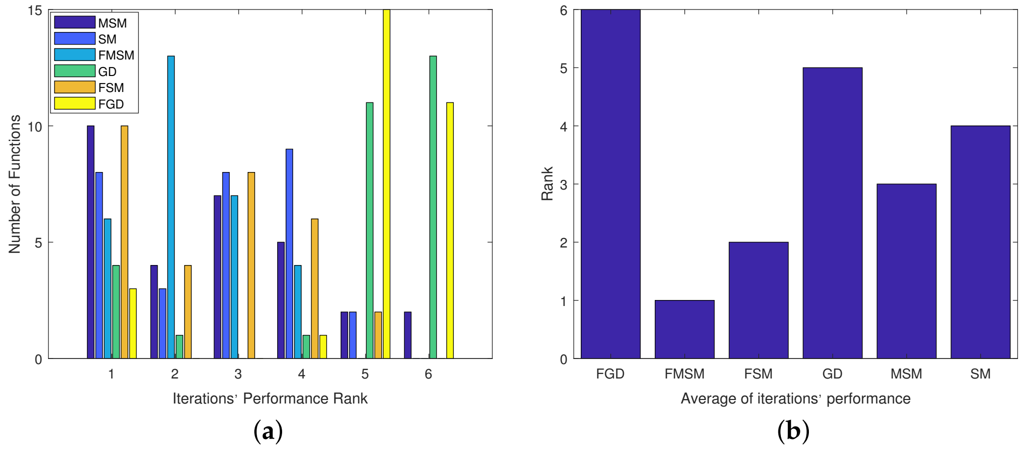


Figure 8. Iterations' performance ranks of the optimization methods on 30 functions and their average rank. (**a**) Iterations' performance. (**b**) Average of iterations' performance.

For example, in Figure 8a, MSM reached rank 1 in the same or a higher number of test functions than FSM and $FMSM$. However, because MSM achieved rank 6 in many more functions than FSM and $FMSM$ in Figure 8b, MSM has an average rank 3, FSM an average rank 2, and $FMSM$ an average rank 1. In other words, $FMSM$ outperforms FSM and MSM in terms of iteration performance. Moreover, the fact that $FMSM$ and FSM iterations outperform their corresponding original methods is another important discovery from Figure 8b.

Figure 9 shows the function evaluations performance ranking on 30 functions and their average rank. Note that a method is regarded as rank 1 if it requires the fewest number of function evaluations out of all the considered methods. If a method has the second-fewest function evaluations compared with all the compared methods, it would be considered rank 2, and so on. Particularly, Figure 9a displays the number of functions in which each method is ranked as rank 1, rank 2, etc., whereas Figure 9b displays the final function evaluation ranks of the methods based on the average of the results presented in Figure 9a.

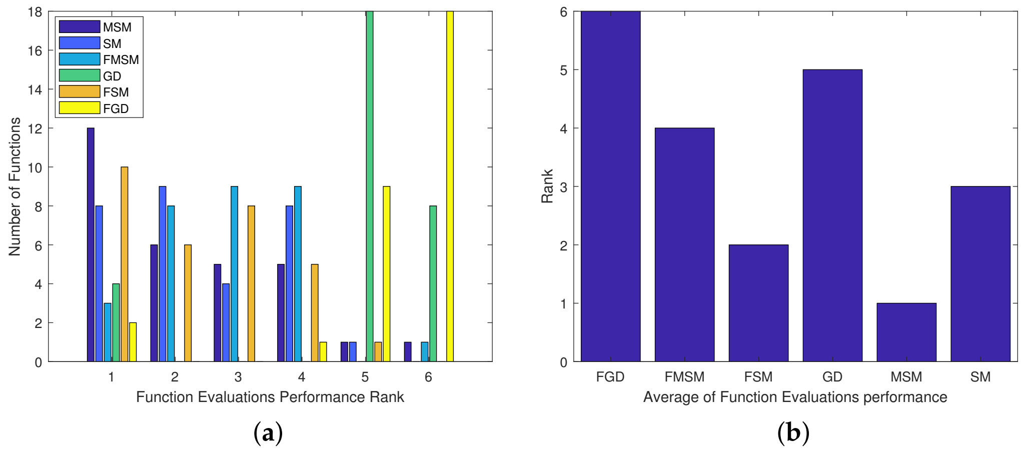


Figure 9. Function evaluation performance ranks of the optimization methods on 30 functions and their average rank. (**a**) Function evaluations performance. (**b**) Average of function evaluation performance.

MSM achieved rank 1 positions in a higher number of functions than all the methods considered in Figure 9a, whereas FGD was considered rank 6 in a higher number of functions than all the methods that were considered. As a result, MSM has the average rank 1, and FGD takes the average rank 6 in Figure 9b. That is, MSM outperforms all the considered methods in terms of function evaluation performance. Moreover, the fact that

FSM, the fuzzy method, outperforms the original SM method is another crucial discovery from Figure 9b.

Figure 10 shows the CPU time consumption performance rank of the optimization methods on 30 functions and their average rank. A method is of rank 1 if it requires the least amount of CPU time compared with all the methods considered. A method achieves rank 2 if it requires the second-least amount of CPU time compared with all the methods, and so on. Particularly, Figure 10a displays the number of functions in which each method is ranked as rank 1, rank 2, etc., whereas Figure 10b displays the final rank of the methods, based on the average of the results presented in Figure 10a.

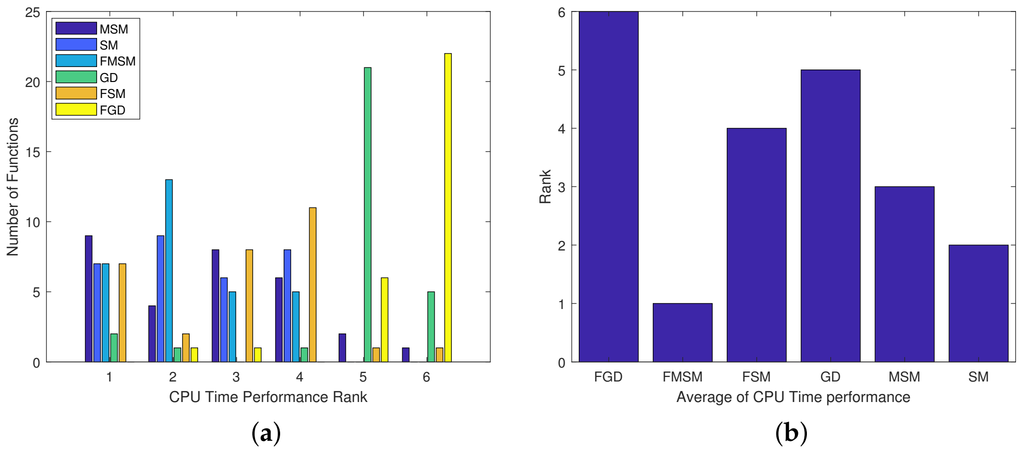


Figure 10. CPU time consumption performance ranks of the optimization methods on 30 functions and their average rank. (**a**) Time consumption's performance. (**b**) Average of time consumption's performance.

MSM is observed as rank 1 in a higher number of functions than all the methods considered in Figure 10a, whereas FGD was considered rank 6 in a higher number of functions than all the compared methods. As a result, MSM has an average rank 3 and FGD an average rank 6 in Figure 10b. If we look at Figure 10b, we can see that $FMSM$ outperforms all the methods considered in terms of CPU time consumption performance.

To summarize, all the fuzzy methods work excellently in finding the minimum of the 30 functions. In general, $FMSM$ has the best iteration performance, MSM has the best function evaluation performance, and $FMSM$ has the best CPU time consumption performance.

We use the notation $\mathcal{M}_i \prec \mathcal{M}_j$ to signify that the method $\mathcal{M}_i$ is ranked better than $\mathcal{M}_j$.

Figure 8b leads to the conclusion $FMSM \prec FSM \prec MSM \prec SM \prec GD \prec FGD$.
Figure 9b leads to the conclusion $MSM \prec FSM \prec SM \prec FMSM \prec GD \prec FGD$.
Figure 10b leads to the conclusion $FMSM \prec SM \prec MSM \prec FSM \prec GD \prec FGD$.

In general, $FMSM$ has the best iteration performance, MSM has the best function evaluation performance, and $FMSM$ has the best CPU time consumption performance. An interesting conclusion is $GD \prec FGD$ in the last positions according to all criteria. A particularly interesting observation is that the proposed fuzzy parameter ν_k improves the SM and MSM methods, but it is not suitable for GD. The logical conclusion is that the fuzzy parameter ν_k is not desirable to use in the role of an isolated parameter, but it is preferable to use it in combination with other scaling parameters.

4.4. Application of the Fuzzy Optimization Methods to Regression Analysis

Regression analysis is an important statistical tool commonly used in the fields of accounting, economics, management, physics, finance, and many more. This tool is used to study the interaction between independent and dependent variables of various data sets. The classical function of regression analysis is defined as

$$y = f(x_1, x_2, \ldots, x_k + \epsilon), \tag{56}$$

where $x_i, i = 1, 2, \ldots, k, k > 0$ are predictor variables, y is the response variable, and ϵ is the error. The linear regression function is obtained by a straight line relationship between y and x

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_kx_k + \epsilon, \tag{57}$$

where $a_0, a_1, \ldots, a_k$ are the parameters of the regression. The main aim of regression analysis is to estimate the parameters $a_0, a_1, \ldots, a_k$ so that the error ϵ is minimized. However, the linear relationship rarely occurs. Thus, a nonlinear regression scheme is frequently used. In this paper, we considered the quadratic regression model. The least squares method is the most popular approach to fitting a regression line and is defined by

$$y = a_0 + a_1x + a_2x^2. \tag{58}$$

The errors for a set of data (x_i, y_i), $i = 1, 2, \ldots, n$ are defined as follows

$$E_i(a) = y_i - (a_0 + a_1x_i + a_2x_i^2), \;\; a = (a_0, a_1, a_2). \tag{59}$$

The main goal would be to fit the "best" line through the data in order to minimize the sum of the residual error squares for all the available data

$$\min_{a\in\mathbb{R}^3} \sum_{i=1}^{n} E_i^2(a), \;\; a = (a_0, a_1, a_2). \tag{60}$$

The data set in Table 7 is a detailed description of people killed in traffic accidents in Serbia from 2012–2021. This set was considered based on the annual reports of the Agency for Traffic Safety of the Republic of Serbia. The ordinal number of the year of data collection is denoted by the x variable and the number of people killed in traffic accidents in Serbia is represented by the y variable. Moreover, only data from 2012–2020 would be considered for the data fitting, while data for 2021 would be reserved for the error analysis.

Table 7. The number of people killed in traffic accidents in Serbia from 2012 to 2021.

Year	Number of Data (x)	The Number of People Killed in Traffic Accidents in Serbia (y)
2012	1	688
2013	2	650
2014	3	536
2015	4	599
2016	5	607
2017	6	579
2018	7	548
2019	8	534
2020	9	492
2021	10	521

The least squares, FMSM, FSM, and FGD methods are used for fitting the regression models to the data collected. The least squares method is frequently used to solve overdetermined linear systems, which usually occurs when the given equations are greater than the number of unknowns [36]. The least squares method includes determining the best approximating line by comparing the total least squares error.

The approximate function for the nonlinear least squares method derived using the data in Table 7 is defined as follows:

$$f(x) = 0.5303030303031x^2 - 24.1030303030320x + 685.1666666666750. \tag{61}$$

For more details on how the approximate function (61) is calculated, see [36]. Let x_i denote the ordinal number of the year and y_i be the number of people killed in traffic accidents in that year. Then, the least squares method (58) is transformed into the following unconstrained minimization problems:

$$\min_{a\in\mathbb{R}^3} f(a) = \min_{a\in\mathbb{R}^3} \sum_{i=1}^{n} E_i^2(a) = \sum_{i=1}^{n} \Big(y_i - (a_0 + a_1 x_i + a_2 x_i^2) \Big)^2, \; a = (a_0, a_1, a_2). \tag{62}$$

where n = 9, i.e., i has values from 1 to 9, corresponding to the years 2012 to 2020. The data from 2012–2020 are utilized to formulate the nonlinear quadratic model for the least square method and the corresponding test function of the unconstrained optimization problem. However, the data for 2021 are excluded from the unconstrained optimization function so that it could be used to compute the relative errors of the predicted data. The relative error is calculated using the following formula to measure the precision of a regression model:

$$Relative\ Error = \frac{|Exact\ value - Approximate\ value|}{|Exact\ value|}. \tag{63}$$

The regression model with the least relative error is considered the best.

The application of the conjugate gradient method in regression analysis to the optimization problems in finding the regression parameters a_0, a_1, …, a_k was considered in [37–40]. To overcome the difficulty of computing the values of a_0, a_1, and a_2 using the matrix inverse, the researchers employed the proposed FMSM, FSM, and FGD methods to solve the test function (62), and the result is presented in Table 8.

Table 8. Test results for optimization of quadratic model for the FMSM, FSM, and FGD methods.

Method	**Initial Point**	**NI**	**NFE**	**CPUts**	**Regression Parameters (a_0, a_1, a_2)**		
					a_0	a_1	a_2
FMSM	(1,1,1)	28,998	119,898	1.484	685.166632504562	−24.1030144870845	0.530301492634611
FSM	(1,1,1)	29,612	120,545	1.609	685.166666629541	−24.1030302889654	0.530303029090458
FGD	(1,1,1)	173,004	7,861,471	35.125	685.161769964723	−24.1009143873562	0.530114238129987
FMSM	(5,5,5)	29,791	126,449	1.750	685.166627004962	−24.102996538241	0.530299060289809
FSM	(5,5,5)	29,504	119,706	1.406	685.166666659503	−24.1030303019929	0.530303030290009
FGD	(5,5,5)	172,876	7,855,584	36.812	685.161745521808	−24.1009038359837	0.530113219772043
FMSM	(−1,−1,−1)	29,259	120,695	1.484	685.166666761033	−24.1030303425383	0.530303033790302
FSM	(−1,−1,−1)	29,513	119,912	1.328	685.166388359794	−24.1029100449169	0.530292483042678
FGD	(−1,−1,−1)	173,698	7,893,030	37.797	685.161987072222	−24.1010082057947	0.530122579942827

The statistics of people killed in traffic accidents in Serbia is estimated using the proposed FMSM, FSM, FGD, least squares, and trend line methods. The trend line is plotted based on the real data obtained from Table 7 using Microsoft Excel and is shown in Figure 11. The equation for the trend line is in the form of a nonlinear quadratic equation

$$y = 0.5303x^2 - 24.103x + 685.17. \tag{64}$$

If we compare the approximation functions (61) and (64), as well as the regression parameters from Table 8 obtained using the FMSM, FSM, and FGD methods, we can see that there are small differences in the values of the parameters a_0, a_1, and a_2.

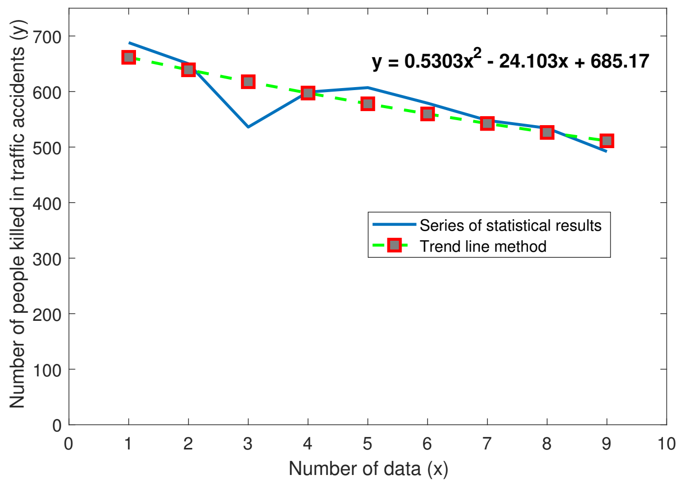


Figure 11. Nonlinear quadratic trend line for people killed in traffic accidents in Serbia.

The functions of the trend line (64) and the least square method (61) are compared with approximation functions from the FMSM, FSM, and FGD methods obtained by substituting the values of the parameters a_0, a_1, and a_2 in (58) for the initial point (1,1,1).

The primary aim of regression analysis is to estimate the parameters $a_0, a_1, \ldots, a_k$ such that the error ϵ is minimized. From Table 9, the proposed FMSM, FSM, and FGD methods have similar relative errors compared with the least square and trend line methods.

Table 9. Estimation point and relative errors for 2021 data.

Method	Estimation Point	Relative Error
FMSM	497.16664	0.045745419
FSM	497.16667	0.045745362
FGD	497.16405	0.045750384
Least Square	497.16667	0.045745361
Trend line	497.17000	0.045738964

Thus, we can conclude that the proposed FMSM, FSM, and FGD methods are applicable to real-life situations.

5. Conclusions

It is known that iterations for solving nonlinear unconstrained minimization are based on the step size defined by the inexact line search. Such step size enables just a sufficient decrease in the value of the objective function. However, after that, there are plenty of possibilities for future adjustments based on the behavior of the objective function. Our goal is to use additional step length parameters to improve convergence. One of these parameters is the $gamma_k$ parameter, which is defined in previous works based on Taylor expansion of the objective function. The second parameter, ν_k, is defined in this paper using neutrosophic logic and the behavior of the objective function in two consecutive iterations. The enhancements of main line search iterations for solving unconstrained optimization are provided based on application of netrosophic logic. Using an appropriate neutrosophic logic, we propose an additional gain parameter ν_k to solve uncertainty in defining parameters of nonlinear optimization methods. The parameter arises as the output from an appropriately defined neutrosophic logic system, and it is usable in various gradient descent methods as a corrective step size.

Performed theoretical analysis reveals convergence of novel iterations under the same conditions as for corresponding original methods. Numerical comparison and statistical ranking point out better results generated by the proposed enhanced methods compared with some existing methods. Moreover, statistical measures reveal advantages of fuzzy and neutrosophic improvements compared with original line search optimization methods.

Precisely, our numerical experience shows that the neutrosophic parameter ν_k is particularly efficient as an additional step size composed with previously defined parameters. Direct application of ν_k is not so effective.

Additional research includes several new directions. First of all, other strategies in neutrosophication and de-neutrosophication are possible, as well as other frameworks parallel to neutrosophic sets, known as picture fuzzy sets and spherical fuzzy sets, discussed in the following articles [41,42]. These can be discussed in future research.

Empirical evaluation shows high sensitivity of the results on the choice of the parameters that define the truth, falsity, and indeterminacy membership functions. Such experience confirms the assumption that a different configuration of parameters, as well as improvements in the neutrosophic logic engine, can lead to further improvements of defined methods. The possibility to define if–then rules in a more sophisticated way based on the history of the obtained values of $f(x)$ remains an open topic for future research. Another topic of future study is the investigation of a neutrosophic approach to enhance stochastic optimization methods. In addition, positive definite matrices B_k are usable as more precise approximations of the Hessian compared with simplest diagonal approximations. Finally, continuous-time nonlinear optimization assumes time-varying scaling parameters inside a selected time interval.

Author Contributions: Conceptualization, P.S.S. and V.N.K.; methodology, P.S.S., V.N.K., and L.A.K.; software, B.I. and S.D.M.; validation,V.N.K., P.S.S., D.S., and L.A.K.; formal analysis, P.S.S., S.D.M. and D.S.; investigation, P.S.S., S.D.M., V.N.K., and L.A.K. ; resources, B.I. and S.D.M.; data curation, B.I. and S.D.M.; writing—original draft preparation, P.S.S., D.S., and S.A.E.; writing—review and editing, P.S.S., S.D.M., and S.A.E.; visualization, B.I. and S.D.M. 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).

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

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

References

1. Sun, W.; Yuan, Y.-X. *Optimization Theory and Methods: Nonlinear Programming*; Springer: Berlin/Heidelberg, Germany, 2006.
2. Brezinski, C. A classification of quasi-Newton methods. *Numer. Algorithms* **2003**, *33*, 123–135. [CrossRef]
3. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: New York, NY, USA, 1999.
4. Petrović, M.J.; Stanimirović, P.S. Accelerated Double Direction method for solving unconstrained optimization problems. *Math. Probl. Eng.* **2014**, *2014*, 965104. [CrossRef]
5. Petrović, M.J.; Rakocević, V.; Kontrec, N.; Panić, S.; Ilić, D. Hybridization of accelerated gradient descent method. *Numer. Algorithms* **2018**, *79*, 769–786. [CrossRef]
6. Stanimirović, P.S.; Miladinović, M.B. Accelerated gradient descent methods with line search. *Numer. Algorithms* **2010**, *54*, 503–520. [CrossRef]
7. Stanimirović, P.S.; Milovanović, G.V.; Petrović, M.J. A transformation of accelerated double step size method for unconstrained optimization. *Math. Probl. Eng.* **2015**, *2015*, 283679. [CrossRef]
8. Petrović, M.J. An accelerated Double Step Size method in unconstrained optimization. *Applied Math. Comput.* **2015**, *250*, 309–319.
9. Ivanov, B.; Stanimirović, P.S.; Milovanović, G.V.; Djordjević, S.; Brajević, I. Accelerated multiple step-size methods for solving unconstrained optimization problems. *Optim. Methods Softw.* **2021**, *36*, 998–1029 [CrossRef]
10. Petrović, M.J.; Stanimirović, P.S.; Kontrec, N.; Mladenović, J. Hybrid modification of Accelerated Double Direction method. *Math. Probl. Eng.* **2018**, *2018*, 1523267. [CrossRef]
11. Picard, E. Memoire sur la theorie des equations aux derivees partielles et la methode des approximations successives. *J. Math. Pures Appl.* **1890**, *6*, 145–210.
12. Ishikawa, S. Fixed points by a new iteration method. *Proc. Am. Math. Soc.* **1974**, *44*, 147–150. [CrossRef]
13. Khan, S.H. A Picard-Mann hybrid iterative process. *Fixed Point Theory Appl.* **2013**, *2013*, 69. [CrossRef]

14. Rakočević, V.; Petrović, M.J. Comparative analysis of accelerated models for solving unconstrained optimization problems with application of Khan's hybrid rule. *Mathematics* **2022**, *10*, 4411. [CrossRef]
15. Humaira, M.S.; Tunç, C. Fuzzy fixed point results via rational type contractions involving control functions in complex–valued metric spaces. *Appl. Math. Inf. Sci.* **2018**, *12*, 861–875. [CrossRef]
16. Vrahatis, M.N.; Androulakis, G.S.; Lambrinos, J.N.; Magoulas, G.D. A class of gradient unconstrained minimization algorithms with adaptive step-size. *J. Comp. Appl. Math.* **2000**, *114*, 367–386. [CrossRef]
17. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
18. Atanassov, K.T. Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **1986**, *20*, 87–96. [CrossRef]
19. Smarandache, F. *A Unifying Field in Logics, Neutrosophy: Neutrosophic Probability, Set and Logic*; American Research Press: Rehoboth, NM, USA, 1999.
20. Wang, H.; Smarandache, F.; Zhang, Y.Q.; Sunderraman, R. Single valued neutrosophic sets. *Multispace Multistruct.* **2010**, *4*, 410–413.
21. Khalil, A.M.; Cao, D.; Azzam, A.; Smarandache, F.; Alharbi, W.R. Combination of the single-valued neutrosophic fuzzy set and the soft set with applications in decision-making. *Symmetry* **2020**, *12*, 1361. [CrossRef]
22. Mishra, K.; Kandasamy, I.; Kandasamy W.B., V.; Smarandache, F. A novel framework using neutrosophy for integrated speech and text sentiment analysis. *Symmetry* **2020**, *12*, 1715. [CrossRef]
23. Tu, A.; Ye, J.; Wang, B. Symmetry measures of simplified neutrosophic sets for multiple attribute decision-making problems. *Symmetry* **2018**, *10*, 144. [CrossRef]
24. Smarandache, F. Neutrosophic Logic—A Generalization of the Intuitionistic Fuzzy Logic. 25 January 2016. Available online: https://ssrn.com/abstract=2721587 (accessed on 1 September 2021).
25. Ansari, A.Q. From fuzzy logic to neutrosophic logic: A paradigme shift and logics. In Proceedings of the 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 22–23 December 2017; pp. 11–15.
26. Guo, Y.; Cheng, H.D.; Zhang, Y. A new neutrosophic approach to image denoising. *New Math. Nat. Comput.* **2009**, *5*, 653–662. [CrossRef]
27. Christianto, V.; Smarandache, F. A Review of Seven Applications of Neutrosophic Logic: In Cultural Psychology, Economics Theorizing, Conflict Resolution, Philosophy of Science, etc. *Multidiscip. Sci. J.* **2019**, *2*, 128–137. [CrossRef]
28. Andrei, N. An acceleration of gradient descent algorithm with backtracking for unconstrained optimization. *Numer. Algorithms* **2006**, *42*, 63–73. [CrossRef]
29. Andrei, N. Relaxed Gradient Descent and a New Gradient Descent Methods for Unconstrained Optimization. Visited 29 November 2022. Available online:https://camo.ici.ro/neculai/newgrad.pdf (accessed on 1 September 2021).
30. Shi, Z.-J. Convergence of line search methods for unconstrained optimization. *App. Math. Comput.* **2004**, *157*, 393–405. [CrossRef]
31. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equation in Several Variables*; Academic Press: New York, NY, USA; London, UK, 1970.
32. Rockafellar, R.T. *Convex Analysis*; Princeton University Press: Princeton, NJ, USA, 1970.
33. Andrei, N. An unconstrained optimization test functions collection. *Adv. Model. Optim.* **2008**, *10*, 147–161.
34. 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]
35. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, **91**, 201–213. [CrossRef]
36. Dawahdeh, M.; Mamat, M.; Rivaie, M.; Sulaiman, I.M. Application of conjugate gradient method for solution of regression models. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 1754–1763.
37. Moyi, A.U.; Leong, W.J.; Saidu, I. On the application of three-term conjugate gradient method in regression analysis. *Int. J. Comput. Appl.* **2014**, *102*, 1–4.
38. Sulaiman, I.M.; Bakar, N.A.; Mamat, M.; Hassan, B.A.; Malik, M.; Ahmed, A.M. A new hybrid conjugate gradient algorithm for optimization models and its application to regression analysis. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *23*, 1100–1109. [CrossRef]
39. 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] [PubMed]
40. Sulaiman, I.M.; Mamat, M. A new conjugate gradient method with descent properties and its application to regression analysis. *J. Numer. Anal. Ind. Appl. Math.* **2020**, *14*, 25–39.
41. Mahmood, T.; Ullah, K.; Khan, Q.; Jan, N. An approach toward decision–making and medical diagnosis problems using the concept of spherical fuzzy sets. *Neural Comput. Appl.* **2019**, *31*, 7041–7053. [CrossRef]
42. Ullah, K. Picture fuzzy maclaurin symmetric mean operators and their applications in solving multiattribute decision–making problems. *Math. Probl. Eng.* **2021**, *2021*, 1098631. [CrossRef]