Received 14 July 2022, accepted 19 July 2022, date of publication 21 July 2022, date of current version 5 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3193094

RESEARCH ARTICLE

Research on Multiaircrafts Cooperative Arraying to Jam Based on Multiobjective Moth-Flame Optimization Algorithm

MINGXI MA[1], **JUN WU**[1], **YUE SHI**[2], **LONG YAN**[3], **AND WEI LU**[4]

[1]Air Control and Navigation College, Air Force Engineering University, Xi'an 710051, China
[2]Equipment Management and UAV Engineering College, Air Force Engineering University, Xi'an 710051, China
[3]93501 Army, Beijing 102600, China
[4]493735 Army, Beijing 102600, China

Corresponding author: Yue Shi (3159098298@qq.com)

ABSTRACT The problem of cooperative arraying to jam is an important part of EW mission planning. Aiming at the problem that multi-objective optimization algorithm is easy to fall into local optimum and converge in three-objective optimization, a multi-aircraft jamming and cooperative arraying method based on improved multi-objective Moth-flame optimization algorithm is proposed. Firstly, the simulation environment is established by using digital elevation map and radar detection model. Then, based on the multi-objective Moth-flame optimization algorithm, the population initialization is completed by using Logistic-Tent chaotic map, which increases the diversity and uniformity of the solution and improves the search ability of the algorithm; Then, the decision factor and Gaussian difference mutation are introduced, which makes the algorithm not only accept the current solution with a certain probability, but also jump out of the current solution and search again according to the disturbance, thus enhancing the search ability of the algorithm; Finally, by comparing with NSGA-II, MOEA/D, MOPSO and NSMFO algorithms on test functions of ZDT and DTLZ series, the performance of the algorithm is verified, and it is proved that multi-objective Moth-flame optimization algorithm is better than other algorithms in both convergence and diversity. In addition, compared with the NSMFO and MOEA/D algorithms in the arraying simulation experiment. The values of the interference power, the width of the route safety zone and the detection area of the radar obtained by the algorithm in this paper, are 117.9 kw, 46 km, and 1727 km^2. Compared with the results of the other two algorithms, the effectiveness of interference is improved by 39.8%, 22.8% and 41.9% respectively.

INDEX TERMS Cooperative arraying to jam, multi-objective optimization, multi-objective Moth-flame optimization algorithm, chaotic mapping, Gaussian difference mutation, widespread reference points.

I. INTRODUCTION

In the current modern and information-based combat environment, the development of radar is changing with each passing day, and its capabilities such as operating range, azimuth angle, elevation angle, resolution and maneuverability are constantly improving [1]–[5], which increases the difficulty for combat aircraft to perform penetration and strike tasks. In order to successfully complete the tasks of penetrating and strike, it is necessary to have jammers to jam the radar, so as to build a safe area suitable for combat aircraft to perform tasks. Because the current radar network construction is intertwined, it is difficult to meet the needs of combat tasks by using only a single jammer to jam the radar network. Therefore, multiple jammers must be used to cooperate and complete the jamming tasks together. Then, in the process of executing the jamming task, how to make the jammer's operational efficiency higher and the operational design more

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

reasonable has formed the problem of multi-aircraft jamming and cooperative arraying.

To solve this problem, scholars at home and abroad have carried out a certain degree of research, but most of the research focuses on the allocation of jamming resources, and the research on jammer 's arraying is relatively few [6]–[7]. The most widely used method is multi-objective intelligent optimization algorithm.

In real life, solving problems often requires not only meeting one goal, so multi-objective optimization model arises at the historic moment. In 1906, Vilfreto Pareto put forward the famous Pareto optimization theory: "The optimal allocation of the resources of a society is not attached so long as it is possible to make at last one independent better off in his own estimation while keeping others as well off as before in their own estimation. "With the wide spread of Pareto theory, multi-objective intelligent optimization algorithms have been continuously proposed and widely used in engineering fields. Common multi-objective intelligent optimization algorithms include Non-dominated Sorting Genetic Algorithm-II algorithm (NSGA-II), Multiple Objective Particle Swarm Optimization (MOPSO), Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) and Non-dominated sorting moth flame optimization (NSMFO) proposed in recent years.

NSGA-II algorithm was proposed by Kalyanmoy Deb and his students in 1994. It adopts the strategies of"non-dominated quick sorting, congestion ranking and introducing elites", which not only reduces the complexity of the algorithm, but also improves the accuracy of the algorithm and ensures the diversity of the population [8]. At the same time, due to its slow convergence speed and easy to fall into local optimum in the later stage of the algorithm, some scholars have improved this kind of problem. Yang Shuai *et al.* put forward that a novel crossover operator is designed to generate new offspring, and two mutation operators are proposed so that the search process can jump out of the local optimal [9].

MOPSO algorithm was proposed by Coello in 2002. The core features of MOPSO algorithm are non-dominated ranking, congestion ranking and adaptive grid method when updating [10]. However, due to the problem that it is easy to fall into local optimum and the population diversity cannot be effectively guaranteed in the later period, some scholars have carried out research on it. Lingjie Li and others put forward a new hybrid multi-objective particle swarm optimization method (HMOPSO-ARA), which proposed adaptive resource allocation strategy, evolutionary search based on files, selection strategy based on decomposition and a new speed update formula, which effectively solved the problems of local optimum and population diversity [11].

MOEA/D algorithm was proposed by Zhang Qingfu and Li Hui in 2006, which creatively decomposes the multi-objective optimization problem and greatly reduces the complexity of the algorithm [12]. The algorithm utilizes the neighborhood relationship among sub-problems and optimizes them simultaneously in a cooperative way. Usually, the neighborhood relationship among sub-problems is determined by calculating the Euclidean distance between their weight vectors. However, at the same time, MOEA/D algorithm has the phenomena of repeated solutions and uneven distribution in the later stage, and can't balance the convergence and population diversity of the algorithm well. Therefore, some scholars have carried out research on this. Xin Zhou and others put forward that "The discretized reproduction and adaptive neighborhood provide a larger search range in solution space to overcome difficulties in duplication and uneven distribution of solutions. Adaptive decomposition method and improved hybrid environment selection promote solutions converge to the optimal direction and further balance convergence and diversity."[13].

TABLE 1. Algorithm comparison.

Algorithm	Advantages	Disadvantages
NSGA-II	Low - dimensional multi - objective optimization has good effect, low computational cost and fast convergence[13]	Crowding distance is not applicable in high dimensional space and the computational complexity is high
MOPSO	The early stage of the search speed, low implementation difficulty	Easy to fall into local optimum
MOEA/D	Fast convergence speed and low computational complexity	Convergence rate and population diversity cannot be well balanced
NSMFO	Less parameters, low computing complexity and global development ability is strong, fast convergence speed	Crowding distance is not applicable in high dimensional space and the computational complexity is high

NSMFO algorithm was proposed by Vimal Savsani *et al.* in 2017, It introduces the elite non-dominant ranking method of NSGA-II algorithm and the crowding ranking method of keeping population diversity, which solves the problem that MFO algorithm cannot solve multi-objective optimization [14]. At the same time, the algorithm performance test and engineering experiment simulation are carried out, and the experimental results show that the performance is better than many classical multi-objective optimization algorithms. However, due to the introduction of NSGA-II algorithm, the algorithm is still unable to deal with the three-objective problem.

To sum up, this paper selects NSMFO algorithm and improves it from the following three aspects:

(1) Firstly, the diversity of initial population is increased by Logistic-Tent chaotic map;

(2) The decision operator is introduced to allow the algorithm to accept the current solution with a certain probability or disturb the moth when the position is updated by Gaussian difference mutation to produce a new solution, which improves the optimization ability of the algorithm;

(3) Using widespread reference points instead of crowded distance sorting, the problem that the algorithm can't

converge on three objectives due to the complexity of computation is solved;

(4) The improved algorithm is tested on ZDT series and DTLZ series, and compared with NSGA-II algorithm, MOEA/D algorithm, MOPSO algorithm and NSMFO algorithm. The experimental results show that LTG-NSMFO algorithm is superior to other algorithms in convergence and diversity

(5) The improved algorithm is used to simulate the multi-aircraft jamming and cooperative arraying problem, and the Pareto front of the problem is obtained. Then, two kinds of jamming effects and jamming positions are given when the jamming power is minimum and the route safety interval is maximum.

The rest of the paper is organized as follows: In the second section, the model of multi-aircraft jamming and cooperative arraying is introduced. The third section introduces the algorithm and its improvement process. In the fourth section, the benchmark function is used to compare the performance of the algorithm. In the fifth section, the simulation experiment analysis of multi-aircraft jamming and cooperative arraying is carried out. Finally, the sixth section gives the conclusion.

II. MODEL CONSTRUCTION

A. RADAR DETECTION MODEL UNDER TERRAIN SHADING

1) DIGITAL ELEVATION MODEL

Digital Elevation Model (DEM) refers to the way of simulating the degree of ground undulation with limited elevation data, which is generally divided into two types: Grid structure and contour structure. In order to facilitate the construction and the solution of subsequent model, Grid structure is adopted in this paper.

2) RADAR DETECTION MODEL

Radar detection area is the most intuitive index when describing radar performance, and radar detection area is directly determined by radar detection radius. Usually, we use radar equation to characterize radar detection model, and characterize radar detection radius as shown in formula 1:

$$R = \left(\frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 (SNR) F_n k T_o B_n L} \right)^{1/4} \tag{1}$$

R is the detection radius of radar; P_t is the maximum value of radar transmission power; G is the radar antenna gain; σ is the radar cross-sectional area of the target; λ is emission wave length for radar; SNR is radar signal-to-noise ratio; F_n is noise figure; k is the Boltzmann constant, $k = 1.38 \times 10^{-23} J/K$; T_0 is the Ambient Kelvin temperature for radar operation; B_n is the noise broadband; L is the system loss.

3) RADAR DETECTION RANGE UNDER TERRAIN SHADE

The radar detection range is usually spherical or hemispherical, but due to the influence of terrain, the radar detection range is obscured in some directions.

For example, in the figure 1, the enemy radar position coordinates are (x_0, y_0, z_0), the position coordinates of our combat aircraft are (x_1, y_1, z_1), the terrain position coordinates at a certain altitude are (x_2, y_2, y_2), the included angle of radar observation targets is α_0, and the included angle of radar observation terrain is α_1. If $\alpha_0 > \alpha_1$, the radar does not have terrain shade in this direction; Otherwise, the radar cannot detect the target aircraft.

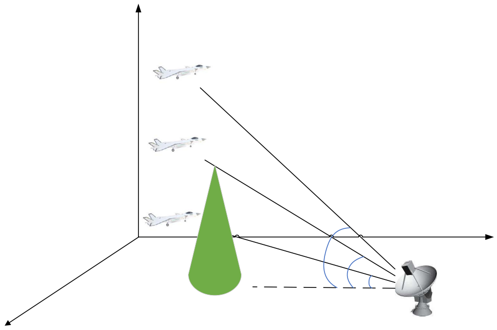


FIGURE 1. Radar detection diagram.

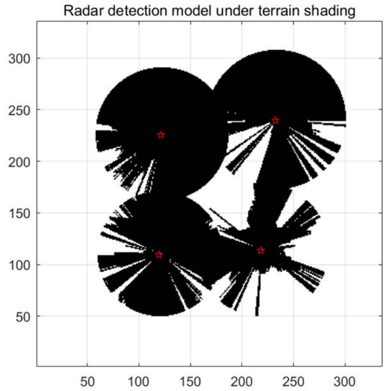


FIGURE 2. Radar detection model under terrain shading.

$$\alpha_0 = \arctan \frac{(z_1 - z_0)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}} \tag{2}$$

$$\alpha_1 = \arctan \frac{(z_2 - z_0)}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} \tag{3}$$

According to the formula 2 and formula 3, every grid in the digital elevation map is traversed, and if $\alpha_0 < \alpha_1$ exists in one grid, there is terrain shading. Therefore, the radar detection range under terrain shade is shown in the figure 2:

B. ELECTRONIC WARFARE JAMMING MODEL

1) JAMMING POWER

In the electronic warfare jamming model, the detection range of radar will change because of jamming. In the case of jamming, the expression of power received by radar receiver is shown in formula 4:

$$P_{radar\_jammer} = \frac{P_j G_j G(\varphi) r_j B_r}{(4\pi)^2 R_j^2 L_j B_j} \tag{4}$$

In which, P_{radar_jammer} represents the received power of the radar receiver; P_j indicates the jamming power of the

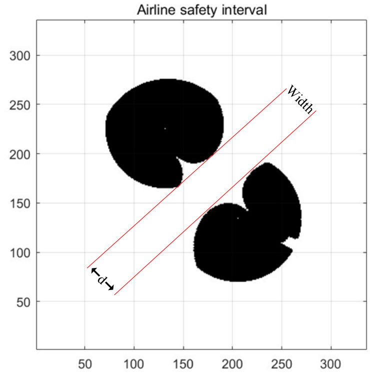


FIGURE 3. Airline safety interval.

jammer; G_j indicates the antenna gain of the jammer in the jamming direction; $G(\varphi)$ represents the gain of the radar antenna in the jamming direction of the jammer, and φ represents the included angle between the main lobe direction of the radar antenna and the jammer; r_j indicates the polarization loss of the jammer; B_r is the signal bandwidth of radar receiver; R_j is the distance between radar and jammer; L_j is the Comprehensive loss of jamming signals generated for jammer; B_j is the bandwidth for jammer signals.

The gain expression of radar antenna in jamming direction of jammer $G(\varphi)$ is shown in formula 5:

$$G(\varphi) = \begin{cases} G & 0 \leq |\varphi| \leq \varphi_{0.5} \\ K\left(2\left(\varphi_{0.5}/\varphi\right)\right)^2 G & \varphi_{0.5} \leq |\varphi| \leq 90^\circ \\ K\left(2\left(\varphi_{0.5}/90^\circ\right)\right)^2 G & 90^\circ \leq |\varphi| \leq 180^\circ \end{cases} \tag{5}$$

In which, K is a constant, usually taken as (0.04, 0.1); G is the gain for enemy radar antenna; $\varphi_{0.5}$ is the lobe width of enemy radar antenna at half power point.

Based on the above formula, the expression of detection radius of enemy radar under the jamming of a single jammer is shown in formula 6:

$$R = \left(\frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 L F_n k T_0 B_n K_j + \frac{4\pi L P_j G_j G(\varphi) r_j B_r}{B_j L_j R_j^2}}\right)^{1/4} \tag{6}$$

Among them, K_j is enemy radar jamming coefficient, and other parameters are consistent with the symbolic meanings in the above formula.

2) ROUTE SAFETY INTERVAL

Through the jamming of enemy radar by jammers, our purpose is to build a route channel that can make combat aircraft pass safely, so as to provide a safe search space for subsequent route planning. Therefore, the width of safe channel is used as one of the evaluation criteria of jamming effect. It is shown in the figure 3:

3) THE AREA OF RADAR DETECTION

The detection area of radar is the most intuitive index of radar detection capability. Generally speaking, radar is deployed near an important asset, and its coverage area directly affects the security of the assets it protects. At the same time, it also affects the warning response time of the assets to the advent of danger. Therefore, this paper takes the detection area of radar as the third objective function, aiming to achieve better suppression effect by minimizing the detection area of radar as much as possible.

C. MULTI-OBJECTIVE OPTIMIZATION MODEL OF JAMMING ARRAY

A typical multi-objective optimization problem is expressed by mathematical expression as formula 7:

$$\begin{aligned} & Z = F(X) = \begin{Bmatrix} \max(\min) f_1(x) \\ \max(\min) f_2(x) \\ \vdots \\ \max(\min) f_n(x) \end{Bmatrix} \\ & s.t.\phi(X) = \begin{Bmatrix} \varphi_1(x) \\ \varphi_1(x) \\ \vdots \\ \varphi_1(x) \end{Bmatrix} \leq Q = \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{Bmatrix} \\ & \forall x_i \in \Omega \end{aligned} \tag{7}$$

Among them, $Z = F(X)$ is the objective function; $s.t.\phi(X)$ is a constraint condition, including equality constraint or inequality constraint; $\forall x_i \in \Omega$ is the decision space for variables. In the final analysis, the problem of cooperative array of multi-aircraft jamming is also a multi-objective optimization problem, aiming at jamming enemy radar positions, obtaining route safety zone and ensuring the safety of jammers to a certain extent.

Therefore, the multi-objective optimization model in this paper is constructed as formula 8, as shown at the bottom of the next page.

The first objective function: $f_1(x) = \min\left(Width_{route-safety-zone}\right)$ represents the width of the route safety zone. In order to solve the problem conveniently, Firstly, the coordinates of the center point of the enemy radar detection model at a certain height section are calculated, and a straight line parallel to the x axis through the center point is made. Then the distance from each point in the radar detection range to the straight line after jamming is calculated, take the nearest point in all distances, and maximize the minimum distance through algorithm iteration, thus ensuring that the width of the route safety zone meets the task requirements.

The second objective function: $f_2(x) = \sum_{j=1}^{n} \frac{P_j G_j G(\varphi) r_j B_r}{(4\pi)^2 R_j^2 L_j B_j}$ represents the sum of jamming power of each jammer to enemy radar. Because the jamming effect of jammer on enemy radar has an upper limit, the jamming effect can be continuously enhanced by increasing the jamming power blindly. When the jamming power reaches the threshold, the jamming effect will no longer increase, but the excessive jamming power will cause the waste of jammer's own resources and increase the probability of jammer being discovered.

Therefore, this paper aims to save jamming resources and ensure the safety of jammers, and chooses minimizing jamming power as the second objective function.

The third objective function: $f_3(x) = \min \left(S_{Radar\_detection\_area}\right)$ represents the detection area of the enemy radar. In order to better achieve the effect of suppression through jamming and ensure the tasks that need to be completed after the suppression is formed, this paper selects the detection area of the radar as the objective function, minimizes the detection area of the radar, and achieves the purpose of improving the safety of subsequent tasks.

III. NSMFO ALGORITHM

A. BASIC MOTH-FLAME OPTIMIZATION ALGORITHM

Moth-flame optimization algorithm (MFO) is a new intelligent optimization algorithm proposed by Seyedali Mirijalili and others in 2015, which is mainly used to solve single-objective optimization problems [15]. The basic Moth-flame optimization algorithm population initialization, moth and flame position update formula is shown in formula 9:

$$
\begin{aligned}
M &= \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1d} \\ m_{21} & m_{22} & \cdots & m_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nd} \end{bmatrix}_{n\times d} \quad M_F = \begin{bmatrix} M_{f1} \\ M_{f2} \\ \vdots \\ M_{fn} \end{bmatrix}_{1\times n} \\
F &= \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1d} \\ f_{21} & f_{22} & \cdots & f_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nd} \end{bmatrix}_{n\times d} \quad F_F = \begin{bmatrix} F_{f1} \\ F_{f2} \\ \vdots \\ F_{fn} \end{bmatrix}_{1\times n}
\end{aligned} \tag{9}
$$

In the above formula, n is the number of moths and d is the characteristic dimension of moths are expressed. M is the initialization population of moth and M_F is the fitness matrix of moth population, which is obtained by solving the value of population on the objective function; F is the initialized population of flame and F_F is the fitness matrix of flame, which is obtained by sorting the fitness of moths, that is, the matrix F_F is the result of sorting the matrix M_F in ascending order, which also shows that flame F is the optimal solution of moths M in the current iterative search. The initial flame set is the same size as the moth population set, but it gradually decreases with the subsequent iteration, and its adaptive renewal mechanism is shown in formula 10:

$$
flameno = round\left(N - iteration \times \frac{N-1}{\max\_iteration}\right) \tag{10}
$$

N represents the maximum number of flames, *iteration* is the current iteration times and max *_iteration* is the maximum iteration times of the algorithm. The change of flame number with the iteration times of the algorithm is shown in the figure 4:

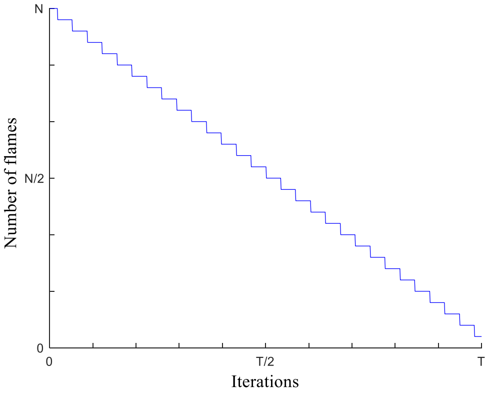


FIGURE 4. Flame adaptive update strategy.

MFO algorithm adopts the flying principle of moths under artificial light source. According to the phototaxis of moths, the flight mode of moths is fitted by logarithmic spiral curve, and moths M_i move around their corresponding flames F_i in logarithmic spiral curve. The updated formula for defining moth position is shown in formula 11

$$
S\left(M_i, F_j\right) = D_i \cdot e^{bt} \cdot \cos\left(2\pi t\right) + F_j \tag{11}
$$

Among them, $S\left(M_i, F_j\right)$ is the updated position of moths M_i, b is a constant parameter, and its value determines the shape of logarithmic equiangular spiral, usually taken as 1; Parameter t is the random numbers in the range $[-1, 1]$, and their values control the distance between moths and flames. The smaller the value t is, the closer the distance between moths and flames. By changing the values of parameters, the position where moths finally arrive around flames is adjusted, which reflects the local optimization ability of the algorithm; $D_i = \left|M_i - F_j\right|$ Is the Euclidean distance between the flame j and the moth i: The moth's trajectory is shown in the figure 5:

$$
\begin{aligned}
Z = F(X) &= \left\{ \begin{array}{l} \max f_1(x) = \min\left(Width_{route-safety-zone}\right) \\ \min f_2(x) = \sum_{j=1}^{n} \frac{P_j G_j G(\varphi) r_j B_r}{(4\pi)^2 R_j^2 L_j B_j} \\ \min f_3(x) = \min\left(S_{Radar\_detection\_area}\right) \end{array} \right\} \\
s.t.\phi(X) &= \left\{ \begin{array}{l} \varphi_1(x) = sqrt\left(\left(x_j - x_{0i}\right)^2 + \left(y_j - y_{0i}\right)^2 + \left(z_j - z_{0i}\right)^2\right) \geq R \\ \varphi_2(x) = \left\{z_{\min} \leq z_j \leq z_{\max}\right\} \end{array} \right\} \\
x &= 1, 2, 3 \ldots N_{radar}; j = 1, 2, 3 \ldots N_{jammer}
\end{aligned} \tag{8}
$$

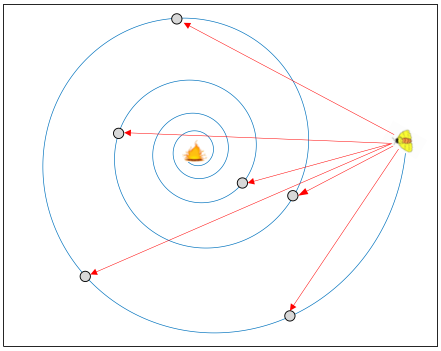


FIGURE 5. The trajectories of moths[16].

B. NON-DOMINANT SORTING MOTH FLAME OPTIMIZATION ALGORITHM

Non-dominant sorting moth flame optimization (NSMFO) takes multiple objective functions as optimization objects, and optimizes these objective functions in order to achieve the optimal state at the same time. Compared with MFO algorithm, NSMFO algorithm introduces the non-dominated quick sorting and congestion distance of NSGA-II algorithm to optimize, which makes the algorithm have the ability to deal with multi-objective problems. However, due to its population initialization, And the updating mechanism has not changed, and there are still some problems such as it is easy to fall into local optimum. At the same time, due to the screening method of crowding distance, the algorithm is prone to repeat search and over-reservation on the boundary, which makes the algorithm slow or even unable to converge when dealing with the three-objective problem. Therefore, the following optimization methods are proposed to optimize NSMFO algorithm.

1) INITIALIZATION OF LOGISTIC-TENT CHAOTIC MAP

In the field of optimization, chaotic mapping is widely used in intelligent optimization algorithms because of its ergodicity, randomness and nonlinearity [17]. At present, Logistic chaotic map and Tent chaotic map are commonly used [18]. Logistic chaotic map has strong global ergodicity, but its iterative sequence distribution is not well-distributed, while Tent chaotic map has good ergodicity [19]–[20]. Therefore, combining the characteristics of both, a Logistic-Tent chaotic model is proposed to improve the NSMFO algorithm. Logistic-Tent chaotic map formula is shown in formula 12:

$$x_{n+1} = \begin{cases} \left[r x_n \left(1 - x_n\right) + \frac{(4-r)}{2} x_n \right] \bmod 1 & \mathit{if}\ x_n < 0.5 \\ \left[r x_n \left(1 - x_n\right) + \frac{(4-r)(1-x_n)}{2} \right] \bmod 1 & \mathit{if}\ x_n \geq 0.5 \end{cases} \tag{12}$$

The distribution of Logistic-Tent chaotic map is shown in the figure 6:

As shown in Figure 6 and table 2, compared with the "Rand" initialization value, the distribution of the Logistic-Tent chaos map is more well-distributed. Therefore, the Logistic-Tent chaos map can make the distribution of the population in a given range more well-distributed, thus enhancing the global search ability of the algorithm.

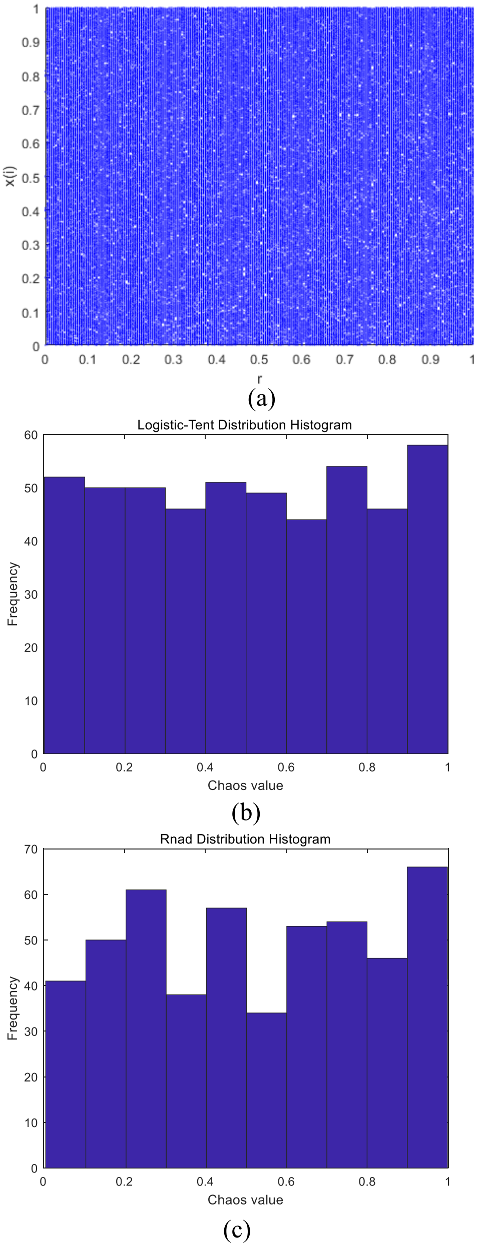


FIGURE 6. (a) Logistic-Tent chaotic map distribution. (b) and (c) Distribution histograms of the values of logistic-tent chaotic initialization and Rand initialization in the interval 0–1.

TABLE 2. Comparison of initialization methods.

Initialization Mode	Average	Variance
Logistic-Tent	0.5046	0.0817
Rand	0.5204	0.0857

2) GAUSSIAN DIFFERENTIAL VARIATION

Because he moth in NSMFO algorithm uses logarithmic equiangular spiral to change its position and its curve characteristics, although the convergence speed of NSMFO algorithm is fast, it is easy to fall into local optimum at

TABLE 3. Gaussian difference variation pseudocode.

```
Gaussian difference variation pseudocode
Algorithm 1 Gaussian difference variation pseudocode
Step 1：Generating decision factor $\xi$
Step 2：Generate random moths and extract the optimal position of moths
Step 3：Cycle to compare moth positions
If rand < $\xi$
Using gaussian difference variation to the current position of moths to disturbance
else
The location updates normally
end
```

the same time. Therefore, Gaussian difference variation is introduced to disturb the position of moths. At the same time, a decision operator is introduced, which allows the algorithm to accept the current inferior solution with a certain probability, so that the algorithm can jump out of the local optimal solution and search for the global optimal solution.

Let one of the moths, each moth $M = [m_{11}, m_{12}, \ldots, m_{1d}]$ chooses whether to produce variation according to the judgment factor, and suppose that the $i - th$ moth produces Gaussian difference variation, and the expression is shown in formula 13:

$$X(t+1) = \omega_1 \cdot f_1 \cdot (X_{best} - X(t)) + \omega_2 \cdot f_2 \cdot (X_{rand} - X(t)) \qquad (13)$$

where the ω_1 and ω_2 are the weight coefficients; f_1 and f_2 represent generating Gaussian distribution random numbers with mean value of 0 and variance of 1; $X(t+1)$ is the new position after Gaussian difference mutation; X_{best} is the optimal moth position in this iteration; X_{rand} is an arbitrarily selected moth position; $X(t)$ is the current position of the moth.

The Gaussian differential mutation pseudo code is shown in table 3:

3) WIDELY DISTRIBUTED REFERENCE POINTS

Widespread reference points are a group of well-distributed points in the decision space [21]. The decision space is evenly divided by reference vectors connected with ideal points. The diversity of the algorithm is ensured by calculating the distance between individuals in the population and each reference vector.

(1) Establish a reference point. Firstly, the reference points are established on the hyperplane of $(M-1)$ dimensions, in which, M represents the dimension of the target space and P is the number of copies to be divided, H is the number of reference points is determined by both of them. It is shown in formula 14:

$$P = C^{H}_{M+H-1} \qquad (14)$$

The set of reference points is. $S_j = \left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}$, $\sum_{j=1}^{M} S_j = 1$,

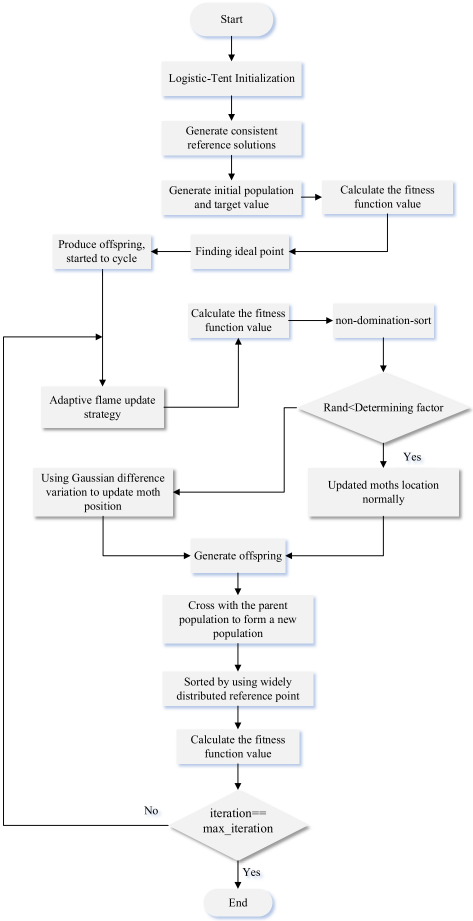

Start
Logistic-Tent Initialization
Generate consistent reference solutions
Generate initial population and target value
Calculate the fitness function value
Finding ideal point
Produce offspring, started to cycle
Calculate the fitness function value
non-domination-sort
Adaptive flame update strategy
Rand<Determining factor
Yes
Using Gaussian difference variation to update moth position
Updated moths location normally
Generate offspring
Cross with the parent population to form a new population
Sorted by using widely distributed reference point
Calculate the fitness function value
No
iteration==max_iteration
Yes
End

FIGURE 7. LTG-NSMFO algorithm pseudocode.

(2) Determine the position coordinates of the reference point. When determining the position reference point coordinates, a set of $(M-1)$ dimensions named x is first constructed, $x \in \left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H+M-2}{H}\right\}$ then updated x according to pairs $x_{ij} = x_{ij} - \frac{j-1}{H}$, and finally the reference point position coordinates are obtained.

(3) Conduct adaptive normalization of individual population. Firstly, the ideal point of the population $z = \left(z_1^{\min}, z_2^{\min}, \ldots, z_n^{\min}\right)$ is constructed, and then the objective function is transformed according to $f_i'(x) = f_i(x) - z_i^{\min}$. Then, the extra points $ASF(x, \omega)$ corresponding to each coordinate axis are obtained by solving, and then the hyperplane is constructed. It is shown in formula 15:

$$ASF(x, \omega) = \max_{i=1}^{M} f_i'(x) \big/ \omega_i \qquad (15)$$

Finally, the intercept between hyperplane and coordinate axis is used to normalize the objective function value adaptively. It is shown in formula 16:

$$f_i^n(x) = \left(f_i'(x) - z_i^{\min}\right) \Big/ \left(a_i - z_i^{\min}\right) \qquad (16)$$

a_i represents the intercept between the hyperplane and the coordinate axis. And the function value of the point of intersection between the normalized plane and the coordinate

TABLE 4. LTG-NSMFO algorithm flow.

```
LTG-NSMFO algorithm pseudocode
Algorithm 2 The pseudocode of LTG-NSMFO
1. Initializing of parameters
2. Initializes the moth population using Logistic-Tent chaos
mapping according to Equation (12)
3. Generate consistent reference solutions
4. Generate initial population and target value
5. Calculate the fitness function value
6. Finding ideal point
7. Produce offspring, started to cycle
8. for i=1: max_iteration
9.     Adaptive flame update strategy according to Equation (10)
10.    Calculate the fitness function value
11.    Non-domination-sort
12.    If rand < $\xi$
13.    Using gaussian difference variation to the current position
of moths to disturbance according to Equation (13)
14.    else
15.    Updated moths location normally
16.    end
17.    Generate offspring
18.    Cross with the parent population to form a new population
19.    Sorted by using widely distributed reference point
20.    Calculate the fitness function value
21.end
22. Return the value of the decision variables and objective
functions
```

axis is 1, then the point on the normalized hyperplane are satisfied $\sum_{i=1}^{m} f_i^n = 1$

(4) Construct the distance relationship between reference point and individuals in the population. We connect the reference point with the origin and construct the ray. The distance between the iindividuals in the population and the ray is regarded as the distance relationship between them. The smaller the distance, the more the population can be selected.

(5) Select the fewer reference points as the basis for individual selection of the last frontier population, and then realize the selection in accordance to the level of front $(l-1)$ frontier and the number of selected reference points.

C. LTG-NSMFO ALGORITHM FLOW

The flow chart of LTG-NSMFO algorithm is shown in the figure 7:

The pseudocode for the LTG-NSMFO algorithm is shown in the table 4:

IV. EXPERIMENTAL VERIFICATION AND ANALYSIS

A. TEST FUNCTIONS

In order to test the effectiveness of this algorithm, eight test functions are selected to test the performance of the algorithm, including five double-objective test functions of ZDT series: ZDT1, ZDT2, ZDT3, ZDT5 and ZDT6; And three three-objective test functions of DTLZ series: DTLZ1, DTLZ2 and DTLZ3. The specific expression of the test function is shown in the table 5 and table 6.

1) ZDT SERIES TEST FUNCTION

TABLE 5. ZDT series test functions.

Test functions	Expression	Dimensions of decision variables	Interval
ZDT1	$f_1(x)=x_1,$ $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x)=1+9\sum_{i=2}^{Dim}x_i/(Dim-1)$	$Dim=30$	$x=(x_1,x_2,K\ ,x_{Dim})^T\in[0,1]$
ZDT2	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-(x_1/g(x))^2\right]$ $g(x)=1+9\sum_{i=2}^{Dim}x_i/(Dim-1)$	$Dim=30$	$x=(x_1,x_2,K\ ,x_{Dim})^T\in[0,1]$
ZDT3	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{\frac{x_1}{g(x)}}-\frac{x_1}{g(x)}*\sin(10\pi x_1)\right]$ $g(x)=1+9\sum_{i=2}^{Dim}x_i/(Dim-1)$	$Dim=30$	$x=(x_1,x_2,K\ ,x_{Dim})^T\in[0,1]$ $x_2,x_3,K\ ,x_{Dim}\in[-5,5]$
ZDT4	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x)=1+10(Dim-1)+\sum_{i=2}^{Dim}[x_i^2-10\cos(4\pi x_i)]$	$Dim=10$	$x_1\in[0,1]$
ZDT6	$f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x)=g(x)\left[1-\left(x_1/g(x)\right)^2\right]$ $g(x)=1+9\left(\sum_{i=2}^{Dim}x_i/(Dim-1)\right)^{0.25}$	$Dim=10$	$x=(x_1,x_2,K\ ,x_{Dim})^T\in[0,1]$

2) DTLZ SERIES TEST FUNCTION

TABLE 6. DTLZ series test functions.

Test functions	Expression	Dimensions of decision variables	Interval
DTLZ1	$f_1(x)=\frac{1}{2}(1+g(x))x_1x_2$ $f_2(x)=\frac{1}{2}(1+g(x))x_1(1-x_2)$ $f_3(x)=\frac{1}{2}(1+g(x))(1-x_1)$ $g(x)=100*\left\{5+\sum_{i=3}^{Dim}\left[(x_i-0.5)^2-\cos\left(20\pi(x_i-0.5)\right)\right]\right\}$	$Dim=7$	$x=(x_1,x_2,K,x_{Dim})^T\in[0,1]$
DTLZ2	$f_1(x)=(1+g(x))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)$ $f_2(x)=(1+g(x))\cos\left(\frac{x_1\pi}{2}\right)\sin\left(\frac{x_2\pi}{2}\right)$ $f_3(x)=(1+g(x))\sin\left(\frac{x_1\pi}{2}\right)$ $g(x)=\sum_{i=3}^{Dim}(x_i-0.5)^2$	$Dim=10$	$x=(x_1,x_2,K,x_{Dim})^T\in[0,1]$
DTLZ3	$f_1(x)=(1+g(x))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)$ $f_2(x)=(1+g(x))\cos\left(\frac{x_1\pi}{2}\right)\sin\left(\frac{x_2\pi}{2}\right)$ $f_3(x)=(1+g(x))\sin\left(\frac{x_1\pi}{2}\right)$ $g(x)=100*\left\{10+\sum_{i=3}^{Dim}\left[(x_i-0.5)^2-\cos\left(20\pi(x_i-0.5)\right)\right]\right\}$	$Dim=12$	$x=(x_1,x_2,K,x_{Dim})^T\in[0,1]$

B. EVALUATION INDICATORS

In order to evaluate the performance of different algorithms, Inverted Generational Distance (IGD) is used as the evaluation index. IGD is a standard to measure the performance of the algorithm by calculating the average Euclidean distance between the points obtained by the algorithm and the points in the real pareto solution set. The smaller the IGD value, the better the performance of the algorithm. It is shown in formula 17:

$$\mathrm{IGD}\left(P,P^{*}\right)=\frac{\sum_{x\in P^{*}}\min_{y\in P}dis\left(x,y\right)}{|P^{*}|}\tag{17}$$

$dis\left(x,y\right)$ is the minimum Euclidean distance from individuals to the real Pareto solution, $|P^{*}|$ is the number of individuals in the point set distributed on the real Pareto surface.

C. EXPERIMENTAL RESULTS

Parameter initialization of different algorithms is shown in Table 7:

1. The test results of different algorithms on ZDT1 function are shown in figure 8: ($Dimension=30$):

2. The test results of different algorithms on ZDT2 function are shown in figure 9: ($Dimension=30$):

3. The test results of different algorithms on ZDT3 function are shown in figure 10: ($Dimension=30$):

4. The test results of different algorithms on ZDT4 function are shown in figure 11: ($Dimension=10$):

5. The test results of different algorithms on ZDT6 function are shown in figure 12: ($Dimension=10$):

6. The test results of different algorithms on DTLZ1 function are shown in figure 13: ($Dimension=7$:)

7. The test results of different algorithms on DTLZ2 function are shown in figure 14: ($Dimension=10$):

8. The test results of different algorithms on DTLZ3 function are shown in figure 15: ($Dimension=12$):

From the figure 8-15 and table 8-9, it can be seen that LTG-NSMFO algorithm can get better results with fewer iterations than the other four algorithms, and its IGD value is smaller than the other four algorithms, so its comprehensive performance is better, which proves that LTG-NSMFO algorithm has better convergence and diversity; In addition, the convergence and diversity of the algorithm are also verified on the functions of DTLZ series with three objectives. MOEA/D algorithm is based on decomposition, and NSMFO algorithm is based on crowding distance, while LTG-NSMFO algorithm is based on widespread reference points to sort the solution after non-dominated sorting. Through comparative

TABLE 7. Parameter Settings for different algorithms.

Algorithm	Parameters
MOEA/D	Population Size=100 Neighbor Size=20 Mutation Probability=0.5
NSGA-II	Population Size=100 Crossover Probability=0.9 Mutation Probability=0.1 Exponential Distribution of Crossover and Mutation Algorithms=20
MOPSO	Population Size=100 Repository Size=20 Inertia Weight=0.5 Inertia Weight Damping Rate=0.99 Personal Learning Coefficient=1 Global Learning Coefficient=2 Inflation Rate=0.1 Leader Selection Pressure=2 Deletion Selection Pressure=2 Mutation Rate=0.1
NSMFO	Population Size=100 Logarithmic spiral parameters=1
LTG-NSMFO	Population Size=100 Logarithmic spiral parameters=1

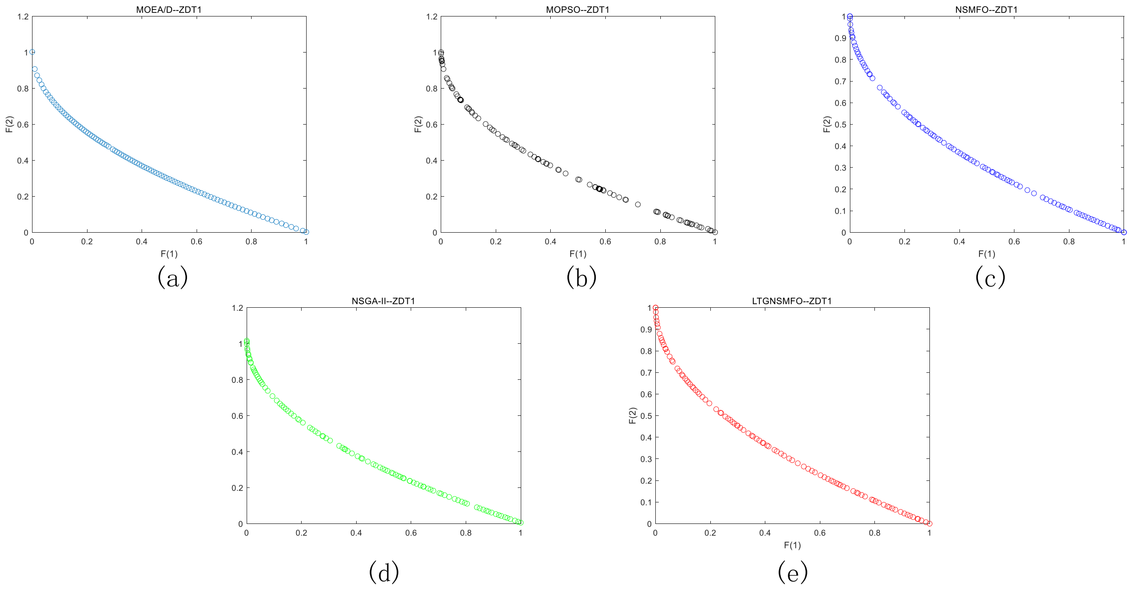
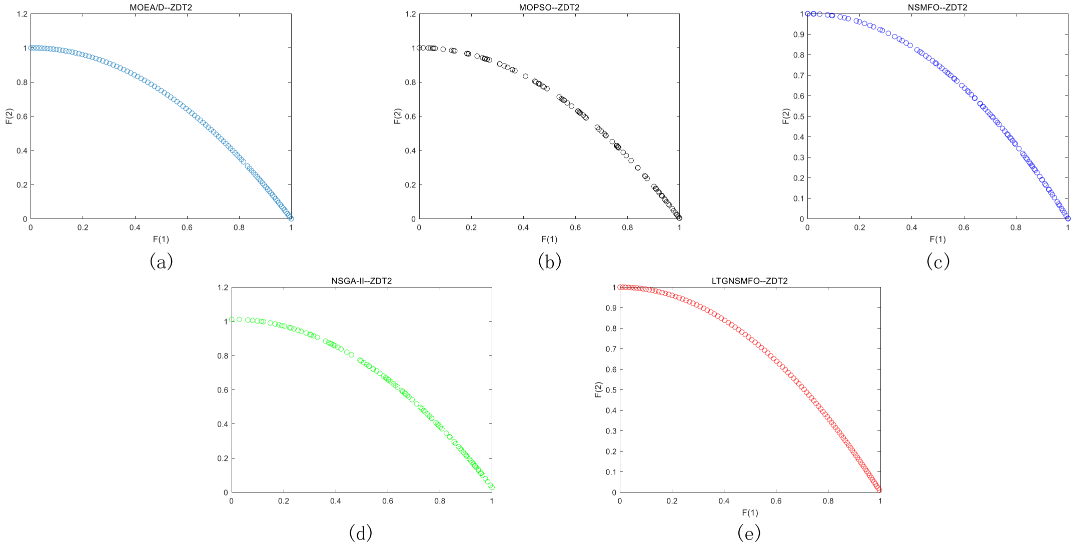


FIGURE 8. Test results of different algorithms on ZDT1 function. (a)–(e) Pareto frontier of MOEA/D, MOPSO, NSMFO, NSGA-II, and LTG-NSMFO algorithms on ZDT1 test function, respectively.

experiments, it can be seen that LTG-NSMFO algorithm is better in sorting, and can effectively solve the problem that NSMFO algorithm can't converge after repeated search on the boundary. The effectiveness of LTG-NSMFO proposed in this paper is proved.

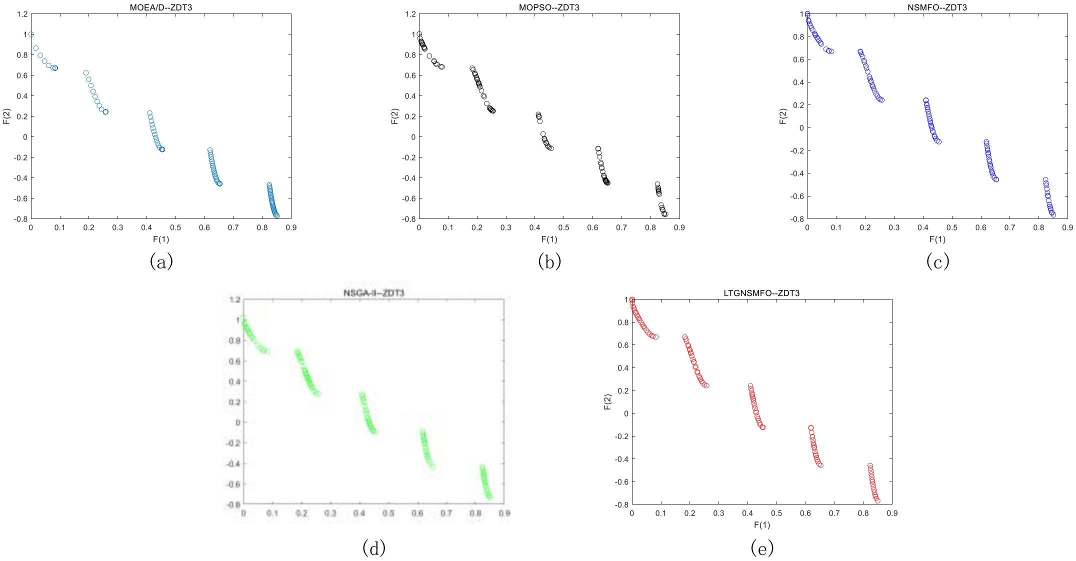
V. SIMULATION EXPERIMENT ANALYSIS OF MULTI-AIRCRAFT JAMMING AND COOPERATIVE ARRAYING

The CPU of the simulation computer is Intel Core i7-9750H CPU 2.60 GHz, the operating system is Windows 10, and



(a) (b) (c) (d) (e)

FIGURE 9. Test results of different algorithms on ZDT2 function. (a)–(e) Pareto frontier of MOEA/D, MOPSO, NSMFO, NSGA-II, and LTG-NSMFO algorithms on ZDT2 test function, respectively.



(a) (b) (c) (d) (e)

FIGURE 10. Test results of different algorithms on ZDT3 function. (a)–(e) Pareto frontier of MOEA/D, MOPSO, NSMFO, NSGA-II, and LTG-NSMFO algorithms on ZDT3 test function, respectively.

the simulation software is MATLAB R2020a. The area of $335km \times 335km$ is selected by the simulation experiment to simulate the combat scene, and the satellite image and Grid structure of this area are shown in the figure16:

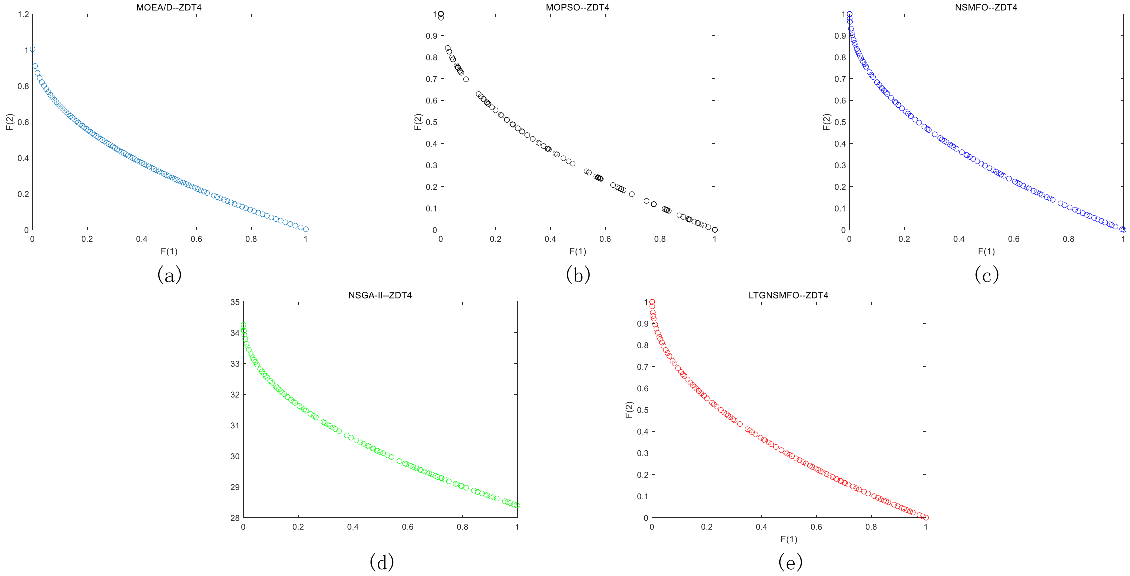
The settings of algorithm parameters and initialization parameters of the modelare shown in Table 10:

Four radars are deployed in this area, and their position coordinates are shown in the above table. The minimum jamming coefficient of these four radars is $K_j = 5$, and for all these four radars, our side adopt the "one-to-one" jamming strategy to jam enemy radars. The performance parameters of radars and jammers are shown in the tables 11 and table 12.

In addition, radar antenna gain is $G = 40$; Jamming signal bandwidth is $B_j = 2 \times 10^6$; Signal bandwidth of radar receiver is $B_r = 2 \times 10^6$; Noise figure is $F_n = 4$; Temperature is $T = 291K$.

According to the above description, the radar detection range obtained under the condition of terrain shading is shown in the figure 17. It can be clearly seen from the figure that the four radars form a relatively strict detection network, showing a situation of covering each other and surrounding the central area, which makes it difficult for our aircraft to penetrate. Therefore, it is necessary to suppress the four radars with the help of jammers.

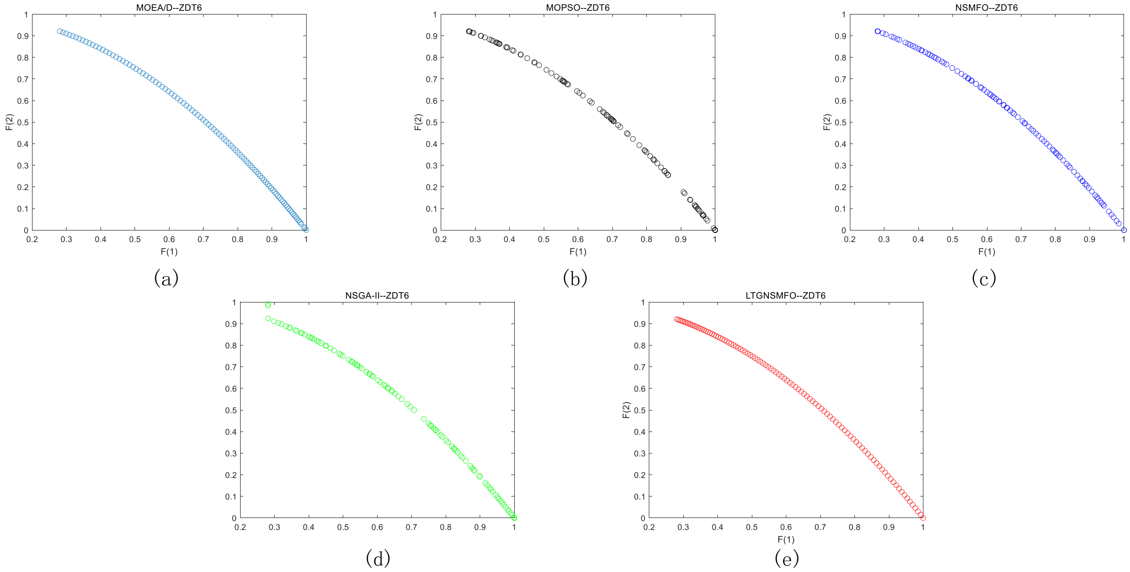
The figure 18 shows the Pareto solution set generated by NSGA-II, MOEA/D and LTG-NSMFO algorithm after 300 iterations, which contains 50 solutions. And its distribution is relatively even, which shows the better search effect of the algorithm. $F(1)$ is the value of the first objective function, which represents the sum of the distances from the four radars to the center line and reflects the width value of the route safety zone; $F(2)$ is the value of the second objective



(a) (b) (c)

(d) (e)

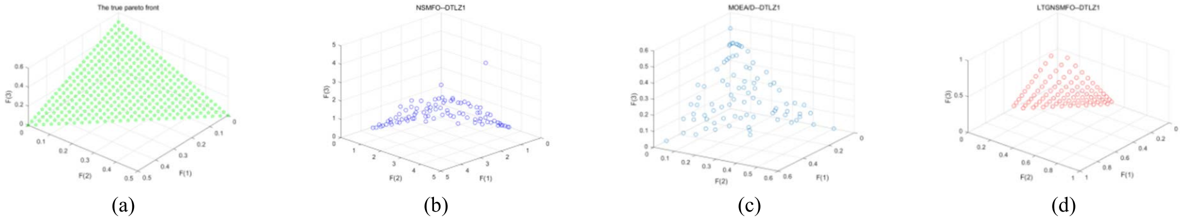
FIGURE 11. Test results of different algorithms on ZDT4 function. (a)–(e) Pareto frontier of MOEA/D, MOPSO, NSMFO, NSGA-II, and LTG-NSMFO algorithms on ZDT4 test function, respectively.



(a) (b) (c)

(d) (e)

FIGURE 12. Test results of different algorithms on ZDT6 function. (a)–(e) Pareto frontier of MOEA/D, MOPSO, NSMFO, NSGA-II, and LTG-NSMFO algorithms on ZDT6 test function.



(a) (b) (c) (d)

FIGURE 13. Test results of different algorithms on DTLZ1 function. (a) Real Pareto front of DTLZ1 test function. (b)–(d) Pareto frontier of MOEA/D, NSMFO, and LTG-NSMFO algorithms on DTLZ1 test function, respectively.

function. In order to facilitate the consistency of calculation, the reciprocal of jamming power is selected; $F(3)$ is the value of the second objective function which represents the detection area of radar; Its mathematical description is shown in formula 18:

$$\max f_2(x) = \sum_{j=1}^{n} \left(1/P_{J-power}\right) \tag{18}$$

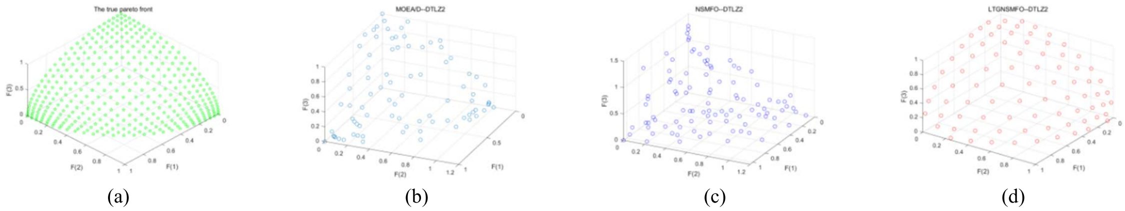


FIGURE 14. Test results of different algorithms on DTLZ2 function. (a) Real Pareto front of DTLZ2 test function. (b)–(d) Pareto frontier of MOEA/D, NSMFO, and LTG-NSMFO algorithms on DTLZ2 test function, respectively.

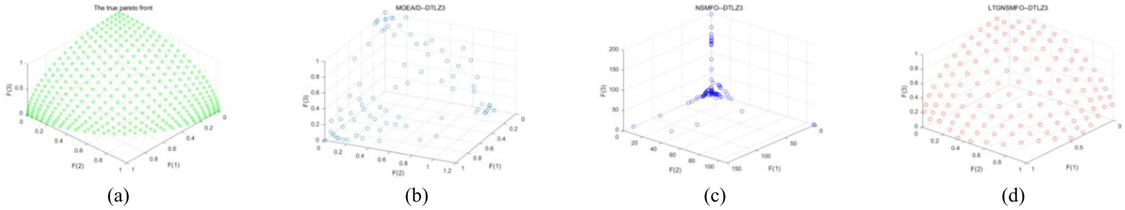


FIGURE 15. Test results of different algorithms on DTLZ3 function. (a) Real Pareto front of DTLZ3 test function. (b)–(d) Pareto frontier of MOEA/D, NSMFO, and LTG-NSMFO algorithms on DTLZ3 test function, respectively.

TABLE 8. Test results of different algorithms on ZDT series functions.

Algorithm	Test functions									
	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
	Convergence / Iterations	IGD	Convergence / Iterations	IGD	Convergence / Iterations	IGD	Convergence / Iterations	IGD	Convergence / Iterations	IGD
MOEA/D	1000/1200	4.6643e-3	1000/1200	3.8042e-3	1000/1200	1.0527e-2	1000/1200	5.2237e-3	1000/1200	1.9999e-3
NSGA-II	1000/1200	1.7160e-2	1000/1200	1.3870e-3	1000/1200	1.0747e-2	1000/1200	28.0635	1000/1200	2.5009e-3
MOPSO	1000/1200	2.2199e-2	1000/1200	8.1529e-3	1000/1200	2.4445e-2	1000/1200	1.6879e-2	1000/1200	3.7702e-3
NSMFO	800/1200	4.5914e-3	800/1200	4.9314e-3	800/1200	5.3316e-3	800/1200	4.3667e-3	1000/1200	3.9665e-3
LTG-NSMFO	500/1200	4.3953e-3	500/1200	2.1130e-3	500/1200	4.7560e-3	500/1200	4.2104e-3	800/1200	2.9835e-3

TABLE 9. Test results of different algorithms on DTLZ series functions.

Algorithm	DTLZ1		DTLZ2		DTLZ3	
	Convergence/Iterations	IGD	Convergence/Iterations	IGD	Convergence/Iterations	IGD
MOEA/D	3000/4000	3.4339e-2	1000/4000	6.7916e-2	3000/4000	8.5052e-2
NSMFO	3000/4000	7.3239e-1	3000/4000	7.5759e-2	3000/4000	6.9871
LTG-NSMFO	2000/4000	3.3095e-2	2000/4000	5.8024e-2	2000/4000	5.1175e-2

Because there are 50 solutions in Pareto front, decision makers need to make choices according to different situations. This paper gives three situations: one is the jamming effect when the jamming power is minimum; the second one is the jamming effect when the safety interval of the air route is the maximum; The third one is the jamming effect when the detection area of the radar is minimum. They are shown in the figure 19, figure 20 and figure 21.

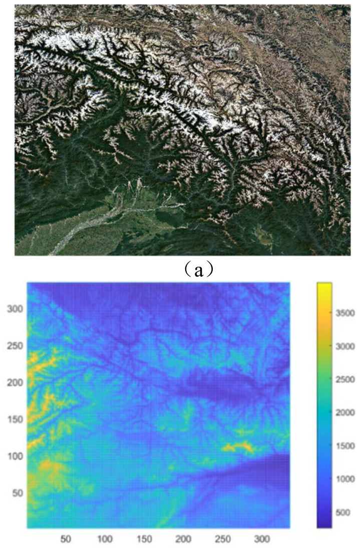
When the decision maker is more inclined to reduce the jamming power to lower the risk of jammers from being discovered, and at the same time, to ensure that there is a certain channel to enable our aircraft to complete the penetration task, the jamming effect is shown in the figure 19. At this time, the minimum interval of route safety zone is 23 km.

At this time, the function values corresponding to the three algorithms are shown in Table 13:

According to Figure 19 and Table 13, when the decision maker hopes to complete the suppression task with a small interference power, the channel widths obtained by the three algorithms are all small. However, the LTG-NSMFO algorithm has an ‘‘open-mouth’’ channel, which is more suitable for the task in terms of width and shape. At the same time, the interference power is reduced by 39.8% compared with the other two algorithms

TABLE 10. Algorithm parameter setting.

The parameter name	number
Population	100
Iterations	300
Logarithmic spiral parameters	1
Radar 1 coordinates	(119.5,109.5,1395.2)
Radar 2 coordinates	(218.5,113.5,949.1)
Radar 3 coordinates	(121.5,225.5,1550.8)
Radar 4 coordinates	(232.5,239.5,1181.9)
Position height of the target	2100



(a)

(b)

FIGURE 16. Topographic map of operational environment. (a) Satellite map and (b) grid map of simulated terrain.

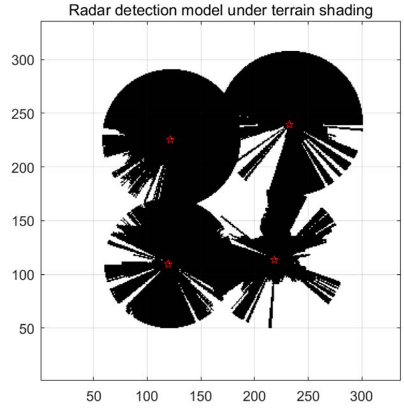


FIGURE 17. Terrain covered cases of radar detection range.

When the decision maker is more inclined to the interval of route safety zone to ensure the flight safety of our combat aircraft, the jamming effect is shown in the figure 20. At this time, the minimum interval of route safety zone is 46 km, and the jamming effect is obviously better than that of the case one.

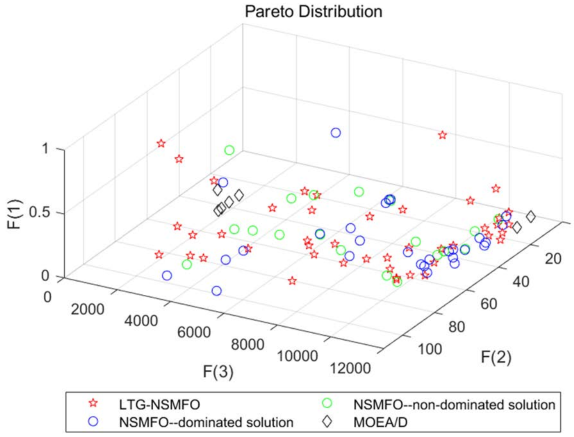
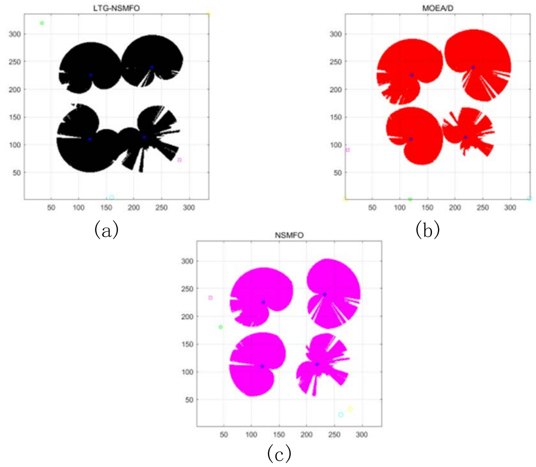


FIGURE 18. The pareto solutions distribution.

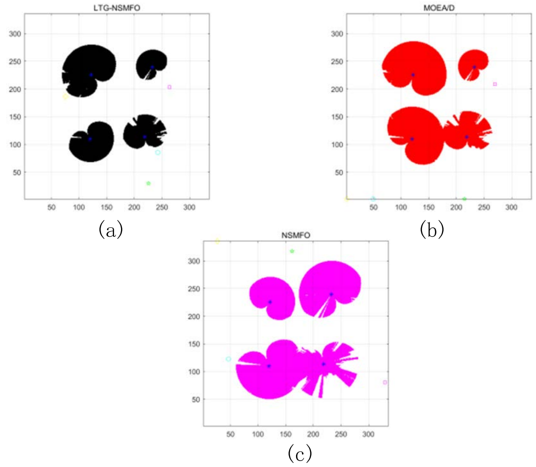


(a)

(b)

(c)

FIGURE 19. Support effect and location distribution of (a) LTG-NSMFO, (b) MOEA/D, and (c) NSMFO algorithms when the jamming power is the minimum.



(a)

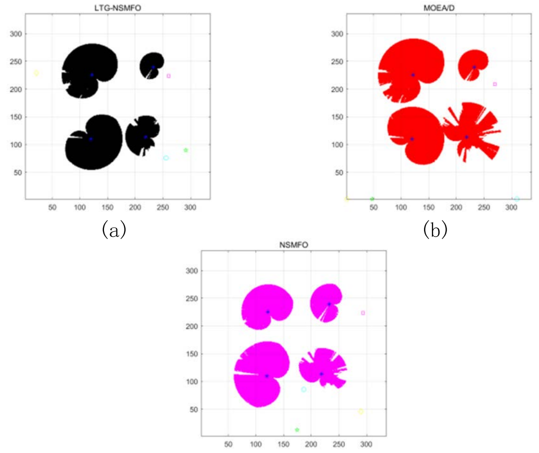
(b)

(c)

FIGURE 20. Support effect and location distribution of (a) LTG-NSMFO, (b) MOEA/D, and (c) NSMFO algorithms when the width of the airline safety zone is the largest.

At this time, the function values corresponding to the three algorithms are shown in Table 14:

According to Figure 20 and Table 14, when the decision maker hopes to form a wider safe zone interval through



(a) (b) (c)

FIGURE 21. Support effect and location distribution of (a) LTG-NSMFO, (b) MOEA/D, and (c) NSMFO algorithms when the radar detection area is the minimum.

TABLE 11. Performance parameters of the radar.

Radar	P_t	L	λ	B_n
Radar 1	1.58e+6	5	0.5	1.80e+5
Radar 2	1.80e+6	4.5	0.65	3.00e+5
Radar 3	1.50e+6	3.8	0.65	2.90e+5
Radar 4	2.30e+6	3.5	0.6	3.50e+5

TABLE 12. Performance parameters of jammers.

Jammer	P_j	G_j	r_j	L_j
Jammer 1	1.00e+6	2	1	20
Jammer 2	1.80e+6	3	1.5	30
Jammer 3	2.00e+6	2.4	2	25
Jammer 4	1.00e+6	4	3	35

TABLE 13. Performance parameters of jammers.

Algorithm	Jamming Power	the width of the airline safety zone	the radar detection area
LTG-NSMFO	117.9kw	23km	8479
MOEA/D	186.5kw	8km	8845
NSMFO	206.5kw	15km	9302

TABLE 14. Performance parameters of jammers.

Algorithm	Jamming Power	the width of the airline safety zone	the radar detection area
LTG-NSMFO	516.2kw	46km	2427
MOEA/D	645.2kw	32km	6845
NSMFO	681.3kw	39km	9240

suppression, the channel width obtained by LTG-NSMFO algorithm is larger, which can form not only the horizontal penetration area, but also the longitudinal penetration channel. Compared with the other two algorithms, the interval value of the route safety zone increases by 22.8% on average.

When the decision maker is more inclined to reduce the radar's detection area to lower the risk of jammers from being discovered, and at the same time, to ensure that there is a wide channel to enable our aircraft to complete the penetration task, the jamming effect is shown in the figure 21. At this time, the minimum interval of route safety zone is 38 km.

TABLE 15. Performance parameters of jammers.

Algorithm	Jamming Power	the width of the airline safety zone	the radar detection area
LTG-NSMFO	847.6kw	38km	1727
MOEA/D	745.2kw	15km	2869
NSMFO	681.3kw	21km	3090

At this time, the function values corresponding to the three algorithms are shown in Table 15:

According to Figure 21 and Table 15, when the decision maker hopes to make the detection area of the enemy radar as small as possible through interference suppression, the suppression effect obtained by the LTG-NSMFO algorithm is better. Compared with the other two algorithms, the detection area of the enemy radar is reduced by an average of 41.9%.

VI. CONCLUSION

In this paper, the problem of multi-aircraft arraying to jam is studied, and a jamming model based on LTG-NSMFO algorithm is proposed. Firstly, the radar threat model is established, and the radar detection model is optimized according to the terrain shadi ng situation. Then, two objective functions of multi-aircraft arraying are determined, one is the width of route safety zone, the other is the size of jamming power, and the multi-objective arraying model is established to provide technical support for practical application in the future. At the same time, the NSMFO algorithm has a slow convergence speed in the search process, and even can't converge when dealing with the three-objective problem:

(1) The initialization of Logistic-tent chaotic map makes the initial search of the algorithm more extensive, enhances the global search ability of the algorithm, and accelerates the convergence speed of the algorithm;

(2) The decision factor and Gaussian difference mutation are introduced to produce disturbance when the moth position is updated, so that the algorithm can not only accept the current inferior solution with a certain probability, but also jump out of the local optimum, thus enhancing the solving ability of the algorithm;

(3) Using widely distributed reference points instead of crowded distance sorting, the algorithm can effectively converge when dealing with three-objective problems.

Finally, the test function proves that LTG-NSMFO algorithm is better than MOEA/D, NSGA-II, MOPSO and NSMFO algorithm, and can complete the task better in multi-aircraft arraying to jam. In the next research, the proposed algorithm is integrated into UAV mission planning software, so as to realize engineering application.

REFERENCES

[1] T. J. Nohara *et al.*, "Low cost, high performance radar networks," 2018.

[2] F. Zhang, B. Gao, and S. Pan, "Photonics-based MIMO radar with high-resolution and fast detection capability," *Opt. Exp.*, vol. 26, no. 13, p. 17529, 2018.

[3] K. V. Mishra and Y. C. Eldar, "Sub-Nyquist radar: Principles and prototypes," in *Compressed Sensing in Radar Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2019, pp. 1–50.

[4] Y. Zhang, Y. Zhang, W. Li, Y. Huang, and J. Yang, "Super-resolution surface mapping for scanning radar: Inverse filtering based on the fast iterative adaptive approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 127–144, Jan. 2018.

[5] E. Peral, E. Im, L. Wye, S. Lee, S. Tanelli, Y. Rahmat-Samii, S. Horst, J. Hoffman, S. H. Yun, T. Imken, and D. Hawkins, "Radar technologies for earth remote sensing from cubesat platforms," *Proc. IEEE*, vol. 106, no. 3, pp. 404–418, Feb. 2018.

[6] Q. Feng, C. Hong, and R. Gao, "Path planning of stand-off jamming electronic warfare aircraft," in *Proc. 39th Chin. Control Conf. (CCC)*, Jul. 2020, pp. 6917–6922.

[7] Q. Wang and D. Yao, "Research on electronic jamming airspace planning," in *Proc. Int. Conf. Comput. Technol., Electron. Commun. (ICCTEC)*, Dec. 2017, pp. 811–816.

[8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 2000, pp. 849–858.

[9] Y. Shuai, S. Yunfeng, and Z. Kai, "An effective method for solving multiple travelling salesman problem based on NSGA-II," *Syst. Sci. Control Eng.*, vol. 7, no. 2, pp. 108–116, Oct. 2019.

[10] C. A. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, Jun. 2003, pp. 1051–1056, doi: 10.1109/CEC.2002.1004388.

[11] L. Li, S. Chen, Z. Gong, Q. Lin, and Z. Ming, "A novel hybrid multi-objective particle swarm optimization algorithm with an adaptive resource allocation strategy," *IEEE Access*, vol. 7, pp. 177082–177100, 2019.

[12] H. Li and Q. Zhang, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Nov. 2007.

[13] X. Zhou, X. Wang, and X. Gu, "Welding robot path planning problem based on discrete MOEA/D with hybrid environment selection," *Neural Comput. Appl.*, vol. 33, no. 19, pp. 12881–12903, Oct. 2021.

[14] V. Savsani and M. A. Tawhid, "Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems," *Eng. Appl. Artif. Intell.*, vol. 63, pp. 20–32, Aug. 2017.

[15] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[16] H. Huang *et al.*, "Algorithm based on improved puhuo to path planning of UAV low-altitude penetration," *J. Chin. Inertial Technol.*, vol. 29, no. 2, pp. 256–263, 2021, doi: 10.13695/j.carol.carrollnki/o3.2021.02.01712-1222.

[17] E. V. Altay and B. Alatas, "Bird swarm algorithms with chaotic mapping," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 1373–1414, Feb. 2020.

[18] H. Bingol and B. Alatas, "Chaotic league championship algorithms," *Arabian J. Sci. Eng.*, vol. 41, no. 12, pp. 5123–5147, Dec. 2016.

[19] B. Alatas, *Chaotic Bee Colony Algorithms for Global Numerical Optimization*. New York, NY, USA: Pergamon, 2010.

[20] B. Alatas, "Chaotic harmony search algorithms," *Appl. Math. Comput.*, vol. 216, no. 3, pp. 2687–2699, 2010.

[21] X. J. Bi and C. Wang, "Based on reference point constraint dominate the NSGA—III algorithm," *Control Decis.*, no. 2, pp. 69–376, 2019.

[22] S. Verma, M. Pant, and V. Snasel, "A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems," *IEEE Access*, vol. 9, pp. 57757–57791, 2021.

[23] X. Zhang, H. Liu, and L. Tu, "A modified particle swarm optimization for multimodal multi-objective optimization," *Eng. Appl. Artif. Intell.*, vol. 95, Oct. 2020, Art. no. 103905.

[24] I. N. Trivedi, P. Jangir, S. A. Parmar, and N. Jangir, "Optimal power flow with voltage stability improvement and loss reduction in power system using moth-flame optimizer," *Neural Comput. Appl.*, vol. 30, no. 6, pp. 1889–1904, Sep. 2018.

[25] X. Zhang, Y. Tan, and Z. Yang, "Resource allocation optimization of equipment development task based on MOPSO algorithm," *J. Syst. Eng. Electron.*, vol. 30, no. 6, pp. 84–95, 2019.

[26] H. Huang *et al.*, "Adaptive interpolation puhuo to optimize more characteristics of particle filter for vehicle tracking algorithm," *J. Shanghai Jiaotong Univ.*, vol. 56, no. 2, pp. 143–155, 2022, doi: 10.16183/j.carolcarroll.nkijsjtu.2021.037.

[27] W. Deng *et al.*, "A decomposition multi-objective evolutionary algorithm based on multi-strategy differential evolution," *Control Decis.*, vol. 37, no. 2, pp. 387–392, 2022, doi: 10.13195/j.kzyjc.2020.1196.

[28] P. F. Diao, S. S. Li, and X. S. Jiang, "Dynamic multi-objective gravity search algorithm based on multi-population decomposition prediction," *Control Decis.*, vol. 36, no. 12, pp. 2910–2918, 2021, doi: 10.13195/j.kzyjc.2020.1002.

[29] Q. H. Gu *et al.*, "Decomposition constrained NSGA-II optimization algorithm for constrained high-dimensional multi-objective problems," *Control Decis.*, vol. 35, no. 10, pp. 2466–2474, 2020, doi: 10.13195/j.kzyjc.2019.0116.

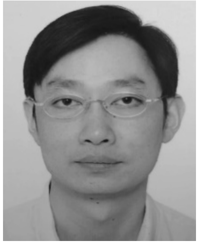
[30] Y. Longfei, Y. Rennong, Z. Yijie, Y. Yang, and Z. Zhenxing, "Tent chaos and simulated annealing improved moth fire suppression optimization algorithm," *J. Harbin Inst. Technol.*, vol. 51, no. 5, p. 49, 2019.

[31] H. Zhang *et al.*, "Research on suppression jamming arrangement for multi-aircraft cooperative electronic warfare planning," *Syst. Eng. Electron.*, vol. 39, no. 3, pp. 542–548, 2017.

MINGXI MA was born in Baotou, Inner Mongolia, in 1998. He is currently pursuing the master's degree with Air Force Engineering University. His major is management science and engineering. His research interests include operation planning and mission planning, and the application of intelligent algorithms.

JUN WU was born in Hunan, China. He is currently a Professor and a Master's Tutor with Air Force Engineering University. His current research interests include dynamic planning, expert systems, and autonomous planning.

YUE SHI was born in Changchun, Jilin, in 1968. She is currently a Professor and a Master's Tutor with Air Force Engineering University. Her major is management science and engineering.



LONG YAN was born in November 1979. He graduated from Air Force Engineering University. He is currently a Senior Engineer. His research interest includes information support of logistics equipment.

WEI LU was born in Chengde, Hebei, in 1990. He received the degree in airborne network communications from Tsinghua University, in 2019.