NSS
Neutrosophic Sets and Systems, Vol. 100, 2026
University of New Mexico

Neutrosophic-Based Optimization Algorithm: A Novel Metaheuristic Algorithm for Global Optimization

Sara Hasan[1] and Reda Ahmed[2]

[1,2] Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt.

*Correspondence: mohreda162018@gmail.com

Abstract

This study presents a novel neutrosophic optimizer, called NOpt, for continuous global optimization. In the proposed algorithm, each candidate solution is represented by a neutrosophic search state comprising truth, indeterminacy, and falsity degrees. The truth component promotes exploitation around promising regions, the indeterminacy component enhances exploration of uncertain regions, and the falsity component facilitates escape from poor or stagnant regions. The refined NOpt further incorporates two mechanisms: a Neutrosophic Elite Refinement Operator, which improves each generated trial vector before its evaluation, and a Neutrosophic Adaptive Local Search Operator, which utilizes a controlled portion of the remaining function-evaluation budget to refine elite solutions. The proposed optimizer is empirically compared with seven established metaheuristic algorithms under a comprehensive experimental protocol. The results demonstrate that NOpt achieves outstanding performance on the fixed-dimensional test functions (TFs), which simulate real-world optimization problems.

Keywords: Neutrosophic set; Optimization problem; Metaheuristic algorithms; Global optimization; Neutrosophic optimizer.

1. Introduction

Global optimization is a fundamental task in many scientific, engineering, and computational applications. The objective is to find the best solution within a search space that may be nonlinear, multimodal, discontinuous, noisy, high-dimensional, or difficult to differentiate [1, 2]. In such cases, exact mathematical optimization methods may become computationally expensive or unsuitable. Metaheuristic algorithms are therefore widely used because they can search complex spaces and provide high-quality approximate solutions within a reasonable computational budget [3]. A central challenge in metaheuristic optimization is the balance between exploration and exploitation [4]. Exploration enables the algorithm to investigate new and diverse regions of the search space, while exploitation refines the search around promising solutions. If exploration dominates for too long, the algorithm may waste function evaluations without sufficient improvement. If exploitation dominates too early, the algorithm may converge prematurely to a local optimum. In addition to these two behaviors, a strong optimizer also requires an effective escape mechanism to move away from poor, stagnant, or misleading regions.

Most existing metaheuristic algorithms address this balance through adaptive parameters, elite guidance, mutation strategies, diversity control, or random perturbation mechanisms [5-7]. However, the search condition of a candidate solution is usually more complex than its fitness value alone. A solution may have good fitness but may be located in an overcrowded or overexploited region. Another solution may have moderate fitness but may lie in a promising uncertain region. A weak or stagnant solution may need a stronger escape behavior instead of ordinary movement. Therefore, a more expressive state representation is needed to describe the search role of each candidate solution during the optimization process.

This paper proposes NOpt, a Truth–Indeterminacy–Falsity-guided Neutrosophic Optimizer for continuous global optimization. The main idea is to represent each candidate solution not only by its position and objective value, but also by a neutrosophic search state [7]:

$$NS_i^t = \langle T_i^t, I_i^t, F_i^t \rangle,$$

where T_i^t denotes the truth degree of the candidate solution X_i at iteration t, I_i^t denotes its indeterminacy degree, and F_i^t denotes its falsity degree. In the proposed optimizer, these three components are not used as descriptive labels after evaluation. They are active search-control components that directly guide the movement of each candidate solution. The truth component controls exploitation around promising regions. The indeterminacy component increases exploration when the search state is unclear, unstable, or insufficiently explored. The falsity component activates escape behavior when a solution is weak, stagnant, or located in an unproductive region. Accordingly, NOpt introduces a unified neutrosophic movement equation in which exploitation, exploration, and escape are controlled adaptively through the three neutrosophic components. The proposed optimizer is designed to be mathematically transparent and directly implementable. Each movement component has a clear role: truth moves the candidate solution toward promising solutions, indeterminacy introduces controlled exploratory variation, and falsity pushes the search away from poor regions. This structure makes the algorithm easier to understand, analyze, and implement compared with optimization methods whose behavior depends on several disconnected operators.

The aim of this work is not to claim universal superiority over all optimization algorithms or all problem classes. Instead, the objective is to introduce a genuinely neutrosophic optimization mechanism and evaluate its ability to improve the balance between exploration, exploitation, and escape in continuous global optimization. The experimental part of this paper evaluates NOpt against seven well-known metaheuristic algorithms: Weighted average algorithm (WAA), Runge-Kutta method (RUN) beyond the metaphor, Rime optimization algorithm (RIME), whale optimization algorithm (WOA), sine-cosine algorithm (SCA), and GOOSE algorithm. The main contributions of this paper are summarized as follows.

- First, a new neutrosophic optimizer, called NOpt, is proposed for continuous global optimization.
- Second, each candidate solution is represented by a neutrosophic search state composed of truth, indeterminacy, and falsity degrees.
- Third, a unified movement equation is developed in which truth guides exploitation, indeterminacy guides exploration, and falsity guides escape from poor or stagnant regions.
- Fourth, the algorithm is formulated in a transparent way that allows direct implementation and analysis.
- Fifth, the proposed optimizer is empirically compared with seven established metaheuristic algorithms using a complete experimental protocol.

The remainder of this paper is organized as follows. Section 2 presents the mathematical preliminaries of global optimization and neutrosophic search states. Section 3 introduces the proposed NOpt algorithm and its main operators. Section 4 presents results and discussion. Section 5 concludes the paper and outlines future work.

2. Mathematical Preliminaries

This section defines the mathematical basis used to construct the proposed NOpt algorithm. The definitions include the continuous global optimization problem, the population structure, the normalized search-quality measure, the diversity indicator, the elite-closeness indicator, the stagnation indicator, and the neutrosophic search state.

2.1. Continuous Global Optimization Problem

Let the optimization problem be a continuous minimization problem defined as [8]:

$$\min f\,(X),\ X = (x_1, x_2, \ldots, x_D) \in \Omega, \tag{1}$$

where $f(X)$ denotes the objective function, X denotes a candidate solution, D denotes the number of decision variables, and Ω is the feasible search space. The feasible search space is bounded by lower and upper limits, as defined below:

$$\Omega = \{X, \in, \mathbb{R}^D, :, L_j \le x_j \le U_j\ j = 12 \ldots D\}, \tag{2}$$

where L_j and U_j are the lower and upper bounds of the j-th decision variable, respectively. Although this paper focuses on minimization problems, a maximization problem can be handled by minimizing $-f(X)$ without changing the structure of the proposed optimizer. At iteration t, the population contains N candidate solutions represented as follows:

$$P^t = \{X_1^t, X_2^t, \ldots, X_N^t\}.$$

The fitness value of the i-th candidate solution is computed as follows:

$$f_i^t = f(X_i^t).$$

For a minimization problem, the best and worst fitness values at iteration t are defined as:

$$f_{\min}^{t} = \min_{1 \le i \le N} f_i^t,\ f_{\max}^{t} = \max_{1 \le i \le N} f_i^t. \tag{5}$$

The current best solution is denoted by:

$$X_*^t = \arg\min_{X_i^t \in P^t} f(X_i^t),$$

and the current worst solution is denoted by:

$$X_w^t = \arg\max_{X_i^t \in P^t} f(X_i^t). \tag{7}$$

If more than one solution has the same best or worst fitness value, any one of them may be selected without affecting the general structure of the algorithm.

2.2. Normalized Search Quality

Raw fitness values may have different scales from one optimization problem to another. Therefore, NOpt uses a normalized search-quality value to describe how good each candidate solution is relative to the current population. For a minimization problem, the normalized search quality of X_i^t is defined as:

$$q_i^t = \begin{cases} \dfrac{f_{\max}^t - f_i^t}{f_{\max}^t - f_{\min}^t}, & f_{\max}^t > f_{\min}^t, \\ 0.5, & f_{\max}^t = f_{\min}^t. \end{cases} \tag{8}$$

The value q_i^t lies in $[0, 1]$. A larger value indicates a better solution relative to the current population. When $f_{\max}^t > f_{\min}^t$, the current best solution has $q_i^t = 1$, while the current worst solution has $q_i^t = 0$. When all solutions have the same fitness value, $q_i^t = 0.5$ is used as a neutral value.

2.3. Population Diversity Indicator

Fitness alone is not sufficient to describe the search condition of a candidate solution. A solution may have good fitness but may be located in an overcrowded region. Another solution may have moderate fitness but may contribute useful diversity. Therefore, NOpt uses a population-diversity indicator. The population center at iteration t is defined as:

$$\bar{X}^t = \frac{1}{N}\sum_{i=1}^{N} X_i^t. \tag{9}$$

The Euclidean distance of X_i^t from the population center is:

$$d_i^t = \|X_i^t - \bar{X}^t\|_2. \tag{10}$$

The normalized diversity indicator is defined as:

$$\delta_i^t = \begin{cases} \dfrac{d_i^t}{\max_{1\le r\le N} d_r^t}, & \max_{1\le r\le N} d_r^t > 0, \\ 0, & \max_{1\le r\le N} d_r^t = 0. \end{cases} \tag{11}$$

The value δ_i^t lies in $[0,1]$. A high value means that the candidate solution is far from the population center and may help preserve search diversity. A low value means that the solution is close to the central region of the population.

2.4. Elite-Closeness Indicator

To support exploitation, NOpt measures how close each candidate solution is to the current best solution. This measure is called the elite-closeness indicator. The average distance from the population to the current best solution is:

$$\bar{d}_*^t = \frac{1}{N}\sum_{i=1}^{N} \|X_i^t - X_*^t\|_2 . \tag{12}$$

The elite-closeness indicator is then defined as:

$$e_i^t = \exp\left(-\frac{\|X_i^t - X_*^t\|_2}{\bar{d}_*^t + \varepsilon}\right), \tag{13}$$

where ε is a very small positive constant used to avoid division by zero. The value e_i^t lies in $(0,1)$. A larger value means that the candidate solution is closer to the current best solution, while a smaller value means that it is farther away from the best solution.

2.5. Stagnation Indicator

A candidate solution may fail to improve for several consecutive iterations. This condition is called stagnation. In NOpt, stagnation is important because a stagnant solution may require stronger exploration or escape behavior. Let a_i^t be the number of consecutive iterations in which solution X_i has not improved up to iteration t. The normalized stagnation indicator is defined as:

$$s_i^t = \min\left(1, \frac{a_i^t}{a_{\max}}\right), \tag{14}$$

where $a_{\max}$ is a predefined stagnation limit. The value s_i^t lies in $[0,1]$. A value close to zero indicates recent improvement, while a value close to one indicates strong stagnation.

2.6. Neutrosophic Search State

The central idea of NOpt is to describe each candidate solution by a neutrosophic search state. This state contains three components [9-11]:

$$NS_i^t = \langle T_i^t, I_i^t, F_i^t \rangle. \tag{15}$$

Here, T_i^t is the truth degree of solution X_i^t. It represents the tendency of the solution to participate in exploitation. I_i^t is the indeterminacy degree of solution X_i^t. It represents the need for exploration caused by uncertainty, diversity, or unstable search behavior. F_i^t is the falsity degree of solution X_i^t. It represents the tendency of the solution to activate escape behavior from weak or stagnant regions. The three components are bounded as follows:

$$0 \leq T_i^t \leq 1,\ 0 \leq I_i^t \leq 1,\ 0 \leq F_i^t \leq 1. \tag{16}$$

The components T_i^t, I_i^t, and F_i^t are not required to sum to one. This is important because a candidate solution may be promising, uncertain, and partially weak at the same time. Therefore, the neutrosophic search state provides a richer description of the search condition than a single fitness-based value.

2.7. Role of the Preliminaries in NOpt

The quantities defined in this section form the mathematical input for the proposed optimizer. The normalized quality q_i^t, diversity indicator δ_i^t, elite-closeness indicator e_i^t, and stagnation indicator s_i^t are used to construct the neutrosophic components T_i^t, I_i^t, and F_i^t. These neutrosophic components control the three main movement behaviors of NOpt: exploitation, exploration, and escape. Therefore, this section provides the mathematical foundation for the full neutrosophic search mechanism introduced in the next section.

3. Proposed NOpt Algorithm

This section presents the final structure of the proposed NOpt algorithm. NOpt is a neutrosophic population-based optimizer in which each candidate solution is guided by three active search components: truth, indeterminacy, and falsity. The truth component strengthens exploitation around promising regions, the indeterminacy component strengthens exploration in uncertain regions, and the falsity component activates escape from poor or stagnant regions. The refined NOpt also includes two additional mechanisms: a Neutrosophic Elite Refinement Operator, which improves the generated trial vector before evaluation, and a Neutrosophic Adaptive Local Search Operator, which uses a controlled part of the remaining function-evaluation budget to refine elite solutions.

3.1. Initialization and Evaluation Budget

The initial population is generated randomly within the feasible search space. For each candidate solution X_i^0, the j-th decision variable is initialized as follows:

$$x_{ij}^0 = L_j + r_{ij}^0\left(U_j - L_j\right),\ i = 1,2,\dots,N,\ j = 1,2,\dots,D, \tag{17}$$

where r_{ij}^0 is a random value in the interval $(0,1)$, L_j is the lower bound, and U_j is the upper bound of the j-th decision variable. After initialization, the objective value of each candidate solution is evaluated as follows:

$$f_i^0 = f(X_i^0). \tag{18}$$

The initial best solution X_*^0, the initial worst solution X_w^0, and the stagnation counters are then initialized. At the beginning of the algorithm, the stagnation counter for each candidate solution is initialized as follows:

$$a_i^0 = 0,\ i = 1,2,\dots,N. \tag{19}$$

The initial global-best solution and its objective value are stored as:

$$X_{best}^{0} = X_{*}^{0},\ f_{best}^{0} = f(X_{*}^{0}). \tag{20}$$

The termination condition of the algorithm is controlled by a maximum number of function evaluations $FEs_{\max}$.

3.2. Search Progress Factors

At iteration t, let FEs^{t} denote the number of objective-function evaluations consumed before generating new trial solutions. The normalized search progress factor is defined as:

$$\tau_t = \min\left(1, \frac{FEs^{t}}{FEs_{\max}}\right). \tag{21}$$

The decreasing exploration factor is:

$$\chi_t = 1 - \tau_t. \tag{22}$$

The value of τ_t increases as the available evaluation budget is consumed, while χ_t decreases during the search. These two factors are used to reduce global exploration gradually and strengthen local refinement in later stages.

3.3. Neutrosophic Search State Construction

At each iteration t, each candidate solution X_i^t is represented by a neutrosophic search state:

$$NS_i^t = \langle T_i^t, I_i^t, F_i^t \rangle. \tag{23}$$

The three components are constructed from the normalized search quality q_i^t, the diversity indicator δ_i^t, the elite-closeness indicator e_i^t, and the stagnation indicator s_i^t, which were defined in Section 2. The uncertainty-quality term is defined as:

$$u_i^t = 4q_i^t(1 - q_i^t). \tag{24}$$

The value u_i^t lies in $[0, 1]$. It reaches its maximum when $q_i^t = 0.5$, which means that the solution is neither clearly strong nor clearly weak relative to the current population. The truth degree is defined as:

$$T_i^t = \text{clip}(\alpha_T q_i^t + \beta_T e_i^t). \tag{25}$$

The truth degree increases when a solution has high normalized quality and is close to the elite region. Therefore, T_i^t controls the exploitation tendency of X_i^t. The indeterminacy degree is defined as:

$$I_i^t = \text{clip}(\alpha_I u_i^t + \beta_I \delta_i^t + \gamma_I s_i^t). \tag{26}$$

The indeterminacy degree increases when a solution has uncertain quality, contributes to population diversity, or shows stagnation. Therefore, I_i^t controls the exploration tendency of X_i^t. The falsity degree is defined as:

$$F_i^t = \text{clip}(\alpha_F(1 - q_i^t) + \beta_F s_i^t). \tag{27}$$

The falsity degree increases when a solution has weak normalized quality or repeated stagnation. Therefore, F_i^t controls the escape tendency of X_i^t. The clipping function is:

$$\text{clip}(z) = \min\big(1, \max(0, z)\big). \tag{28}$$

All weighting coefficients used in the neutrosophic state construction are non-negative. The default values are:

$$\alpha_T = 0.70,\ \beta_T = 0.30, \tag{29}$$

$$\alpha_I = 0.50,\ \beta_I = 0.30,\ \gamma_I = 0.20, \tag{30}$$

$$\alpha_F = 0.70,\ \beta_F = 0.30. \tag{31}$$

These coefficients satisfy the following constraint:

$$\alpha_T + \beta_T = 1,\ \alpha_I + \beta_I + \gamma_I = 1,\ \alpha_F + \beta_F = 1. \tag{32}$$

Thus, the components T_i^t, I_i^t, and F_i^t remain interpretable within $[0, 1]$, while the clipping function provides additional numerical safety.

3.4. Normalized Neutrosophic Movement Weights

The neutrosophic components are not required to sum to one. However, the movement equation requires proportional control. Therefore, NOpt converts them into normalized movement weights:

$$\omega_{T,i}^t = \frac{T_i^t}{T_i^t + I_i^t + F_i^t + \varepsilon}, \tag{33}$$

$$\omega_{I,i}^t = \frac{I_i^t}{T_i^t + I_i^t + F_i^t + \varepsilon}, \tag{34}$$

$$\omega_{F,i}^t = \frac{F_i^t}{T_i^t + I_i^t + F_i^t + \varepsilon}, \tag{35}$$

where $\varepsilon > 0$ is a very small constant used to avoid division by zero. The weight $\omega_{T,i}^t$ controls the influence of exploitation, $\omega_{I,i}^t$ controls the influence of exploration, and $\omega_{F,i}^t$ controls the influence of escape.

3.5. Adaptive Elite Set

NOpt uses an adaptive elite set to guide exploitation and refinement. The elite set is larger at the beginning of the search to preserve diversity and smaller near the end to focus the search around the best regions. The adaptive elite ratio is defined as:

$$\rho_t = \rho_{\min} + (\rho_{\max} - \rho_{\min})\chi_t, \tag{36}$$

where $\rho_{\max}$ is the initial elite ratio and $\rho_{\min}$ is the final elite ratio. The number of elite solutions is:

$$n_e^t = \max(2, \lceil \rho_t N \rceil). \tag{37}$$

After sorting the population in ascending order according to objective value, the elite set is:

$$\mathcal{P}_{el}^t = \left\{X_{(1)}^t, X_{(2)}^t, \dots, X_{(n_e^t)}^t\right\}, \tag{38}$$

where $X_{(1)}^t = X_*^t$ is the current best solution.

3.6. Adaptive Differential Scale and Crossover Rate

To improve search flexibility, NOpt assigns an adaptive differential scale and an adaptive crossover rate to each candidate solution. The adaptive differential scale is:

$$F_{DE,i}^{t} = F_{\min}^{DE} + (F_{\max}^{DE} - F_{\min}^{DE})\big(0.55\omega_{I,i}^{t} + 0.45\omega_{F,i}^{t}\big). \tag{39}$$

Thus, solutions with high indeterminacy or high falsity receive stronger differential movement. The adaptive crossover rate is:

$$CR_{i}^{t} = CR_{\min} + (CR_{\max} - CR_{\min})\big(0.70\omega_{T,i}^{t} + 0.30\omega_{I,i}^{t}\big). \tag{40}$$

Thus, promising and uncertain solutions inherit more information from the generated trial vector, while weaker solutions are controlled mainly through the escape mechanism.

3.7. Truth-Guided Exploitation Operator

The truth-guided exploitation operator moves a candidate solution toward promising regions. It uses the current best solution X_{*}^{t} and one elite solution X_{p}^{t} randomly selected from $\mathcal{P}_{el}^{t}$:

$$A_{i}^{t} = r_{1,i}^{t} \odot (X_{*}^{t} - X_{i}^{t}) + r_{2,i}^{t} \odot (X_{p}^{t} - X_{i}^{t}), \tag{41}$$

where $r_{1,i}^{t}, r_{2,i}^{t} \in [0, 1]^{D}$ are random vectors, and $\odot$ denotes element-wise multiplication. The first term attracts the solution toward the current best position, while the second term attracts it toward another elite solution. This improves exploitation while reducing excessive dependence on a single best point.

3.8. Indeterminacy-Driven Exploration Operator

The indeterminacy-driven exploration operator allows uncertain solutions to explore new regions. It combines a differential population step with a controlled Gaussian perturbation:

$$B_{i}^{t} = F_{DE,i}^{t}\big(X_{m_i}^{t} - X_{n_i}^{t}\big) + \eta_{t}(U - L) \odot Z_{i}^{t}, \tag{42}$$

where Z_{i}^{t} is a random vector based on the normal distribution, and $X_{m_i}^{t}$ and $X_{n_i}^{t}$ are two randomly selected solutions from the current population. Their indices satisfy:

$$m_i \neq n_i,\ m_i \neq i,\ n_i \neq i. \tag{43}$$

The exploration amplitude is:

$$\eta_{t} = \eta_{\min} + (\eta_{\max} - \eta_{\min})\chi_{t}^{\kappa_{\eta}}, \tag{44}$$

where $\eta_{\max}$ and $\eta_{\min}$ are the maximum and minimum exploration amplitudes, and $\kappa_{\eta} > 0$ controls the decay rate. The differential term produces directed exploration based on population variation. The Gaussian perturbation decreases gradually as the search approaches the final stages, while the differential term remains adaptive through $F_{DE,i}^{t}$, allowing population-based directional exploration to continue when needed.

3.9. Falsity-Based Escape Operator

The falsity-based escape operator moves weak or stagnant solutions away from poor search regions. It contains a repulsion term from the current worst solution and a clipped Cauchy perturbation:

$$C_{i}^{t} = r_{5,i}^{t} \odot (X_{i}^{t} - X_{w}^{t}) + \sigma_{t}(U - L) \odot K_{i}^{t}, \tag{45}$$

where $r_{5,i}^{t} \in [0, 1]^{D}$, X_{w}^{t} is the current worst solution, and K_{i}^{t} is a clipped Cauchy random vector. Each component of K_{i}^{t} is generated as:

$$K_{ij}^{t} = max\left(-c_{max}, \min\left(c_{max}, K_{ij}^{t}\right)\right),\ j = 1,2,\dots,D. \tag{46}$$

The escape amplitude is:

$$\sigma_t = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min})\chi_t^{\kappa_\sigma}, \tag{47}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the maximum and minimum escape amplitudes, and $\kappa_\sigma > 0$ controls the decay rate. The repulsion term moves the solution away from the worst region, while the Cauchy perturbation provides occasional long-range movement to reduce stagnation.

3.10. Neutrosophic Elite Refinement Operator

To improve local accuracy in later stages, NOpt introduces a Neutrosophic Elite Refinement Operator. This operator strengthens refinement around the elite region while preserving the neutrosophic search structure. The refinement activation factor is:

$$\ell_t = \tau_t^{\kappa_R}, \tag{48}$$

where $\kappa_R > 0$ controls how quickly refinement becomes active. Since ℓ_t increases as the consumed evaluation budget increases, the refinement operator has weak influence at the beginning and stronger influence near the end. The refinement perturbation scale is:

$$\mu_t = \mu_{\min} + (\mu_{\max} - \mu_{\min})(1 - \ell_t). \tag{49}$$

Thus, the perturbation radius becomes smaller as the algorithm approaches the final stages. Let $X_{u_i}^t$ and $X_{v_i}^t$ be two elite solutions selected from $\mathcal{P}_{el}^t$, where:

$$u_i \neq v_i. \tag{50}$$

The refinement vector is defined as:

$$R_i^t = \ell_t\left[r_{6,i}^t \odot (X_*^t - X_i^t) + F_{DE,i}^t\left(X_{u_i}^t - X_{v_i}^t\right) + \mu_t(U - L) \odot Z_{R,i}^t\right], \tag{51}$$

where $r_{6,i}^t \in [0, 1]^D$, and $Z_{R,i}^t$ is a random vector based on the normal distribution. The first term refines the solution toward the current best solution. The second term uses elite difference information to preserve local directional variation. The third term adds a small perturbation around the elite region. Since this operator is later multiplied by the truth weight, it mainly affects solutions with stronger exploitation potential. This refinement operator does not require an additional objective-function evaluation. It modifies the trial vector before the single trial evaluation of each candidate solution.

3.11. Unified Neutrosophic Trial Vector

The complete trial vector is generated by combining exploitation, exploration, escape, and elite refinement:

$$V_i^t = X_i^t + \omega_{T,i}^t A_i^t + \omega_{I,i}^t B_i^t + \omega_{F,i}^t C_i^t + \omega_{T,i}^t R_i^t. \tag{52}$$

Equation (52) is the core movement equation of NOpt. It shows that the movement of each candidate solution is controlled by its own neutrosophic state. If $\omega_{T,i}^t$ is dominant, the candidate mainly exploits and refines the elite region. If $\omega_{I,i}^t$ is dominant, the candidate explores uncertain regions. If $\omega_{F,i}^{t}$ is dominant, the candidate activates escape from weak or stagnant regions.

3.12. Component-Preserving Crossover

After constructing V_i^t, NOpt applies component-preserving crossover between the current solution X_i^t and the generated trial vector V_i^t. For each dimension j, the trial solution is generated as:

$$y_{ij}^t = \begin{cases} v_{ij}^t, & rand_{ij}^t \le CR_i^t \text{ or } j = j_{rand}, \\ x_{ij}^t, & \text{otherwise}, \end{cases} \tag{53}$$

where $rand_{ij}^t \sim U(0, 1)$, and j_{rand} is a randomly selected dimension from $\{1, 2, \dots, D\}$.

The condition $j = j_{rand}$ ensures that at least one component is inherited from the generated trial vector.

This crossover step improves stability because not all dimensions are forced to change at every iteration.

3.13. Greedy Selection and Stagnation Update

The objective value $f(\tilde{Y}_i^t)$ is evaluated once. The feasible trial solution replaces the current solution only if it improves the objective value:

$$X_i^{t+1} = \begin{cases} \tilde{Y}_i^t, & f(\tilde{Y}_i^t) < f(X_i^t), \\ X_i^t, & \text{otherwise}. \end{cases} \tag{54}$$

The stagnation counter is updated as:

$$a_i^{t+1} = \begin{cases} 0, & f(\tilde{Y}_i^t) < f(X_i^t), \\ a_i^t + 1, & \text{otherwise}. \end{cases} \tag{55}$$

After the complete population update, the number of function evaluations consumed is updated as follows:

$$FEs_{pop}^{t+1} = FEs^t + N. \tag{56}$$

Successful improvement resets the stagnation counter, while failure to improve increases it. The updated stagnation value is then used in the next iteration to influence both indeterminacy and falsity.

3.14. Population Best, Worst, and Global-Best Update Before Local Search

After all candidate solutions are processed, the population before local search is:

$$P_{pop}^{t+1} = \{X_1^{t+1}, X_2^{t+1}, \dots, X_N^{t+1}\}. \tag{57}$$

The index of the best solution in this population is:

$$p_{pop}^{t+1} = \arg\min_{1 \le i \le N} f(X_i^{t+1}), \tag{58}$$

and the index of the worst solution is:

$$w_{pop}^{t+1} = \arg\max_{1 \le i \le N} f(X_i^{t+1}). \tag{59}$$

Accordingly, the best and worst solutions before local search are:

$$X_{*,pop}^{t+1} = X_{p_{pop}^{t+1}}^{t+1},\ X_{w,pop}^{t+1} = X_{w_{pop}^{t+1}}^{t+1}. \tag{60}$$

The global-best solution before local search is updated as:

$$X_{best,pop}^{t+1} = \begin{cases} X_{*,pop}^{t+1}, & f\left(X_{*,pop}^{t+1}\right) < f(X_{best}^{t}), \\ X_{best}^{t}, & \text{otherwise.} \end{cases} \tag{61}$$

The corresponding objective value is:

$$f_{best,pop}^{t+1} = f\left(X_{best,pop}^{t+1}\right). \tag{62}$$

3.15. Neutrosophic Adaptive Local Search Operator

The Neutrosophic Adaptive Local Search Operator improves late-stage accuracy by allocating a controlled part of the remaining evaluation budget to elite-region refinement. Unlike the elite refinement vector in Eq. (51), this operator performs additional objective-function evaluations. Therefore, every local-search trial is counted within $FEs_{\max}$. The local-search activation factor is:

$$\lambda_t = \tau_t^{\kappa_{LS}}, \tag{63}$$

where $\kappa_{LS} > 0$ controls how quickly local search becomes active. The local-search radius is:

$$h_t = h_{\min} + (h_{\max} - h_{\min})(1 - \tau_t)^{\kappa_h}, \tag{64}$$

where $h_{\max}$ is the initial local-search radius, $h_{\min}$ is the minimum local-search radius, and $\kappa_h > 0$ controls the decay rate. The maximum number of local-search trials allowed at iteration t is:

$$B_{LS}^{t} = \min\left(\lceil \lambda_t \rho_{LS} N \rceil, FEs_{\max} - FEs_{pop}^{t+1}\right), \tag{65}$$

where $\rho_{LS} \in [0, 1]$ is the local-search budget ratio. If $B_{LS}^{t} = 0$, the local-search phase is skipped.

For local search, an elite set is reconstructed from P_{pop}^{t+1} using the same adaptive elite ratio ρ_t. This elite set is denoted by:

$$\mathcal{P}_{el,pop}^{t+1}. \tag{66}$$

For each local-search trial $b = 1,2, \ldots, B_{LS}^{t}$, one elite solution $X_{z_b}^{t+1}$ is selected from $\mathcal{P}_{el,pop}^{t+1}$. A local trial solution is generated as:

$$L_b^t = X_{z_b}^{t+1} + h_t(U - L) \odot Z_b^t + \lambda_t r_b^t \odot \left(X_{best,pop}^{t+1} - X_{z_b}^{t+1}\right), \tag{67}$$

where Z_b^t s a random vector based on the normal distribution, and $r_b^t \in [0, 1]^D$. The first perturbation term searches locally around the selected elite solution. The second term directs the local trial toward the best region found so far. Boundary control is applied as:

$$\tilde{L}_{b,j}^t = \min\left(U_j, \max\left(L_j, L_{b,j}^t\right)\right),\ j = 1,2,\dots,D. \tag{68}$$

When $B_{LS}^t > 0$, the objective values of all local-search trials are evaluated and stored. Therefore, the value $f\left(\tilde{L}_{best}^t\right)$ is reused in the update rules without an additional function evaluation. The best local trial is defined as:

$$\tilde{L}_{best}^t = \arg\min_{1 \le b \le B_{LS}^t} f\left(\tilde{L}_b^t\right). \tag{69}$$

The global-best solution after local search is updated as:

$$X_{best}^{t+1} = \begin{cases} \tilde{L}_{best}^t, & B_{LS}^t > 0 \text{ and } f\left(\tilde{L}_{best}^t\right) < f_{best,pop}^{t+1}, \\ X_{best,pop}^{t+1}, & \text{otherwise.} \end{cases} \tag{70}$$

The corresponding objective value is:

$$f_{best}^{t+1} = f(X_{best}^{t+1}). \tag{71}$$

If the best local trial improves the global-best solution, it is inserted into the population by replacing the worst member of P_{pop}^{t+1}:

$$X_{w_{pop}^{t+1}}^{t+1} \leftarrow \tilde{L}_{best}^t, \text{ if } B_{LS}^t > 0 \text{ and } f\left(\tilde{L}_{best}^t\right) < f_{best,pop}^{t+1}. \tag{72}$$

The stored objective value of $\tilde{L}_{best}^t$is assigned to the replaced population member, so no additional objective-function evaluation is required for this replacement. After this possible replacement, the final population of iteration $t + 1$ is denoted by:

$$P^{t+1}. \tag{73}$$

The best and worst solutions are then recomputed from the final population:

$$X_*^{t+1} = \arg\min_{X_i^{t+1} \in P^{t+1}} f(X_i^{t+1}), \tag{74}$$

$$X_w^{t+1} = \arg\max_{X_i^{t+1} \in P^{t+1}} f(X_i^{t+1}). \tag{75}$$

The number of function evaluations consumed after local search is:

$$FEs^{t+1} = FEs^{t+1} + B_{LS}^t. \tag{76}$$

This operator strengthens late-stage refinement while preserving a fair evaluation budget. It does not give NOpt extra evaluations beyond the predefined maximum because each local trial is explicitly counted.

3.16. Pseudocode of the proposed NOpt

The key parameters of the refined NOpt algorithm are summarized in Table 1. These parameters are fixed across benchmark functions in the experimental study unless otherwise stated. Finally, the pseudocode of the proposed algorithm is provided in Algorithm 1. The refined NOpt performs adaptive search at the level of each candidate solution. A promising solution receives stronger truth-guided exploitation and elite refinement. An uncertain solution receives stronger indeterminacy-driven exploration. A weak or stagnant solution receives stronger falsity-based escape. The adaptive elite set supports broader guidance at the beginning of the search and more focused refinement near the end. The adaptive differential scale and crossover rate allow each candidate solution to receive a movement pattern consistent with its neutrosophic state. The elite refinement operator improves the generated trial vector before evaluation, while the adaptive local-search operator allocates part of the remaining evaluation budget to direct refinement around elite solutions. Therefore, NOpt does not rely on one fixed movement rule for the whole population. Instead, each solution receives its own movement balance according to its truth, indeterminacy, and falsity degrees, while the algorithm remains bounded by the same objective-function evaluation budget used for experimental comparison.

Table 1. Main parameters of the refined NOpt algorithm

Parameter	Meaning	Default value	Parameter	Meaning	Default value
N	Population size	Problem-dependent	$c_{\max}$	Cauchy clipping limit	3
D	Number of decision variables	Problem-dependent	κ_η	Exploration decay exponent	1.5
$FEs_{\max}$	Maximum number of function evaluations	Problem-dependent	κ_σ	Escape decay exponent	2.0
$\rho_{\max}$	Initial elite ratio	0.30	κ_R	Elite-refinement activation exponent	1.5
$\rho_{\min}$	Final elite ratio	0.05	κ_{LS}	Local-search activation exponent	2.0
ρ_{LS}	Local-search budget ratio	0.20	κ_h	Local-search radius decay exponent	2.0
$a_{\max}$	Maximum stagnation limit	20	ε	Small positive constant	10^{-12}
$\eta_{\max}$	Maximum exploration amplitude	0.03	$h_{\max}$	Maximum local-search radius	0.05
$\eta_{\min}$	Minimum exploration amplitude	10^{-5}	$h_{\min}$	Minimum local-search radius	10^{-10}
$\sigma_{\max}$	Maximum escape amplitude	0.02	$F_{\min}^{DE}$	Minimum adaptive differential scale	0.40
$\sigma_{\min}$	Minimum escape amplitude	10^{-6}	$F_{\max}^{DE}$	Maximum adaptive differential scale	1.00
$CR_{\max}$	Maximum crossover rate	1.00	$CR_{\min}$	Minimum crossover rate	0.05

Algorithm 1: Pseudocode of NOpt

Output : X_{best}

1. Initialize N solutions using Eq. (17)
2. Evaluate each X_i^0 using Eq. (18)

3. Identify X_*^0 and X_w^0
4. $X_{best}^0 = X_*^0$
5. $t = 0$
6. $FEs^t = N$
7. $a_i^0 = 0, \forall i \in \{1{:}N\}$
8. **while** $(FEs^t < FEs_{\max})$
9. Compute τ_t and χ_t using (21) and (22).
10. Compute q_i^t, δ_i^t, e_i^t, and s_i^t for all solutions.
11. Compute u_i^t using (24).
12. Compute T_i^t, I_i^t, and F_i^t using (25), (26), and (27).
13. Compute $\omega_{T,i}^t$, $\omega_{I,i}^t$, and $\omega_{F,i}^t$ using (33), (34), and (35).
14. Construct the adaptive elite set using (36), (37), and (38).
15. Compute $F_{DE,i}^t$ and CR_i^t using (39) and (40).
16. **for** $i = 1{:}N$
17. Compute the exploitation vector A_i^t using (41).
18. Compute the exploration vector B_i^t using (42).
19. Compute the escape vector C_i^t using (45).
20. Compute the elite-refinement vector R_i^t using (51).
21. Generate the trial vector V_i^t using (52).
22. Apply component-preserving crossover using (53).
23. Evaluate $\tilde{Y}_i^t$
24. Apply greedy selection using (54).
25. Update the stagnation counter using (55).
26. Update the best and worst solutions.
27. $FEs^t = FEs^t + 1$
28. **End for**
29. Reconstruct the elite set
30. Compute λ_t, h_t, and B_{LS}^t using (63), (64), and (65).
31. $\boldsymbol{if}$ $B_{LS}^t > 0$
32. Generate B_{LS}^t local-search trials using (67).
33. Apply boundary control to the local-search trials using (68).
34. Evaluate and store the objective values of all local-search trials.
35. Select the best local trial using (69).
36. Update the best solution using (70).

37. If the best local trial improves the global best, replace the worst population member using (72).
38. $\boldsymbol{End\ \ If}$
39. Define the final population P^{t+1} using (73).
40. Recompute X_*^{t+1} and X_w^{t+1} from P^{t+1} using (74) and (75).
41. $FEs^t = FEs^t + B_{LS}^t$
42. $t = t + 1$
43. **End while**

4. Results and discussion

In this study, the proposed NOpt optimizer is evaluated on 21 standard test functions (TFs) listed in Table 2 [4, 12]. These test functions are widely used in the literature to assess the exploratory and exploitative performance of newly proposed algorithms, which is the main reason for using them to evaluate NOpt. The proposed NOpt is compared with several competing algorithms, including Weighted average algorithm (WAA) [13], Runge-Kutta method (RUN) beyond the metaphor [14], Rime optimization algorithm (RIME) [15], whale optimization algorithm (WOA) [3], sine-cosine algorithm (SCA) [16], and GOOSE algorithm [17], to demonstrate its effectiveness and assess whether it provides meaningful contributions to the optimization field. The control parameters of the competing algorithms, except for the maximum number of function evaluations and population size, are set as recommended in the original references. The maximum number of function evaluations and population size are set to 100,000 and 20, respectively, for all algorithms to ensure a fair comparison. After executing each algorithm independently 30 times using different random seeds, the resulting fitness values are analyzed in terms of several performance metrics, including best fitness (best), average fitness (Avg), standard deviation (SD), and Friedman mean rank (FK). All experiments are conducted on a device with the following hardware and software specifications: 11th Gen Intel(R) Core(TM) i7-11850H @ 2.50 GHz, 32 GB of RAM, and a 64-bit Windows 11 Professional operating system. All algorithms are implemented in MATLAB R2024b under the same computational environment to ensure a fair comparison.

Each algorithm is independently executed 30 times on each standard TF using different random seeds. The resulting fitness values are then analyzed in terms of the aforementioned performance metrics and provided in Table 3. As can be observed from this table, NOpt outperforms all algorithms on 3 of the 21 TFs, achieves competitive performance on four TFs, and is inferior to some algorithms, such as WAA, RUN, and NRBO, on the remaining TFs. To assess the overall performance of the algorithms, their average FK values across all TFs are computed and presented in Fig. 1. As shown in this figure, RUN achieves the best overall performance with an average FK of 2.54, followed by NRBO, whereas NOpt ranks fourth with an average FK of 4.00. Although the proposed algorithm in this study does not achieve the best overall performance, the obtained results provide empirical evidence that relying solely on neutrosophic set behaviors does not provide sufficient diversity in the search process, thereby limiting the algorithm's ability to achieve outstanding performance across the majority of TFs. In addition, the results show that NOpt can achieve outstanding performance on the fixed-dimensional TFs, which simulate real-world optimization problems, where it demonstrates competitive performance against some algorithms and outperforms others, indicating its potential for effectively addressing real-world optimization problems. Therefore, investigating the performance of NOpt on a broader range of real-world problems is considered an important direction for future research.

Table 2. Description of the standard test functions.

ID	Functions	D	Search range	Global opt
TF1	$\boldsymbol{F1} = \sum_{i=1}^{D} x_i^2$	100	[-100,100]	0
TF2	$\boldsymbol{F2} = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	100	[-100,100]	0
TF3	$\boldsymbol{F3} = \sum_{i=1}^{D} \left(\sum_{j=1}^{D} x_j \right)^2$	100	[-100,100]	0
TF4	$\boldsymbol{F4} = max(\lvert x_i \rvert) \ \lvert 1 \le i \le D$	100	[-100,100]	0
TF5	$\boldsymbol{F5} = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	100	[-30,30]	0
TF6	$\boldsymbol{F6} = \sum_{i=1}^{D-1} [x_i + 0.5]^2$	100	[-100,100]	0
TF7	$\boldsymbol{F7} = \sum_{i=1}^{D} i x_i^4 + random[0,1]$	100	[-1.28,1.28]	0
TF8	$\boldsymbol{F8} = \sum_{i=1}^{D-1} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	100	[-500,500]	$-418.98 \times D$
TF9	$\boldsymbol{F9} = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	100	[-5.12,5.12]	0
TF10	$\boldsymbol{F10} = -20exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e$	100	[-32,32]	0
TF11	$\boldsymbol{F11} = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	100	[-600,600]	0
TF12	$\boldsymbol{F12} = \frac{\pi}{D} \left(10 \ sin(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10\,sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right) + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$	100	[-50,50]	0

TF13	$\boldsymbol{F13} = \frac{\boldsymbol{\pi}}{\boldsymbol{d}}\left(\boldsymbol{10\,sin^2(3x_1)} + \sum_{\boldsymbol{i=1}}^{\boldsymbol{D-1}}(\boldsymbol{x_i} - \boldsymbol{1})^2\,[\boldsymbol{1} + \boldsymbol{sin^2(3\pi x_i)}] + (\boldsymbol{x_d} - \boldsymbol{1})^2[\boldsymbol{1} + \boldsymbol{sin^2(2\pi x_D)}]\right) + \sum_{\boldsymbol{i=1}}^{\boldsymbol{d}} \boldsymbol{u(x_i\ ,5,100,4)}$	100	[-50,50]	0
TF14	$\boldsymbol{F14} = \left(\frac{\boldsymbol{1}}{\boldsymbol{500}} + \sum_{\boldsymbol{j=1}}^{\boldsymbol{25}} \frac{\boldsymbol{1}}{\boldsymbol{j} + \sum_{\boldsymbol{i=1}}^{\boldsymbol{2}}(\boldsymbol{x_i} - \boldsymbol{a_{ij}})^2}\right)^{-1}$	2	[-65,65]	1
TF15	$\boldsymbol{F15} = \sum_{i=1}^{11}\left[\boldsymbol{a_i} - \frac{\boldsymbol{x_{1(b_i^2+b_ix_2)}}}{\boldsymbol{b_i^2 + b_ix_3 + x_4}}\right]^2$	4	[-5,5]	0.00030
TF16	$\boldsymbol{F16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
TF17	$\boldsymbol{F17} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[-5,5]	0.398
TF18	$\boldsymbol{F18} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
TF19	$\boldsymbol{F19} = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2}$	3	[1,3]	-3.86
TF20	$\boldsymbol{F20} = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2}$	6	[0,1]	-3.32
TF21	$\boldsymbol{F21} = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532

Table 3: Results of the algorithms across 21 standard TFs.

	TF1				TF2				TF3			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	1.38E-01	2.12E-01	6.28E-02	5.70	8.24E-01	1.49E+00	3.27E-01	6.00	6.24E+03	9.82E+03	1.95E+03	5.00
WOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	9.11E+04	4.15E+05	1.37E+05	7.93
WAA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.00**
SCA	9.78E-01	9.71E+02	1.83E+03	7.97	1.27E-08	3.22E-03	1.43E-02	5.00	5.85E+04	1.31E+05	3.53E+04	7.07
RUN	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.00**
GOOSE	1.10E-01	1.82E-01	4.24E-02	5.30	2.17E+02	4.18E+16	2.29E+17	8.00	1.62E+03	2.26E+03	3.79E+02	4.00
RIME	1.46E+00	2.00E+00	3.81E-01	7.03	9.72E+00	1.79E+01	5.32E+00	7.00	1.97E+04	2.39E+04	2.72E+03	6.00
NRBO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.00**
	TF4				TF5				TF6			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	1.12E+01	1.49E+01	2.31E+00	4.03	1.04E+02	5.26E+02	6.59E+02	5.73	1.13E-01	2.10E-01	5.77E-02	4.30
WOA	1.26E+01	6.90E+01	2.59E+01	7.03	9.55E+01	9.63E+01	6.11E-01	2.50	7.17E-02	2.37E-01	1.41E-01	4.00

WAA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.00**	**2.53E-04**	**5.42E-02**	**9.09E-02**	**1.00**	1.84E-06	4.17E-03	5.25E-03	2.00
SCA	6.85E+01	7.79E+01	4.38E+00	7.53	1.29E+06	1.27E+07	1.26E+07	8.00	2.19E+01	1.34E+03	1.80E+03	8.00
RUN	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.00**	9.43E+01	9.65E+01	1.39E+00	2.63	**2.07E-08**	**5.65E-08**	**5.91E-08**	**1.00**
GOOSE	3.06E+01	3.92E+01	5.13E+00	5.70	1.22E+02	5.40E+02	6.97E+02	5.83	1.24E-01	1.82E-01	3.56E-02	3.70
RIME	2.83E+01	3.94E+01	4.83E+00	5.70	2.17E+02	7.99E+02	7.47E+02	6.43	1.23E+00	1.88E+00	3.41E-01	6.00
NRBO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.00**	9.71E+01	9.84E+01	4.33E-01	3.87	1.58E+01	1.71E+01	7.69E-01	7.00
	TF7				TF8				TF9			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	5.44E-02	9.33E-02	2.74E-02	5.07	-2.64E+04	-2.43E+04	1.12E+03	4.63	1.59E+02	1.98E+02	2.42E+01	5.80
WOA	1.99E-05	5.55E-04	6.40E-04	3.87	-4.19E+04	-3.95E+04	4.26E+03	2.13	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**
WAA	**9.89E-07**	**2.16E-05**	**2.43E-05**	**1.43**	**-4.19E+04**	**-4.19E+04**	**5.62E-02**	**1.00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**
SCA	4.57E-01	1.02E+01	9.84E+00	7.83	-9.43E+03	-7.69E+03	5.15E+02	7.90	1.51E+01	1.28E+02	8.64E+01	5.23
RUN	5.37E-06	6.70E-05	7.24E-05	2.60	-2.75E+04	-2.27E+04	2.69E+03	5.17	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**
GOOSE	7.52E-01	1.59E+00	3.49E-01	7.17	-2.58E+04	-2.30E+04	1.28E+03	5.20	5.03E+02	6.52E+02	9.53E+01	8.00
RIME	9.82E-02	1.52E-01	2.79E-02	5.93	-3.38E+04	-3.12E+04	1.22E+03	2.87	1.88E+02	2.99E+02	4.64E+01	6.97
NRBO	7.87E-06	3.60E-05	2.80E-05	2.10	-1.14E+04	-9.19E+03	1.11E+03	7.10	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**
	TF10				TF11				TF12			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	4.69E-01	1.42E+00	3.46E-01	5.07	1.95E-01	3.11E-01	5.13E-02	6.00	6.17E-03	2.11E+00	1.75E+00	4.67
WOA	4.44E-16	3.52E-15	2.42E-15	3.55	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	7.81E-04	1.93E-03	1.23E-03	3.00
WAA	**4.44E-16**	**4.44E-16**	**0.00E+00**	**2.15**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	2.39E-08	3.78E-06	4.59E-06	2.00
SCA	9.45E-01	1.75E+01	6.86E+00	7.67	5.32E-01	8.93E+00	1.05E+01	7.97	7.46E+05	2.62E+07	2.17E+07	8.00
RUN	**4.44E-16**	**4.44E-16**	**0.00E+00**	**2.15**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	**1.11E-09**	**1.74E-09**	**3.79E-10**	**1.00**
GOOSE	1.80E+01	1.93E+01	3.29E-01	7.20	9.39E-03	1.55E-02	6.01E-03	5.00	3.19E+00	5.13E+00	1.05E+00	6.03
RIME	2.39E+00	2.95E+00	3.50E-01	6.07	6.13E-01	7.66E-01	6.94E-02	7.03	6.72E+00	1.04E+01	2.83E+00	6.97
NRBO	**4.44E-16**	**4.44E-16**	**0.00E+00**	**2.15**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.50**	5.54E-01	6.33E-01	4.92E-02	4.33
	TF13				TF14				TF15			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	1.10E-01	3.39E-01	1.69E-01	3.53	**9.98E-01**	**9.98E-01**	**4.70E-16**	**1.85**	**3.07E-04**	4.22E-04	3.01E-04	2.80
WOA	1.55E-01	6.24E-01	4.04E-01	4.20	9.98E-01	1.26E+00	6.86E-01	4.52	3.08E-04	5.04E-04	2.39E-04	4.83
WAA	**2.26E-07**	**1.81E-04**	**2.89E-04**	**1.37**	9.98E-01	9.98E-01	2.03E-13	4.23	3.11E-04	**3.49E-04**	**3.90E-05**	4.10
SCA	4.68E+05	4.99E+07	6.75E+07	8.00	9.98E-01	1.13E+00	5.03E-01	5.90	3.13E-04	9.01E-04	4.38E-04	5.87
RUN	5.79E-08	1.99E-02	3.81E-02	1.63	9.98E-01	4.53E+00	3.91E+00	5.77	3.07E-04	6.13E-04	4.39E-04	2.87
GOOSE	4.04E-02	3.30E+00	7.34E+00	4.37	9.98E-01	1.36E+01	6.46E+00	7.48	3.08E-04	7.23E-03	9.45E-03	6.43
RIME	1.84E+00	1.02E+02	6.47E+01	6.80	9.98E-01	9.98E-01	7.91E-14	3.32	3.08E-04	6.37E-03	1.24E-02	5.40

NRBO	9.69E+00	9.87E+00	1.00E-01	6.10	9.98E-01	2.44E+00	2.94E+00	2.93	3.08E-04	1.79E-03	5.06E-03	3.70
	TF16				TF17				TF18			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	**-1.03E+00**	**-1.03E+00**	7.00E-13	3.57	**3.98E-01**	**3.98E-01**	2.47E-12	2.57	**3.00E+00**	**3.00E+00**	7.15E-12	3.00
WOA	**-1.03E+00**	**-1.03E+00**	4.52E-13	2.90	3.98E-01	3.98E-01	1.20E-08	5.13	**3.00E+00**	**3.00E+00**	2.57E-06	6.30
WAA	**-1.03E+00**	**-1.03E+00**	3.83E-07	7.03	3.98E-01	3.98E-01	3.10E-07	6.97	**3.00E+00**	**3.00E+00**	1.71E-05	7.83
SCA	**-1.03E+00**	**-1.03E+00**	6.89E-06	7.90	3.98E-01	3.98E-01	2.88E-04	8.00	**3.00E+00**	**3.00E+00**	2.92E-06	6.57
RUN	**-1.03E+00**	**-1.03E+00**	**1.66E-13**	**2.53**	**3.98E-01**	**3.98E-01**	**2.45E-12**	**2.50**	**3.00E+00**	**3.00E+00**	**4.54E-14**	**1.97**
GOOSE	**-1.03E+00**	-1.00E+00	1.49E-01	5.63	3.98E-01	3.98E-01	3.99E-10	4.67	**3.00E+00**	**3.00E+00**	3.90E-08	5.27
RIME	**-1.03E+00**	**-1.03E+00**	8.58E-10	5.43	3.98E-01	3.98E-01	1.11E-09	5.13	**3.00E+00**	**3.00E+00**	4.08E-09	4.00
NRBO	**-1.03E+00**	-1.03E+00	6.39E-16	1.00	3.98E-01	3.98E-01	0.00E+00	1.03	**3.00E+00**	**3.00E+00**	1.19E-14	1.07
	TF19				TF20				TF21			
	Best	Avg	SD	FK	Best	Avg	SD	FK	Best	Avg	SD	FK
NOpt	**-3.86E+00**	**-3.86E+00**	**2.46E-11**	2.10	**-3.32E+00**	**-3.32E+00**	**2.74E-11**	**1.07**	**-1.02E+01**	**-1.02E+01**	6.27E-10	1.53
WOA	**-3.86E+00**	**-3.86E+00**	2.57E-03	6.67	-3.32E+00	-3.27E+00	6.93E-02	4.60	**-1.02E+01**	-9.67E+00	1.90E+00	4.67
WAA	**-3.86E+00**	**-3.86E+00**	7.26E-05	6.33	-3.32E+00	-3.26E+00	5.30E-02	5.37	**-1.02E+01**	**-1.02E+01**	1.04E-04	4.23
SCA	**-3.86E+00**	**-3.86E+00**	1.46E-03	8.00	-3.13E+00	-2.76E+00	5.16E-01	8.00	-7.08E+00	-2.91E+00	2.43E+00	7.60
RUN	**-3.86E+00**	**-3.86E+00**	4.63E-08	2.97	-3.32E+00	-3.28E+00	5.83E-02	3.00	**-1.02E+01**	**-1.02E+01**	**1.60E-10**	**1.47**
GOOSE	**-3.86E+00**	**-3.86E+00**	4.61E-07	4.93	-3.32E+00	-3.21E+00	3.04E-02	5.83	-1.02E+01	-4.73E+00	2.94E+00	6.43
RIME	**-3.86E+00**	**-3.86E+00**	1.47E-09	3.67	-3.32E+00	-3.31E+00	4.11E-02	2.90	-1.02E+01	-9.14E+00	2.06E+00	4.03
NRBO	**-3.86E+00**	**-3.86E+00**	2.68E-06	**1.33**	-3.32E+00	-3.25E+00	6.34E-02	5.23	-1.02E+01	-9.03E+00	1.36E+00	6.03

Bold values represent the best findings.

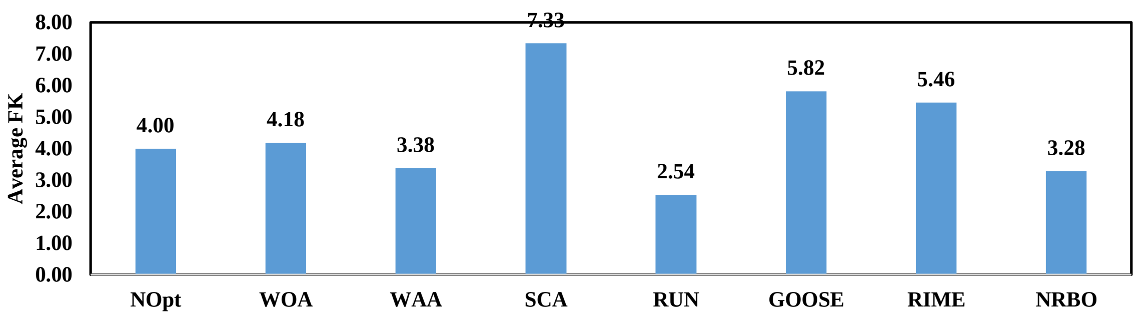

Fig. 1: Average FK values obtained by the algorithms across all standard TFs.

5. Conclusion and future work

This study introduced a novel neutrosophic optimizer, termed NOpt, for continuous global optimization. The proposed algorithm represents each candidate solution using a neutrosophic search state comprising truth, indeterminacy, and falsity degrees. These three components are designed to enhance the different search behaviors required during optimization. Specifically, the truth component promotes exploitation around promising regions, the indeterminacy component enhances exploration of uncertain regions, and the falsity component facilitates escape from poor or stagnant regions. In addition to the main neutrosophic search mechanism, two complementary operators were incorporated into NOpt to further improve its search performance. The Neutrosophic Elite Refinement Operator improves each generated trial vector before its evaluation, whereas the Neutrosophic Adaptive Local Search Operator utilizes a controlled portion of the remaining function-evaluation budget to refine elite solutions. The performance of NOpt was comprehensively assessed against seven established metaheuristic algorithms using a rigorous experimental protocol. The obtained results demonstrated that NOpt can achieve outstanding performance on the fixed-dimensional test functions, which are designed to represent real-world optimization problems. These results indicate that incorporating truth, indeterminacy, and falsity degrees into the search process can provide a promising framework for balancing exploration and exploitation and improving the ability of a metaheuristic algorithm to locate high-quality solutions. Despite these promising results, the proposed algorithm has several potential directions for further investigation. Future work will focus on extending NOpt to multi-objective, constrained, and discrete optimization problems to examine its adaptability to different optimization scenarios. Incorporating adaptive or learning-based mechanisms into the neutrosophic search process may provide additional improvements in convergence and exploration–exploitation balance. Finally, applying NOpt to a broader range of real-world engineering and practical optimization problems will be considered to further validate its effectiveness and practical applicability.

References

1. Faramarzi, A., et al., *Equilibrium optimizer: A novel optimization algorithm.* Knowledge-Based Systems, 2020. 191: p. 105190.
2. Mirjalili, S., S.M. Mirjalili, and A. Lewis, *Grey Wolf Optimizer.* Advances in Engineering Software, 2014. 69: p. 46-61.
3. Mirjalili, S. and A. Lewis, *The Whale Optimization Algorithm.* Advances in Engineering Software, 2016. 95: p. 51-67.
4. Li, S., et al., *Slime mould algorithm: A new method for stochastic optimization.* Future Generation Computer Systems, 2020. 111: p. 300-323.
5. Rajwar, K., K. Deep, and S. Das, *An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges.* Artificial Intelligence Review, 2023. 56(11): p. 13187-13257.
6. Zhang, W., et al., *Reinforcement learning-enhanced metaheuristic algorithm for a battery recycling closed-loop supply chain network design problem.* Research in Transportation Business & Management, 2026. 68: p. 101768.
7. Paknahad, M., P. Hosseini, and S.J.S. Hakim, *Enhancing optimization efficiency through discrete integer metaheuristic optimization algorithm (DIMOA): a novel metaheuristic approach for truss optimization.* Neural Computing and Applications, 2026. 38(4): p. 64.
8. Abdel-Basset, M., et al., *Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems.* Computer Methods in Applied Mechanics and Engineering, 2023. 415: p. 116200.
9. Biju, E., et al., *Deep neuro-neutrosophic systems for detection of online sexism.* Engineering Applications of Artificial Intelligence, 2025. 159: p. 111714.
10. Özçil, A., et al., *The enhanced ARAS method within the interval-valued neutrosophic set framework: a novel approach for climate change performance assessment.* Expert Systems with Applications, 2027. 333: p. 134192.
11. Ejegwa, P.A., et al., *Spearman-based correlation methods under single-valued neutrosophic sets with application to wheat farmland selection.* Scientific Reports, 2026.
12. Abdel-Basset, M., et al., *Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems.* Knowledge-Based Systems, 2023. 262: p. 110248.
13. Cheng, J. and W. De Waele, *Weighted average algorithm: A novel meta-heuristic optimization algorithm based on the weighted average position concept.* Knowledge-Based Systems, 2024. 305: p. 112564.
14. Ahmadianfar, I., et al., *RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method.* Expert Systems with Applications, 2021. 181: p. 115079.
15. Su, H., et al., *RIME: A physics-based optimization.* Neurocomputing, 2023. 532: p. 183-214.
16. Mirjalili, S., *SCA: A Sine Cosine Algorithm for solving optimization problems.* Knowledge-Based Systems, 2016. 96: p. 120-133.
17. Hamad, R.K. and T.A. Rashid, *GOOSE algorithm: A powerful optimization tool for real-world engineering challenges and beyond.* Evolving Systems, 2024: p. 1-26.

Received: Jan 19, 2026. Accepted: Aug 29, 2026