Journal of Systems Engineering and Electronics
Vol. 34, No. 4, August 2023, pp.1007 – 1019

Shuffled frog leaping algorithm with non-dominated sorting for dynamic weapon-target assignment

ZHAO Yang*, LIU Jicheng, JIANG Ju, and ZHEN Ziyang

College of Automation Engineering, Nanjing University of Aeronautics and Astronauts, Nanjing 211106, China

Abstract: The dynamic weapon target assignment (DWTA) problem is of great significance in modern air combat. However, DWTA is a highly complex constrained multi-objective combinatorial optimization problem. An improved elitist non-dominated sorting genetic algorithm-II (NSGA-II) called the non-dominated shuffled frog leaping algorithm (NSFLA) is proposed to maximize damage to enemy targets and minimize the self-threat in air combat constraints. In NSFLA, the shuffled frog leaping algorithm (SFLA) is introduced to NSGA-II to replace the inside evolutionary scheme of the genetic algorithm (GA), displaying low optimization speed and heterogeneous space search defects. Two improvements have also been raised to promote the internal optimization performance of SFLA. Firstly, the local evolution scheme, a novel crossover mechanism, ensures that each individual participates in updating instead of only the worst ones, which can expand the diversity of the population. Secondly, a discrete adaptive mutation algorithm based on the function change rate is applied to balance the global and local search. Finally, the scheme is verified in various air combat scenarios. The results show that the proposed NSFLA has apparent advantages in solution quality and efficiency, especially in many aircraft and the dynamic air combat environment.

Keywords: dynamic weapon-target assignment (DWTA) problem, shuffled frog leaping algorithm (SFLA), air combat research.

DOI: 10.23919/JSEE.2023.000102

1. Introduction

Modern air combat technology is developing rapidly in the track of autonomous and intelligent combat. The weapon-target assignment (WTA) has always been a fundamental problem in battlefield decisions for firepower strikes [1]. Especially with the advent of the unmanned aerial vehicle (UAV) era, each aircraft can make intelligent decisions independently and quickly and conduct cooperative air combat in the form of a flying formation. Therefore, how to intelligently carry out WTA in complex air combat environments plays a crucial role in modern air combat systems [2–5].

The WTA method provides an optimization scheme for improving the overall attack effects of multi-weapon and multi-target firepower coordinated attacks. The WTA problem refers to the problem of optimally allocating defensive weapons to enemy targets to minimize the total expected survival value of the targets or to maximize the total expected survival value of the protected assets [6–9]. Essentially, the WTA problem can be formulated as a nonlinear integer programming problem and is known to be non-deterministic polynomical (NP)-complete [10]. The WTA model can be divided into the static WTA (SWTA) model and the dynamic WTA (DWTA) model from the perspective of whether time is considered as a factor [11,12]. SWTA launches all defense weapons at a certain stage to find the best allocation for temporary defense tasks [13]. In the static version, all inputs to the problem are fixed. The number of targets and weapons is known and fixed, and all weapons engage targets in a single stage [14]. Compared with SWTA, DWTA launches weapons in stages, which can solve a supersaturation attack. When new attacking targets appear, there is no need to recalculate the allocation scheme because the combat results of the previous stage have been observed before launching, Therefore, DWTA is more in line with the real air combat environment and its simulation effect is more realistic [1,15].

As far as the research algorithms are concerned, researchers have proposed many algorithms to solve the DWTA problem, such as the Lagrangian relaxation method [16], the rule-based approach [17], the approximate dynamic programming approach, and the geometric-based approach [18]. These methods exhibit better performance for smaller dimensions but cannot effectively solve large-scale problems [19]. As an improvement, Wang et al. [20] improved the particle initialization and inertia weight selection methods of the particle swarm

Manuscript received March 07, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61673209; 71971115).

optimization (PSO) algorithm and effectively improved the optimization efficiency and allocation results for the large-scale single-object WTA (SOWTA) problem. Mei et al. [21] constructed a SOWTA model based on the killing region of a weapon platform, and proposed a combinatorial algorithm derived from a heuristic algorithm and receding horizon control. However, these algorithms only solve static single-objective optimization, and the applicability of the algorithms is poor. Gao et al. [22] noted that traditional algorithms, such as the goal programming algorithm [23], the minimum-maximum method [24], and the linear weighting method, consume considerable time and resources. Therefore, we proposed an algorithm suitable for static multi-objective target assignment to improve these methods. Time-dependent SWTA (TSWTA) was proposed in [25]. This WTA algorithm was numerically tested and showed excellent performance in a computation time of a few seconds. However, in static algorithms, all inputs to the problem are fixed. The number of targets and weapons is known and cannot be changed, and all weapons engage targets in a single stage. In addition, the optimal results of these static algorithms are not ideal, and the calculation time is long. In terms of complexity, although the proposed model takes both antagonism and uncertainty into account [26], the information-sharing mode of utilizing only one fighter in a team leads to unfavorable stability and error-tolerance rates when identifying the target type [27]. From an overall perspective, most existing methods are not highly applicable to DWTA models in multi-target, multi-weapon, multi-stage, large-scale, and complex air combat situations.

The shuffled frog leaping algorithm (SFLA) is a meta-heuristic algorithm that seeks global optimization in complex problems and has been successfully applied in many fields [28]. SFLA has a stronger global search and optimization ability when compared with the evolutionary algorithm such as the genetic algorithm (GA). The problem of DWTA is transformed into a multi-objective optimization problem of maximizing damage to enemy targets and minimizing self-threat under constraints. The contributions of this paper are given as follows:

(i) The evolutionary mechanism of SFLA is introduced to the non-dominated sorting genetic algorithm-II (NSGA-II) to replace the inside evolutionary scheme of GA, which is named as non-dominated SFLA (NSFLA). Experiments prove that this improvement is more effective in solving the DWTA problem.

(ii) Unlike traditional update law in SFLA, in the part of local evolution in NSFLA, every two individuals, not only the worst one in the memeplex, are included to update, which can improve the local search ability of the algorithm. This means the diversity of the population is expanded.

(iii) A discrete adaptive mutation scheme is creatively put forward based on the function change rate, which combines Gaussian mutation and Cauchy mutation to accelerate the convergence of the algorithm to the optimal solution. The local search of NSFLA obtains the ideal optimal solution by improving the local search ability and convergence speed, guiding individuals to jump out of the local optimal region, and balancing the relationship between global search and local search.

(iv) A series of experiments from various aspects are carried out and effectively demonstrate the applicability and efficiency of NSFLA in solving DWTA problems.

The following parts of this paper are structured. In Section 2, the mathematical model of the DWTA is described. Section 3 gives a brief introduction to the SFLA and describes the design process of NSFLA. Section 4 tests the scheme and analyzes the test results. Section 5 presents conclusions.

2. Problem description

The following notations [29] are used to formulate the DWTA:

r: The total number of remaining weapons at starting stage s.

n: The total number of remaining targets at starting stage s.

m: The total number of aircraft that constitute the UAV formation.

v: The total number of stages considered in the problem.

Y: The set of targets at stage s, $Y = \{1, 2, \cdots, n\}$.

x_{kj}^h: $\boldsymbol{X}^h = [x_{kj}^h]_{r\times n}$ is the decision matrix at the stage h, and $x_{kj}^h = 1$ if weapon k is assigned to target j at stage h; otherwise, $x_{kj}^h = 0$.

p_{kj}^h: $\boldsymbol{P}^h = [p_{kj}^h]_{r\times n}$ is the kill probability matrix at stage h, and p_{kj}^h is the probability that the weapon k destroys the target j at the stage h.

t_{kj}^h: $\boldsymbol{T}^h = [t_{kj}^h]_{r\times n}$ is the threat matrix at stage h, and t_{kj}^h is the threat value of target j against allied aircraft i at the stage h.

This work mainly focuses on the target-based DWTA problem. At stage h, if the kth weapon is assigned to the jth target, then the survival probability of the jth target is $(1-p_{kj}^h)$.

After performing a coordinated attack, at all stages, the survival probability of the jth target becomes $\prod_{h=s}^{v}\prod_{k=1}^{r}(1-p_{kj}^h)^{x_{kj}^h}$. Moreover, let T_{ji} be the threat value of the jth target to the ith allied aircraft; then, the remaining threat can be written as $T_{ji}\cdot\prod_{h=s}^{v}\prod_{k=1}^{r}(1-p_{kj}^{h})^{x_{kj}^{h}}$. Based on the above assumptions and the background, the mathematical model of DWTA is established as follows:

$$\begin{cases}\min F=\sum_{j=1}^{n}\sum_{i=1}^{m}\left[T_{ji}\cdot\prod_{h=s}^{v}\prod_{k=1}^{r}(1-p_{kj}^{h})^{x_{kj}^{h}}\right]\\ \max E=\sum_{j=1}^{n}\left[1-\prod_{h=s}^{v}\prod_{k=1}^{r}(1-p_{kj}^{h})^{x_{kj}^{h}}\right]\end{cases}\tag{1}$$

where F indicates the minimum total expected threat value of all residual enemy targets from the starting stage to the final stage and E represents destroying as many enemy targets as possible. The constraints are given as follows:

$$\sum_{j=1}^{n}x_{kj}^{h}\leqslant n_{k}^{h},\ \forall h\in\{s,s+1,s+2,\cdots,v\},\forall k\in\{1,2,\cdots,r\},\tag{2}$$

$$\sum_{k=1}^{r}x_{kj}^{h}\leqslant m_{j}^{h},\ \forall h\in\{s,s+1,s+2,\cdots,v\},\forall j\in\{1,2,\cdots,n\},\tag{3}$$

$$\sum_{h=s}^{v}\sum_{j=1}^{n}x_{kj}^{h}\leqslant a_{k},\ \forall k\in\{1,2,\cdots,r\},\tag{4}$$

$$x_{kj}^{h}\leqslant f_{kj}^{h},\forall h\in\{s,s+1,s+2,\cdots,v\},\ \forall k\in\{1,2,\cdots,r\},\ \forall j\in\{1,2,\cdots,n\},\tag{5}$$

$$x_{kj}^{h}\in\{0,1\},\ \forall h\in\{s,s+1,s+2,\cdots,v\},\ \forall k\in\{1,2,\cdots,r\},\ \forall j\in\{1,2,\cdots,n\}.\tag{6}$$

Among the above constraints, (2) is the capacity constraint for weapons, which limits the maximum number of targets weapon j can engage at the stage h; (3) indicates that most m_j^h weapons can be assigned to target j at the stage h; (4) ensures that the amount of engagement for a particular weapon k through all stages cannot exceed its predefined allowable amount a_k; (5) is the time window constraint, where the influence of the time window on weapon engagement feasibility is considered, and it ensures that any weapon firing must meet the engagement feasibility at each stage; (6) ensures that all the decision variables x_{kj}^h must be binary. Therefore, the dynamic weapon-target assignment problem of air combat is transformed into the multi-objective optimization in (1) under constraints in (2)–(6).

Remark 1 The values of kill probability p_{kj}^h and threat matrix T_{ji} can refer to [13], which are considered as known variables in this paper.

3. Proposed optimization algorithms

3.1 SFLA

SFLA is a memetic metaheuristic designed to seek a globally optimal solution by performing an informed heuristic search using a heuristic function [30]. SFLA has been tested on several combinatorial problems and found to be efficient in finding global solutions [31].

The main function of the SFLA is to update the position of the worst-performing frog through iterative operation in each memeplex in a memetic evolution, where the frogs in jth memeplex can be given by

$$\Omega_j=\left\{X_{j+\mathrm{Nmp}(l-1)}\in X\,\middle|\,1\leqslant l\leqslant \mathrm{Nem}\right\},\ \ 1\leqslant j\leqslant \mathrm{Nmp}\tag{7}$$

where Nmp is the total number of memeplexes in each evolution; Nem is the number of frogs in each memeplex; X is the initial population. $P=\mathrm{Nem}\times\mathrm{Nmp}$ represents the size of X.

The position of the worst-performing frog is improved by learning from the best frog in the memeplex or from the population and position of itself. The learning rule can be written as

$$\begin{cases}D_i=\mathrm{rand}(\cdot)\times(x_b-x_w)\\ x_{\mathrm{new}}=x_w+D_i\end{cases},\ \ D_{\min}\leqslant D_i\leqslant D_{\max}\tag{8}$$

where x_b and x_w represent the best and the worst frog in a memeplex; D_i is the step size of ith update of x_w; $D_{\max}$ is the maximum step of movement; rand$(\cdot)$ means a random number between 0 and 1.

The basic procedures of the SFLA can be summarized as Algorithm 1.

Algorithm 1 SFLA procedure

Step 1 Randomly initialize the original population, generating the first generation frogs $X^0=\{x_1^0,x_2^0,\cdots,x_P^0\}$; the ith frog can be written as $x_i^0=\left\{x_{i,1}^0,x_{i,2}^0,\cdots,x_{i,s}^0\right\}$ $(i=1,2,\cdots,P)$. Let $l=0$ represents the original generation.

Evolution phase

Step 2 Calculate the fitness value matrix $F(\boldsymbol{X}^l)=\{F(x_1^l),F(x_2^l),\cdots,F(x_P^l)\}$ and arrange which in descending order. Denote $x_g^l=\arg\max F(\boldsymbol{X}^l)$.

Step 3 Partition frogs into Nmp memeplexes, with each memeplex containing Nem frogs, that is, $\left\{x_j^l,x_{\mathrm{Nmp}+j}^l,\cdots,x_{(\mathrm{Nem}-1)\cdot\mathrm{Nmp}+j}^l\right\}\in\Omega_j^l$ $(j=1,2,\cdots,\mathrm{Nmp})$. Denote $x_{j,b}^l=\Omega_j^l(1)$, $x_{j,w}^l=\Omega_j^l(\mathrm{Nem})$.

Step 4 Update the worst frog $x_{j,w}^l$ according to (8), and obtain the new $x_{j,\mathrm{new}}^l$.

Step 5 If $F(x_{j,\mathrm{new}}^l)>F(x_{j,w}^l)$, replace $x_{j,w}^l$ with $x_{j,\mathrm{new}}^l$. Otherwise, keep remained.

Step 6 Shuffle the memeplexes.

Step 7 If the stopping criterion is met, halt and denote x_g^l as the final solution. Otherwise, let $l = l+1$ and go to Step 2.

3.2 Elitist non-dominated sorting SFLA

Elitist NSGA-II is a multi-objective genetic algorithm with the most significant influence and most comprehensive application range [32,33]. Considering the optimization goals in (1) and constraints in (2)–(6), and the complexity expansion caused by increasing flights, missiles, and stages in air combat. Classical NSGA-II may encounter low optimization speed, heterogeneous space search, and blind search direction, which have strongly related to the inevitable flaws of GA. Thus, in this section, based on the research of SFLA and inspired by the works in multi-objective optimization, a novel elitist NSFLA is proposed to solve the dynamic weapon-target assignment.

Before introducing the details of NSFLA, several definitions need to explain.

Definition 1 (Multi-objective optimization) The multi-objective optimization problem is defined by f, and the function maps the decision variables x of m-dimensions to the objective vector of n-dimension. The mathematical description of the multi-objective optimization problem is as follows:

$$\begin{cases} \min Y = \boldsymbol{f}(x) = [f_1(x), f_2(x), \cdots, f_n(x)]^{\mathrm{T}} \\ \boldsymbol{g}(x) = [g_1(x), g_2(x), \cdots, g_r(x)]^{\mathrm{T}} \end{cases} \tag{9}$$

where $x_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,m}]$ is a decision vector with m variables; Y consists of n objective functions $f_i(X)$ that need to be optimized. The constraint $g(x)$ consists of r equalities or inequalities.

Definition 2 (Concept of domination) Assume $f: \mathbf{R}^m \to \mathbf{R}^k, \boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{\Omega} \subseteq \mathbf{R}^m$; it is said that decision variables $\boldsymbol{x}_1$ dominate another vector $\boldsymbol{x}_2$ if and only if $f(\boldsymbol{x}_1)$ is partially better than $f(\boldsymbol{x}_2)$. That is, $f_i(\boldsymbol{x}_1) \leqslant f_i(\boldsymbol{x}_2)$, $\forall i \in \{1,2,\cdots,k\}$ and $\exists i \in \{1,2,\cdots,k\}$, s.t. $f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$. This is denoted by $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$.

Definition 3 (Concept of non-dominated set) Among a set of solutions X, the non-dominated set of solutions X' are not dominated by any other set member X.

Definition 4 (Concept of crowding distance) The average distance between two points on either side of a given point along each objective is defined as the crowding distance. It serves to estimate the perimeter of the cuboid formed by using the nearest neighbors as the vertices.

Remark 2 According to the definitions above, we know that every solution involved in the evolution has two attributes: (i) the non-domination rank i_{rank}; (ii) the crowding distance i_{dist}. The crowded-comparisons operator requires both the non-dominated rank and crowding distance of each solution.

Hence, the details and procedures of how NSFLA is applied to solve DWTA problems are interpreted in pseudo-code as shown in Algorithm 2.

Algorithm 2 Procedures of NSFLA for DWTA

Step 1 Generate initial population $\boldsymbol{X}^0 = \{\boldsymbol{X}_1^0, \boldsymbol{X}_2^0, \cdots, \boldsymbol{X}_P^0\}$, where $\boldsymbol{X}_i^0 = \left\{\left[x_{kj}^0\right]_{r\times n}\right\}_i$ indicates the ith initial target assignment matrix individual and x_{kj}^0 needs to meet (5) and (6), let $l = 0$.

Step 2 Calculate the values of total expected threat $f(\boldsymbol{X}_i^l)$ and damage index $e(\boldsymbol{X}_i^l)$ for each individual under constraints (2)–(4).

Step 3 Perform non-dominated sorting of $\boldsymbol{X}^l$ based on $f(\boldsymbol{X}_i^l)$ and $e(\boldsymbol{X}_i^l)$, then calculate the crowding distance for each $\boldsymbol{X}_i^l$, and only the first p individuals are retained. Denote the first as well the best individual as $\boldsymbol{X}_b^l$.

Step 4 If $\boldsymbol{X}_b^l$ satisfy the stopping criterion, halt and denote $\boldsymbol{X}_b^l$ as the final solution.Otherwise, go to Step 5.

Step 5 Partition the non-dominated sorted $\boldsymbol{X}^l$ into Nmp memeplexes, that is, $\boldsymbol{X}^l = \left\{\boldsymbol{\Omega}_1^l, \boldsymbol{\Omega}_2^l, \cdots, \boldsymbol{\Omega}_{\text{Nmp}}^l\right\}$, for each $\boldsymbol{\Omega}_j^l$, there is $\left\{\boldsymbol{X}_j^l, \boldsymbol{X}_{\text{Nmp}+j}^l, \cdots, \boldsymbol{X}_{(\text{Nem}-1)\cdot\text{Nmp}+j}^l\right\} \in \boldsymbol{\Omega}_j^l$.

Step 6 Update $\boldsymbol{\Omega}_j^l$ based on **local evolution.** Then, we will get the evolutional memeplex, denotes as $\boldsymbol{R}_j^l$.

Step 7 Shuffle the memeplexes $\boldsymbol{R}_j^l (j = 1,2,\cdots,\text{Nmp})$, then we get $\boldsymbol{R}^l$, that is $\boldsymbol{X}^{l+1}$.

Step 8 If $l+1 \geqslant l_{\max}$, go to Step 9; Otherwise$l = l+1$, go to Step 3.

Step 9 Output $\boldsymbol{X}_b^l = \left\{\left[x_{kj}^l\right]_{r\times n}\right\}_b$, that is the optimal assignment matrix.

Remark 3 The local evolution in Step 6 of Algorithm 2 is explained in detail in the following, which is another contribution in this paper. This method can improve the local search ability of the algorithm when compared with the GA in NSGA- II.

There are several improvements in the proposed local evolution. Firstly, every $\boldsymbol{X}_j^l$ involved in $\boldsymbol{\Omega}_j^l$ is participated in evolution; that is, the combinations of $\left(\boldsymbol{X}_{(a-1)\cdot\text{Nmp}+j}^l, \boldsymbol{X}_{(\text{Nem}-a)\cdot\text{Nmp}+j}^l\right), (a = 1,2,\cdots,\text{Nem}/2; j = 1,2,\cdots,\text{Nmp})$ are update, the law can refer to [13]. Secondly, the mutation operation is introduced to improve the diversity of the population. A discrete adaptive mutation algorithm based on the function change rate is proposed, which combines Gaussian mutation and Cauchy mutation to balance the relationship between global search and local search. So, it is necessary to explain the discrete adaptive mutation operators.

The discrete Gaussian mutation operator is obtained as follows:

$$\mathrm{mu}(\lambda)=\begin{cases}0, & \lambda\in(-0.3,0.3)\\ 1, & \lambda\in(-1,-0.3)\cup(0.3,1)\\ 2, & \lambda\in(-2,-1)\cup(1,2)\\ 3, & \lambda\in(-3,-2)\cup(2,3)\\ 4, & \lambda\in(-\infty,-3)\cup(3,+\infty)\end{cases} \tag{10}$$

where $\mathrm{mu}(\lambda)$ means the number of mutation bits, and λ is a random number that obeys the Gaussian distribution $\varphi(\lambda)$ as follows:

$$\varphi(\lambda)=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{\lambda^2}{2}},\ -\infty<\lambda<+\infty. \tag{11}$$

The discrete Cauchy mutation operator is also obtained from (10). The difference is that the random number λ is obeyed the following Cauchy distribution rather than (11):

$$\phi(x)=\frac{1}{\pi\,(x^2+1)},\ -\infty<x<+\infty. \tag{12}$$

Fig. 1 describes the visual relationship between the two distributions and the mutation bit numbers. We can conclude that the Cauchy mutation has a higher probability of producing a greater random number as well as mutation bits when compared with Gaussian mutation. In conclusion, a discrete Gaussian mutation has strong local search ability, which can improve the convergence speed of the algorithm. Discrete Cauchy mutation possesses a strong global search ability; when the algorithm becomes trapped in the local optimum, it can guide individuals to jump out of the local optimal region with a great probability of success.

Deciding how to select the type of mutation is a practical problem in the local evolution of NSFLA. A novel adaptive mechanism according to the change rate of the objective function is proposed. Define a numerical index η_a^l to indicate the degree of the mutation of $\boldsymbol{R}^l$.

$$\eta_j^l=\exp\left(\sum_{a=1}^{\mathrm{Nem}}\left(\frac{|\Delta f_a^l|-\Delta f_{\min}^l}{\Delta f_{\max}^l-\Delta f_{\min}^l}+\frac{|\Delta e_a^l|-\Delta e_{\min}^l}{\Delta e_{\max}^l-\Delta e_{\min}^l}\right)\right) \tag{13}$$

where $\Delta f_a^l=f(\boldsymbol{X}_{(a-1)\cdot\mathrm{Nmp}+j}^l)-f(\boldsymbol{X}_{(a-1)\cdot\mathrm{Nmp}+j}^{l-1})$; $\Delta f_{\max}^l=\max\{\Delta f_a^l\}$; $\Delta f_{\min}^l=\min\{\Delta f_a^l\}$.

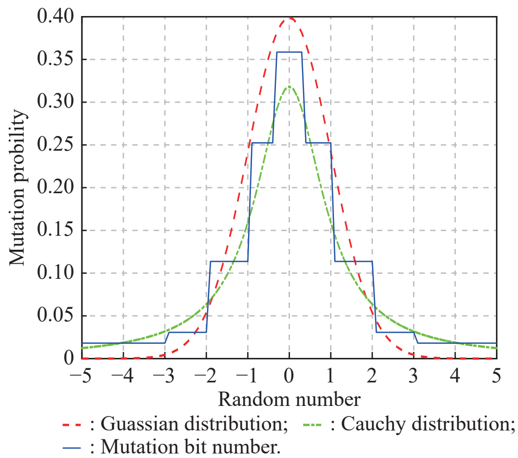


Fig. 1 Gaussian distribution and Cauchy distribution

The definition of Δe_j^l, $\Delta e_{\max}^l$, and $\Delta e_{\min}^l$ are similar to above. $f(\boldsymbol{X}_j^l)$ and $e(\boldsymbol{X}_j^l)$ are calculated by (1). If the value of η^l is less than the set threshold ε, the algorithm may fall into local convergence. At this time, Cauchy mutation is used to guide individuals to jump out of local convergence; otherwise, Gaussian mutation is selected to conduct a local search to improve the convergence speed. Then the procedures of local evolution are revealed in Algorithm 3, and the flow chart of NSFLA is given in Fig. 2.

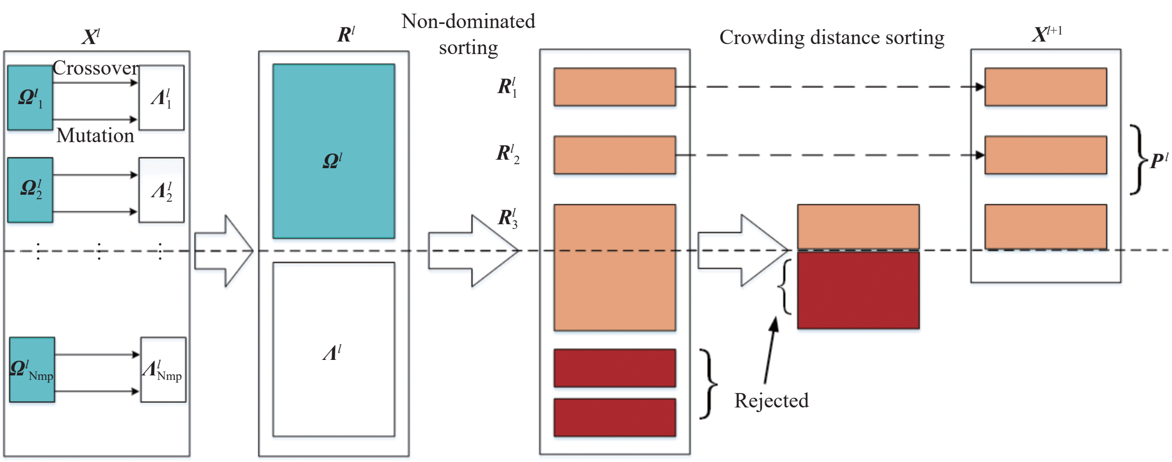


Fig. 2 Flow chart of NSFLA

Algorithm 3 Local evolution procedure

Step 1 Randomly generate $\rho \in (0,1)$. Initialize p_{cross} and ε.

Step 2 Combine every two individuals in $\boldsymbol{\Omega}_j^l$ $\left(\boldsymbol{X}^l_{(a-1)\text{Nmp}+j}, \boldsymbol{X}^l_{(\text{Nem}-a)\text{Nmp}+j}\right)$, $(a=1,2,\cdots,\text{Nem}; j=1,2,\cdots,\text{Nmp})$.

Step 3 If $\rho \geqslant p_{\text{cross}}$, $\left(\boldsymbol{X}^l_{(a-1)\cdot\text{Nmp}+j},\ \boldsymbol{X}^l_{(\text{Nem}-a)\cdot\text{Nmp}+j}\right)$ in $\boldsymbol{\Omega}_j^l$ perform crossover operation, generate new offspring, denoted as $\boldsymbol{\Lambda}_j^l$. Otherwise, let $\boldsymbol{\Lambda}_j^l = \boldsymbol{\Omega}_j^l$.

Step 4 Update η_j^l based on (13).

Step 5 If $\eta_j^l < \varepsilon$, perform Cauchy mutation. Otherwise, perform Gaussian mutation. Update the mutation number based on (10).

Step 6 Denote $\boldsymbol{R}_j^l = \boldsymbol{\Lambda}_j^l \cup \boldsymbol{\Omega}_j^l$.

3.3 Complexity analysis

The computation cost of the classical NSGA-II algorithm includes the initialization (T_i), evaluation (T_e), and update (T_u) for each solution. Assume d is the dimensionality of the search and p represents the upper bound of the quantity of the evolved solutions. Thus, the time complexity of NSGA-II can be estimated as $T(d) = T_i + (T_e + T_u)\times p = d + (d+d)\times p = d(1+2\cdot p)$. Therefore, $O(d\cdot p)$ is the time complexity of the classical NSGA-II algorithm. Compared with NSGA-II, NSFLA introduces SFLA to replace the inside evolutionary scheme of GA, which divides the population into Nmp memeplexes, Nem individuals are included in each memeplex, that is $\text{Nmp}\times\text{Nem} = P$, and two improved schemes, including crossover (T_c) and discrete adaptive mutation (T_m). In addition, the change rate of the objective function (T_r) should be considered. Therefore, the worst-case time complexity of NSFLA can be calculated as $T(d) = T_i + (T_c + T_m)\times\text{Nem}\times\text{Nmp} + T_r\times\text{Nmp}$, due to $T_r\times\text{Nmp} < T_r\times\text{Nmp}\times\text{Nem} = T_r\times p$, one can get $T(d) < T_i + (T_c + T_m + T_r)\times p = d(1+3\cdot p)$, and the computation cost of NSFLA can be expressed as $O(d\cdot p)$, which is the same as that of the classical NSGA-II.

4. Simulation results and discussions

To evaluate the performance of the proposed NSFLA for the DWTA problem and the influence of parameters on the algorithm, five experiments, including EX-1—EX-5 are carried out on the MATLAB/Simulink platform. The EX-1 and EX-2 are basic exploratory experiments; the aims are to obtain the optimal parameters of the improved evolution scheme of NSFLA. In EX-3, three multi-objective standard test functions are selected to verify the effectiveness of NSFLA when compared with NSGA-II. The EX-4 applies the proposed NSFLA to the actual air combat scenes. The optimization processes of residual target expected value F and killing target expected value are depicted in this experiment. The results of EX-5 are mainly about weapon-target assignment results optimized by NSFLA.

4.1 Problem formulation

In order to test the performance of NSFLA in solving DWTA, a set of problems is artificially generated based on three factors that affect problem complexity, m, r, and n. The meaning of these parameters is shown in Section 2. According to Table 1, four cases of air combat scenario are set as

Case I: $(m,r,n,v) = (4,8,6,2)$,
Case II: $(m,r,n,v) = (6,12,10,2)$,
Case III: $(m,r,n,v) = (12,24,20,3)$,
Case IV: $(m,r,n,v) = (18,36,36,3)$.

For instance, the parameters of Case III indicate that there are 12 flights and 20 enemy targets in air combat, each flight equipped with two missiles, and the combat includes three stages. Furthermore, for each stage, the corresponding calculation method of the kill probability matrix and threat matrix can refer to [13].

Table 1 Parameters of Cases I−IV

Parameter	Value	Type
m	{4, 6, 12, 18}	Integer
r	$\{2\times m, 2\times m, 2\times m, 2\times m\}$	Integer
n	{6, 10, 20, 36}	Integer
v	{2, 2, 3, 3}	Integer
n_k^h	1	Integer
m_j^h	{2, 2, 3, 3}	Integer
a_k	{2, 2, 2, 2}	Integer
f_{kj}^h	$\begin{cases}1, & \rho<\theta\\ 0, & \text{otherwise}\end{cases}$ where $\theta = 0.8 - 0.1h$	Float

4.2 Results analysis of EX-1

In the application of NSFLA, it is vital to observe the relationship between the number of memeplexes and the individuals in each memeplex of local evolution. Thus, a factorial experiment is conducted in EX-1, which considers six schemes under the restriction of a fixed population $P = 300$ with Nmp at six levels: 5, 10, 15, 20, 30, and 60, corresponding to the coefficients of Nem. Case III is taken as the research object, where the numbers of flights, enemies, and missiles are initial as 12, 20, and 24, respectively. The number of iterations N is 300. A total of three stages are considered in this air combat. In addition, it is assumed that no other aircraft join or withdraw during the interval. The main effects of EX-1 are summarized in Table 2, which F_{avg} and E_{avg} represent the average optimal values over 20 replications.

Table 2 Main effects of EX-1

(Nmp, Nem)	(F_{avg}, E_{avg})	(F_{std}, E_{std})	T_{avg}	T_{std}	N_{avg}	Ind
(5,60)	(38.24,10.27)	(0.005 31,0.005 07)	16.293 1	1.468 2	63.024 1	0.318 4
(10,30)	(37.37,11.51)	(0.004 06,0.004 01)	15.674 2	0.925 7	58.285 2	0.762 7
(15,20)	**(37.12,12.01)**	**(0.003 52,0.003 37)**	**14.720 3**	**0.506 0**	**52.750 3**	**1.924 1**
(20,15)	(37.13,11.79)	(0.003 47,0.003 64)	14.926 8	0.740 1	58.951 1	1.795 6
(30,10)	(37.45,11.23)	(0.005 09,0.004 98)	15.647 2	0.836 4	60.204 7	0.650 7
(60,5)	(38.03,11.02)	(0.006 46,0.006 32)	15.935 6	1.267 9	62.605 9	0.316 6

The corresponding standard deviations are represented by F_{std} and E_{std}. T_{avg} and T_{std} denote the mean value and standard deviation of convergence time. In addition, N_{avg} represents the mean convergence iteration. Ind is a numerical index of performance that reflects the comprehensive performance under the corresponding combination, which defined as

$$\text{Ind} = \lambda \frac{E_{avg}}{F_{avg}F_{std}E_{std}} \cdot \frac{P \cdot N}{T_{std}T_{avg}N_{avg}}. \tag{14}$$

It can be seen from the experimental results in Table 2 that different combinations of (Nmp, Nem) can result in various performances, where the best one among all schemes is presented in bold type. When the initial population number P of solutions is 300, (Nmp, Nem) = (15, 20) obtains the best fitness function, convergence rate and stability. A natural conclusion can be drawn from the analysis that an approximate memeplex division is essential to individual evolution in NSFLA. In addition, the comprehensive performance of the combined pools of $P = 150$, $P = 200$, and $P = 500$ are given in Table 3, and the bold values also the optimal combination are adopted in the following few experiments.

Table 3 Comprehensive performance of the combination pools

P=150		P=200		P=500	
(Nmp, Nem)	Ind	(Nmp, Nem)	Ind	(Nmp, Nem)	Ind
(5,30)	0.266 8	(5,40)	1.450 2	(10,50)	4.934 1
(10,15)	0.295 5	(10,20)	1.588 4	(20,25)	5.175 5
(15,10)	**0.303 1**	**(20,10)**	**1.603 2**	**(25,20)**	**5.302 2**
(30,5)	0.276 4	(40,5)	1.479 4	(50,10)	4.981 6

4.3 Results analysis of EX-2

EX-2 aims to test the influence of the promoted crossover and mutation scheme. Parameters including the crossover probability p_{cross} and mutation threshold value ε are taken into consideration. The detailed local evolution process of the memeplexes is shown in Subsection 3.2. The selection scheme of mutation types is based on Fig. 1 and (10). Based on EX-1, Case II and Case III are taken into account, and the most effective combination of Nmp and Nem is chosen from Table 3. The scenario is the same as in EX-1.

The main effects of EX-2 are summarized in Table 4. For each problem, three different crossover probabilities p_{cross} are given as 0.2, 0.5, and 0.8, and three levels of the mutation threshold ε: 2.0, 6.0, and 10.0 corresponding to each p_{cross}. The optimal index in each combination is presented in bold.

Table 4 Main effects of EX-2

Case	Criteria	ε								
		$p_{cross}=0.2$			$p_{cross}=0.5$			$p_{cross}=0.8$		
		2.0	6.0	10.0	2.0	6.0	10.0	2.0	6.0	10.0
II	F_{avg}	8.247 3	8.247 2	8.247 3	8.246 1	8.245 8	8.246 0	8.247 0	8.247 2	8.247 2
	E_{avg}	4.805 1	4.805 0	4.805 5	4.807 9	4.808 2	4.806 8	4.808 1	4.811 2	4.807 8
	T_{avg}	9.600 5	9.131 7	8.905 3	9.522 4	9.012 7	8.905 0	9.673 2	9.126 0	9.107 1

Continued

Case	Criteria	ε: $p_{cross}=0.2$, 2.0	6.0	10.0	$p_{cross}=0.5$, 2.0	6.0	10.0	$p_{cross}=0.8$, 2.0	6.0	10.0
	N_{avg}	18.5207	15.3214	15.8553	17.0524	13.3285	16.5227	14.0952	13.4100	16.2571
	Ind	13.4931	18.4212	**19.3138**	11.7050	**21.5326**	16.6947	15.4371	**19.0545**	18.9267
III	F_{avg}	37.2723	37.2715	37.2771	37.1365	37.1361	37.1473	37.1642	37.1637	37.1810
	E_{avg}	12.0548	12.1036	12.0872	12.1429	12.2550	12.2546	12.2510	12.2649	12.2588
	T_{avg}	14.3106	13.5501	13.1864	13.3601	12.9197	13.0985	14.6062	13.0089	13.4272
	N_{avg}	65	59	61	62	55	56	66	62	62
	Ind	0.4060	**0.5009**	0.4978	0.5329	**0.6220**	0.6212	0.4576	**0.0538**	0.5296

As seen from the three sets of data in Table 4, for Case II, when the mutation threshold ε is set to 6.0, regardless of whether p_{cross} is 0.2, 0.5, or 0.8, the algorithm will obtain relatively better operation results, including for the fitness function, convergence iteration, computing time and stability. Similarly, when ε is fixed, the NSFLA gets more effective solutions when p_{cross} equals 0.5. For Case III, when the number of flights, missiles, and stages are expanded, the experimental data show the same result, that is, the crossover rate p_{cross} is 0.5, the mutation threshold is $\varepsilon = 6.0$, and the NSFLA obtains the optimal performance. In addition, the index variation tendencies of Case I and Case IV are presented in Fig. 3, which are obtained by fitting the experimental data.

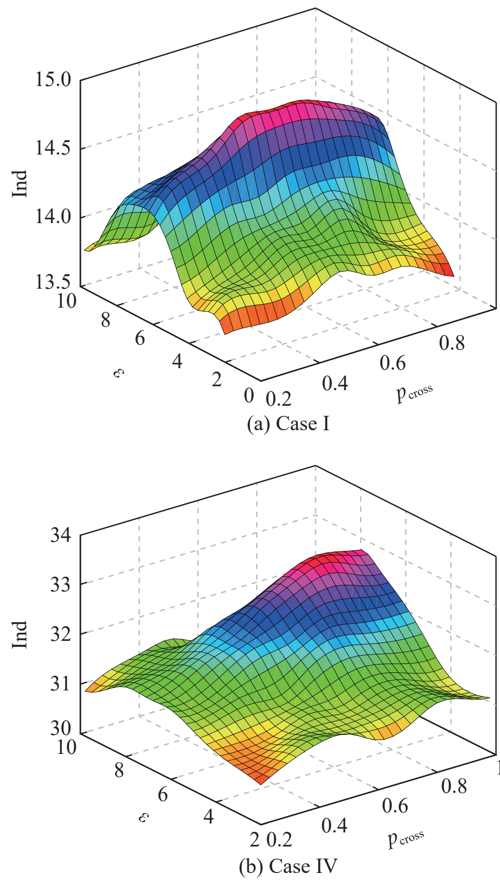


Fig. 3 Index variation of Case I and Case IV

It can be seen from Fig. 3 that both figures show the phenomenon of highs in the middle and lows in the surrounding areas, which can be explained by the following. When the mutation threshold and cross-rate are selected too small, a small search range and an algorithm that quickly falls into local convergence will result. In contrast, values for these two parameters that are too large may cause instability. Therefore, according to Fig. 3, $\varepsilon = 6.2$ and $p_{cross} = 0.65$ are the optimal combination for Case I, and $\varepsilon = 8.1$, $p_{cross} = 0.76$ for Case IV.

4.4 Results analysis of EX-3

In order to further verify the general optimization performance of NSFLA, three standard test functions including Binh1, Fonseca2, and Poloni have been selected. The number of initial population individuals for both NSFLA and NSGA-II is 100. The total evolution iteration N is 100, and the corresponding parameters of NSFLA refer to the optimal solution of EX-1 and EX-2. Each test function runs 20 times. The relevant performance parameters include (i) average convergence degree C_{avg}; (ii) average convergence iteration N_{avg}; (iii) convergence standard deviation C_{std} . The results are given in Table 5.

Table 5 Main effects of EX-3

Parameter	Binh1 NSFLA	Binh1 NSGA-II	Fonseca2 NSFLA	Fonseca2 NSGA-II	Poloni NSFLA	Poloni NSGA-II
C_{avg}	0.00015	0.00020	0.00012	0.00077	0.00038	0.00170
N_{avg}	6.68508	7.00131	5.24771	5.78068	6.52404	7.13399
C_{std}	0.02091	0.02417	0.00096	0.00154	0.11474	0.13550

The results from Table 5 prove that the convergence performance has indeed been improved for NSFLA. The proposed SFLA scheme replaces GA as the internal evolution mechanism improves the depth, breadth, and speed of evolution. Especially the convergence degree has been improved at least 30% according to statistical results. The convergence rate is also accelerated for the three test functions. From the standard deviation of the optimal solution distribution, NSFLA presents more reliable stability.

4.5 Results analysis of EX-4

The aim of EX-3 is to compare the performance of the NSFLA with NSGA-II toward the multi-objective optimization problem of weapon-target assignment in air combat. The relevance between the residual target expected value F and iteration times for NSFLA and NSGA-II are given in Fig. 4. The curves of the killing target expected value E are depicted in Fig. 5. Both methods execute 20 runs for each case, and the numbers of flights, missiles, stages, and any other parameters are determined according to Table 1. To ensure a comprehensive comparison, the threat matrix $\boldsymbol{T}$, kill probability matrix $\boldsymbol{P}$, and time window constraint is set to each algorithm's same values. The number of initial population individuals for both algorithms from Case I to Case IV are set to 150, 200, 300, and 500, and the iteration times are set to 50, 50, 150, and 300, respectively. In addition, the related parameters of NSFLA refer to the optimal solution of EX-1 and EX-2. In terms of NSGA-II, the crossover probability and mutation probability are initialized to 0.5 and 0.1, respectively.

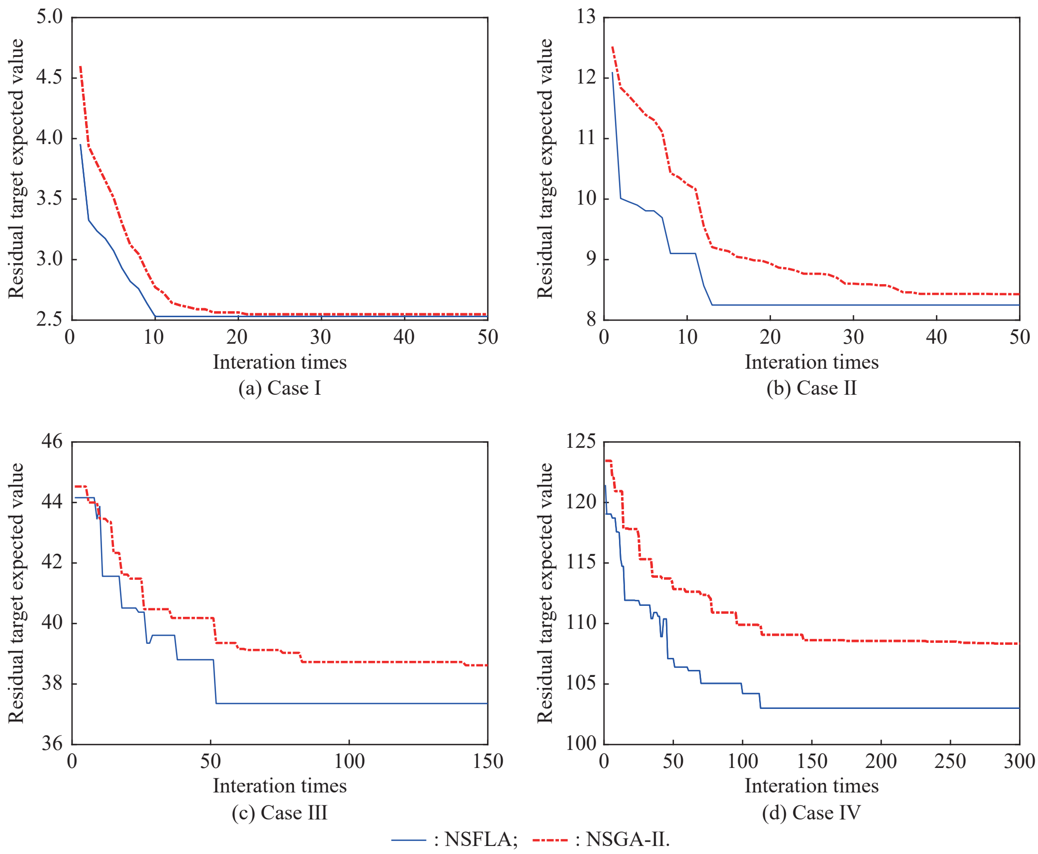
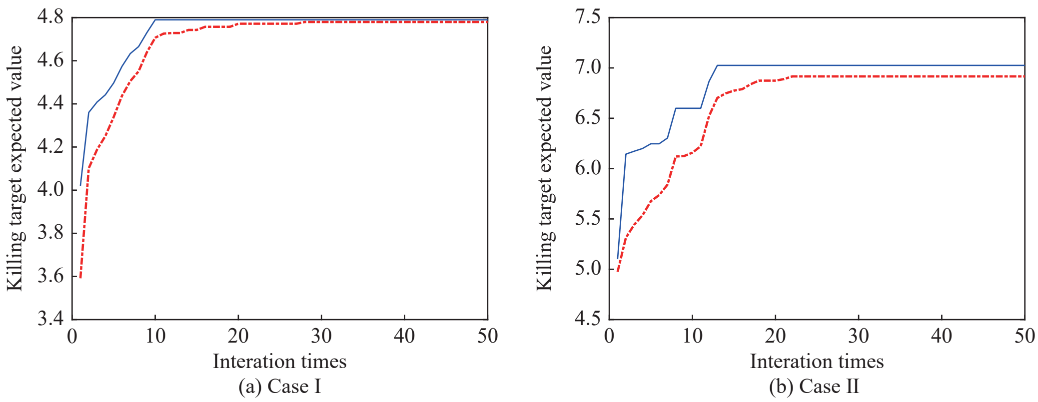


Fig. 4 Residual target expected value *F*



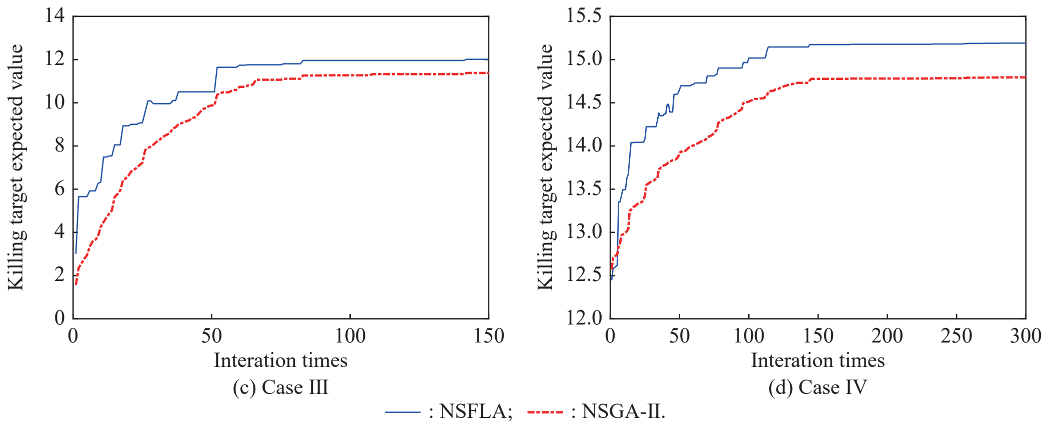


Fig. 5 Target killing expected value *E*

From Fig. 4 and Fig. 5, we can arrive at the following conclusions.

(i) In terms of the stable values of optimization objective F and E. NSFLA and NSGA-II act out resemble optimization ability concerning small-scale air combat. With the increase in the number of participating aircraft and the engagement stage, NSFLA can better solve large-scale multi-objective optimization problems and guide the problem in a better direction.

(ii) With regard to the convergence rate, we can see that NSFLA converges around the 12th, 15th, 53rd, and 127th generations. While the convergence generation for NSGA-II is 18th, 23th, 70th, and 160th, respectively, which all lag behind NSFLA and the degree are more remarkable with the expansion of air combats.

(iii) For the evolution curves of F and E under these two algorithms, it is obvious that the curves of NSGA-II display abnormal fluctuations. The same phenomenon is not so significant for NSFLA, which means NSFLA is more stable than NSGA-II.

4.6 Results analysis of EX-5

In order to verify the adjustment ability of NSFLA in solving the DWTA problem, that means, in this experiment, the battlefield environment may change during the stage interval. This experiment will also give the optimal weapon-target assignment results of Case II and Case III. For each case, the following three situations are considered:

Situation A: The number of enemy targets remains unchanged;

Situation B: Several other enemy targets engaged in;

Situation C: Some targets exit during the interval. The optimal assignment schemes for Situation A of Case II are given in Fig. 6.

	1	2	3	4	5	6
1	7	10	9	5	8	2
2	9	5	3	1	4	6

(a) Scheme 1

	1	2	3	4	5	6
1	7	4	9	5	1	3
2	9	5	2	8	10	6

(b) Scheme 2

Fig. 6 Assignment results for Situation A of Case II

Fig. 6 reveals the two optimal assignment schemes. The residual and killing target values E are 8.2447 and 7.9056 for scheme 1 and 8.2417 and 7.9005 for scheme 2, constituting a non-dominated relationship. The element k in the ith$(i=1,2)$ row and jth$(j=1,2,\cdots,6)$ column of the table means that the missile of flight j is allocated to the target k in the ith stage.

Then, the assignment results for Situation B are given in Fig. 7, and the added three enemy targets are codes as No.11, No.12, and No.13.

	1	2	3	4	5	6
1	7	10	4	5	8	2
2	13	9	6	1	11	12

(a) Scheme 1

	1	2	3	4	5	6
1	4	10	8	5	1	2
2	7	13	3	11	12	6

(b) Scheme 2

	1	2	3	4	5	6
1	4	10	8	3	1	2
2	13	6	5	11	12	7

(c) Scheme 3

Fig. 7 Assignment results for Situation B of Case II

From the results of Fig. 7, those above three schemes constitute the optimal allocation non dominated solution set. We can see that the algorithm can adapt to the dynamic changes of the battlefield and truly realize dynamic weapon-target assignment, which is reflected in that the newly added targets are all be assigned in stage 2. The participation of enemy targets dynamically changes the parameter value of optimization objectives. But owing to the self-adjustment capability of NSFLA, the complex air combat problem can be well solved.

The optimal assignment results for Situation C of Case II are given in Fig. 8, and the withdrawn two enemy targets are No.7 and No.8. The assignment scheme indicates that every missile is launched to maximize the expected killing values as much as possible.

	1	2	3	4	5	6
1	4	10	8	5	7	2
2	5	6	3	1	2	9

(a) Scheme 1

	1	2	3	4	5	6
1	4	10	8	5	7	1
2	3	6	5	2	2	9

(b) Scheme 2

Fig. 8 Assignment results for Situation C of Case II

Then the assignment results for Case III under the above three Situations A−C are given in Figs. 9−11. There are three combat stages; the dynamic process is considered during each interval. The specific combat processes are shown below the figures. For simplify, only one non-dominated solution is presented. In Fig. 10, two enemy targets are added in interval 1, numbered 21 and 22; three enemy targets are added in interval 2, numbered 23, 24, and 25. In Fig. 11, two enemy targets are withdrawn from interval 1, numbered 14 and 16; one enemy target is withdrawn from interval 2, numbered 7.

	1	2	3	4	5	6	7	8
1	18	11	16	19	12	2	5	9
2	12	13	14	6	20	15	10	1
3	5	18	4	17	3	7	3	8

Fig. 9 Assignment results for Situation A of Case III

	1	2	3	4	5	6	7	8
1	18	11	4	19	14	2	5	9
2	12	13	21	6	20	15	10	16
3	25	3	22	17	23	7	1	8

Fig. 10 Assignment results for Situation B of Case III

	1	2	3	4	5	6	7	8
1	7	11	4	19	12	2	5	9
2	12	13	1	11	20	15	10	17
3	1	13	4	17	6	18	3	8

Fig. 11 Assignment results for Situation C of Case III

The above comprehensive simulation analysis shows that the proposed NSFLA possesses good performance in adjustment ability, and the DWTA can be well solved by applying NSFLA.

5. Conclusions

This paper proposes the NSFLA to effectively solve the DWTA problem of cooperative air combat. The NSFLA introduces the methods of non-dominated sorting and crowding distance in NSGA-II to the evolution process of the SFLA. It decomposes the population into several subgroups and then performs non-dominated sorting on the two optimization indexes of maximum damage and minimum threat in each subgroup. When the subgroups evolve, crowding distance is used to screen individuals, which can effectively improve the distribution, convergence, and efficiency of the algorithm. At the same time, the internal evolutionary algorithm of the leapfrog algorithm is also improved to ensure that each individual, rather than only the worst individual, participates in the evolution to enrich the diversity of understanding. In additions, a discrete adaptive mutation algorithm based on the function change rate is proposed to accelerate the convergence of the algorithm to the optimal solution, improve the local search ability and convergence speed of the algorithm, and guide individuals to jump out of the local optimal region.

A series of experimental studies on the DWTA problem based on different algorithms is conducted. The results indicate that the modifications made above effectively improve the performance of the algorithm, especially the introduction of non-dominated sorting and congestion distance. The NSFLA achieves good performance in computing speed, convergence, and handling large-scale and dynamic target allocation problems.

References

[1] KLINE A, AHNER D, HILL R. The weapon-target assignment problem. Computers & Operations Research, 2019, 105(1): 226–236.

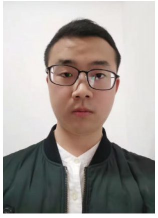
[2] LEE Z J, LEE C Y, SU S F. An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem. Applied Soft Computing, 2002, 2(1): 39–47.

[3] XIN B, CHEN J, ZHANG J, et al. Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. IEEE Trans. on Systems Man and Cybernetics Part C-Applications and Reviews,

2010, 40(6): 649–662.
[4] ZHANG Y, YANG R N, ZUO J L, et al. Weapon-target assignment based on decomposition-based evolutionary multi-objective optimization algorithms. Systems Engineering and Electronics, 2014, 36(12): 2435–2441. (in Chinese)
[5] LI N, HUAI W Q, WANG S D. The solution of target assignment problem in command and control decision-making behaviour simulation. Enterprise Information Systems, 2016, 11(7): 1059–1077.
[6] AHUJA R K, KUMAR A, JHA K C, et al. Exact and heuristic algorithms for the weapon-target assignment problem. Operations Research, 2007, 55(6): 1136–1146.
[7] KARASAKAL O, OZDEMIREL NE, KANDILLER L. Anti-ship missile defense for a naval task group. Naval Research Logistics, 2011, 58(3): 305–322.
[8] KORSAH G A, STENTZ A, DIAS M B. A comprehensive taxonomy for multi-robot task allocation. International Journal of Robotics Research, 2013, 32(12): 1495–1512.
[9] LUO L, CHAKRABORTY N, SYCARA K. Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks. IEEE Trans. on Robotics, 2015, 31(1): 19–30.
[10] LLOYD S P, WITSENHAUSEN H S. Weapons allocation is NP-complete. Proc. of the Summer Computer Simulation Conference, 1986: 1054−1061.
[11] SAHIN M A, LEBLEBICIOGLU K. Approximating the optimal mapping for weapon target assignment by fuzzy reasoning. Information Sciences, 2014, 255(1): 30–44.
[12] CHANG T Q, KONG D P, HAO N, et al. Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization. Applied Soft Computing, 2018, 70(1): 845–863.
[13] ZHAO Y, CHEN Y F, ZHEN Z Y, et al. Multi-weapon multi-target assignment based on hybrid genetic algorithm in uncertain environment. International Journal of Advanced Robotic Systems, 2020, 17(2): 1054–1058.
[14] LI X, LUO J, CHEN M R, et al. An improved shuffled frog-leaping algorithm with extremal optimization for continuous optimization. Information Sciences, 2012, 192(1): 143–151.
[15] CAI H P, LIU J X, CHEN Y W, et al. Survey of the research on dynamic weapon-target assignment problem. Journal of Systems Engineering and Electronics, 2006, 17(3): 559–565.
[16] NI M F, YU Z K, MA F, et al. A Lagrange relaxation method for solving weapon-target assignment problem. Mathematical Problems in Engineering, 2011, 4(1): 264–265.
[17] SAHIN M, LEBLEBICIOGLU K. Rule-based weapon target assignment on the battlefield. Proc. of the 18th IFAC World Congress, 2011: 13600−13605.
[18] DAVIS M T, ROBBINS M J, LUNDAY B J. Approximate dynamic programming for missile defense interceptor fire control. European Journal of Operational Research, 2017, 259(3): 873–886.
[19] BOARDMAN N T, LUNDAY B J, ROBBINS M J. Heterogeneous surface-to-air missile defense battery location: a game theoretic approach. Journal of Heuristics, 2017, 23(6): 417–447.
[20] WANG S H, YANG Q S, WANG R H, et al. Particle swarm optimization based weapon-target assignment for attacking ground targets. Electronics Optics & Control, 2017, 24(3): 36–40.
[21] MEI Z J, PENG Z H, ZHANG X L. Optimal dynamic weapon-target assignment based on receding horizon control heuristic. Proc. of the 13th IEEE International Conference on Control & Automation, 2017: 876−881.
[22] GAO C Q, KOU Y X, LI Y, et al. Multi-objective weapon target assignment based on D-NSGA-III-A. IEEE Access, 2019, 7: 50240−50254.
[23] STEUER R E. Multiple criteria optimization: theory, computation, and application. Malabar: Krieger, 1988.
[24] ZHA W Z, CHEN J, PENG Z H. Dynamic multi-team antagonistic games model with incomplete information and its application to multi-UAV. IEEE/CAA Journal of Automatica Sinica, 2015, 2(1): 74–84.
[25] WANG Y, GARCIA E, CASBEER D, et al. Cooperative control of multi-agent systems: theory and applications. New York: John Wiley & Sons, 2017.
[26] OSYCZKA A. An approach to multicriterion optimization problems for engineering design. Computer Methods in Applied Mechanics and Engineering, 1978, 15(3): 309–333.
[27] SARABI J A, ARAABI B N. How to decide when the sources of evidence are unreliable: a multi-criteria discounting approach in the Dempster-Shafer theory. Information Sciences, 2018, 448(1): 233–248.
[28] ZHOU D Y, PAN Q, ZHANG K. An improved discrete shuffled frog leaping algorithm for cooperative multi-target assignment of BVR air combat. Proc. of the IEEE International Conference on Signal Processing, Communications and Computing, 2014: 686−691.
[29] LAI C M, WU T H. Simplified swarm optimization with initialization scheme for dynamic weapon-target assignment problem. Applied Soft Computing, 2019, 82: 105542.
[30] EUSUFF M, LANSEY K. Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management, 2003, 129(3): 210–225.
[31] EUSUFF M, LANSEY E, PASHA F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Engineering Optimization, 2006, 38(2): 129–154.
[32] ROY P, CHAKRABARTI A. Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect. Applied Soft Computing, 2013, 13(11): 4244–4252.
[33] FETTAKA S, THIBAULT J, GUPTA Y. A new algorithm using front prediction and NSGA-II for solving two and three-objective optimization problems. Optimization and Engineering, 2015, 16(4): 713–736.

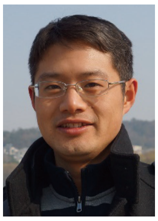
Biographies

ZHAO Yang was born in 1987. He received his M.S. degree in the College of Automation Engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. He is pursuing his Ph.D. degree in the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics. His research interests include advanced aircraft flight control and intelligent decision-making for modern air combat.
E-mail: zy@nuaa.edu.cn



LIU Jicheng was born in 1996. He received his B.S. degree in the College of Automation Engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2019. He is pursuing his M.S. degree in the College of Automation Engineering from Nanjing University of Aeronautics and Astronautics. His main research interests include advanced flight control theory and its application to multi-unmanned aerial vehicle flight control systems.
E-mail: ljc_uav@nuaa.edu.com

JIANG Ju was born in 1963. He received his M.S. and Ph.D. degrees in navigation, guidance and control from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1988 and 2007. He is a professor at Nanjing University of Aeronautics and Astronautics. His research interest includes navigation and flight control of aircraft systems.
E-mail: jiangju@nuaa.edu.cn



ZHEN Ziyang was born in 1981. He received his M.S. and Ph.D. degrees in control theory and control engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007 and 2010. He is a professor at Nanjing University of Aeronautics and Astronautics. His research interests include advanced aircraft flight control, unmanned aerial vehicle swarm cooperative control, and artificial intelligence.
E-mail: zhenziyang@nuaa.edu.cn