An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.16, No.1, pp.17-39 (2026)
https://doi.org/10.36922/IJOCTA025280120

RESEARCH ARTICLE

Multi-graph network population evolutionary optimization algorithm with migration and best hunter crossover strategies for cross-field applications

Zhaoyang Lian and Bailu S*

School of Systems Science, Beijing Normal University, Beijing, China
lianzhaoyang@bnu.edu.cn, bailusi@bnu.edu.cn

ARTICLE INFO

Article History:
Received: July 7, 2025
Revised: August 27, 2025
Accepted: September 22, 2025
Available Online: October 29, 2025

Keywords:
Swarm intelligence algorithm
Multi-graph network
Evolutionary optimization algorithm
Cross field applications
Heuristic algorithm

AMS Classification 2010:
26A33; 34A08; 35H15; 34K50
47H10; 60H10

ABSTRACT

Although swarm intelligence optimization algorithms, such as simulating biological bionic behaviors or natural laws have been relatively mature, there are relatively few algorithms considering multi-graph network evolutionary behaviors and the algorithms combining graph network structure with biomimetic behaviors are worth studying. In this paper, a multi-graph network population optimization algorithm with migration and best hunter crossover strategies was proposed for cross-field applications. The gorgeous central radial multi-graph matrices were rotated and deformed to change different formations while hunting prey. The global graph population of a strong group was adopted to explore prey in a large range and the local graph population of a weak group was adopted to guard food in a small range near their prey or home. The migration strategy was aimed at reducing overexploitation by hunters and the best hunter crossover strategy was aimed to retain the excellent genes of the best hunter while also preserving the vitality of new individuals. Furthermore, the proposed algorithm was applied to open-source function optimization problems, and extended to four engineering applications and design problems such as multi-sector aviation scheduling, flexible workshop scheduling optimization, unmanned aerial vehicle routing optimization of oil plants in three-dimensional maps, and power system bus type optimization achieving competitive results.

1. Introduction

Swarm intelligence algorithms are bionic random probability heuristic search algorithms based on the study on collective behaviors in decentralized and self-organizing systems. In each iteration of swarm intelligence algorithms, multiple individuals jointly explore, and the defects of a single individual are not expanded to the whole population. Therefore, compared to the algorithm with a single individual update, multiple individuals are easier to escape the local optimal solution.

They can be roughly divided into biological behavior simulation algorithms and natural law simulation algorithms according to the different types of simulated behaviors. The biological behaviors of simulation algorithms can be divided into human and animal behaviors. Although these algorithms are similar in to the way they constantly update the current individuals through iteration to make them gradually move toward the optimal solution, they update current individuals in different ways. Many swarm intelligence algorithms integrate mathematical functions or network graph with biological bionic behaviors or natural laws. The update rules for different swarm intelligence algorithms are shown in Figure 1. Some algorithms update speed or position according to different shapes and specific rules, such as updates of linear, chain, sine/cosine, heap structure, arithmetic operator, parabolic curve,

*Corresponding Author

cardioid curve, lightning attachment shape, vertical normal vector, spiral curve, random type within radius, rotating, contradictory splitting, tumbling, and golden sine. Some algorithms update the acceleration and velocity according to the force relationship such as Archimedes Law, Henry's Law, the molecular dynamics model, and the oscillation circuit. Some algorithms are updated according to the network graph, such as the updates of the Lichtenberg graph.

Although swarm intelligence optimization algorithms, such as simulating biological bionic behaviors or natural laws have been relatively mature,[1] there are relatively few algorithms considering multi-graph network evolutionary behaviors and the algorithms that combine graph network structure with biomimetic behaviors are worth studying. The Lichtenberg algorithm[2] only refers to the basic Lichtenberg graph, which may lack some agility in certain hunting situations. In this paper, a multi-graph network population evolutionary optimization algorithm was proposed for cross-field applications. The algorithm dynamically simulates the population predation process through a graph network. Considering that the population will change different formations while hunting prey, sampling graph matrix was rotated and deformed in the algorithm. Strong groups in the population explore prey in a large range, and weak groups develop and guard food in a small range near their prey or home. After updating individual positions based on multi graph networks, the exponential migration strategy is adopted to reduce overexploitation by hunters, and the best hunter crossover strategy is introduced to retain the excellent genes of the best hunter while preserving the vitality of new individuals. Therefore, the multi-scale updating strategy of global graph population and the local graph population were adopted. Moreover, there is relatively more research in a single field[34] or some cross fields,[5] but the research on general algorithms for other cross fields is also important. To improve the usefulness of the algorithm, the proposed bionic multi-graph network optimization algorithms were extended to multi-sector aviation scheduling (MSAS), Flexible workshop scheduling (FWS), Unmanned aerial vehicle routing optimization (UAVRO) of oil plant in three-dimensional (3D) map, and Power system bus types optimization (PSBTO).

The rest of this paper is organized as follows. The related works are described in **Section 2**. The proposed multi-graph network population evolutionary optimization algorithms with migration and best hunter crossover strategies for cross-field applications are described in **Section 3**. The algorithm implementation and the experiment results compared with other swarm intelligence algorithms are described in **Section 4**. The main findings of this study are concluded in **Section 5**.

2. Related methods

The overview of swarm intelligence algorithms is shown in Figure 2.

2.1. Swarm intelligence algorithm

2.1.1. Biological behavior simulation algorithm

The smell agent optimization[6] simulates the behaviors of odor molecules and human agents. The sniffing model simulates the ability of the agent's olfaction to perceive odor molecules when odor molecules diffuse from the odor sources to the agents. The speeds and individual positions are updated by simulating the effects of temperature and mass on the kinetic energy of odor molecules. Tracking model simulates the traceability ability of agents to track odor molecules. The pathfinder algorithm[7] simulates the behaviors of pathfinders and followers. The pathfinders are the leaders of the followers and the explorers of the global search directions. The Sperm swarm optimization mainly simulates the process of egg fertilization by sperm based on fertilization activity. In the algorithm, temperature, and pH value affect the update of individual speed. The Brain storm optimization [9] simulates the discussion behaviors of human brainstorming, adopts clustering ideas to search for local optimal values, and obtains global optimal values by Comparing local optimal values.[10] The social network Search optimization[11] simulates the imitations, dialogues, debates, and innovation emotions when users express their opinions in social network. The Gaining sharing knowledge[12] optimization simulates the process of acquiring and sharing knowledge in primary homogeneous acquisition stage and Advanced hierarchical acquisition stage.[13] The cooperation search algorithm[14] is inspired by the cooperative behaviors of modern enterprise teams, which simulates the interaction behaviors of the chairman, supervisors, board of directors, and staff. This algorithm updates individuals by referencing the average of the best and elite individuals, as well as the current individual's mirror point. The heap based optimization algorithm[15] simulates the tree hierarchical heap structure of the company, forms the interaction between individuals with the concept of

Figure 1. The update rules for different swarm intelligence algorithms

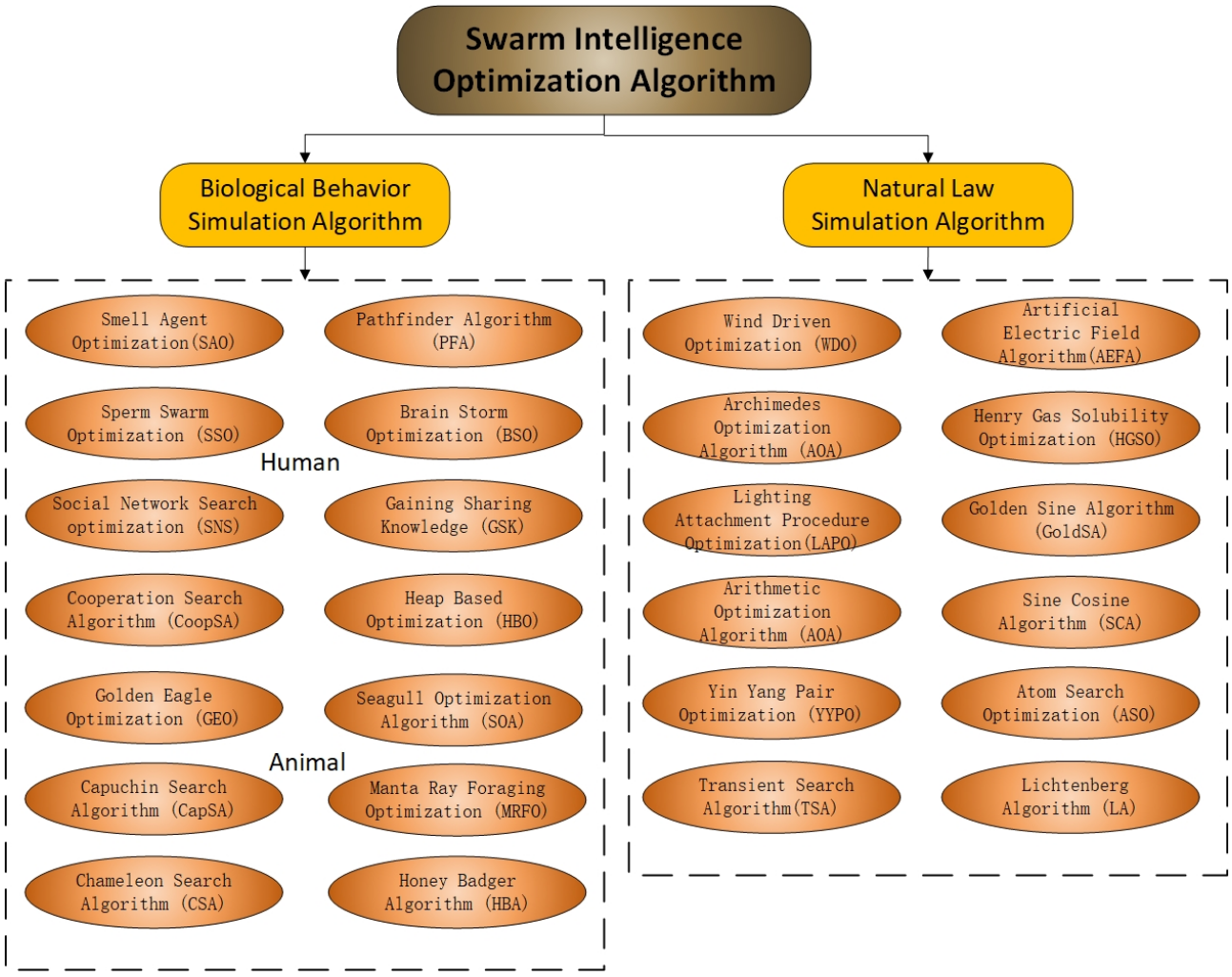


Figure 2. The overview of swarm intelligence algorithms

heap, and constructs three-mathematical models to develop new solutions.? The golden eagle optimization (GEO)[16] simulates the interactive hunting between the attack vector toward the prey and the circular cruise normal vector. They show more tendency to cruise and look for prey in the early stage of hunting, but they show more tendencies to attack in the final stage. The Seagull optimization algorithm[17] simulates the spiral-shaped predatory behaviors of seagulls that often use bread crumbs to attract fish, and make sounds to attract underground earthworms by their feet. The Capuchin search algorithm[18] simulates the behaviors of capuchin monkeys, such as jumping between the trees, jumping among the ground, walking on the ground, normal walking, local climbing, and chasing. The Manta ray foraging optimization[19] simulates chain foraging, spiral foraging, and Tumbling foraging behaviors of manta rays.[20] The Chameleon search algorithm[21] simulates the behaviors of a chameleon, which rotates nearly 360° to locate prey and captures prey with a high-speed launched sticky tongue according to the motion equation. The Honey badger algorithm[22] mainly simulates the cardioid curve mining behavior and honey gathering behavior of honey badgers for optimization.

2.1.2. Natural law simulation algorithm

The wind-driven optimization[23] simulates the forced movements of air particles in the atmosphere, and the position is updated by combination with Newton's Second Law and the ideal gas equation of state.[24] The artificial electric field algorithm[25] simulates the movements of charged particles in the electrostatic field, which attract or repel each other under the action of the electric force. Particles with large charges can attract all other particles with low charges closer to them. The Archimedes optimization algorithm[26] simulates the Archimedes' buoyancy law, which updates the density, volume, buoyancy, and acceleration of the immersed object according to the collision with other adjacent objects. The Henry gas solubility optimization[27] simulates Henry's Law, which deals with the relationship among temperature, liquid type, volume and the amount of gas dissolved in the liquid [28]. The lighting attachment procedure optimization[29] simulates the lightning upward leaders in nature to approach the optimal and downward leaders to increase the exploration connection process. The golden sine algorithm[30] simulates the unit circle scanning of the sine function and reduces the search space by the golden section to approximate the optimal solution for updating the search space of the candidate solution.[31] The arithmetic optimization algorithm[32] simulates the global search through a multiplication strategy and a division strategy to enhance the global optimization ability of the algorithm and overcome premature convergence. The addition strategy and subtraction strategy are used in the development stage to reduce the dispersion of the solution.? The sine cosine algorithm[33] fluctuates outward or toward the direction of the optimal solution based on the mathematical model of sine and cosine.? The yin yang pair optimization (YYPO)[34] achieves balance through the coordination and supplement between the two contradictory behaviors and updates the solution by one-way splitting or D-direction splitting.[35] The atom search optimization[36] simulates the phenomenon that atoms move due to the mutual attraction and repulsion forces under the system constraints based on the molecular dynamics model.[37] The transient search algorithm[38] is inspired by the oscillation circuit. When the switch changes the transient state, the capacitor or inductor needs time to charge or discharge through the first circuit and the second-order circuit until it reaches the steady-state value. The Lichtenberg algorithm[2] is an algorithm that adopts the Lichtenberg graph with branches and feathers to search and update.

2.2. Lichtenberg algorithm

The Lichtenberg pattern, a sculpture about catching lightning discharge pattern, was first created by the German physicist Lichtenberg. The LA[2] is a natural law simulation algorithm for searching and updating the Lichtenberg pattern, which has branches with a radiating center that looks like snowflakes or feathers. The current best individual is at the center, and the individual is updated by reference to the rotated random sampling Lichtenberg pattern. Although Lichtenberg pattern-based algorithms have achieved some results, algorithms that combine other graphs and biomimetic strategies are still worth studying.

3. Multi-graph network population optimization with migration and best hunter crossover strategies

The proposed framework of the multi-graph network population optimization algorithm (MGNPOMC) with migration and best hunter crossover strategies (MC) for cross-fields applications is shown in Figure 3. Considering too many subgraphs, only three representative subgraphs of MGNPOMC1, MGNPOMC3, and MGNPOMC4 are presented in Figure 3, while the other three

subgraphs of MGNPOMC2, MGNPOMC5, and MGNPOMC6 are represented by ellipses and described in Figures 4, 5, and 6. In the migration strategy, the MGNPOMC4 representative subgraphs is represented in Figure 3, while the other five subgraphs were also used in the experiment. The proposed framework includes two parts: bionic multi-graph network population optimization algorithm and cross field applications.

(i) The MGNPOMC was obtained by referring to Julia graphs with different shapes and scales. By reference to the population Julia graph with a central radiation shape, prey was taken as the center and random sampling of different patterns was used as the basis graphs, which rotated and deformed randomly. The positions of candidate individuals were updated according to the generated graph library with global and local scales. After scaling and rotating in the local and global population graph network optimization, the migration strategy was adopted to reduce overexploitation by hunters, and the best hunter crossover strategy was introduced to retain the excellent genes of the best hunter while preserving the vitality of new individuals.

(ii) The proposed bionic MGNPOMC algorithm was extended to cross-field application and design problems. The algorithm was applied to open-source function optimization problems such as F1-F23, CEC 2019 (cecF1-cecF10), and F1- F50 test functions, and to four engineering applications and design problems such as MSAS, FWS, UAVRO of oil plant in 3D map, and PSBTO, achieving competitive results.

The prey in the algorithm corresponds to the best individual of the current iteration. The change of population formation during predation corresponds to the rotation and deformation of the basis graph. In the process of predator chasing prey, the current optimal solution in the algorithm updates with iteration and approaches to the global optimal solution gradually. The scaling of global and local graph population in each iteration corresponds to the large-scale exploration of groups with strong physique of the population and the small-scale utilization of individuals with poor physique.

The proposed algorithm process is as follows:

3.1. Generation of basis multi-graph

Different original Julia graphs were obtained. Julia graphs with different gorgeous shapes in Figure 4 were obtained by different $C_{cons}(i_g)$ and iteration $N_{iter_g}(i_g)$. The i_g is the index of Julia graph and the N_{iter_g} is the total number of Julia graphs. The C_{cons} and N_{iter_g} in different Julia graphs are shown in **Table** A1. The initial value of complex matrix Z_{new} was divided into real and imaginary parts. The real part remained constant across each column, while the imaginary part increased from –1.5 to 1.5. Conversely, the imaginary part remained constant across each row, while the real part increased from –1.5 to 1.5. If the modulus of the complex element in the matrix $|Z(i_{row}, j_{col})|$ is less than 2, the corresponding element in Z_{data} that the record of the color information of the graph is updated to i_{iter_g}. Different C_{cons} and iterations N_{iter_g} generate different matrices Z_{data}. Figure 4 obtained by different matrices Z_{data} and corlormaps.

$$\begin{cases} \boldsymbol{Z}_{new}(i_{row}, j_{col}) = \boldsymbol{Z}_{his}^{P_{power}}(i_{row}, j_{col}) + C_{cons}(k) \\ \boldsymbol{Z}_{data}(i_{row}, j_{col}) = i_{iter_g}, |\boldsymbol{Z}_{new}(i_{row}, j_{col})| < 2 \end{cases} \quad (1)$$

The multiple different cropped basis Julia graphs were obtained as the basis of individual hunting formations. The graph obtained by Z_{data} was cropped near the center with the max value in the matrix Z_{data} and used as the basis for individual sampling and updating of the network. The brightness of the gray background graph in Figure 5 was controlled by scaling Z_{bri}. The reference sampling graph matrix M was obtained. Matrix M is the position coordinates of elements in basis cropped graphs that meet the conditions $Z_{data}(i_{row}, j_{col}) > 0.5 \cdot max(Z_{data})$. The points in the matrix M are represented by the red scattered points in Figure 5.

$$\boldsymbol{Z}_{bri}(i_{row}, j_{col}) = \frac{200}{max(\boldsymbol{Z}_{data})} \cdot \boldsymbol{Z}_{data}(i_{row}, j_{col}) \quad (2)$$

3.2. Hunting by multi-graph network

Individuals X_{ind} were randomly initialized to X_{init} within the range of L_{lb} to L_{ub}, and the best individual of the population was taken as the starting point X_{start} at iteration t_{iter}.

$$X_{init}(i, j) = L_{lb} + (L_{ub} - L_{lb}) \cdot r_{rnd} \quad (3)$$

The sampling graph matrix was randomly rotated and scaled to obtain the local and global candidate point library according to the graph network.

In MGNPO(01-06) and MGNPOMC(01-06), the 01-06 graph network were adopted. In MGNPOP and MGNPOMCP, graph networks with probability were randomly selected from six

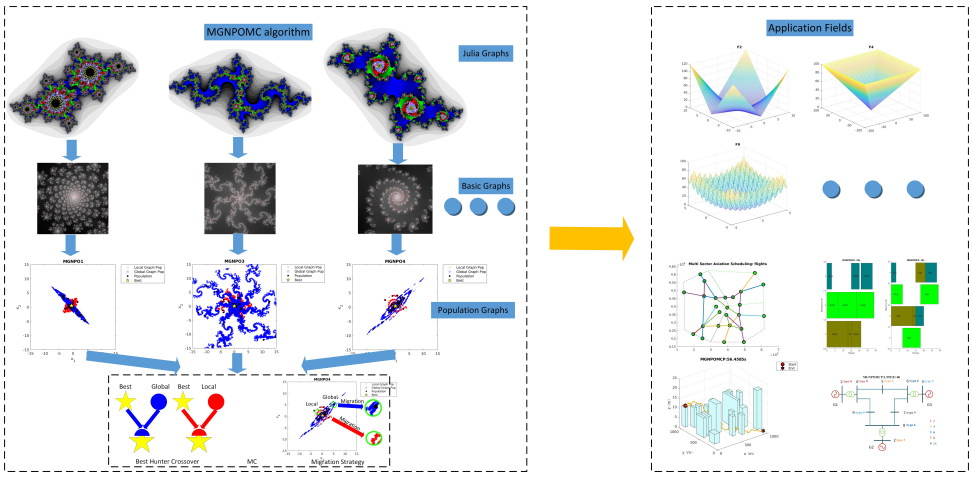


Figure 3. The multi-graph network population optimization with migration and best hunter crossover strategies

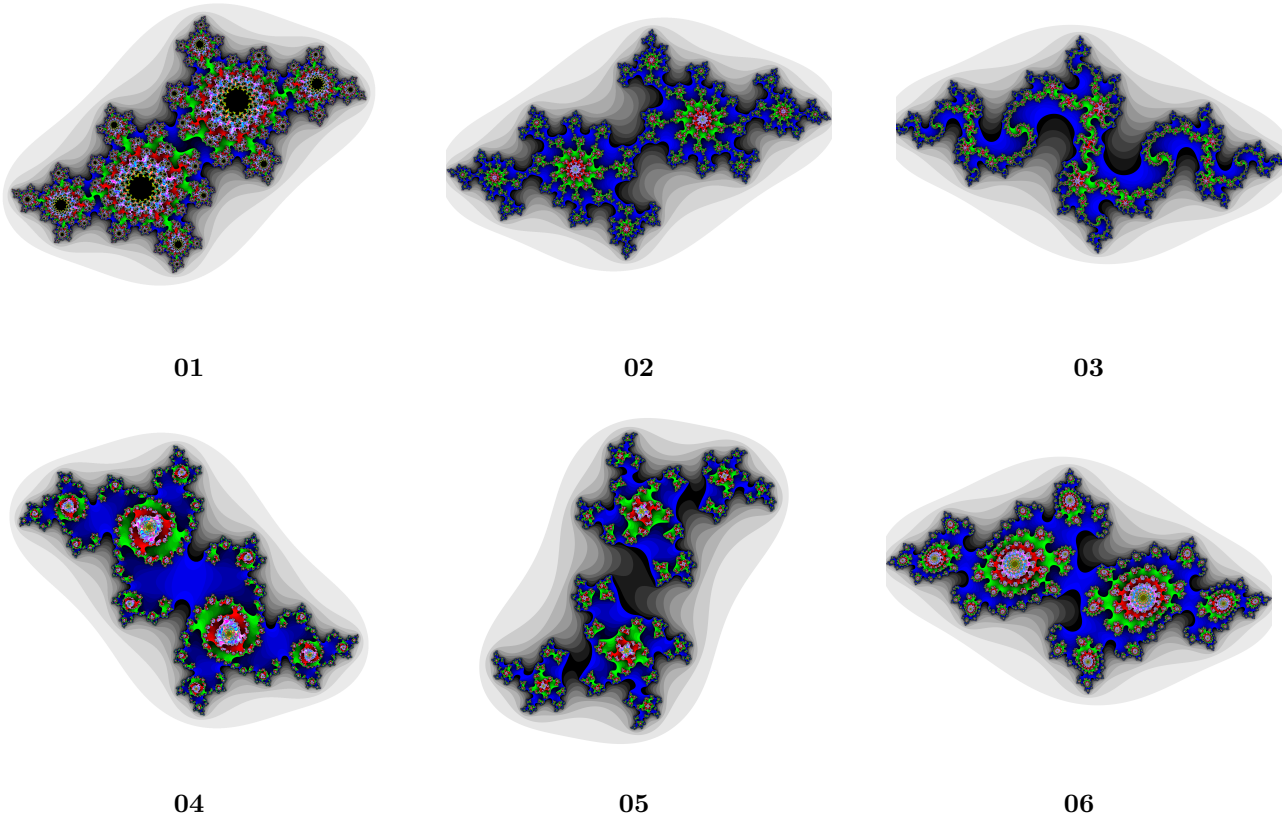


Figure 4. Julia graphs with different gorgeous shapes by different C_{cons} and iterations N_{iter_g}

graphs. The reference sampling graph matrix M was randomly rotated.

$$\begin{cases} \theta_r = \theta_0 \cdot r_{rnd} \\ M(i_p,1) = M(i_p,1) \cdot cos(\theta_r) - M(i_p,2) \cdot sin(\theta_r) \\ M(i_p,2) = M(i_p,1) \cdot sin(\theta_r) + M(i_p,2) \cdot cos(\theta_r) \end{cases} \quad (4)$$

If the individual dimension is greater than 2, the values of other dimensions are updated as follows. Index j_d is an odd number between 1 and N_d.

$$\begin{cases} \gamma = \theta_0 \cdot r_{rnd} \\ X_1(i_p, j_d) = M(i_p, j_d) \cdot cos(\gamma) - M(i_p, 2) \cdot sin(\gamma) \\ X_1(i_p, j_d + 1) = M(i_p, j_d) \cdot sin(\gamma) + M(i_p, 2) \cdot cos(\theta) \end{cases} \quad (5)$$

The values X_1 of each dimension j_d were normalized and scaled. Parameter $X_c(j_d)$ is the center of matrix X_2 in dimension j_d. The N_{X_2} is the number of points in X_2. The middle position of $X_2(\frac{N_{X_2}}{2}, j_d)$ is updated to the center X_c. Library X_{lib} is the candidate point library updated according to the graph network.

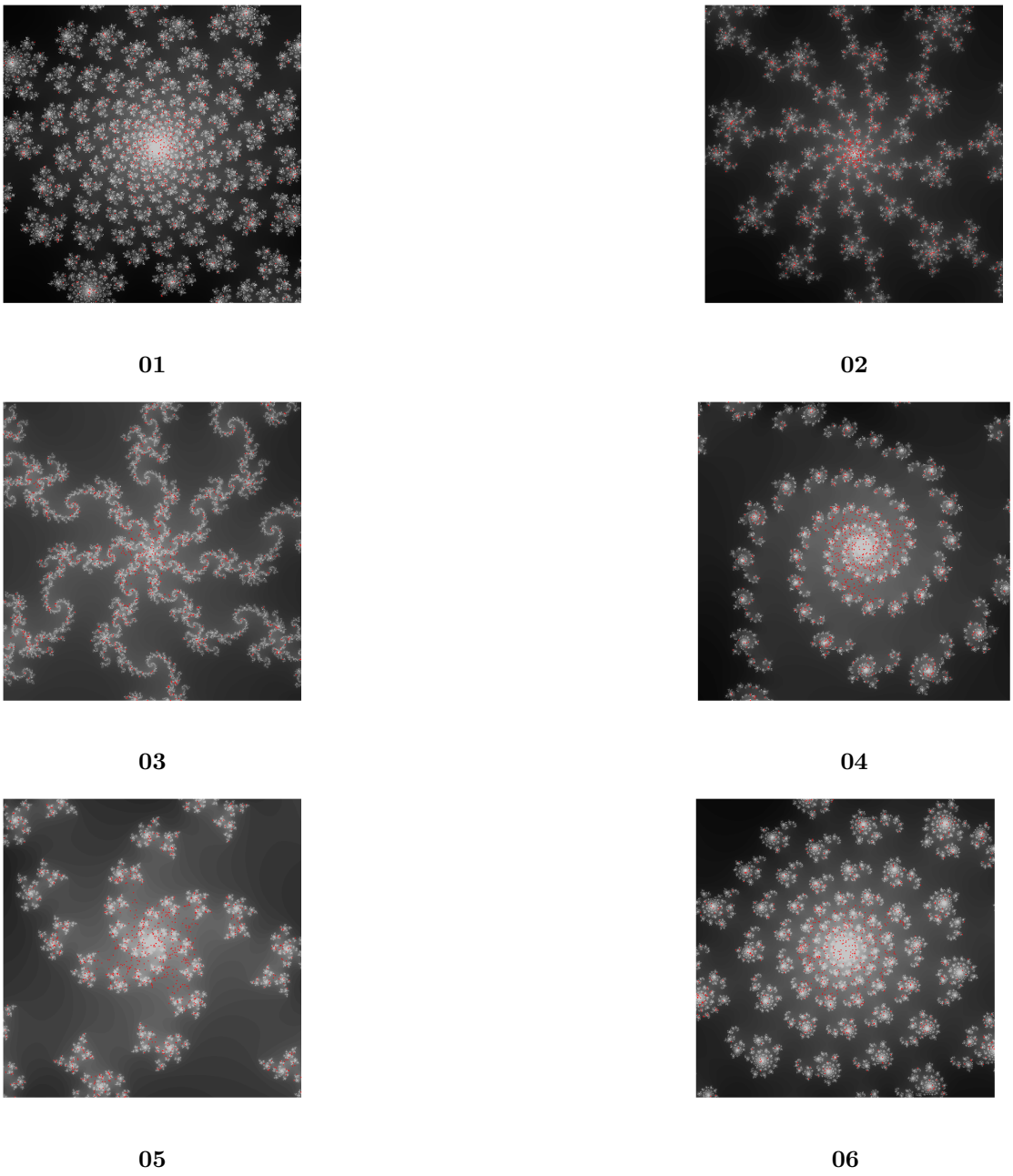


Figure 5. The basis graphs for individual sampling and updating in the network

$$\begin{cases} r_{scale} = 1.2 \cdot r_{rnd} \\ w_{scale}(j_d) = c_{scale} \cdot r_{scale} \cdot \frac{L_{ub}(j_d) - L_{lb}(j_d)}{max(X_1(j_d)) - min(X_1(j_d))} \\ X_2(i_p, j_d) = w_{scale}(j_d) \cdot X_1(i_p, j_d) \\ X_c(j_d) = \frac{max(X_c(j_d)) - min(X_c(j_d))}{2} + min(X_c(j_d)) \\ X_2(\frac{N_{X_2}}{2}, j_d) = X_c(j_d) \end{cases} \tag{6}$$

Coefficient c_{scale} adjusts the local and global update in the graph network. The $c_{scale} = 0.4$ and $c_{scale} = 1$ correspond to local update $X_{localLib}$ of X_{lib} and global update $X_{globalLib}$ of X_{lib}, respectively.

Global and local individuals were randomly sampled to form the updated population. $N_{pop} \cdot 0.4$ global individuals from $X_{globalLib}$ were randomly sampled to obtain $X_{globalS}$, $N_{pop} \cdot 0.6$ global individuals from $X_{localLib}$ were randomly sampled to obtain X_{localS}, and then $X_{globalS}$ and X_{localS} were combined to form the updated population X_{new}. The local and global population graph optimization are shown in Figure 6.

$$\begin{cases} V(j_d) = X_c(j_d) - X_{start}(j_d) \\ X_{lib}(i_p, j_d) = X_2(i_p, j_d) - V(j_d) \\ X_{new} = [X_{globalS}; X_{localS}] \end{cases} \tag{7}$$

3.3. Migration and best hunter crossover strategies

It is necessary to evaluate whether adopting the best hunter crossover and migration strategies is required. In original MGNPO(01-06) and

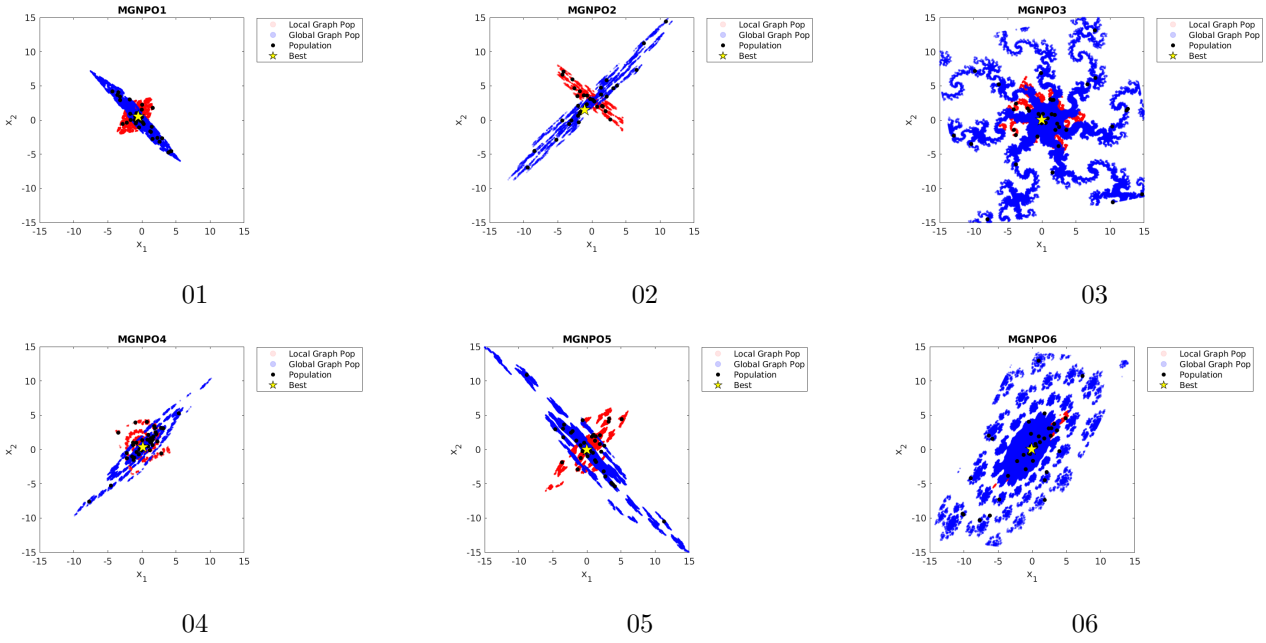


Figure 6. Local and global population graph optimization

MGNPOP, the best hunter crossover and migration strategies are both not adopted. In MGNPOMC(01-06) and MGNPOMCP, the best hunter crossover and migration strategies are both adopted. The best hunter crossover strategy and migration strategy are shown in Figure 7. Although only the MGNPOMC4 representative subgraph is represented in Figure 7, the other five subgraphs were also selected in the experiment.

$$X_{new}(j_{cd}) = \begin{cases} X_{new}(j_{cd}) & r_{rnd} < p_{cd} \\ X_{best}(j_{cd}) & otherwise \end{cases} \quad (8)$$

After scaling and rotating Figure 5 in local and global population graph network optimization, the individuals intersect with the best hunter to retain the excellent genes of the best hunter while also preserving the vitality of new individuals. If $r_{rnd} < p_{cd}$, the current individual retains the genes of the new individual X_{new} in dimension j_{cd}; otherwise, the retain the genes of the best hunter X_{best} in dimension j_{cd}.

When the prey in some areas is overexploited by hunters, some hunters need to undergo an exponentially large-scale migration in search of a new prey. If $r_{rnd} < p_{mig}$, we assume that the individual requires a large-scale migration. If $r_{rnd} < p_{md}$, the j_{md} dimensional value of the migration individual undergoes significant scaling and updating. $c_{migscale}$ is the coefficient of migration scaling. If the new position after migration exceeds the boundary value, try again. If it exceeds the maximum number of migration attempts and still exceeds the boundary value, $X_{new}(j_{md})$ is updated to the boundary value L_{ub} or L_{lb}.

$$X_{new}(j_{md}) = e^{2 \cdot c_{migscale} \cdot (r_{rnd} - 0.5)} \cdot X_{new}(j_{md})) \quad (9)$$

3.4. Pseudo code

The pseudocode of the MGNPOMC algorithm is shown in **Algorithm** 1.

4. Results

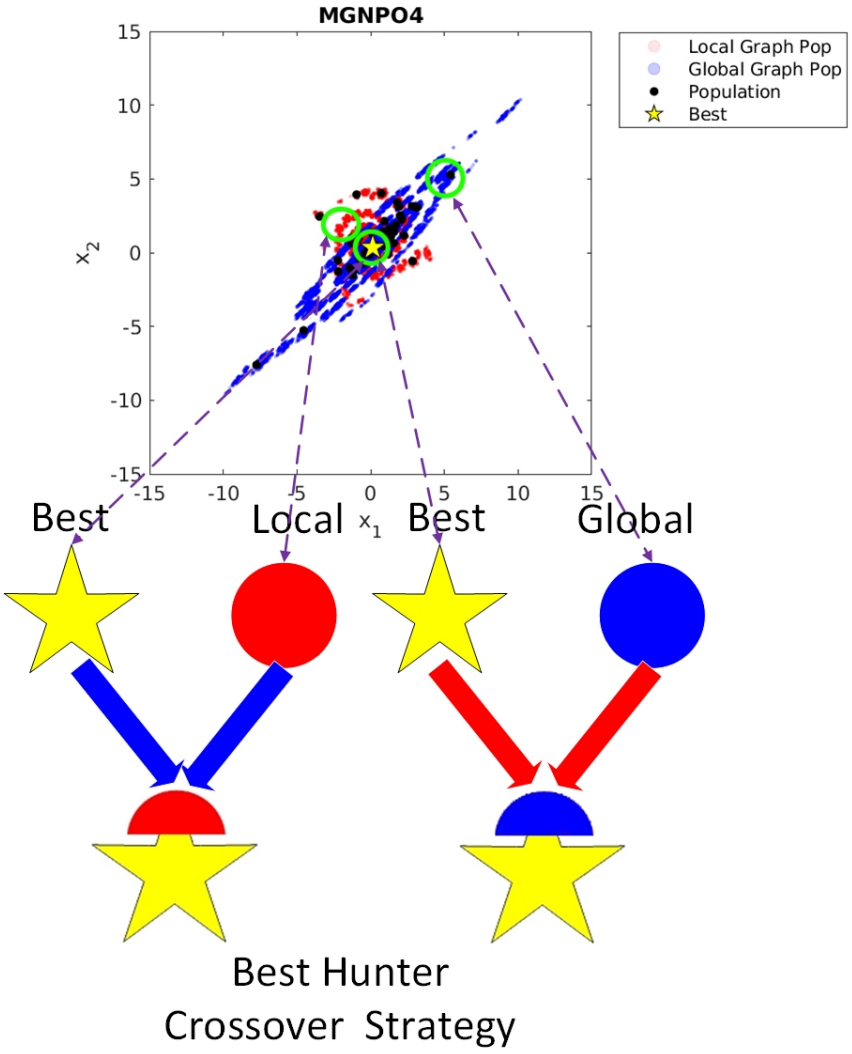
Because of the randomness of the experimental results, 10 random experiments were conducted to calculate the average value. On the four extended application data sets, the random seed was fixed as the default value of 1.

4.1. Function optimization problems

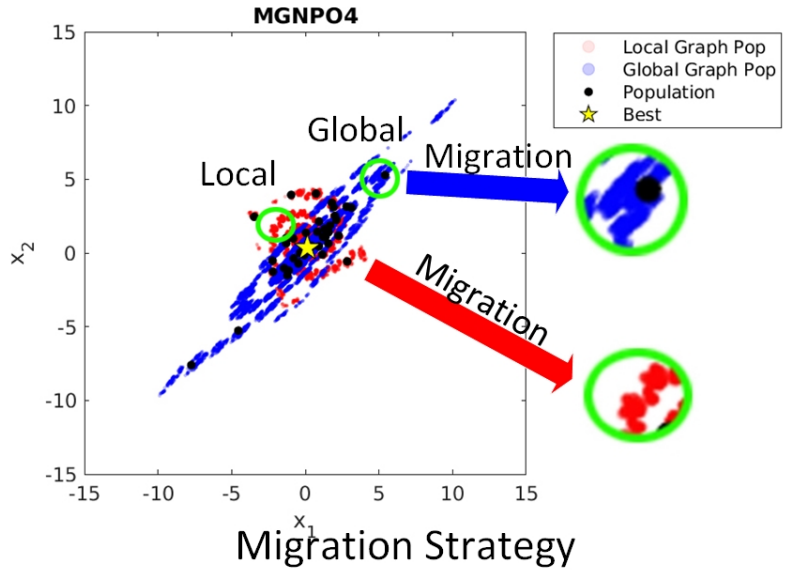
The first open-source dataset included the optimization problems of F1-F23[39] and cecF1-cecF10. The cecF1-cecF10 are "100-digit challenge" of CEC 2019 optimization problems.[40] The second open-source dataset is the optimization problems of F1-F50.[4142]

4.1.1. MGNPO on F1-F23 and cecF1-cecF10

Firstly, contrast experiments were conducted to verify the performance of multi-graph network population evolutionary optimization algorithm in the F1-F23[39] and cecF1-cecF10[40] test functions. The average results are presented in **Table** 1. In F1-F23, N_{iter} is set to 100 and N_{pop} is set to 200. In cecF1-cecF10, N_{iter} is set to 50



A. The best hunter crossover strategy



B. The migration strategy

Figure 7. The best hunter crossover strategy and migration strategy

Algorithm 1 The MGNPOMC algorithm.

Input: Encoded individual X and related application data.
Output: The optimized individual and related application results.
Firstly, start generating basis multi-graph.
for $i_{iter} = 1, ..., N_{iter_g}$ **do**
    Set initial parameters such as $C_{cons}(i_g)$ and $N_{iter_g}(i_g)$.
    for $i_{iter_g} = 1, ..., N_{iter_g}(i_g)$ **do**
        Update $\boldsymbol{Z}_{new}$ and $\boldsymbol{Z}_{data}$ by (1).
    end for
    Obtain original Julia graphs.
    Update the matrix $\boldsymbol{Z}_{bri}$ by (2) and obtain the gray background graph in Figure 5.
    Update the matrix M and obtain the red scattered points in Figure 5.
    Obtain cropped basis Julia graphs.
end for
Then the individuals begin to refer to the multi-graph network for hunting.
Individuals X_{ind} are randomly initialized to X_{init} by (3).
for $t_{iter} = 1, ..., N_{iter}$ **do**
    The best individual of the population was taken as the starting point X_{start}.
    Select the graph network and corresponding M.
    Start randomly rotating the reference sampling graph matrix M.
    Update the $M(i_p, 1)$ and $M(i_p, 2)$ by (4).
    Update the X_1 by (5).
    Set $c_{scale} = 0.4$ as local scale factor and $c_{scale} = 1$ as global scale factor.
    Normalize and scale the X_1 by (6) and different c_{scale} to obtain X_2.
    The middle position of $X_2(\frac{N_{X_2}}{2}, j_d)$ is updated to the center X_c.
    Update the deviation V between the center X_c of the graph network and the starting point X_{start} of the population.
    Update the X_{lib} by (7) and some as $X_{globalLib}$ and some as $X_{localLib}$.
    for $i = 1, ..., N_{pop}$
        Randomly sample to obtain $X_{globalS}$ and X_{localS} and obtain X_{new}.
        Determine whether the new individual X_{new} intersects with the best hunter X_{best} by (8).
        Determine whether the new individual X_{new} needs to migrate to other areas by (9).
        If this X_{new} is better, update the current individual.
    end for
end for
Find the global optimal individual and related application results.

and N_{pop} is set to 200. Among the 33 test functions, the average results of the reference LA only had advantages in four test functions, and these of MGNPO had advantages in 29 test functions. The experimental results indicate that referring to different graph networks for hunting formations has some impacts on the results. A better result means a better prey or a better combat performance. Different test functions correspond to different prey or different enemy forces in different enemy situations and terrains. Using different formations to hunt different formations to deal with enemy forces in different enemy situations and terrains is also in line with natural laws. In most test cases, there is always one type of MGNPO network in MGNPO(01-06) that performs relatively well when hunting specific prey or knowing the enemy situation. But sometimes predators do not know the strength or enemy situation of their prey, and they cannot know which graph network of formation is suitable for hunting this prey or solve this test function. A random method is adopted to select the graph network from 01 to 06 at each iteration in MGNPOP algorithm. Among the 33 test functions, the average results of the reference LA exhibited advantages in 14 test functions, and those of MGNPOP had advantages in 19 test functions.

4.1.2. MGNPOMC on F1-F23 and cecF1-cecF10

Secondly, after scaling and rotating in local and global population graph networks, the individuals intersect with the best hunter to retain the excellent genes of the best hunter while also preserving the vitality of new individuals. When the prey in some areas is overexploited by hunters, some hunters need to undergo an exponential large-scale migration in search of new prey, which gives the algorithm the opportunity to quickly jump out of local optima in such situations. In

Table 1. The average results comparison of different swarm intelligence algorithms

Fun	LA(Ref)[2]	MGNPO						
		01	02	03	04	05	06	MGNPOP
F1	2.32E+00	1.30E+00	7.51E-01	2.97E+00	1.72E+00	1.17E+00	1.85E+00	1.24E+00
F2	3.46E+00	3.28E+00	3.30E+00	6.38E+00	3.33E+00	2.89E+00	3.44E+00	2.62E+00
F3	1.35E+02	4.38E+01	4.24E+01	9.54E+01	5.63E+01	4.69E+01	8.28E+01	3.64E+01
F4	2.04E+01	2.04E+01	1.40E+01	2.46E+01	1.73E+01	1.71E+01	2.18E+01	2.14E+01
F5	1.18E+03	1.11E+03	3.39E+02	9.55E+03	9.36E+02	5.18E+02	1.86E+03	6.26E+02
F6	2.00E+00	1.18E+00	1.11E+00	2.94E+00	1.24E+00	1.16E+00	2.02E+00	1.59E+00
F7	4.43E-02	3.67E-02	2.82E-02	6.49E-02	3.50E-02	2.04E-02	3.32E-02	3.61E-02
F8	-2.74E+03	-2.63E+03	-2.54E+03	-2.64E+03	-2.60E+03	-2.56E+03	-2.55E+03	-2.58E+03
F9	1.84E+01	1.70E+01	1.98E+01	2.42E+01	1.84E+01	1.78E+01	1.66E+01	1.60E+01
F10	4.33E+00	3.73E+00	2.99E+00	7.03E+00	4.58E+00	3.66E+00	3.20E+00	3.13E+00
F11	8.68E-01	9.28E-01	8.06E-01	1.01E+00	8.59E-01	7.71E-01	1.00E+00	8.95E-01
F12	1.24E+01	1.41E+01	1.25E+01	2.63E+01	9.74E+00	1.08E+01	1.16E+01	1.26E+01
F13	7.20E-01	1.11E+00	8.36E-01	1.16E+01	8.91E-01	3.48E-01	3.10E+00	2.29E+00
F14	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
F15	4.65E-03	2.82E-03	2.89E-03	2.89E-03	2.81E-03	2.94E-03	3.12E-03	2.91E-03
F16	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
F17	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01
F18	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
F19	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
F20	-3.24E+00	-3.24E+00	-3.25E+00	-3.24E+00	-3.24E+00	-3.24E+00	-3.24E+00	-3.24E+00
F21	-1.02E+01	-1.02E+01	-1.00E+01	-1.01E+01	-9.40E+00	-9.40E+00	-1.02E+01	-1.01E+01
F22	-9.59E+00	-1.04E+01	-9.69E+00	-1.04E+01	-8.21E+00	-1.04E+01	-1.04E+01	-9.64E+00
F23	-9.86E+00	-9.86E+00	-9.86E+00	-9.86E+00	-9.86E+00	-9.86E+00	-8.52E+00	-9.86E+00
cecF1	3.29E+07	3.48E+07	2.01E+07	4.59E+07	3.06E+07	2.46E+07	3.38E+07	3.23E+07
cecF2	7.63E+03	7.21E+03	6.94E+03	7.42E+03	7.18E+03	6.71E+03	7.18E+03	7.18E+03
cecF3	7.44E+00	7.22E+00	6.96E+00	7.71E+00	8.23E+00	6.71E+00	8.56E+00	8.40E+00
cecF4	4.32E+01	4.79E+01	5.18E+01	5.04E+01	5.42E+01	4.35E+01	5.27E+01	4.55E+01
cecF5	2.12E+00	2.03E+00	1.90E+00	2.15E+00	1.94E+00	1.99E+00	2.00E+00	2.01E+00
cecF6	9.34E+00	9.49E+00	9.60E+00	8.35E+00	8.76E+00	8.89E+00	9.18E+00	9.65E+00
cecF7	1.19E+03	1.18E+03	1.13E+03	1.27E+03	1.22E+03	1.23E+03	1.20E+03	1.29E+03
cecF8	4.42E+00	4.50E+00	4.52E+00	4.58E+00	4.54E+00	4.49E+00	4.64E+00	4.39E+00
cecF9	1.53E+00	1.48E+00	1.48E+00	1.55E+00	1.49E+00	1.50E+00	1.54E+00	1.55E+00
cecF10	2.14E+01	2.14E+01	2.14E+01	2.14E+01	2.13E+01	2.13E+01	2.14E+01	2.14E+01
VS(win)	4	29						
VS(win)	14							19

MGNPOMC(01-06) and MGNPOMCP, the best hunter crossover and migration strategies are both adopted. The average results comparison of different swarm intelligence algorithms with migration strategies is presented in **Table** 2. Among the 33 test functions, the average results of the reference LA only had advantages in four test functions, and those of MGNPOMC had advantages in 29 test functions. The experimental results indicate that referring to different graph network hunting formations has some impacts on the results. Different formations are used to hunt different formations to deal with enemy forces in different enemy situations and terrains. But sometimes predators do not know the strength or enemy situation of their prey, a random method is adopted to select the graph network from 01 to 06 at each iteration in MGNPOMCP algorithm to deal with unknown enemy situations.

4.1.3. Comparative experiments with other classic algorithms

Thirdly, we also conducted comparative experiments with other algorithms on the F1-F23[39] and cecF1-cecF10[40] test functions. The average results comparison of the proposed method with the Giza pyramids construction,[43] the chimp optimization algorithm,[44] the Henry gas solubility optimization,[27] the particle swarm optimization [45], the simplified the human learning optimization,[46] the rider optimization algorithm,[47] and the whale optimization algorithm[39] are shown in **Table** 3. VS1 is a comparison marker between the MGNPOMC method and the reference method in the current column, where 1 represents MGNPOMC victory, -1 represents reference method victory, and 0 represents the similarities. VS2 is a comparison marker

Table 2. The average results comparison of different swarm intelligence algorithms with best hunter crossover and migration strategies

Fun	LA(Ref)	MGNPOMC						
		01	02	03	04	05	06	MGNPOMCP
F1	2.32E+00	7.00E-86	1.02E-88	3.08E-96	1.72E-77	2.54E-86	2.53E-96	1.51E-106
F2	3.46E+00	1.01E-43	2.28E-60	5.95E-56	4.95E-55	1.22E-54	1.37E-51	5.13E-57
F3	1.35E+02	5.86E-29	6.52E-42	1.04E-15	1.45E-17	1.98E-26	7.95E-18	1.37E-34
F4	2.04E+01	2.88E-23	3.56E-37	9.72E-32	1.36E-28	3.39E-38	1.81E-27	1.70E-32
F5	1.18E+03	7.92E+00	7.95E+00	7.87E+00	7.86E+00	7.96E+00	7.80E+00	7.90E+00
F6	2.00E+00	8.49E-03	8.35E-03	1.06E-02	7.27E-03	1.12E-02	1.40E-02	9.85E-03
F7	4.43E-02	9.82E-04	9.52E-04	9.97E-04	5.00E-04	9.33E-04	6.38E-04	4.23E-04
F8	-2.74E+03	-3.68E+03	-3.63E+03	-3.64E+03	-3.63E+03	-3.64E+03	-3.66E+03	-3.68E+03
F9	1.84E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	4.33E+00	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
F11	8.68E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	1.24E+01	1.06E-04	5.52E-05	9.74E-05	5.06E-05	1.24E-04	1.23E-04	1.72E-04
F13	7.20E-01	7.38E-04	1.60E-03	2.66E-03	3.57E-04	1.01E-03	5.95E-04	1.11E-03
F14	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
F15	4.65E-03	2.66E-03	2.90E-03	2.93E-03	2.60E-03	7.20E-04	7.55E-04	2.91E-03
F16	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
F17	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01	3.17E-01
F18	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
F19	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
F20	-3.24E+00	-3.26E+00	-3.26E+00	-3.26E+00	-3.26E+00	-3.26E+00	-3.26E+00	-3.26E+00
F21	-1.02E+01	-7.12E+00	-5.63E+00	-6.37E+00	-5.87E+00	-5.12E+00	-7.40E+00	-5.87E+00
F22	-9.59E+00	-6.72E+00	-5.52E+00	-6.05E+00	-6.34E+00	-6.48E+00	-5.52E+00	-6.48E+00
F23	-9.86E+00	-7.15E+00	-6.92E+00	-7.69E+00	-6.38E+00	-6.48E+00	-7.15E+00	-6.38E+00
cecF1	3.29E+07	1.60E+05	2.71E+03	9.23E+04	6.85E+05	8.13E+05	1.30E+05	9.05E+04
cecF2	7.63E+03	3.59E+02	2.28E+02	5.20E+02	1.67E+02	2.21E+02	9.05E+01	3.31E+02
cecF3	7.44E+00	5.48E+00	5.21E+00	6.13E+00	6.26E+00	5.66E+00	5.26E+00	5.12E+00
cecF4	4.32E+01	2.46E+01	2.34E+01	2.61E+01	2.37E+01	2.27E+01	2.92E+01	2.41E+01
cecF5	2.12E+00	1.38E+00	1.24E+00	1.31E+00	1.26E+00	1.31E+00	1.25E+00	1.29E+00
cecF6	9.34E+00	4.27E+00	4.01E+00	4.08E+00	4.64E+00	4.11E+00	4.84E+00	4.08E+00
cecF7	1.19E+03	6.68E+02	7.55E+02	8.76E+02	7.98E+02	7.61E+02	7.53E+02	7.19E+02
cecF8	4.42E+00	4.03E+00	3.98E+00	3.83E+00	4.09E+00	3.84E+00	4.12E+00	3.97E+00
cecF9	1.53E+00	1.33E+00	1.35E+00	1.35E+00	1.31E+00	1.34E+00	1.32E+00	1.34E+00
cecF10	2.14E+01	2.11E+01	2.11E+01	2.10E+01	2.10E+01	2.11E+01	2.11E+01	2.10E+01
VS(win)	4				29			
VS(win)	4							29

between the MGNPOMCP method and the reference method in the current column. Count1 and Count2 are the sum of VS1 and VS2. If Count1 or Count2 is greater than 0, the proposed MGNPOMC or MGNPOMCP is better. Compared to GPC, the average results of the reference GPC have advantages in five test functions, and those of MGNPOMC have advantages in 26 test functions. The MGNPOMC and MGNPOMCP both have 21 more test functions of obtaining advantages than GPC. Compared to ChOA, MGNPOMC has 31 more test functions than ChOA of obtaining advantages, MGNPOMCP has 29 more test functions than ChOA of obtaining advantages. Compared to HGSO, the MGNPOMC and MGNPOMCP both have 12 more test functions of obtaining advantages than HSGO. Compared to PSO, MGNPOMC has 13 more test functions of obtaining advantages than PSO, MGNPOMCP has 11 more test functions of obtaining advantages than PSO. Compared to SHLO, the average results of the reference MGNPOMC and MGNPOMCP have advantages in all 33 test functions. Compared to ROA, the MGNPOMC and MGNPOMCP both have 27 more test functions of obtaining advantages than ROA. Compared to WOA, MGNPOMC has 21 more test functions of obtaining advantages than WOA, MGNPOMCP has 17 more test functions of obtaining advantages than WOA.

4.1.4. The significance p-value analysis

Fourthly, we conducted a statistical analysis of the significance of p-values between the proposed method and the original reference LA in the F1-F23 and cecF1-cecF10 test functions. In each test function, we took the results of 10 random runs of the original reference LA as the first row of matrix M_1, and the results of 10 random runs of the MGNPOMC as the second row of matrix M_1. Then, statistical analysis was performed on the matrix M_1 of each test function **Table**

Table 3. The average results comparison of the proposed method with GPC, ChOA, HGSO, PSO, SHLO, ROA, and WOA

Fun	MGNPOMC	MGNPOMCP	GPC[43]	ChOA[44]	HGSO[27]	PSO[45]	SHLO[46]	ROA[47]	WOA[39]
F1	2.53E-96	1.51E-106	7.3211E-22	1.98E-11	3.39E-88	3.71E-03	2.55E+03	6.60E-08	6.63E-40
F2	2.28E-60	5.13E-57	4.4235E-12	1.78E-08	2.70E-46	1.30E-01	1.21E+01	3.68E-04	1.89E-25
F3	6.52E-42	1.37E-34	1.7463E-21	3.08E-05	4.41E-87	2.66E-01	1.23E+04	7.20E-07	2.98E+02
F4	3.39E-38	1.70E-32	1.4084E-11	4.59E-04	4.04E-46	1.40E-01	5.40E+01	1.30E-03	1.80E+00
F5	7.80E+00	7.90E+00	7.5112E+00	8.89E+00	7.52E+00	1.85E+01	3.48E+06	8.21E+00	6.55E+00
F6	7.27E-03	9.85E-03	5.9401E-01	1.41E-01	2.89E-01	5.32E-03	2.08E+03	2.50E+00	3.54E-04
F7	5.00E-04	4.23E-04	4.3785E-05	1.27E-03	1.32E-04	5.21E-02	1.73E+00	1.68E-02	1.96E-03
F8	-3.68E+03	-3.68E+03	-2.6107E+03	-2.22E+03	-4.70E+06	-2.53E+03	-2.27E+03	-1.81E+03	-3.59E+03
F9	0.00E+00	0.00E+00	0.0000E+00	8.90E+00	0.00E+00	2.21E+01	5.12E+01	8.01E-06	2.84E-15
F10	8.88E-16	8.88E-16	7.8108E-12	1.36E+01	8.88E-16	1.32E-01	1.37E+01	1.44E-04	5.51E-15
F11	0.00E+00	0.00E+00	0.0000E+00	1.74E-01	0.00E+00	1.45E-01	3.57E+01	3.57E-09	3.45E-02
F12	5.06E-05	1.72E-04	9.2657E-02	2.65E-02	6.08E-02	2.07E-04	1.15E+07	2.65E+00	5.48E-03
F13	3.57E-04	1.11E-03	4.7575E-01	9.57E-01	2.49E-01	2.95E-03	6.36E+06	2.05E-01	9.41E-03
F14	9.98E-01	9.98E-01	2.8147E+00	9.98E-01	1.31E+00	1.40E+00	7.49E+00	1.27E+01	1.39E+00
F15	7.20E-04	2.91E-03	5.6851E-04	1.27E-03	3.56E-04	9.73E-04	7.86E-02	7.12E-04	7.40E-04
F16	-1.03E+00	-1.03E+00	-1.0242E+00	-1.03E+00	-1.03E+00	-1.03E+00	-9.81E-01	-1.00E+00	-1.03E+00
F17	3.17E-01	3.17E-01	3.3275E-01	3.04E-01	-3.30E+08	-2.98E+23	3.23E-01	3.64E-01	3.04E-01
F18	3.00E+00	3.00E+00	4.4707E+00	3.00E+00	3.00E+00	3.00E+00	8.40E+01	2.07E+02	3.00E+00
F19	-3.86E+00	-3.86E+00	-3.6465E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.48E+00	-3.78E+00	-3.86E+00
F20	-3.26E+00	-3.26E+00	-2.4489E+00	-2.72E+00	-3.03E+00	-3.27E+00	-2.71E+00	-2.70E+00	-3.19E+00
F21	-7.40E+00	-5.87E+00	-1.6667E+00	-4.14E+00	-4.60E+00	-8.40E+00	-1.29E+00	-3.59E+00	-8.65E+00
F22	-6.72E+00	-6.48E+00	-1.7501E+00	-4.89E+00	-5.29E+00	-5.94E+00	-1.33E+00	-3.01E+00	-7.37E+00
F23	-7.69E+00	-6.38E+00	-2.1078E+00	-5.06E+00	-4.69E+00	-8.39E+00	-1.98E+00	-3.38E+00	-7.50E+00
cecF1	2.71E+03	9.05E+04	1.0000E+00	4.66E+06	1.00E+00	7.96E+08	3.53E+08	1.00E+00	3.89E+07
cecF2	9.05E+01	3.31E+02	5.0284E+00	4.10E+03	4.59E+00	2.82E+04	1.74E+04	5.00E+00	8.95E+03
cecF3	5.21E+00	5.12E+00	7.0772E+00	5.58E+00	7.57E+00	1.05E+01	1.12E+01	1.22E+01	6.96E+00
cecF4	2.27E+01	2.41E+01	5.7901E+01	6.53E+01	5.87E+01	4.03E+01	1.21E+02	1.41E+02	6.06E+01
cecF5	1.24E+00	1.29E+00	2.8859E+01	3.72E+01	1.57E+01	1.63E+00	7.36E+01	1.35E+02	2.72E+00
cecF6	4.01E+00	4.08E+00	6.9294E+00	9.75E+00	8.10E+00	3.51E+00	1.24E+01	1.54E+01	9.50E+00
cecF7	6.68E+02	7.19E+02	1.2290E+03	1.96E+03	1.72E+03	1.08E+03	2.06E+03	2.91E+03	1.40E+03
cecF8	3.83E+00	3.97E+00	4.5297E+00	4.84E+00	4.87E+00	4.14E+00	5.29E+00	5.52E+00	4.86E+00
cecF9	1.31E+00	1.34E+00	1.4834E+00	1.62E+00	1.71E+00	1.28E+00	3.37E+00	7.29E+00	1.34E+00
cecF10	2.10E+01	2.10E+01	2.1064E+01	2.15E+01	2.13E+01	2.14E+01	2.13E+01	2.18E+01	2.14E+01
Count1			21	31	12	13	33	27	21
Count2			21	29	12	11	33	27	17

A2. $P_{flag} = 1$ represents $P_{Prob>F} < 0.05$ and $F_{flag} = 1$ represents $F > 5.11$. If P_{flag} and F_{flag} are 1, there are significant differences between different methods. The number of functions with $P_{Prob>F}(rows) < 0.05$ between LA and MGNPOMC is 32. The number of functions with $F > 5.11$ between LA and MGNPOMC is 32. That is to say, there is a significant difference between LA and MGNPOMC in most test functions. In each test function, we took the results of 10 random runs of the original reference LA as the first row of matrix M_2, and the results of 10 random runs of MGNPOMCP as the second row of matrix M_2. Then, statistical analysis was performed on the matrix M_2 of each test function and obtain **Table** Appendix **A2**. The number of functions with $P_{Prob>F<0.05}$ between LA and MGNPOMCP is 25. The number of functions with $F > 5.11$ between LA and MGNPOMCP is 25. That is to say, there is a significant difference between LA and MGNPOMCP in most test functions. Compared to the original reference LA, the proposed multi-graph network population evolutionary optimization algorithm with best hunter crossover and migration strategies showed a significant improvement.

4.1.5. The MGNPOMC on the F1-F50

Finally, contrast experiments with $N_{iter} = 100, N_{pop} = 200$ were conducted on the F1-F50 test functions.[41][42] As shown in Appendix **Table** A3, among the 50 test functions, the average results of the reference LA only had advantages in five test functions, those of MGNPOMC have advantages in 44 test functions, and those of both are the same in one test function. The average results of the reference LA had advantages in seven test functions, those of MGNPOMCP have advantages in 42 test functions, and those of both were the same in one test function.

Table 4. The results comparison of different swarm intelligence algorithms in four extended applications

Fun	LA(Ref)[2]	MGNPOMC						
		01	02	03	04	05	06	MGNPOMCP
MSAS	2.1801E+06	2.1895E+06	2.2207E+06	2.1773E+06	2.2090E+06	2.1910E+06	2.2217E+06	2.1621E+06
FWS	32	29	30	29	30	29	30	29
UAVRO	81.3860	93.5753	93.6252	71.6294	70.4079	67.5296	70.9937	56.4585
PSBTO	1.5923E+06	1.5921E+06	1.5921E+06	1.5921E+06	1.5921E+06	1.5921E+06	1.5921E+06	1.5921E+06

So, the proposed multi-graph network population evolutionary optimization algorithm with the best hunter crossover and migration strategies methods has achieved competitive results in function optimization.

4.2. Extended applications

The results comparison of different swarm intelligence algorithms in the four extended applications is shown in **Table** 4.

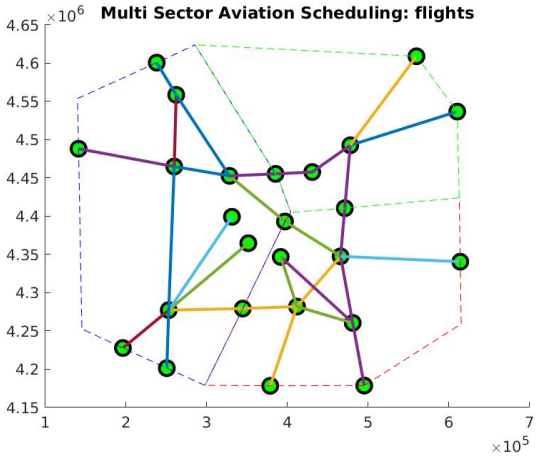


Figure 8. The cities and flights of the multi-sector aviation scheduling

4.2.1. Multi-sector aviation scheduling

Swarm intelligence individuals X_{ind} correspond to aviation control parameters. The individual is a 3D vector; the first dimension controls the flight priority P_{flight}, the second dimension controls speed P_{speed}, and the third dimension controls the route type P_{rt}. Individual fitness Y_{fit} is related to the actual time deviation f_{fit1} and the flight's actual arrival time f_{fit2}.

The aircraft P_{pos} was found to dispatch. The schedulable flight was sorted to obtain the priority index P_{idx} and the sorted flight P_{spos} according to flight priority P_{flight} and P_{pos}. The speed of this flight v_{flight} was updated, which was related to maximum speed v_{max}, minimum speed v_{min}, flight number N_f, and speed priority P_{speed}. Some route types have multiple heights N_{rf} and parameter h_{idx} controls, which height can be selected.

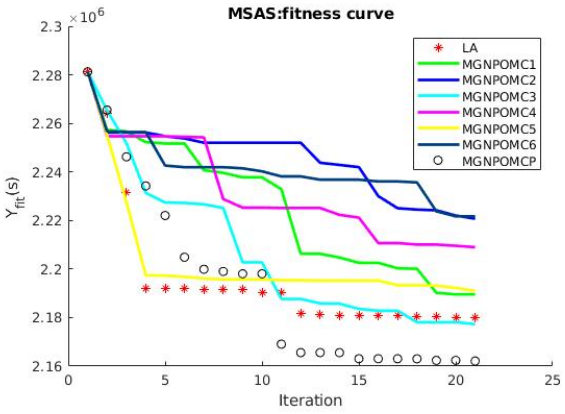


Figure 9. The fitness curves comparison of different swarm intelligence algorithms in the multi-sector aviation scheduling application

$$\begin{cases} P_{spos} = P_{pos}(P_{idx}) \\ v_{flight} = (v_{max} - v_{min}) \cdot P_{speed}(N_f) + v_{min} \\ h_{idx} = f_{ceil}(N_{rf} \cdot P_{rt}(N_f)) \end{cases} \quad (10)$$

f_{fit1} is the actual entry time minus the scheduled entry time $T_{scheduled}$ of the flight, which means early or late. The flight's actual arrival time f_{fit2} is the flight start time T_{start} and air navigation time T_{flight}.

$$\begin{cases} f_{fit1} = T_{start} - T_{scheduled} \\ f_{fit2} = T_{start} + T_{flight} \\ Y_{fit} = w_1 \cdot f_{fit1} + w_2 \cdot f_{fit2} \end{cases} \quad (11)$$

The cities and MSAS are shown in Figure 8. The small green dots are cities, the lines with different colors are routes of different flights, and the dashed lines with different colors are different sectors. The fitness curves and results comparison of different swarm intelligence algorithms with $N_{iter} = 10, N_{pop} = 20$ in MSAS application are shown in Figure 9 and **Table** 4. MGNPOMCP took 2.1621E The results of MGNPOMC3 and MGNPOMCP are better than that of LA in MSAS application.

4.2.2. Flexible workshop scheduling optimization

The swarm intelligence individuals correspond to the priority weight matrix of the process and machine and control whether the process and machine are selected. There are four machines, three

workpieces, three processes for each workpiece, and a total of nine processes. The dimension of the individual X_{ind} is $9 \times 4 = 36$. Individual fitness Y_{fit} is the time consumption after completing all processes.

T_{temp} is the actual time consumption. T_{Mt} is the machine theoretical processing time matrix of different workpieces and different machines in the current process. W_{Pri} is the priority weight matrix from X_{ind} in the current process. T_{St} is the temp start time matrix of different workpieces and different machines in the current process. C_{Pt} is the process coefficient matrix of different machines with different workpieces, and the coefficients of different machines with different workpieces are different. C_W is the workpiece coefficient matrix of different processes and machines, and the coefficients of different workpieces with the same machine are different, but the coefficients of different processes with the same machine and workpiece are the same. T_{Mt}, T_{St}, W_{Pri}, C_{Pt}, and C_{Wt} are all 3×4 dimensions. T_M is the current end time vector of all four machines. n_M ,T_S, and T_T are the index of the machine, the actual starting time, and the theoretical processing time corresponding to the minimum value in the temp matrix, T_{temp}, respectively. T_P is the current end time vector of the temp process. Y_{fit} is the max T_P of all processes. Since the start time of the first process in all workpieces is initialized to 0, Y_{fit} is also the time consumption after completing all processes.

$$\begin{cases} \boldsymbol{T_{temp}} = \frac{\boldsymbol{T_{Mt}} \cdot \boldsymbol{W_{Pri}} \cdot \boldsymbol{T_{St}}}{\boldsymbol{C_{Pt}} \cdot \boldsymbol{C_{Wt}} } \\ \boldsymbol{T_S} = max(\boldsymbol{T_S}, \boldsymbol{T_M}(n_M)) \\ Y_{fit} = max(\boldsymbol{T_P}) = max(\boldsymbol{T_S} + \boldsymbol{T_T}) \end{cases} \tag{12}$$

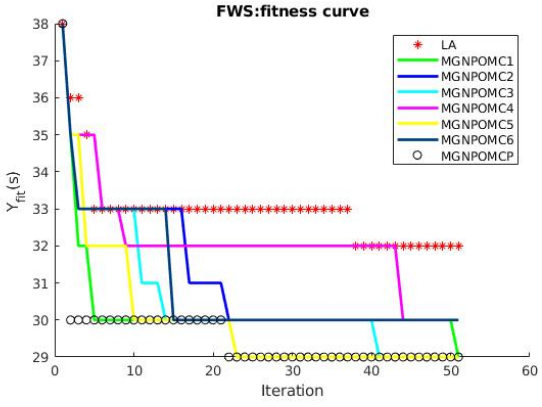


Figure 10. The fitness curves comparison of different swarm intelligence algorithms in the flexible workshop scheduling (FWS) application

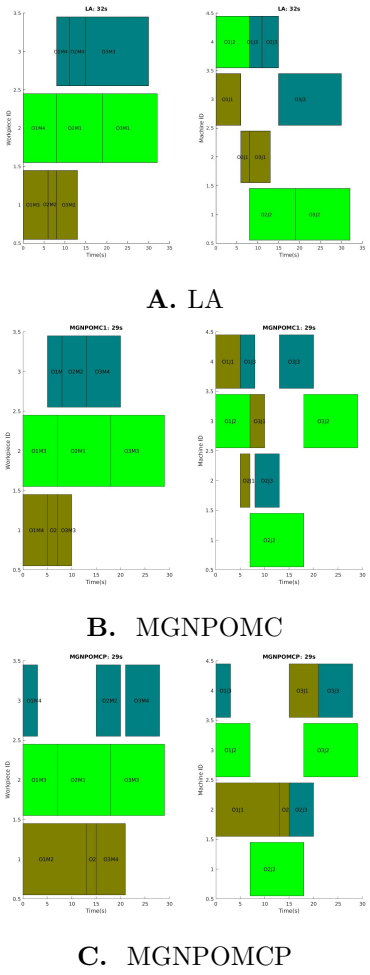


Figure 11. The machining sequences optimized by different swarm intelligence algorithms in the the flexible workshop scheduling (FWS) application

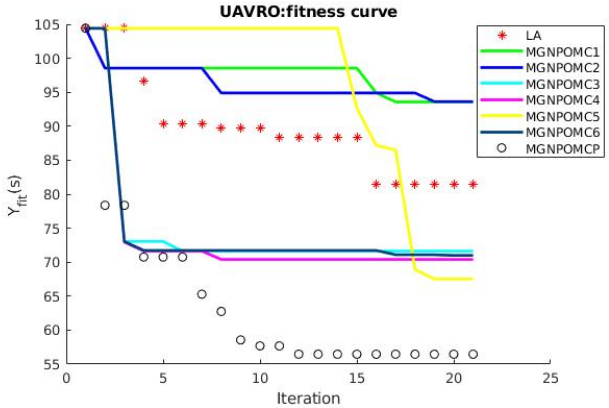


Figure 12. The fitness curves comparison of different swarm intelligence algorithms in unmanned aerial vehicle routing optimization (UAVRO) application

The fitness curves and results comparison of different swarm intelligence algorithms with $N_{iter} = 5, N_{pop} = 50$ in the the FWS optimization application are shown in **Table** 4. The machining sequences are optimized by different swarm

intelligence algorithms in the FWS application is shown in Figure 11. The machining sequences of the workpieces and the machines are shown on the left and right of the subfigure, respectively. The $O1M3$ indicates are machine 3 and workpiece 1, and the $O1J2$ means are workpiece 1 and process 2. The results of MGNPOMC1, MGNPOMC3, MGNPOMC5, and MGNPOMCP are 29. The results of MGNPOMC1, MGNPOMC2, MGNPOMC3, MGNPOMC4, MGNPOMC5, MGNP0MC6, and MGNPOMCP are better than that of LA in FWS application.

4.2.3. Unmanned aerial vehicle routing optimization of oil plant in three-dimensional map

There are 19 obstacles, one starting point, and one ending point in UAV routing optimization of an oil plant. Each obstacle is represented by a 6D vector, indicating its XYZ position, length, width, and height. Coordinate position and length are discretized by unit length (50, 50, 1), such as speed or flight distance per unit of expend, and then the discrete point tensor in $20 \times 20 \times 20$ discrete space of an obstacle is obtained. There are three cases in each dimension: forward, stop, and backward [1, 0, -1]. After removing the fixed point (0, 0, 0) in the 3D space, the $(3^3 - 1) = 26$ movable directions P_{dir} are obtained. Swarm intelligence individuals X_{ind} with 8000D vectors are the selection priorities W_{pri} of discrete tensor space points. Individual fitness Y_{fit} corresponds to the minimum consumption of UAV routing from the start point to the end point in a 3D map. The small red circle is the starting point, the small red five-star is the ending point, the box is the obstacle, and the black point is the discrete point of the obstacle.

The candidate movable point set P_{nextN} is the current point P_{cur} plus each direction vector P_{dir}. The candidate points that exceed the boundary ($P_{nextN}(i,:) < 1$ or $P_{nextN}(i,:) > 20$) is eliminated. The immovable points, such as obstacles and passing points, that are marked with 1 in the map ($M_{map}(P_{nextN}(i,1), P_{nextN}(i,2), P_{nextN}(i,3)) = 1$) are eliminated. If there is no available point and the destination is not reached ($P_{nextN} = empty$), the algorithm traces back to the previous node ($P_{cur} = path(end-1,:)$) and the last node on the path is deleted ($path(end,:) = empty$). L_{path} is the number of nodes on the path. $P_{nextN}(no)$ is the candidate point in P_{nextN} with the minimum the distance $D_{wcur}(no)$, and this point $P_{nextN}(no)$ is added to the path. The point $P_{nextN}(no)$ is updated to the current point P_{cur}, and the position in the map is marked as 1 ($M_{map}(P_{cur}(1), P_{cur}(2), P_{cur}(3)) = 1$).

$$\begin{cases} \boldsymbol{P}_{nextN} = P_{cur} + \boldsymbol{P}_{dir} \\ D_{cur}(i) = \|\boldsymbol{P}_{nextN}(i,:) - P_{cur})\| + \|\boldsymbol{P}_{nextN}(i,:) \\ -P_{end})\| \\ P_{pri}(i) = \boldsymbol{W}_{pri}(\boldsymbol{P}_{nextN}(i,1), \boldsymbol{P}_{nextN}(i,2), \\ \boldsymbol{P}_{nextN}(i,3)) \\ D_{wcur}(i) = D_{cur}(i) \cdot \sqrt{P_{pri}(i)} \\ Y_{fit} = \sum_{i=1}^{L_{path}-1} \sqrt{\sum_{j=1}^{dim}(path(i,j) - path(i+1,j))^2} \end{cases} \tag{13}$$

The fitness curves and results comparison of different swarm intelligence algorithms with $N_{iter} = 10, N_{pop} = 20$ in UAVRO application are shown in Figure 12 and **Table** 4. MGNPOMCP took 56.458 seconds, and the least consumption. The results of MGNPOMC3, MGNPOMC4, MGNPOMC5, MGNP0MC6, and MGNPOMCP are better than that of LA in UAVRO application. The 3D UAV paths optimized by different swarm intelligence algorithms in UAVRO application are shown in Figure 13.

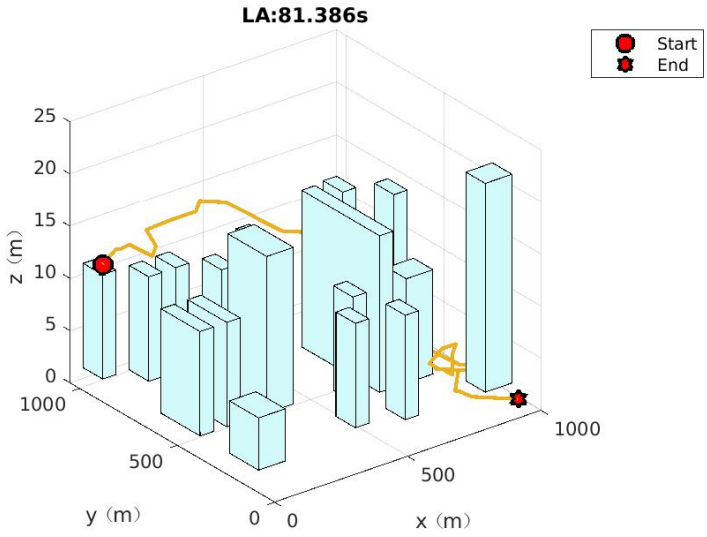
4.2.4. Power system bus types optimization

Swarm intelligence individuals X_{ind} with 9D vectors are the selection priorities of the line types at different bus positions. Individual fitness Y_{fit} corresponds to the total cost of load cost f_{load}, the line loss cost f_{loss}, the line expend cost f_{line}, and the punishment constraint f_{punish}.

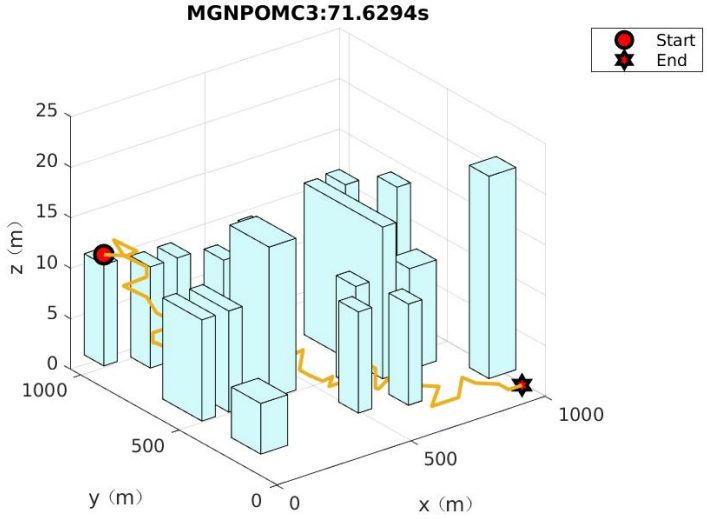
There are nine buses, and 10 types of lines T_L can be selected for each bus. The resistance, reactance, and cost per unit length L_C of each line are different. The line type of each bus is optimized to minimize the cost of the whole power system. The C is the cost at each bus T_L. The load cost f_{load} is the cumulative sum of the product of the active power mismatch Lagrange multiplier c_{λ_P} and the active power $P(i)$. The line loss cost f_{loss} is the cumulative sum of the product of the real part of line loss $L_{realLoss}(i)$ and the loss coefficient c_{λ_L}. The line expend cost f_{line} is the cumulative sum of all branch expend $C(T_L(i))$. If the operation in the tide circuit is not successful, the loss function f_{punish} will be punished.

$$\begin{cases} T_L = round(x \cdot (N_L - 1)) + 1 \\ C = L_C(T_L) \cdot D \\ f_{load} = \sum_{i=1}^{N_L} P(i) \cdot c_{\lambda_P} \\ f_{loss} = \sum_{i=1}^{N_L} L_{realLoss}(i) \cdot c_{\lambda_L} \\ f_{line} = \sum_{i=1}^{N_L} C(T_L(i)) \\ Y_{fit} = (f_{load} + f_{loss}) \cdot 200 + f_{line} + f_{punish}. \end{cases} \tag{14}$$

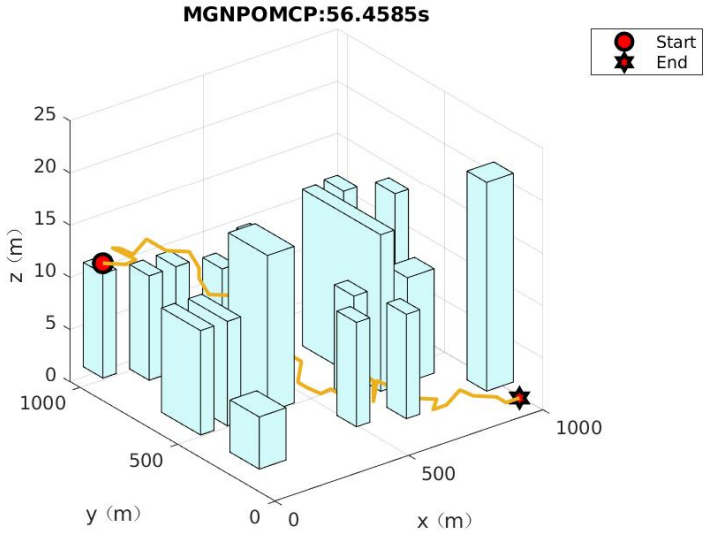
The fitness curves and results comparison of different swarm intelligence algorithms with



A. LA

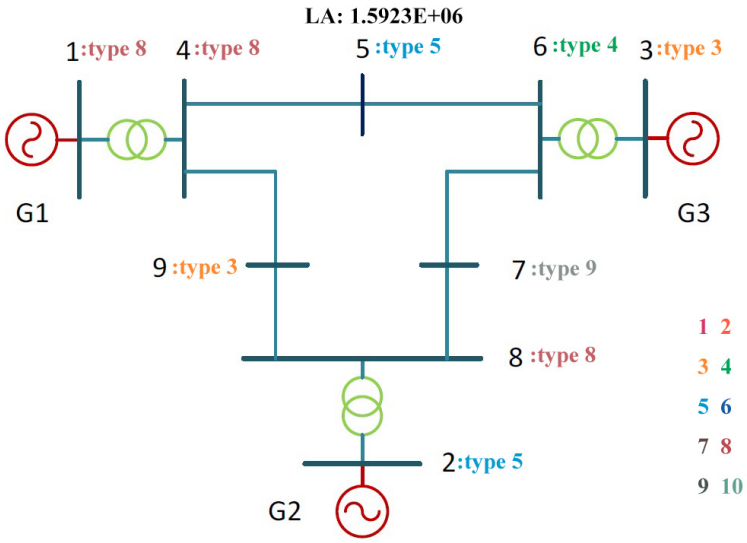


B. MGNPOMC

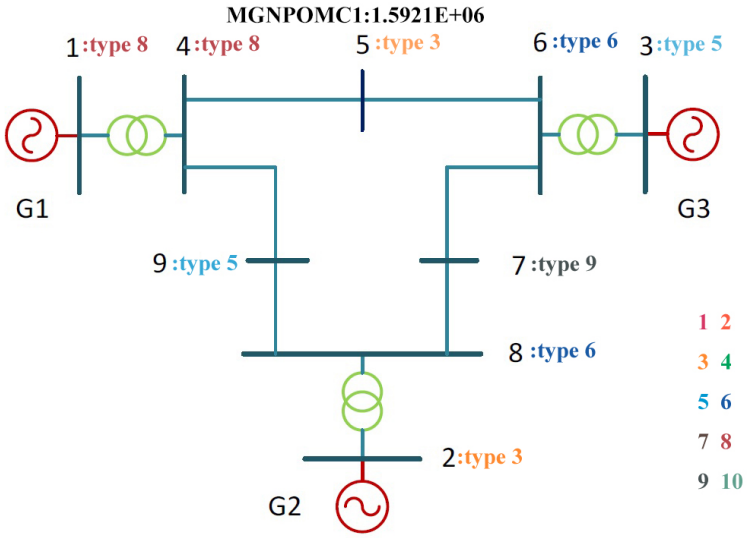


C. MGNPOMCP

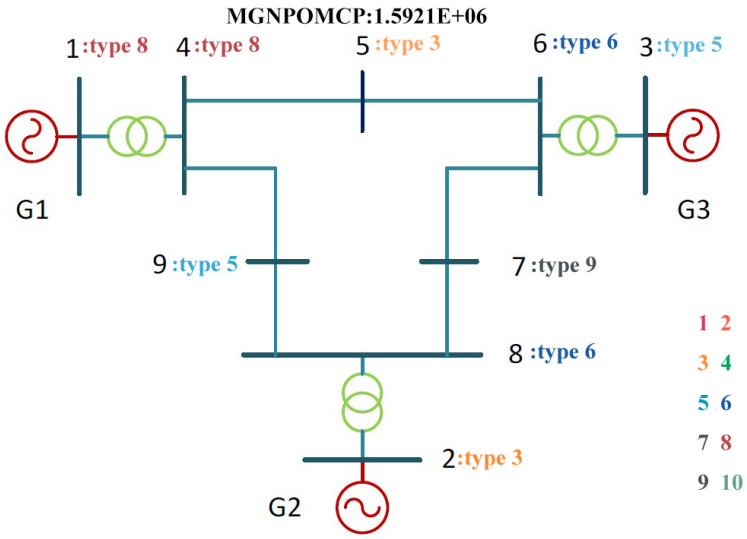
Figure 13. The three-dimensional unmanned aerial vehicle paths optimized by different swarm intelligence algorithms in the unmanned aerial vehicle routing optimization application



A. LA



B.MGNPOMC



C. MGNPOMCP

Figure 14. The layouts and bus types optimized by different swarm intelligence algorithms in the power system bus types optimization application

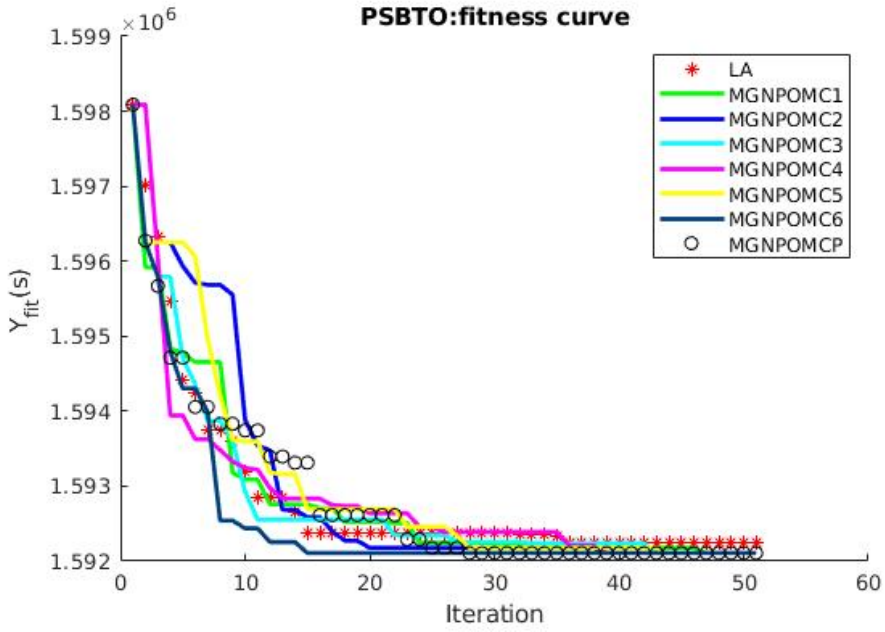


Figure 15. The fitness curves comparison of different swarm intelligence algorithms in the power system bus types optimization (PSBTO) application

$N_{iter} = 20, N_{pop} = 50$ the PSBTO application are shown in Figure 15 and **Table** 4. The results of MGNPOMC1, MGNPOMC2, MGNPOMC3, MGNPOMC4, MGNPOMC5, MGNP06, and MGNPOMCP are all better than that of LA in PSBTO application. The layouts and bus types optimized by different swarm intelligence algorithms in PSBTO application are shown in Figure 14. The G1, G2, and G3 are generators, and different bus types are represented by different color fonts. The $1 : type9$ means the bus 1 with a type 9 line in the power system.

5. Conclusion

In this paper, a MGNPOMC algorithm was proposed for cross-field engineering applications and design problems. By reference to the population Julia graph with central radiation shape, prey was taken as the center and random sampling of different patterns as the basis graphs, which were rotated and deformed randomly. The positions of candidate individuals were updated according to the generated graph library with global and local scales. Furthermore, the MGNPOMC algorithm was applied to open CEC dataset, and extended to MSAS, FWSO, UAVRO, and PSBTO, and achieving competitive results. However, this article also has some limitations. Only animal hunting formations related to multi-graph networks were referenced, and other biomimetic strategies of organisms still need to be explored. Only four types of engineering applications were extended with swarm intelligence algorithms, and there are other engineering applications that need to be expanded. In the near future, the brain-like navigation behavior of grid cells will be introduced into swarm intelligence optimization, and it will be expanded to other fields such as sparse problem of a phased array antenna, wireless sensor network optimization, and subway system optimization.

Acknowledgments

None.

Funding

This work was supported by National Science and Technology Innovation 2030 Major Program of China (2022ZD0205005) and (National Natural Science Foundation of China (42576280).

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization: Bailu Si
Formal analysis: Zhaoyang Lian
Investigation: Zhaoyang Lian
Methodology: Zhaoyang Lian
Writing - original draft: Zhaoyang Lian
Writing - review & editing: Bailu Si

Availability of data

The data is partially available from the authors upon reasonable request.

AI tools statement

All authors confirm that no AI tools were used in the preparation of this manuscript.

References

1. Xing A, Chen Y, Suo J, Zhang J. Improving teaching-learning-based optimization algorithm with goldensine and multi-population for global optimization. *Math Comput Simul.* 2024;221:94-134.
2. Pereira JLJ, Francisco MB, Diniz CA, Oliver GA, Cunha SS Jr, Gomes GF. Lichtenberg algorithm: a novel hybrid physics-based metaheuristic for global optimization. *Expert Syst Appl.* 2021;170:114522.
3. Zeng T, Tang F, Ji D, Si B. Neurobayesslam: neurobiologically inspired Bayesian integration of multisensory information for robot navigation. *Neural Netw.* 2020;126:21-35.
4. Duan L, Lian Z, Qiao Y, Chen J, Miao J, Li M. A novel feature fusion approach for classification of motor imagery EEG based on hierarchical extreme learning machine. *Cogn Comput.* 2024;16(2):566-580.
5. Lian Z, Si B. Multigroup cooperative evolutionary optimization algorithm combined with quantum entanglement for cross-field applications. *Artif Intell Rev.* 2025;58(10):1-32.
6. Salawudeen AT, Muiazu MB, Yusuf A, Adedokun AE. A novel smell agent optimization (SAO): an extensive CEC study and engineering application. *Knowl Based Syst.* 2021;232:107486.
7. Yapici H, Cetinkaya N. A new meta-heuristic optimizer: pathfinder algorithm. *Appl Soft Comput.* 2019;78:545-568.
8. Shehadeh HA. A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization. *Neural Comput Appl.* 2021;33(18):11739-11752.
9. Shi Y. Brain storm optimization algorithm. In: Advances in Swarm Intelligence: Second International Conference, ICSI 2011; June 12-15, 2011; Chongqing, China. Part I 2. Springer; 2011:303-309.
10. Cai Z, Gao S, Yang X, Yang G, Cheng S, Shi Y. Alternate search pattern-based brain storm optimization. *Knowl Based Syst.* 2022;238:107896.
11. Zamli KZ, Alhadawi HS, Din F. Utilizing the roulette wheel based social network search algorithm for substitution box construction and optimization. *Neural Comput Appl.* 2023;35(5):4051-4071.
12. Mohamed AW, Hadi AA, Mohamed AK. Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. *Int J Mach Learn Cybern.* 2020;11(7):1501-1529.
13. Sallam KM, Hossain MA, Chakrabortty RK, Ryan MJ. An improved gaining-sharing knowledge algorithm for parameter extraction of photovoltaic models. *Energy Convers Manag.* 2021;237:114030.
14. Feng ZK, Niu WJ, Liu S. Cooperation search algorithm: a novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Appl Soft Comput.* 2021;98:106734.
15. Askari Q, Saeed M, Younas I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst Appl.* 2020;161:113702.
16. Mohammadi-Balani A, Dehghan Nayeri M, Azar A, Taghizadeh-Yazdi M. Golden eagle optimizer: a nature-inspired metaheuristic algorithm. *Comput Ind Eng.* 2021;152:107050.
17. Dhiman G, Kumar V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl Based Syst.* 2019;165:169-196.
18. Braik M, Sheta A, Al-Hiary H. A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural Comput Appl.* 2021;33(7):2515-2547.
19. Zhao W, Zhang Z, Wang L. Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. *Eng Appl Artif Intell.* 2020;87:103300.
20. Hu G, Li M, Wang X, Wei G, Chang CT. An enhanced manta ray foraging optimization algorithm for shape optimization of complex CCG-ball curves. *Knowl Based Syst.* 2022;240:108071.
21. Braik MS. Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. *Expert Syst Appl.* 2021;174:114685.
22. Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W. Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. *Math Comput Simul.* 2022;192:84-110.
23. Bayraktar Z, Komurcu M, Bossard JA, Werner DH. The wind driven optimization technique and its application in electromagnetics. *IEEE Trans Antennas Propag.* 2013;61(5):2745-2757.

24. Ala'F K, Alqammaz A, Khasawneh AM, Abualigah L, Darabkh KA, Zinonos Z. An environmental remote sensing and prediction model for an IoT smart irrigation system based on an enhanced wind-driven optimization algorithm. *Comput Electr Eng.* 2025;122:109889.
25. Anita, Yadav A, et al. AEFA: artificial electric field algorithm for global optimization. *Swarm Evol Comput.* 2019;48:93-108.
26. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl Intell.* 2021;51:1531-1551.
27. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: a novel physics-based algorithm. *Future Gener Comput Syst.* 2019;101:646-667.
28. Karasu S, Altan A. Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. *Energy.* 2022;242:122964.
29. Mohamed M, Youssef A-R, Kamel S, Ebeed M. Lightning attachment procedure optimization algorithm for nonlinear non-convex short-term hydrothermal generation scheduling. *Soft Comput.* 2020;24:16225-16248.
30. Liu Q, Li N, Jia H, Qi Q, Abualigah L, Liu Y. A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems. *Mathematics.* 2022;10(9):1567.
31. Han M, Du Z, Zhu H, Li Y, Yuan Q, Zhu H. Golden-sine dynamic marine predator algorithm for addressing engineering design optimization. *Expert Syst Appl.* 2022;210:118460.
32. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng.* 2021;376:113609.
33. Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst.* 2016;96:120-133.
34. Punnathanam V, Kotecha P. Yin-yang-pair optimization: a novel lightweight optimization algorithm. *Eng Appl Artif Intell.* 2016;54:62-79.
35. Wang W-C, Xu L, Chau K-W, Zhao Y, Xu D-M. An orthogonal opposition-based-learning yin–yang-pair optimization algorithm for engineering optimization. *Eng Comput.* 2021:1-35.
36. Zhao W, Wang L, Zhang Z. A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Gener Comput Syst.* 2019;91:601-610.
37. Hua L, Zhang C, Peng T, Ji C, Nazir MS. Integrated framework of extreme learning machine (ELM) based on improved atom search optimization for short-term wind speed prediction. *Energy Convers Manag.* 2022;252:115102.
38. Qais MH, Hasanien HM, Alghuwainem S. Transient search optimization: a new meta-heuristic optimization algorithm. *Appl Intell.* 2020;50:3926-3941.
39. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw.* 2016;95:51-67.
40. Price KV, Awad NH, Ali MZ, Suganthan PN. The 100-digit challenge: problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. *Nanyang Technol Univ.* 2018;1:1-21.
41. Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. *Appl Math Comput.* 2009;214(1):108-132.
42. Zhao W, Wang L, Mirjalili S. Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications. *Comput Methods Appl Mech Eng.* 2022;388:114194.
43. Harifi S, Mohammadzadeh J, Khalilian M, Ebrahimnejad S. Giza pyramids construction: An ancient-inspired metaheuristic algorithm for optimization. *Evol Intell.* 2021;14(4):1743-1761.
44. Khishe M, Mosavi MR. Chimp optimization algorithm. *Expert Syst Appl.* 2020;149:113338.
45. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proc ICNN'95 - Int Conf Neural Netw.* Vol 4. IEEE; 1995:1942-1948.
46. Wang L, Ni H, Yang R, Pardalos PM, Du X, Fei M. An adaptive simplified human learning optimization algorithm. *Inf Sci.* 2015;320:126-139.
47. Binu D, Kariyappa BS. RIDENN: A new rider optimization algorithm-based neural network for fault diagnosis in analog circuits. *IEEE Trans Instrum Meas.* 2018;68(1):2-26.

Zhaoyang Lian *received a Ph.D. degree in Computer Science and Technology from Beijing University of Technology. Currently, he is a postdoctoral fellow in the School of Systems Science at Beijing Normal University. His email is lianzhaoyang@bnu.edu.cn.*

https://orcid.org/0009-0002-9121-1526

Bailu Si *is a professor in the School of System Science at Beijing Normal University, the Director of the Brain and Intelligent Robot Laboratory, a doctoral supervisor, and a national outstanding young researcher. His email is bailusi@bnu.edu.cn.*

https://orcid.org/0000-0002-0260-3433

Appendix file

Table A1. The C_{cons} and N_{iter_g} in different Julia graphs

Julia graph	$\boldsymbol{C_{cons}}$	$\boldsymbol{N_{iter_g}}$
01	$C_{cons}(1) = -0.4000 + 0.6000i$	$N_{iter_g}(1) = 200$
02	$C_{cons}(2) = -0.7017 + 0.3842i$	$N_{iter_g}(2) = 100$
03	$C_{cons}(3) = -0.8350 - 0.2321i$	$N_{iter_g}(3) = 100$
04	$C_{cons}(4) = -0.2365 - 0.6721i$	$N_{iter_g}(4) = 100$
05	$C_{cons}(5) = +0.2311 + 0.6068i$	$N_{iter_g}(5) = 50$
06	$C_{cons}(6) = -0.7322 - 0.2628i$	$N_{iter_g}(6) = 100$

Table A2. The significant p-value analysis of methods LA, MGNPOMC, and MGNPOMCP

Fun	LA and MGNPOMC				LA and MGNPOMCP			
	$\boldsymbol{P_{Prob>F}}$	$\boldsymbol{P_{flag}}$	F	$\boldsymbol{F_{flag}}$	$\boldsymbol{P_{Prob>F}}$	$\boldsymbol{P_{flag}}$	F	$\boldsymbol{F_{flag}}$
F1	1.45E-02	1	9.12E+00	1	1.45E-02	1	9.12E+00	1
F2	7.45E-03	1	1.18E+01	1	7.45E-03	1	1.18E+01	1
F3	4.91E-03	1	1.37E+01	1	4.91E-03	1	1.37E+01	1
F4	1.93E-04	1	3.65E+01	1	1.93E-04	1	3.65E+01	1
F5	5.07E-02	1	5.08E+00	1	5.07E-02	0	5.08E+00	0
F6	2.14E-03	1	1.81E+01	1	2.19E-03	1	1.79E+01	1
F7	6.18E-04	1	2.63E+01	1	6.27E-04	1	2.62E+01	1
F8	1.11E-03	1	2.22E+01	1	1.06E-03	1	2.25E+01	1
F9	1.07E-04	1	4.27E+01	1	1.07E-04	1	4.27E+01	1
F10	4.36E-04	1	2.91E+01	1	4.36E-04	1	2.91E+01	1
F11	1.59E-05	1	6.95E+01	1	1.59E-05	1	6.95E+01	1
F12	5.30E-03	1	1.33E+01	1	5.30E-03	1	1.33E+01	1
F13	1.27E-04	1	4.08E+01	1	1.30E-04	1	4.06E+01	1
F14	1.00E+00	1	-2.32E+00	1	9.76E-01	0	9.64E-04	0
F15	1.54E-01	1	2.42E+00	1	6.22E-01	0	2.60E-01	0
F16	2.32E-02	1	7.46E+00	1	2.32E-02	1	7.45E+00	1
F17	1.00E+00	1	1.67E-13	1	1.00E+00	0	1.32E-12	0
F18	4.74E-02	1	5.27E+00	1	4.99E-02	1	5.12E+00	1
F19	5.83E-02	1	4.70E+00	1	5.83E-02	0	4.70E+00	0
F20	4.68E-01	1	5.75E-01	1	4.68E-01	0	5.75E-01	0
F21	3.98E-02	1	5.77E+00	1	1.96E-03	1	1.86E+01	1
F22	3.57E-02	1	6.09E+00	1	3.24E-02	1	6.39E+00	1
F23	1.09E-01	1	3.16E+00	1	2.40E-02	1	7.35E+00	1
cecF1	4.16E-04	1	2.95E+01	1	4.32E-04	1	2.92E+01	1
cecF2	1.82E-05	1	6.73E+01	1	2.71E-05	1	6.08E+01	1
cecF3	8.33E-02	0	3.79E+00	0	1.03E-01	0	3.29E+00	0
cecF4	2.04E-03	1	1.84E+01	1	8.52E-03	1	1.12E+01	1
cecF5	1.91E-06	1	1.16E+02	1	1.81E-07	1	2.02E+02	1
cecF6	3.60E-04	1	3.07E+01	1	2.31E-05	1	6.34E+01	1
cecF7	1.25E-03	1	2.14E+01	1	2.41E-03	1	1.74E+01	1
cecF8	3.71E-02	1	5.98E+00	1	3.22E-02	1	6.40E+00	1
cecF9	4.98E-02	1	5.13E+00	1	7.90E-02	0	3.92E+00	0
cecF10	3.11E-06	1	1.03E+02	1	4.49E-06	1	9.47E+01	1
Count		32		32		25		25

Table A3. The average results comparison of different swarm intelligence algorithms with best hunter crossover and migration strategies

Fun	LA(Ref)[2]	MGNPOMC						
		01	02	03	04	05	06	MGNPOMCP
F1	-5.00E+00	-5.00E+00	-5.00E+00	-5.00E+00	-5.00E+00	-5.00E+00	-5.00E+00	-5.00E+00
F2	7.87E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	7.53E+02	2.49E-02	1.41E-02	3.44E-02	1.45E-02	2.82E-02	2.43E-02	1.26E-02
F4	2.23E+02	3.16E-03	1.11E-03	3.20E-03	3.38E-03	2.07E-03	3.88E-03	2.29E-03
F5	8.58E-01	2.55E-03	2.29E-03	2.47E-03	2.44E-03	2.23E-03	2.60E-03	2.25E-03
F6	7.62E-02	7.62E-02	1.42E-05	7.62E-02	8.02E-02	7.62E-02	2.06E-05	7.62E-02
F7	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
F8	2.11E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F9	4.87E+00	2.38E-01	4.12E-01	3.31E-01	2.49E-01	1.86E-01	3.35E-01	2.73E-01
F10	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01
F11	-1.98E+02	-2.08E+02	-2.09E+02	-2.09E+02	-2.09E+02	-2.08E+02	-2.09E+02	-2.08E+02
F12	2.85E+00	3.91E-26	2.55E-38	1.78E-32	7.00E-27	6.17E-37	1.88E-23	1.85E-19
F13	6.33E+01	2.74E-04	1.52E-04	1.10E-04	3.95E-04	1.63E-04	6.21E-05	5.90E-05
F14	6.21E+04	3.69E-02	2.78E-02	2.13E-02	1.92E-02	2.66E-02	2.57E-02	3.17E-02
F15	1.29E+04	4.68E+00	1.00E+01	3.40E+00	4.75E+00	4.47E+00	2.82E+00	4.06E+00
F16	1.77E+05	2.91E+01	2.90E+01	2.89E+01	2.90E+01	2.92E+01	2.91E+01	2.91E+01
F17	1.96E+03	6.98E-01	6.91E-01	7.28E-01	7.19E-01	6.87E-01	7.06E-01	6.92E-01
F18	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
F19	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
F20	9.71E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F21	1.03E-05	2.19E-06	2.59E-06	9.08E-06	2.43E-06	2.94E-06	6.89E-06	3.34E-06
F22	1.62E+02	1.35E-02	7.46E-03	5.65E-03	9.32E-03	1.60E-02	1.02E-02	1.12E-02
F23	-6.91E+03	-8.90E+03	-8.75E+03	-8.91E+03	-8.75E+03	-8.80E+03	-8.75E+03	-8.41E+03
F24	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00	-1.80E+00
F25	-3.96E+00	-4.68E+00	-4.68E+00	-4.69E+00	-4.68E+00	-4.68E+00	-4.69E+00	-4.68E+00
F26	-6.32E+00	-9.44E+00	-9.43E+00	-9.47E+00	-9.40E+00	-9.50E+00	-9.38E+00	-9.36E+00
F27	9.91E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F28	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
F29	8.64E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F30	1.97E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F31	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02
F32	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
F33	1.09E-02	6.38E-04	6.64E-04	2.58E-03	6.83E-04	6.56E-04	6.17E-04	6.95E-04
F34	-9.40E+00	-6.62E+00	-6.62E+00	-5.60E+00	-6.11E+00	-6.62E+00	-6.38E+00	-5.87E+00
F35	-9.06E+00	-4.96E+00	-4.47E+00	-4.24E+00	-5.20E+00	-4.62E+00	-6.29E+00	-5.01E+00
F36	-7.85E+00	-6.84E+00	-7.69E+00	-9.86E+00	-8.02E+00	-7.65E+00	-7.11E+00	-7.65E+00
F37	4.43E-01	1.03E+00	4.86E-01	8.56E-01	1.06E+00	6.29E-01	6.29E-01	6.53E-01
F38	1.42E-02	3.38E-02	2.28E-02	4.01E-02	3.13E-02	3.49E-02	4.21E-02	2.42E-02
F39	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
F40	-3.24E+00	-3.23E+00	-3.23E+00	-3.23E+00	-3.23E+00	-3.23E+00	-3.23E+00	-3.23E+00
F41	5.89E+00	4.40E-02	4.73E-02	8.70E-02	5.33E-02	4.78E-02	3.33E-02	3.60E-02
F42	1.78E+01	4.39E-02	2.75E-02	2.85E-02	3.22E-02	4.09E-02	3.51E-02	3.98E-02
F43	3.46E+04	1.07E-02	7.64E-03	2.22E-02	7.60E-03	9.61E-03	1.27E-02	8.34E-03
F44	3.29E+04	1.92E-01	1.26E-01	1.25E-01	1.33E-01	1.59E-01	1.59E-01	1.71E-01
F45	-1.08E+00	-1.08E+00	-1.08E+00	-1.08E+00	-1.08E+00	-1.08E+00	-1.08E+00	-1.08E+00
F46	-9.17E-01	-9.83E-01	-7.89E-01	-9.07E-01	-1.03E+00	-9.04E-01	-1.07E+00	-9.98E-01
F47	-4.22E-01	-4.78E-01	-5.48E-01	-4.78E-01	-4.78E-01	-4.78E-01	-4.78E-01	-4.78E-01
F48	1.23E-03	2.02E-05	3.07E-05	2.76E-05	2.27E-05	2.90E-05	3.76E-05	3.65E-05
F49	4.45E+02	9.05E+01	2.98E+02	9.03E+01	8.95E+01	9.34E+01	9.06E+01	8.96E+01
F50	9.30E+03	8.06E+03	8.04E+03	7.33E+03	6.62E+03	3.86E+03	7.30E+03	4.54E+03
VS(win)	5				44			
VS(win)	7							42