Unscented Sigma Point-Guided Quasi-Opposition Slime Mould Algorithm and Its Engineering Applications: Postprint

Authors: Liu Yusong, Liu Sheng

Date: 2022-05-10T00:00:00+00:00

Abstract

To address issues such as search stagnation and poor algorithmic stability in the slime mold algorithm, this paper proposes an unscented sigma point guided quasi-oppositional slime mold algorithm. First, two opposition-based learning processes—quasi-oppositional learning and quasi-reflection learning—are employed to generate a comprehensive opposition population containing both quasi-opposite and quasi-reflected individuals based on the timing of exploration and exploitation behaviors in the original slime mold algorithm, thereby expanding the search scope. Second, the diversity degree of the population is utilized to determine whether to reconstruct the original population using the opposition population for subsequent calculations, avoiding the disruption of the population's inherent search characteristics by a fixed opposition process and improving search precision. Finally, sigma points from the unscented transform are introduced to improve the fundamental movement pattern of the slime mold algorithm, enabling unscented sigma points to guide the search of the slime mold algorithm and accelerate convergence speed. The experimental section employs CEC2017 benchmark functions, utilizes traditional statistical characteristics, MAE ranking, and Wilcoxon rank-sum test to verify algorithm effectiveness, and applies the proposed algorithm to solve a real-world engineering optimization problem of car side impact. Comparative tests are conducted against novel high-level swarm intelligence algorithms, improved algorithms, and incomplete algorithms. Experimental results demonstrate that the proposed improvement strategies are effective and mutually complementary, with the improved algorithm exhibiting superior competitiveness in solution accuracy and robustness.

Full Text

Preamble

Vol. 39 No. 9
Application Research of Computers
ChinaXiv Cooperative Journal
Accepted Paper

Unscented Sigma Point Guided Quasi-Opposite Slime Mould Algorithm and Its Engineering Application

Liu Yusong, Liu Sheng†
(School of Management Studies, Shanghai University of Engineering Science, Shanghai 201600, China)

Abstract: To address the issues of search stagnation and poor stability in the Slime Mould Algorithm (SMA), this paper proposes an Unscented Sigma Point Guided Quasi-Opposite Slime Mould Algorithm (UQSMA). First, two opposition-based learning processes—quasi-opposition learning and quasi-reflection learning—are employed to generate a comprehensive opposite population containing both quasi-opposite and quasi-reflected individuals based on the exploration and exploitation behaviors of the original SMA, thereby expanding the search space. Second, the diversity degree of the population is used to determine whether to reconstruct the original population using the opposite population for subsequent calculations, avoiding the destruction of the population's inherent search characteristics by a fixed opposition process and improving search accuracy. Finally, unscented transformation sigma points are introduced to improve the basic movement pattern of SMA, enabling unscented sigma points to guide the search process and accelerate convergence speed. The experimental section employs CEC2017 benchmark test functions, using traditional statistical features, MAE ranking, and Wilcoxon rank-sum test to verify algorithm effectiveness. The algorithm is also applied to solve a real-world engineering optimization problem of car side impact design, with comparative tests conducted against novel high-level swarm intelligence algorithms, improved algorithms, and incomplete algorithms. Experimental results demonstrate that the proposed improvement strategies are effective and that the combination of strategies complements each other, making the improved algorithm more competitive in solution accuracy and robustness.

Keywords: slime mould algorithm; quasi-opposite learning; quasi-reflection learning; unscented transformation; CEC2017

---

0 Introduction

In the context of big data and artificial intelligence, swarm intelligence heuristic algorithms have become increasingly popular in recent decades due to their excellent performance on large-scale complex optimization problems and low computational requirements. The Slime Mould Algorithm (SMA) [?], proposed by Shimin Li et al. in 2020, is a stochastic optimization algorithm with simple movement principles but incorporating dynamic self-adaptation, feedback weighting, and other improvement mechanisms, along with extensive customization space, attracting widespread attention from researchers worldwide. SMA is inspired by the oscillation patterns of slime mould in nature, which uses biological oscillations to transmit positive or negative feedback about food information at different locations, thereby finding the optimal food source. The algorithm has been successfully applied to photovoltaic model parameter optimization [?], machine learning feature selection [?], medical X-ray image segmentation [?], computer vision image thresholding [?], and other practical problems, demonstrating broad development prospects in financial technology, data mining, and industrial engineering.

However, SMA still suffers from search stagnation, poor stability, and low solution accuracy. Although current research on SMA is relatively limited, scholars domestically and internationally have employed different strategies to achieve various improvements. In the direction of using opposition-based learning frameworks to improve SMA, reference [?] utilized a chaotic opposition learning framework to expand the algorithm's search space; reference [?] employed a random opposition learning framework to help the algorithm escape local optima; reference [?] improved SMA using a combined strategy of elite opposition and quadratic interpolation, achieving better overall performance. In the direction of improving SMA's basic movement pattern, reference [?] used a strategy called "attacking-feeding" based on greedy strategy and Gaussian mutation to attempt new positions for individual solutions in each iteration, significantly improving solution accuracy; reference [?] introduced the crossover operator from the Artificial Bee Colony algorithm to accelerate convergence; reference [?] incorporated the search mechanism of the Sine Cosine Algorithm to improve search efficiency; reference [?] replaced SMA's basic movement pattern with quantum rotation gates to address search stagnation and local optima issues.

In studies using opposition-based learning frameworks to improve SMA, most employ single opposition strategies and lack judgment and control over the timing of opposite population reconstruction, leaving room for further optimization in preserving the algorithm's original characteristics and reducing computational complexity. In studies improving SMA's basic movement pattern, the main approaches involve using classical improvement strategies or core movement patterns from other algorithms to replace SMA's movement mechanism, lacking differentiated and targeted adjustments for different search behavior stages of SMA, leaving room for further optimization in compensating for algorithm limitations while preserving its original advantages.

To address the problems of search stagnation and poor stability in SMA, this paper employs an opposition-based learning strategy framework to improve SMA, adjusting and combining strategies based on SMA's characteristics to propose the Unscented Sigma Point Guided Quasi-Opposite Slime Mould Algorithm (UQSMA). The algorithm uses combined quasi-opposite and quasi-reflection processes to escape local optima and maintain population diversity; simultaneously, it reconstructs the original population with opposite populations based on population diversity to reduce computational overhead from fixed opposition processes and enhance algorithm stability; finally, it uses unscented sigma points to guide the population toward the theoretical optimum, applying different improvement strategies for different search stages of SMA to compensate for its poor stability when facing different experiments and problems. The combination of multiple strategies balances the exploration and exploitation phases, with each strategy complementing others to overcome individual limitations while ensuring effectiveness. Experiments on CEC2017 benchmark test functions verify the positive impact of the improvement strategies on solution accuracy and robustness.

---

1 Basic Slime Mould Algorithm

The slime mould in SMA generally refers to *Physarum polycephalum*, originally classified as a fungus, hence the name "slime mould." It is a eukaryotic organism living in cold, damp places, with its active dynamic phase being the primary focus of SMA research. In this phase, slime mould uses organic matter to locate food, surround it, and secrete enzymes to digest it. During migration, its front extends into a fan shape while the rear connects to a vein network ensuring cytoplasm flow for information communication. Based on food quality and density, it dynamically adjusts vein diameters, ultimately positioning the entire organism at the optimal food source. Inspired by this behavior, Shimin Li and his team developed SMA.

The abstract mathematical description of basic SMA is as follows: Assume N slime mould individuals are distributed in a D-dimensional search space. The position of the i-th slime mould at iteration t can be expressed as $X_i(t)$. $T_{\max}$ is the maximum iteration number, and the global best position X_b represents the food location. The foraging process consists of three stages:

1) Approaching Food: Slime mould can approach food based on scent in the air. The following equation simulates this process:

$$X_i(t+1) = \begin{cases} X_i(t) + v_b \cdot (W \cdot X_A(t) - X_B(t)), & r_p < z \\ v_c \cdot X_i(t), & r_p \geq z \end{cases}$$

where v_b is a coefficient following uniform distribution $\mathcal{U}[-a, a]$, v_c linearly decreases from 1 to 0, t represents the current iteration, X_b represents the position of the individual with the highest food concentration found so far (the current global best solution), X_i represents the slime mould position, X_A and X_B represent two randomly selected individuals from the population, and W represents the weight of slime mould individuals.

The weight W is calculated as:

$$W_i = \begin{cases} 1 + r \cdot \log\left(\frac{bF - S_i}{bF - wF} + 1\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{S_i - wF}{bF - wF} + 1\right), & \text{otherwise} \end{cases}$$

where S_i represents the fitness value of the i-th slime mould individual, bF represents the best fitness found in all iterations, and wF represents the worst fitness in the current iteration.

The parameter v_c is calculated as:

$$v_c = \tanh |S_i - DF|$$

where DF is the optimal fitness value in the current iteration. The condition branch determines whether the individual belongs to the population with fitness ranking in the top half.

2) Wrapping Food: This stage simulates the contraction pattern of vein structures during food search. When slime mould veins contact higher food concentrations, stronger biological oscillation waves are generated, faster cytoplasm flows, and higher vein cytoplasm viscosity. Equation (5) describes the feedback obtained based on food concentration or quality. Positions with food concentration in the upper half of all search results receive positive feedback, while lower positions receive negative feedback, utilizing fitness calculation information to adjust weights at different positions and modulate the population's search behavior. The random number r ensures individual diversity.

Based on the above theory, the position update formula is:

$$X_i(t+1) = \begin{cases} r \cdot (UB - LB) + LB, & r_p < z \\ X_i(t) + v_c \cdot (W \cdot X_A(t) - X_B(t)), & r_p \geq z \end{cases}$$

where UB and LB represent the upper and lower boundary constraints of the search space, and r represents a random number in $[0, 1]$. The transition probability z was experimentally verified by Shimin Li to yield the best algorithm performance when set to 0.03, so this paper also fixes z at 0.03.

3) Oscillation Reaction: Slime mould adjusts cytoplasm flow in veins through biological oscillation reactions to search for optimal food positions. Parameters v_b, v_c, and W in SMA simulate this oscillation reaction. The behavior of selecting different oscillation modes based on fitness simulates how individuals choose their next actions based on food found at other locations. The dynamic adjustment of v_c simulates vein diameter changes, while the linear variation of v_b simulates changes in the retention degree of historical information.

2.1 Opposition-Based Learning Strategy

Tizhoosh [?] first proposed the concept of Opposition-Based Learning (OBL), which is widely applied in swarm intelligence algorithm improvement strategies and has many extensions and variants. Opposition-based learning strategies help expand search space, accelerate algorithm convergence, and maintain population diversity. The core mechanism of OBL is to simultaneously evaluate the fitness of both original and opposite populations and select superior individuals to form a new population for subsequent calculations. The mathematical expression for traditional opposite point generation is:

$$OX_j = LB_j + UB_j - x_j$$

Quasi-Opposition-Based Learning (QOBL) [?] and Quasi-Reflection-Based Learning (QRBL) [?] are two OBL variants. Their mathematical expressions are:

$$OQ_j = \mathrm{rand}\left(\frac{LB_j + UB_j}{2}, LB_j + UB_j - x_j\right)$$

$$QR_j = \mathrm{rand}\left(x_j, \frac{LB_j + UB_j}{2}\right)$$

Reference [?] used parameters from the Harris Hawks Optimization (HHO) algorithm to control the implementation of different search behaviors in the population through quasi-opposite and quasi-reflection processes. It demonstrated that QOBL can avoid the low precision caused by excessive position transformation in traditional OBL while relatively expanding search space to accelerate convergence. QRBL can expand the search space near individual positions and improve population search precision. [Figure 1: see original paper] shows the search spaces of the three OBL variants in one-dimensional space.

Standard swarm intelligence algorithm frameworks have clear distinctions between exploration and exploitation behaviors. Using a single opposition strategy generates opposite populations with uniform characteristics, resulting in different auxiliary effects on exploration and exploitation phases, with one phase being relatively less effective. To construct opposite populations more effectively, this paper uses the QOBL process with large-step rough search during SMA's exploration phase and the QRBL process with small-step fine search during exploitation, ultimately generating a comprehensive opposite population to enhance the optimization effect of the opposition learning framework for different search space characteristics.

In SMA, parameter z controls population exploration and exploitation. This paper uses the QOBL process during exploration and QRBL during exploitation, with mathematical expressions:

$$OX_i(t) = \begin{cases} LB + \text{rand} \cdot (UB - LB) - x_i(t), & r_p < z \\ LB + UB - x_i(t), & r_p \geq z \end{cases}$$

where parameter z decreases with iteration rounds, meaning the comprehensive opposite population consists mainly of quasi-opposite individuals in early stages and quasi-reflected individuals in later stages, dynamically balancing the ratio of exploration and exploitation individuals in the opposite population.

2.2 Linear Diversity Measurement

Reference [?] proposed a method for measuring population diversity in evolutionary algorithms:

$$D_t = \frac{1}{N \cdot D} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{i,j}(t) - \bar{x}_j(t))^2}$$

where D is problem dimension, j is current dimension, and N is population size.

In traditional OBL, opposite populations generated in each iteration are compared with original populations to form new populations, which not only significantly increases computational time overhead but also interrupts the convergence process through position resetting, causing instability in multiple independent experiments. Therefore, timely opposition-based population reconstruction is necessary to maximize the effectiveness of opposition learning strategies.

Inspired by reference [?], this paper sets a diversity threshold (DT) to regulate whether the current population needs to generate a new population through the comprehensive opposite population. If calculated population diversity exceeds the threshold, a new population is generated; otherwise, no new population is created. Since SMA inherently maintains population diversity, low diversity may indicate convergence. Experiments show that even when SMA falls into local optima, population diversity does not remain consistently low but oscillates between high and low levels. When diversity exceeds the threshold, the comprehensive opposite population can help the original population expand search space and escape local optima. This mechanism achieves reduced opposition reconstruction with minimal computation, significantly decreasing time overhead. Moreover, since population diversity strongly correlates with convergence status, threshold-based screening enables new populations to correspond to different convergence states, providing assistance when stagnation occurs while avoiding interference during rapid convergence.

2.3 Unscented Sigma Point Guidance Strategy

Reference [?] first proposed using Unscented Transformation (UT) to generate a series of unscented sigma points to help the original population approximate the theoretical optimal position. By modifying the original algorithm's step mechanism, the population's search patterns are enriched, improving algorithm generality for different problems and avoiding stagnation. The mathematical expression for unscented sigma points is:

$$\begin{cases} \xi_0 = \bar{x} \\ \xi_i = \bar{x} + \sqrt{D+\lambda} \cdot \sigma_i, \quad i = 1, 2, \dots, D \\ \xi_i = \bar{x} - \sqrt{D+\lambda} \cdot \sigma_i, \quad i = D+1, D+2, \dots, 2D \end{cases}$$

where $\bar{x}$ is the mean position in dimension D, σ_i is the standard deviation in dimension i, and there are $2D+1$ sigma points. This paper uses the first sigma point (population mean) to guide SMA's exploration phase and the remaining sigma points (new positions calculated from standard deviation and mean) to guide the exploitation phase, helping the algorithm approximate the theoretical optimum in the expanded search space from opposition learning.

Equation (7) is SMA's core search framework. When $r_p < z$, the population exhibits exploration behavior, with the basic movement pattern using partial information from the best position and two random individuals to generate step sizes. This paper replaces X_A and X_B with ξ_i to reduce randomness in step generation, using the mean to guide rapid convergence during exploration. When $r_p \geq z$, the population shows exploitation behavior, with the basic pattern being position decay. This paper replaces it with new positions generated from mean and variance while retaining parameter v_c to ensure some randomness, helping SMA approximate the theoretical optimum during exploitation. A transition probability is set to avoid computing all $2D+1$ sigma points, saving computational overhead. The improved equation (7) is:

$$X_i(t+1) = \begin{cases} \xi_0 + v_b \cdot (W \cdot \xi_i - \xi_j), & r_p < z \\ v_c \cdot \xi_i, & r_p \geq z \end{cases}$$

2.4 Unscented Sigma Point Guided Quasi-Opposite Slime Mould Algorithm

This paper proposes the Unscented Sigma Point Guided Quasi-Opposite Slime Mould Algorithm (UQSMA). First, based on SMA's exploration and exploitation phases, QOBL and QRBL processes generate a comprehensive opposite population. Using different opposition processes during different spatial characteristic search stages helps amplify SMA's behavioral tendencies, strengthening its inherent characteristics and accelerating algorithm startup. Second, linear diversity measurement calculates population diversity, and real-time diversity determines whether to integrate the comprehensive opposite population with the original population to generate a new population, ensuring algorithm accuracy, saving computational overhead, and improving stability. Finally, unscented sigma points replace SMA's basic movement pattern, guiding the population toward the theoretical optimum, enriching search paths, maintaining optimal solution updates, and avoiding stagnation.

2.5 Algorithm Steps of the Improved Algorithm

The flowchart of UQSMA is shown in [Figure 2: see original paper].

2.6 Time Complexity Analysis

Assuming population size N, search space dimension D, and maximum iterations T, the time complexity of original SMA mainly comes from initialization, fitness calculation, and position update: $O(T \cdot N \cdot D)$. UQSMA consists of two improvement strategies: the comprehensive opposition learning framework (QOBL and QRBL) and the unscented sigma point guidance mechanism. The opposition process requires generating N opposite individuals from N individuals, increasing fitness calculation for $2N$ individuals, sorting, and generating new populations, with complexity $O(T \cdot (2N \cdot D + N \log N))$. The unscented sigma point generation does not increase time complexity. Thus, UQSMA's time complexity is:

$$O(T \cdot (2N \cdot D + N \log N + N \cdot D)) = O(T \cdot N \cdot (3D + \log N))$$

In actual operation, since fitness threshold judgment determines whether to perform the comprehensive opposition process, the total time complexity varies across different problems. The $2N$ term in practice is smaller, making this the theoretical maximum complexity.

2.7 Convergence Analysis

UQSMA is essentially a random search algorithm whose global convergence can be proven using probability measure theory. The global convergence criteria and theorem [?] state that an algorithm must satisfy two conditions:

Condition 1: For the solution space S and algorithm A, the optimal solution set R_ϵ is defined as $R_\epsilon = \{x \in S \mid f(x) < f(x^*) + \epsilon\}$, where x^* is the global optimum. The algorithm must satisfy:

$$\prod_{k=0}^{\infty}(1 - \mu_k(A_k(R_\epsilon))) = 0$$

where μ_k is the probability measure of algorithm A at iteration k.

Condition 2: For any Borel subset B of S with positive measure, the probability of searching in B must be positive:

$$\sum_{k=0}^{\infty}\mu_k(B) = \infty$$

Lemma 1 (Global Convergence Theorem for Random Optimization Algorithms [**?**]): If function f is measurable, search space S is a measurable subset of $\mathbb{R}^n$, and algorithm A satisfies Conditions 1 and 2, then algorithm A converges globally.

Theorem 1: UQSMA satisfies Condition 1.
Proof: The optimal position update in UQSMA includes both original SMA position update and quasi-opposite/quasi-reflection position update. The algorithm can be abstracted as:

$$G_{t+1} = \begin{cases} g(X_i(t)), & f(g(X_i(t))) \geq f(G_t) \\ h(X_i(t)), & f(h(X_i(t))) < f(G_t) \end{cases}$$

where $g(\cdot)$ represents the original SMA update, $h(\cdot)$ represents the quasi-opposite/quasi-reflection update, and G_t is the global best at iteration t. The global best fitness is monotonically non-increasing and gradually approaches the infimum of the solution space, thus satisfying Condition 1.

Theorem 2: UQSMA satisfies Condition 2.
Proof: For algorithm A with population size N to satisfy Condition 2, the sample space must contain R_ϵ, i.e., $\bigcup_{i=1}^{N} S_{i,k} \supseteq R_\epsilon$, where $S_{i,k}$ is the support set of individual i at iteration k.

For the quasi-opposite and quasi-reflection update mechanism in UQSMA, the sample space after update reaches $S_{i,k} \cup O_{i,k}$, where $O_{i,k}$ is the sample space from opposition learning. The probability of reaching any point in $O_{i,k}$ is greater than zero. As iteration increases, each $S_{i,k}$ gradually expands, so there exists an integer K such that when $k > K$, $\bigcup_{i=1}^{N} S_{i,k} \supseteq R_\epsilon$. Similarly, the original SMA update also expands sample space. Therefore, Condition 2 is satisfied.

In conclusion, by Lemma 1, UQSMA has global convergence. Moreover, since quasi-opposite and quasi-reflection processes generate new sample spaces different from the original SMA sample space, this strategy expands the population's search space per iteration, thus accelerating convergence compared to original SMA.

3 Simulation Experiments and Results Analysis

The experimental environment is Windows 10 64-bit OS, AMD Ryzen 7 5800H CPU @ 3.20GHz, 16GB RAM, with algorithms implemented in MATLAB 2020b.

Experiment 1: To fully verify UQSMA's solution accuracy and stability on function optimization problems, it is compared with other algorithms on CEC2017 test functions. Comparison algorithms include the latest improved SMA (AFSMA) [?], Dynamic Opposition Learning SMA (DOLSMA), Elite Opposition-based improved Whale Optimization Algorithm (EGSWOA) [?], original SMA [?], Harris Hawks Optimization (HHO) [?], and classic Whale Optimization Algorithm (WOA) [?].

For fairness, population size is set to 30, maximum iterations to 500, and each algorithm runs independently 30 times. Parameter settings for comparison algorithms follow their original papers, as shown in .

lists the 29 CEC2017 benchmark functions used, including Unimodal (UM), Multimodal (MM), Hybrid (H), and Composition (C) types with dimensions of 10, 30, 50, and 100. The standard 30-dimensional setting is used here; Function 02 is typically excluded due to high-dimensional instability.

shows experimental results comparing UQSMA with other algorithms. Overall, UQSMA achieves better mean solution accuracy than all comparison algorithms on 21 of 29 CEC2017 functions, demonstrating that the improvement strategies enable sustained search effectiveness and efficiency, overcoming original SMA's search stagnation limitations. UQSMA's solution accuracy is higher than DOLSMA, indicating that the combination of QOBL and QRBL is more effective than traditional dynamic opposition learning for SMA. Additionally, UQSMA's variance is generally lower than other algorithms, proving that the strategy combination effectively improves stability and positively impacts final solution accuracy and robustness.

For further validation, statistical tests are performed using Wilcoxon rank-sum test at 5% significance level [?]. shows pairwise comparison results, where "+" indicates UQSMA is statistically better, "=" indicates no significant difference, and "-" indicates worse performance. The results show UQSMA is never worse than DOLSMA, HHO, or WOA; it is only inferior to EGSWOA on 1 function and similar on 1 function; it is only inferior to AFSMA and original SMA on a few functions. Compared to original SMA, nearly half the test functions show no significant distribution difference, indicating that UQSMA preserves original SMA's search characteristics through diversity-threshold-based reconstruction while achieving better final accuracy, proving its superior sustained search performance on CEC2017 benchmarks.

Convergence curves and boxplots visualize algorithm performance. [Figure 3:

see original paper] shows convergence curves for selected functions (log-scale y-axis for clarity). [Figure 4: see original paper] shows boxplots with means. UQSMA exhibits fast early convergence and sustained late search, demonstrating that QOBL and QRBL expand search space in early-middle stages to quickly find better directions, while unscented sigma point guidance helps maintain refinement in later stages, avoiding stagnation. UQSMA shows smaller variance across repeated runs, indicating that the diversity-based reconstruction strategy maintains stability despite increased diversity from opposition learning and new randomness from sigma points.

presents Mean Absolute Error (MAE) ranking [?] across all 29 functions:

$$MAE = \frac{1}{N_f} \sum_{i=1}^{N_f} |f_i - f_i^*|$$

where N_f is the number of functions, f_i is the mean result on function i, and f_i^* is the theoretical optimum. UQSMA ranks first with significantly lower MAE, comprehensively demonstrating its effectiveness and competitiveness.

Experiment 2: To verify UQSMA's improvement effectiveness, it is compared with incomplete algorithms: Unscented Sigma point guided SMA (USMA), Quasi-Opposition SMA (QOSMA), and Quasi-Reflection SMA (QRSMA). shows MAE rankings. UQSMA ranks first, QOSMA second (better than original SMA), while USMA and QRSMA rank third and fourth (worse than original SMA). This shows that unscented sigma points alone introduce excessive randomness reducing accuracy, and fixed continuous quasi-opposition processes also degrade search performance. However, the combination of QOBL/QRBL with unscented sigma points yields optimal performance: the comprehensive opposition process and timely reconstruction compensate for sigma point randomness, improving stability, while sigma points provide refined search behavior to overcome the blindness of expanded search space, preventing stagnation. The strategies complement each other, avoiding individual defects and achieving better combined effects.

Experiment 3: To validate UQSMA's effectiveness in real-world engineering, it is applied to the car side impact design optimization problem [?]. This problem considers design variables including B-pillar inner/outer reinforcement thickness, floor inner/outer thickness, crossbeam thickness, door beam thickness, door belt reinforcement thickness, roof rail thickness, barrier height, and impact position. It features higher dimensionality and more complex constraints than classical engineering problems. The objective function, constraints, and variable ranges are:

Objective function and constraints:

$$\begin{aligned} f(X) &= 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \\ \text{s.t.} \quad & g_1(X) = 1.16 - 0.3717x_{2x}4 - 0.00931x_{2x_{10}} - 0.484x_{3x}9 + 0.01343x_{6x_{10}} \leq 0 \\ & g_2(X) = 28.98 + 3.818x_3 - 4.2x_{1x}2 + 0.0207x_{5x_{10}} + 6.63x_{6x}9 - 7.7x_{7x}8 + 0.32x_{9x_{10}} \leq 0 \\ & \ldots \\ & x_1 \in [0.5, 1.5], x_8 \in [-0.5, 0.5], x_9 \in [0.19, 0.345], x_{10} \in [-30, 30] \end{aligned}$$

Population size is 30, maximum iterations 500, with 30 independent runs. Comparison algorithms and parameters match Experiment 1 (EGSWOA excluded due to different boundary handling). Results are shown in . While UQSMA's best value is slightly worse than original SMA and its improvements, its mean value across multiple runs outperforms all comparison algorithms with smaller variance, demonstrating better stability and robustness in practical optimization problems.

---

4 Conclusion

To address SMA's limitations, this paper proposes UQSMA using combined quasi-opposite/quasi-reflection learning and unscented sigma point guidance strategies, applying it to CEC2017 test functions and real-world problems. The improved opposition learning framework integrates different opposition generation processes into SMA's exploration and exploitation phases, expanding search space, accelerating early convergence, improving stability, and demonstrating universal applicability across diverse test functions. During iterations, unscented sigma points guide the population's basic movement toward the theoretical optimum, enriching search paths and avoiding stagnation. For comprehensive evaluation, Wilcoxon rank-sum test, MAE metrics, and car side impact engineering application comparisons demonstrate the proposed algorithm's solution accuracy, robustness, and practicality.

Future research will further improve and test SMA for more diverse and complex problems, applying it to machine learning model parameter optimization and portfolio optimization to further realize practical swarm intelligence applications.

References

[1] Li Shimin, Chen Huiling, Wang Mingjing, et al. Slime mould algorithm: A new method for stochastic optimization [J]. Future Generation Computer Systems, 2020, 111: 300–323.

[2] Mostafa M, Rezk H, Aly M, et al. A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel [J]. Sustainable Energy Technologies and Assessments, 2020, 42(November): 100849.

[3] Abdel-Basset M, Mohamed R, Chakrabortty R K, et al. An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection [J]. Computers & Industrial Engineering, 2021, 153(November 2020): 107078.

[4] Abdel-Basset M, Chang V, Mohamed R. HSMA_{WOA}: A hybrid novel Slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images [J]. Applied Soft Computing Journal, 2020, 95: 106642.

[5] Naik M K, Panda R, Abraham A. Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm [J]. Journal of King Saud University-Computer and Information Sciences, 2020.

[6] Rizk-Allah R M, Hassanien A E, Song D. Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites [J]. ISA transactions, 2022, 121: 191-205.

[7] Jia Heming, Liu Yuxiang, Liu Qingxin, et al. Hybrid algorithm of slime mould algorithm and arithmetic optimization algorithm based on random opposition-based learning [J/OL]. Journal of Frontiers of Computer Science and Technology: 1-12 [2021-12-02]. http://kns.cnki.net/kcms/detail/11.5602.TP.20210714.1817.006.html.

[8] Guo Yuxin, Liu Sheng, Zhang Lei, et al. Elite opposition-based learning quadratic interpolation slime mould algorithm [J/OL]. Application Research of Computers: 1-7 [2021-12-02]. https://doi.org/10.19734/j.issn.1001-3695.2021.02.0175.

[9] Liu Chenghan, He Qing. Adaptive artificial bee colony slime mould algorithm with improved crossover operator [J/OL]. Journal of Chinese Computer Systems: 1-8 [2021-12-02]. http://kns.cnki.net/kcms/detail/21.1106.TP.20211114.1427.002.html.

[10] Örnek B N, Aydemir S B, Düzenli T, et al. A novel version of slime mould algorithm for global optimization and real world engineering problems: Enhanced slime mould algorithm [J]. Mathematics and Computers in Simulation, 2022, 198: 253-288.

[11] Yu Caiyang, Heidari A A, Xue Xiao, et al. Boosting quantum rotation gate embedded slime mould algorithm [J]. Expert Systems with Applications, 2021, 181: 115082.

[12] Abualigah L, Diabat A, Mirjalili S, et al. The arithmetic optimization algorithm [J]. Computer methods in applied mechanics and engineering, 2021, 376: 113609.

[13] Tizhoosh H R. Opposition-based learning: a new scheme for machine

intelligence [C]// International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). IEEE, 2005, 1: 695-701.

[14] Rahnamayan S, Tizhoosh H R, Salama M M A. Quasi-oppositional differential evolution [C]// 2007 IEEE congress on evolutionary computation. IEEE, 2007: 2229-2236.

[15] Ewees A A, Abd Elaziz M, Houssein E H. Improved grasshopper optimization algorithm using opposition-based learning [J]. Expert Systems with Applications, 2018, 112: 156-172.

[16] Fan Qian, Chen Zhenjian, Xia Zhanghua. A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems [J]. Soft Computing, 2020, 24(19): 14825-14843.

[17] Wineberg M, Oppacher F. The underlying similarity of diversity measures used in evolutionary computation [C]// Genetic and evolutionary computation conference. Springer, Berlin, Heidelberg, 2003: 495-502.

[18] Solis F J, Wets J B. Minimization by Random Search Techniques [J]. Mathematics of Operations Research, 1981, 6(1): 19-30.

[19] Choi T J, Togelius J, Cheong Y G. A fast and efficient stochastic opposition-based learning for differential evolution in numerical optimization [J]. Swarm and Evolutionary Computation, 2021, 60: 100797.

[20] Guo Wenyan, Xu Peng, Dai Fengqun, et al. Improved harris hawks optimization algorithm based on random unscented sigma point mutation strategy [J]. Applied Soft Computing, 2021: 108012.

[21] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in engineering software, 2016, 95: 51-67.

[22] Xiao Ziya, Liu Sheng. Study on elite opposition-based golden-sine whale optimization algorithm and its application of project optimization [J]. Acta Electronica Sinica, 2019, 47(10): 2177-2186.

[23] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications [J]. Future generation computer systems, 2019, 97: 849-872.

[24] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms [J]. Swarm and Evolutionary Computation, 2011, 1(1): 3-18.

[25] Li Shouyu, He Qing, Du Nisuo. Butterfly optimization algorithm for chaotic feedback sharing and group synergy [J/OL]. Journal of Frontiers of Computer Science and Technology: 1-12 [2021-05-19]. http://kns.cnki.net/kcms/detail/11.5602.TP.20210128.1109.014.htm

[26] Zhang Xinmin, Jiang Yun, Liu Shangwang, et al. Hybrid coyote optimization algorithm with grey wolf optimizer and its application to clustering optimization [J/OL]. Acta Automat. Sin.: 1-17 [2021-05-19]. https://doi.org/10.16383/j.aas.c190617.

[27] Youn B D, Choi K K, Yang R J, et al. Reliability-based design optimization for crashworthiness of vehicle side impact [J]. Structural and Multidisciplinary Optimization, 2004, 26(3): 272-283.

Note: Figure translations are in progress. See original paper for figures.