Journal of Systems Engineering and Electronics
Vol. 29, No. 3, June 2018, pp.519 – 529



# Approach to WTA in air combat using IAFSA-IHS algorithm

LI Zhanwu[1], CHANG Yizhe[2,*], KOU Yingxin[2], YANG Haiyan[2], XU An[2], and LI You[2]

1. College of Electronic Communication, Northwestern Polytechnical University, Xi'an 710072, China;
2. Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China

**Abstract:** In this paper, a static weapon target assignment (WTA) problem is studied. As a critical problem in cooperative air combat, outcome of WTA directly influences the battle. Along with the cost of weapons rising rapidly, it is indispensable to design a target assignment model that can ensure minimizing targets survivability and weapons consumption simultaneously. Afterwards an algorithm named as improved artificial fish swarm algorithm-improved harmony search algorithm (IAFSA-IHS) is proposed to solve the problem. The effect of the proposed algorithm is demonstrated in numerical simulations, and results show that it performs positively in searching the optimal solution and solving the WTA problem.

**Keywords:** air combat, weapon target assignment, improved artificial fish swarm algorithm-improved harmony search algorithm (IAFSA-IHS), artificial fish swarm algorithm (AFSA), harmony search (HS).

**DOI:** 10.21629/JSEE.2018.03.09

## 1. Introduction

Weapon target assignment (WTA) is known to be a typical non-polynomial complete (NPC) problem [1] such that any enumeration-based solver faces exponential computational complexity as the problem size increases. In other words, while any enumeration method may ensure a global solution, its computational requirement grows exponentially as the number of targets or weapons grows. To address the computational burden of exact solvers and the search times of global search solvers, several sophisticated search algorithms and heuristics have been proposed, such as genetic algorithms (GA) [2], simulated annealing (SA) [3], discrete particle swarm optimization (PSO) [4], permutation and tabu-search heuristics [5], a hybrid algorithm of a GA with an ant colony optimization (ACO) [6], artificial bee colony algorithm [7], differential evolutionary algorithm [8], backing tracking search optimization algorithm [9], cutting plane techniques as well as branch and bound algorithms [10], and auction algorithms [11 – 14]. However, as outstanding optimization algorithms with unique advantages, artificial fish swarm algorithm (AFSA) and harmony search (HS) are seldom used for solving WTA.

AFSA simulating the behavior of a fish inside water was proposed by Li [15] and has been applied in an engineering context [16 – 18]. The main artificial fish behaviors are the following: random, chasing, swarming and searching, which allows the algorithm to act better in global searching but poor in local searching.

HS, mimicking the improvisation process of music players, was originated by Geem [19]. HS performs excellently in quality of the solutions but with low search speed and strongly dependent on the initial solution.

Considering the distinctive features of each algorithm and less utility in WTA, a new algorithm organized by the improved AFSA and improved HS (IAFSA-IHS) is proposed in this paper and applied to solve the WTA problem.

This paper is organized as follows. In Section 2, formulation of the WTA problem is studied. The proposed algorithm named IAFSA-IHS is analyzed and designed in Section 3. Section 4 tests the performance of the proposed approach through numerical simulation on the WTA problem in three conditions. Section 5 concludes the paper.

## 2. WTA problem formulation

WTA refers to the reactive assignment of defensive weapons to engage or counter identified threats [20]. The primary concern is minimizing the total expected lethality of the enemy fighters with a limited number of weapons [20 – 26]. Considering the short time of air combat, the problem must be solved as close to the real-time as possible. Another characteristic is quality of the optimal solution, which causes critical effect on the following combat deployment.

Research on WTA modeling issues was firstly investigated in 1950s [20], since then the mathematical models have significantly improved [2,4,10,25]. In this paper, the problem is formulated as a combinatorial optimization problem, with the objective of minimizing the total lethality of the targets. The problem parameters are the kill

Manuscript received December 30, 2016.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61472441).

probabilities of the weapon-target pairs and the cost of the weapons. The kill probability is the weapon's probability of destroying a target if engaged. It depends on all aspects of engagement, such as the type of weapons, the type of state and location of the targets. The cost of weapons depends on the cost of each weapon and the engagement.

Consider the assignment problem of $m$ weapons to $n$ targets: let $p_{ij} \in [0,1]$ be the kill probability of weapon $i$ for target $j$, where $i \in \{1,2,\ldots,m\}$ and $j \in \{1,2,\ldots,n\}$. The WTA problem is formulated as follows:

$$J = \min\sum_{j=1}^{n}\left(\prod_{i=1}^{m}((1-p_{ij})\boldsymbol{C}_i)^{x_{ij}}\right)$$

$$\text{s.t.} \sum_{i=1}^{m} x_{ij} \geqslant 1,$$

$$\sum_{j=1}^{n} x_{ij} \leqslant 1,$$

$$\sum_{\substack{j=1\\ i\in \boldsymbol{A}(k)}}^{n} x_{ij} \leqslant N_{\boldsymbol{A}(k)}, \quad P_j \leqslant P_J \tag{1}$$

where $x_{ij}$ is the decision value indicating if weapon $i$ is assigned to target $j$, $\boldsymbol{C}_i$ denotes the cost of missile $i$, $\boldsymbol{A}(k)$ stands for the missiles classified by $k$, $N_{\boldsymbol{A}(k)}$ is the quantity of missiles marked by $\boldsymbol{A}(k)$ carried by the fighters, $P_J$ denotes the preseted threshold of targets survival probability of target $j$, and $P_j$, formulated as $P_j = \prod_{i=1}^{m}(1-p_{ij})^{x_{ij}}$, is the survival probability of target $j$.

In the above formulation, we minimize the sum of the weighted cumulative probability of threat lethality while ensuring all constraints are satisfied. Four constraints respectively denote that each target must be assigned by a weapon, each missile can be assigned to one target at most, missiles that already assigned in class $\boldsymbol{A}(k)$ must be fewer than or equal to $N_{\boldsymbol{A}(k)}$, the survival probability of target $j$ marked as $P_j$ should not be greater than the threshold of targets survival probability of target $j$ named $P_J$.

## 3. Construction of IAFSA-IHS

### 3.1 Improvements on AFSA

To improve the performance of the algorithm, certain modifications are introduced into AFSA, including initialization, visual, moving strategy and leap behaviour. Concrete implementations are presented in the following.

#### 3.1.1 Initialization

The initial solution causes crucial influences on convergence performance of the algorithm. The initial solution is randomly generated in most of the current study, by which it is difficult to get the solution of high quality. Therefore a hybrid initialization method involving chaos, information entropy and opposition-based learning methods is proposed and described in Fig. 1.

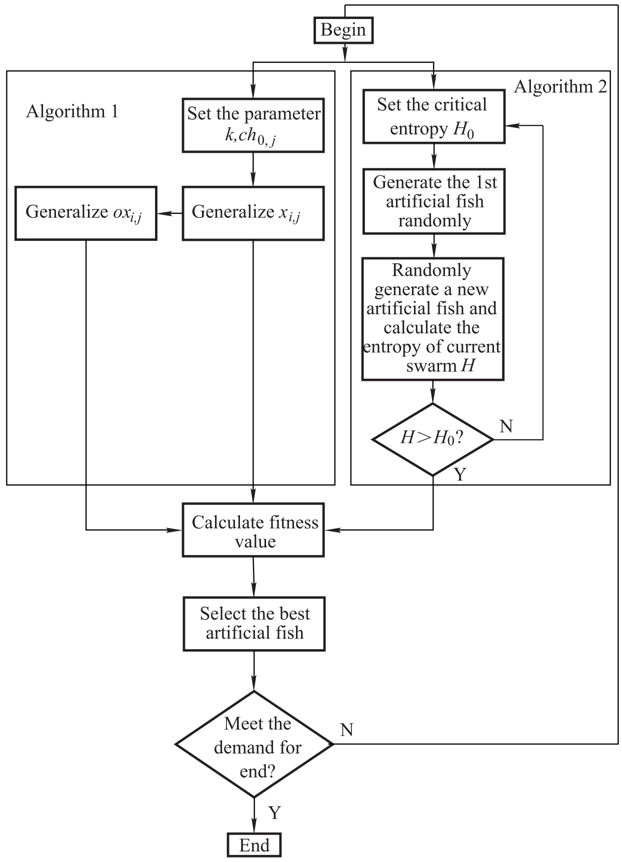



**Fig. 1 Process of initialization**

Assume that there are $p$ artificial fishes and each one has $q$ variables in the swarm. $X_{rs}$ denotes the parameter $s$ inside fish $r$, $r \in \{1,\ldots,p\}$, $s \in \{1,\ldots,q\}$. The pseudocode for the above two algorithms is described as follows:

**Algorithm 1** Initialization based on logistic chaos & opposition-based learning

Initialize parameter: set the maximum chaotic iteration number $ingen$

for $r = 1 : p$ do
  for $s = 1 : q$ do
    for $o = 1 : ingen$ do
      $ch_{os} = \sin(\pi \cdot ch_{o-1,s})$ // Update the chaotic factor
      $X_{os} = X_{\min,s} + ch_{os}(X_{\max,s} - X_{\min,s})$ // Generate initial chaotic solution
    end for
    $OX_{os} = X_{\min,s} + X_{\max,s} - X_{rs}$ // Generate initial opposition-based learning solution
  end for
end for

**Algorithm 2** Initialization based on information entropy

Set the critical entropy $H_0$
$H = 0$
while $H < H_0$
  for $r = 1 : p$
    for $t = r + 1 : p$
      for $s = 1 : q$
        $X_{ts} = X_{\min,s} + rand \cdot (X_{\max,s} - X_{\min,s})$
        $P_{rt}^{s} = 1 - \dfrac{|X_{rs} - X_{ts}|}{A_s^{\max} - A_s^{\min}}$ // Difference ratio between $X_{rs}$ and $X_{ts}$, $A_s^{\max}$ and $A_s^{\min}$ are the maximum and minimum values of the variable $s$ respectively
        $H_s = \sum\limits_{r=1}^{p}\sum\limits_{t=2}^{p}(-P_{rt}^{s}\lg P_{rt}^{s})$ // Entropy of variable $s$ in artificial fish swarm
        $H = \dfrac{1}{q}\sum\limits_{s=1}^{q} H_s$ // Entropy of the whole swarm
      end for
    end for
  end for
end while

3.1.2 Pseudo-parallel evolution

In the late period of search, artificial fish will move more slowly and be mislead to local optimums. A pseudo-parallel evolution strategy that divides the artificial fish swarm into multiple sub-swarm is introduced for solving the problem. Each sub-swarm executes search for optimum independently and exchanges the optimal solution with others at every iteration to maintain the search space and prevent premature.

3.1.3 Artificial fish moving strategy

To improve the search ability of the artificial fish, a moving strategy which combines maintaining the self advantage with absorbing the advantage from others is proposed below:

$$X_r^{gen+1} = \frac{gen}{\max gen}X_r^{gen} + \frac{\max gen - gen}{\max gen}(X_u^{gen} - X_r^{gen}) \quad (2)$$

where $gen$ is the current iteration. $\max gen$ is the maximum iteration. $X_r^{gen+1}$, $X_r^{gen}$ are the positions of artificial fish $r$ at iteration $gen+1$ and, $gen$ respectively. $X_u^{gen}$ is the position of artificial fish $u_{at}gen$ iteration. The first part, $\dfrac{gen}{\max gen}X_r^{gen}$, describes artificial fish keeping the self superiority, the second part inspired by the neighborhood search strategy, $\dfrac{\max gen - gen}{\max gen}(X_u^{gen} - X_r^{gen})$, indicates the absorb for advantage from another artificial fish $u$ within the visual range.

3.1.4 Self adaptive visual procedure

Visual, which has been deeply studied, acts as a key factor that directly influences the convergence speed and accuracy of AFSA.

In initial iterations, the visual varies in accordance with the upper formulation for ensuring good performance in convergence speed and global searching, thereafter changes by means of the next one for better convergence accuracy. However, there are two deficiencies existed, the switch condition from the upper formulation to the one below is not quite clear and the visual at the last iteration is not easy to obtain.

The main purpose is to ensure the artificial fish searching with larger visual within initial iterations and with smaller visual afterward, therefore a modified visual model concerned with iteration and convergence speed is constructed as follows:

$$visual_r^{gen} = \begin{cases} \max\left\{visual \cdot |\boldsymbol{D}_r(gen) - \boldsymbol{D}_r(gen-1)| \cdot \mathrm{e}^{-30\left(\frac{gen}{\max gen}\right)^S}, 1\right\}, & \boldsymbol{D}_r(gen) - \boldsymbol{D}_r(gen-1) \neq 0 \\ visual \cdot |\boldsymbol{D}_r(gen - gen') - \boldsymbol{D}_r(gen - gen' - 1)| \cdot \mathrm{e}^{-30\left(\frac{gen}{\max gen}\right)^S}, & \boldsymbol{D}_r(gen) - \boldsymbol{D}_r(gen-1) = 0 \end{cases} \quad (3)$$

where $gen'$ measures the iterations at which the optimal solution stays unchanged. $visual$ is the initial visual. $\boldsymbol{D}_r(gen_{last})$ is the solution at the last iteration before $\boldsymbol{D}_r(gen) - \boldsymbol{D}_r(gen-1) = 0$. $\boldsymbol{D}_r(gen_{last}-1)$ is the solution at the previous iteration before $gen_{last}$. $S$ is a constant number that could be 1, 3, 5, 10.

From (3) we can see that adjustment in visual changes one dimension at least, and gets smaller as $\boldsymbol{D}_r(gen) - \boldsymbol{D}_r(gen-1)$ decreases, which helps to improve the accuracy of the solutions. When $\boldsymbol{D}_r(gen) - \boldsymbol{D}_r(gen-1) = 0$, the visual obtained by (3) will get larger, by which artificial fish is capable of escaping from local optimal regions.

3.1.5 Improvements on leap behaviour

For preventing the artificial fish from converging to local optimal solutions, leap behaviour that forces the artificial fish to change its location is adopted in certain references [18,27]. In this paper, artificial fishes execute leap behaviour when the optimal solution stands still after one iteration. Meanwhile, it is thought that the more iterations at which the optimal solution stagnates, the more extent to which the optimal solution varies, and the higher the probability of leap behaviour execution is. Accordingly, improvements on leap behaviour probability and leap behaviour strategy are designed as follows.

The leap behaviour is modified as

$$X_r^{gen+1} = X_r^{gen} + visual_r^{gen} \cdot \boldsymbol{\lambda} \cdot rand \tag{4}$$

where $\boldsymbol{\lambda}$ is a bool vector, and the quantity of 1 in vector is obtained from

$$\theta = \max\left\{1, q - q \cdot round\left(\mathrm{e}^{-30\left(\frac{gen'}{gen}\right)^S} \cdot \frac{\boldsymbol{D}_r(gen-gen')}{\boldsymbol{D}_r(gen)}\right)\right\}. \tag{5}$$

Formula (5) makes it certain that one dimension at least, $q$ dimensions at most in location of artificial fish $r$ change by leap behaviour at each iteration. The bigger the $\dfrac{\boldsymbol{D}_r(gen-gen')}{\boldsymbol{D}_r(gen)}$ and $\dfrac{gen'}{gen}$ are, the more the dimension changes. By this way, leap behaviour is able to adjust the location of artificial fish dynamically and to achieve the goal of avoiding premature convergence.

## 3.2 Improvement on HS

In this article, improvements on HS contain two aspects: harmony memory consider rate (HMCR) and new harmony generating strategy.

### 3.2.1 Improvements on HMCR

Modifications on HMCR have been studied in many researches [28 – 30]. Through references, it can be concluded that HMCR should be given a small value to increase the diversity of the harmony memory and global search when solutions differ greatly. On the other hand, when the solutions are relatively resembled, HMCR should be raised for performing search more accurately and reduce the time for searching. In this paper, it is considered that iteration has effect on HMCR as solution does, especially the harmony memory consider rate is supposed to increase gradually at final iterations for acting better in global search. The HMCR is formulated from the perspectives of current iteration and deviations between harmony vectors as below:

$$\boldsymbol{HMCR}(gen) = \frac{\boldsymbol{E}_{\min}(gen)}{\boldsymbol{E}_{\max}(gen)} \mathrm{e}^{\frac{gen}{\max gen}-1} \tag{6}$$

where $\boldsymbol{E}_{\min}(gen)$ and $\boldsymbol{E}_{\max}(gen)$ represent the best and worst solution in harmony memory at iteration $gen$ respectively. The HMCR in (6) utilizes dynamic mechanism to select the harmony vector under various circumstance. The larger the $\dfrac{\boldsymbol{E}_{\min}(gen)}{\boldsymbol{E}_{\max}(gen)}$ is, the smaller the difference among the harmony vectors in harmony memory is, which indicates that $\boldsymbol{HMCR}(gen)$ becomes larger to converge to the optimum fastly. Likewise, the larger the $gen$ is, the larger the $\boldsymbol{HMCR}(gen)$ is, and the algorithm is able to have a faster convergence to the optimum.

### 3.2.2 New harmony improvisation

New harmony improvisation which is a critical factor influencing performance, is also seen as the key point in modifications on HS. A new method for harmony improvisation in terms of learning strategy is defined in this paper, which is described as below:

$$u_s^{gen} = \begin{cases} u_{r_1,s}^{gen} + \min\{c_1,c_2\} \cdot (u_{best,s}^{gen} - u_{r_1,s}^{gen}) + \max\{c_1,c_2\} \cdot (u_{r_2,s}^{gen} - u_{r_1,s}^{gen}), & \boldsymbol{D}_{best}(gen) - \boldsymbol{D}_{best}(gen-1) = 0 \\ u_{r_1,s}^{gen} + \max\{c_1,c_2\} \cdot (u_{best,s}^{gen} - u_{r_1,s}^{gen}) + \min\{c_1,c_2\} \cdot (u_{r_2,s}^{gen} - u_{r_1,s}^{gen}), & \boldsymbol{D}_{best}(gen) - \boldsymbol{D}_{best}(gen-1) \neq 0 \end{cases} \tag{7}$$

where $u_s^{gen}$ is the $s$th decision variable in harmony vector generated at iteration $gen$. $u_{r_1,s}^{gen}$, $u_{r_2,s}^{gen}$, $u_{best,s}^{gen}$ are the $s$th decision variables in harmony vector $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and the optimal harmony vector in harmony memory generated at iteration $gen$. $c_1$ and $c_2$ are learning coefficients, $0 \leqslant c_1, c_2 \leqslant 1, c_1 \neq c_2$.

The bigger one between $c_1$ and $c_2$ is selected to ensure the harmony affected by the neighborhood solution inside the harmony memory and avoiding premature when $\boldsymbol{D}_{best}(gen) - \boldsymbol{D}_{best}(gen-1) = 0$. Otherwise, the harmony improvisation is more influenced by the optimal solution, which helps converge to optimum more quickly.

## 3.3 Switch condition

### 3.3.1 Switch condition analysis

The switch condition acting as a key part of the proposed algorithm causes a great influence on speed and accuracy of convergence. It is dicussed by two aspects as switching iteration and individual stand error of artificial fish.

(i) Switching iteration

Switching iteration $gen_{switch}$ is used to make the proposed algorithm switch from improved AFSA to improved HS at a certain iteration, which is obtained by multiple concrete experiments.

(ii) Individual stand error of artificial fish

This factor reflects the deviation among artificial fish in the swarm. It should be switched to the HS for the following search while the individual stand error of artificial fish getting smaller, which means the search space getting smaller. The model of individual stand error of artificial fish, with regard to location and fitness value, is proposed as follows.

The stand error of location on artificial fish $lse$ is calculated by

$$\begin{cases} lse = \sqrt{\dfrac{1}{q-1}\displaystyle\sum_{s=1}^{q} lse_s^2} \\ lse_s^2 = \dfrac{1}{p-1}\displaystyle\sum_{r=1}^{p}(X_{rs} - \bar{X}_s)^2 \end{cases} \tag{8}$$

where $\bar{X}_s$ is the average location of all artificial fish in dimension $s$. The artificial fish is of high probability to gather in the local optimal region while each fish getting closer, as well as $lse$ being smaller.

The stand error of fitness value on artificial fish $fse$ is calculated by

$$fse = \sqrt{\sum_{r=1}^{p} \frac{(\boldsymbol{D}_r(gen) - \bar{\boldsymbol{D}}(gen))^2}{p-1}} \tag{9}$$

where $\bar{D}(gen)$ is the average fitness value. Likewise, the artificial fish seems to get trapped in the local optimum area while the fitness value of each fish resembling, as well as $fse$ being smaller.

From the mentioned above, the individual stand error of artificial fish is derived by

$$\begin{aligned} & se = \varphi_1 \cdot lse + \varphi_2 \cdot fse \\ & \text{s.t. } 0 \leqslant \varphi_1, \varphi_2 \leqslant 1 \\ & \varphi_1 + \varphi_2 = 1 \end{aligned} \tag{10}$$

where $\varphi_1$ and $\varphi_2$ represent the coefficients about $lse$ and $fse$ respectively. The switch to improved HS should be executed for global optimum if $se$ is smaller than the preseted value $se_{def}$, which means the stand error of fitness value and location fluctuates to a smaller extent, therefore the solution is weak.

3.3.2 Design for switch step

For the sake of fast convergence and quality of solutions, it is ruled in this paper that the switch is executed only if any condition is satisfied. Due to the different performances of the algorithms when facing various problems or benchmark functions, it is also regulated that the individual stand error of artificial fish is seen as the prior factor, and the switch flow is described in Fig. 2.

It can be seen from Fig. 2 that individual stand error of artificial fish is investigated at first, switch is executed if satisfied; otherwise check whether the algorithms are qualified to switch from the perspective of switching iteration, the switch is carried on if satisfied, or else the algorithm runs back to the improved AFSA. No matter which condition the algorithm is in, the artificial fish swarm turns into the initial solutions in HS and $gen = gen + 1$ only if it meets the demand for switch.

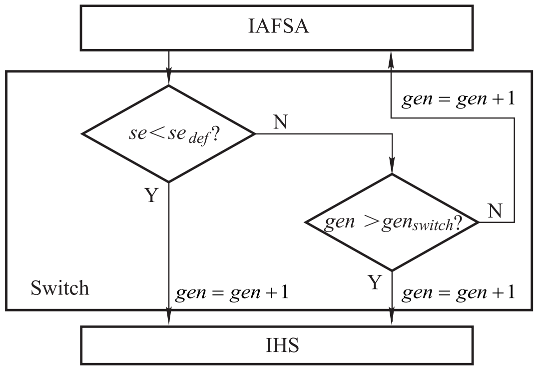



**Fig. 2 Switch flow for algorithm**

### 3.4 Flow for IAFSA-IHS

The concrete flow for the proposed algorithm is depicted in Fig. 3.

## 4. Experimental analysis

### 4.1 Parameter settings and simulation condition

For testing the performance of IAFSA-IHS comprehensively, quite a few algorithms are adopted into comparison. Parameters of chosen algorithms including GA, PSO, ACO, SA, AFSA, HS, improved binary AFSA (IBAFSA), global dynamic harmony search (GDHS), improved adaptive binary harmony search (IABHS), IAFSA-IHS, are listed in detail as Table 1.

**Table 1 Parameter settings for each algorithm**

| Algorithm | Parameter |
|---|---|
| GA | Population size $= 50, P_c = 0.8, P_m = 0.05$ |
| PSO | Particle swarm size $= 50, v_m = 0.8, \omega = 0.8$ |
| ACO | Ant swarm size $= 50, \alpha = 0.8, \beta = 0.8, \rho = 0.8$ |
| SA | $T_0 = 1\,000, T_{end} = 0.000\,1, \lambda = 0.6, L = 100$ |
| AFSA | Fish swarm size $= 50, visual = 1, step = 0,$ $trynumber = 10, \delta = 0.5$ |
| HS | $\mathrm{HMS} = 50, \mathrm{HMCR} = 0.8, \mathrm{PAR} = 0.3, bw = 0.02$ |
| IBAFSA | Fish swarm size $= 50, P_m = 0.01, \theta = 0.8, L = 10$ |
| GDHS | $\mathrm{HMS} = 50$ |
| IABHS | $\mathrm{HMS} = 50, \mathrm{HMCR} = 0.8, \mathrm{PAR} = 0.3, bw = 0.02,$ $\mathrm{NGC} = 30, C = 20$ |
| IAFSA-IHS | Fish swarm size $= 50, ingen = 50, H_0 = 0.3, S = 5,$ $visual = 1.5, \gamma_1 = 0.3, \gamma_2 = 0.5, c_1 = 0.6, c_2 = 0.4,$ $\varphi_1 = 0.6, \varphi_2 = 0.4, gen_{switch} = 20, se_{def} = 5$ |

Parameters in run are uniform as $\max gen = 100$. Simulation is run at Matlab 2011b (7. 0. 13. 564), and the CPU is Pentium(R) Dual-Core with 2.50 GHz and 2.00 GB RAM.

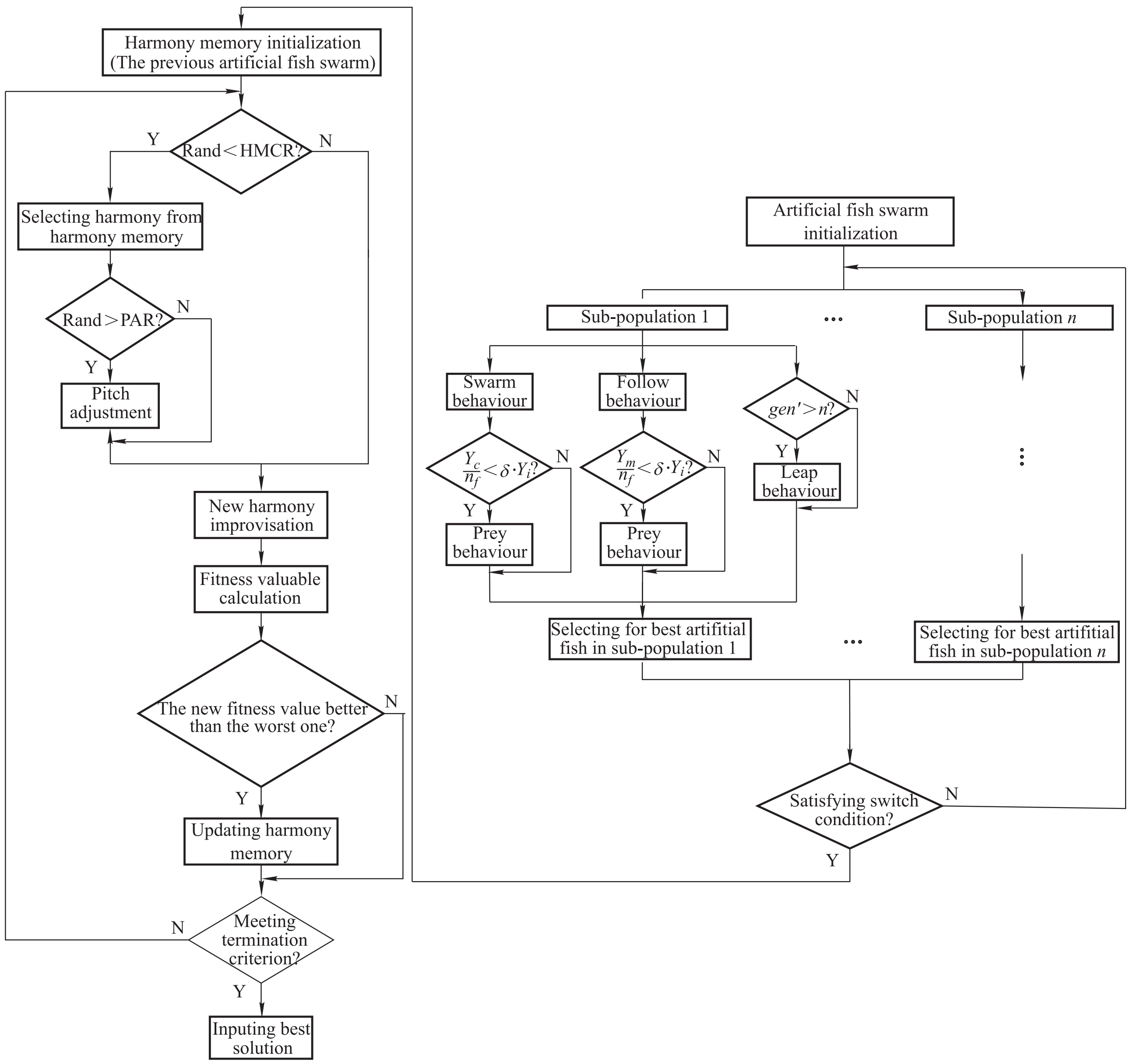



**Fig. 3 Flow for the IAFSA-IHS**

### 4.2 Test on WTA in air combat

#### 4.2.1 WTA problem while $P_J = 0.1$

There is red formation consisting of four fighters confronting with blue formation including six targets, and each fighter in red formation carries two missiles. The kill probability of the $i$th missile to the $j$th target $p_{ij}$ $(1 \leqslant i \leqslant 8, 1 \leqslant j \leqslant 6)$, is given in Table 2.

**Table 2 Kill probability of missiles to targets while $P_J = 0.1$**

| $T_j$ | $F_1$ | | $F_2$ | | $F_3$ | | $F_4$ | |
|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ |
| $T_1$ | 0.914 7 | 0.905 7 | 0.126 9 | 0.913 3 | 0.632 3 | 0.097 5 | 0.278 4 | 0.546 8 |
| $T_1$ | 0.957 5 | 0.964 8 | 0.157 6 | 0.970 5 | 0.957 1 | 0.485 3 | 0.800 2 | 0.141 8 |
| $T_3$ | 0.421 7 | 0.915 7 | 0.902 2 | 0.959 4 | 0.655 7 | 0.035 7 | 0.849 1 | 0.933 9 |
| $T_4$ | 0.678 7 | 0.757 7 | 0.743 1 | 0.392 2 | 0.655 4 | 0.171 1 | 0.706 0 | 0.831 8 |
| $T_5$ | 0.276 9 | 0.046 1 | 0.097 1 | 0.823 4 | 0.694 8 | 0.317 0 | 0.950 2 | 0.034 4 |
| $T_6$ | 0.438 7 | 0.981 5 | 0.765 5 | 0.795 1 | 0.186 8 | 0.509 7 | 0.445 5 | 0.646 3 |

$\boldsymbol{F}_k$ $(1 \leqslant k \leqslant 4)$ denotes the fighter $k$. $\boldsymbol{M}_i$ $(1 \leqslant i \leqslant 8)$ denotes the $i$th missile. $\boldsymbol{T}_j$ $(1 \leqslant j \leqslant 6)$ is the $j$th missile. Two kinds of missiles carried by fighters are marked as $N_{A(1)} = 5$, $N_{A(2)} = 3$, of which the cost can be normalized as $\boldsymbol{C} = [0.6 \quad 0.6 \quad 1 \quad 0.6 \quad 0.6 \quad 1 \quad 1 \quad 0.6]$. Threshold of targets survival probability is uniformly set as $P_J = 0.1$ $(1 \leqslant J \leqslant 6)$. Concrete parameters are set as in Table 1.

Compared with the algorithms mentioned above, the optimization process containing ten algorithms and the final assignment solution are shown in Fig. 4 and Tables 3–4.

The switch happens in the 20th generation and $se = 8.531\ 8$, IAFSA-IHS does not meet the switch condition regarding $se$ until triggering the switch on generation, thus in the 21st generation, the algorithm continues the search in IHS mode.

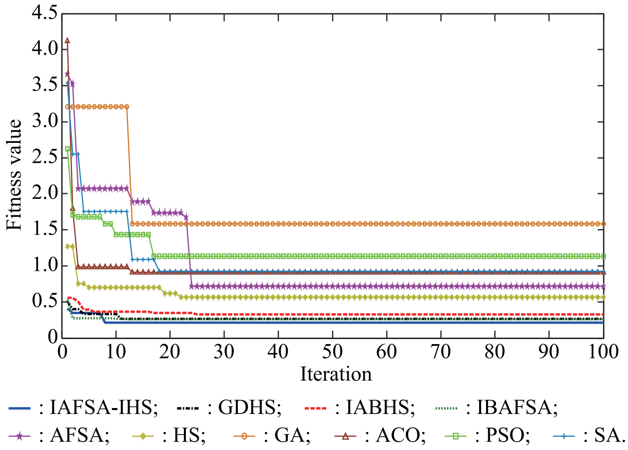



**Fig. 4 Optimization process while $P_J = 0.1$**

**Table 3 Comparison of tested algorithms while $P_J = 0.1$**

| Algorithm | $F_1$ | | $F_2$ | | $F_3$ | | $F_4$ | | Fitness value | Probability of target survivability | Cost | $t$/s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ | | | | |
| IAFSA-IHS | $T_2$ | $T_6$ | $T_4$ | $T_1$ | $T_4$ | $T_5$ | $T_5$ | $T_3$ | 0.215 4 | 0.336 3 | 6 | 2.368 7 |
| GDHS | $T_4$ | $T_6$ | $T_3$ | $T_1$ | $T_2$ | 0 | $T_5$ | $T_4$ | 0.247 4 | 0.326 2 | 5 | 2.966 3 |
| IABHS | $T_1$ | $T_6$ | $T_3$ | $T_5$ | $T_2$ | 0 | $T_4$ | $T_4$ | 0.326 7 | 0.470 6 | 5 | 2.522 4 |
| IBAFSA | $T_2$ | $T_6$ | $T_3$ | $T_1$ | $T_4$ | 0 | $T_5$ | $T_4$ | 0.257 1 | 0.353 3 | 5 | 3.001 6 |
| AFSA | $T_1$ | $T_2$ | $T_4$ | $T_5$ | $T_5$ | 0 | $T_3$ | $T_6$ | 0.711 7 | 0.935 9 | 5 | 3.539 1 |
| HS | 0 | $T_2$ | $T_6$ | $T_1$ | $T_4$ | $T_4$ | $T_5$ | $T_3$ | 0.568 5 | 0.757 9 | 5.4 | 4.039 1 |
| GA | $T_1$ | $T_2$ | $T_6$ | $T_5$ | $T_5$ | $T_3$ | $T_4$ | 0 | 1.584 5 | 1.6672 | 5.4 | 4.437 8 |
| ACO | $T_4$ | $T_1$ | $T_1$ | $T_2$ | $T_5$ | $T_6$ | $T_3$ | $T_4$ | 0.910 8 | 1.1123 | 6 | 4.189 |
| PSO | $T_3$ | $T_1$ | $T_4$ | $T_6$ | $T_5$ | $T_2$ | $T_1$ | $T_3$ | 1.132 2 | 1.3880 | 6 | 3.658 4 |
| SA | $T_5$ | $T_1$ | $T_4$ | $T_2$ | $T_5$ | $T_6$ | $T_1$ | $T_3$ | 0.924 8 | 1.1315 | 6 | 4.672 2 |

**Table 4 Remaining threat value of targets while $P_J = 0.1$**

| Algorithm | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ |
|---|---|---|---|---|---|---|
| IAFSA-IHS | 0.086 7 | 0.042 5 | 0.066 1 | 0.088 5 | 0.034 | 0.018 5 |
| GDHS | 0.086 7 | 0.042 9 | 0.097 8 | 0.030 5 | 0.049 8 | 0.018 5 |
| IABHS | 0.085 3 | 0.042 9 | 0.097 8 | 0.049 5 | 0.176 6 | 0.018 5 |
| IBAFSA | 0.086 7 | 0.042 5 | 0.097 8 | 0.058 | 0.049 8 | 0.018 5 |
| AFSA | 0.085 3 | 0.035 2 | 0.150 9 | 0.256 9 | 0.053 9 | 0.353 7 |
| HS | 0.086 7 | 0.035 2 | 0.066 1 | 0.285 6 | 0.049 8 | 0.234 5 |
| GA | 0.085 3 | 0.035 2 | 0.964 3 | 0.294 0 | 0.053 9 | 0.234 5 |
| ACO | 0.082 3 | 0.029 5 | 0.150 9 | 0.054 0 | 0.305 2 | 0.490 3 |
| PSO | 0.068 0 | 0.514 7 | 0.038 2 | 0.256 9 | 0.305 2 | 0.204 9 |
| SA | 0.068 0 | 0.029 5 | 0.066 1 | 0.256 9 | 0.220 7 | 0.490 3 |

It is obviously observed from Fig. 4 and Tables 3–4 that solution obtained by IAFSA-IHS gets the best fitness value with the highest speed. By comparison, the results derived by improved algorithms, such as GDHS, IBAFSA, IABHS, achieve higher fitness value and probability of target survivability with less cost, and the time spending is a bit more. Common standard algorithms without modifications, including AFSA, HS, GA, ACO, PSO, SA, act poorly in terms of the problem. Moreover, HS, GA, ACO and SA complete the process by spending too much time. Whatever the solution quality they derive, those algorithms are not fit for the above case due to the time spending.

#### 4.2.2 WTA problem while $P_J = 0.2$

To test its performance widely, threshold of targets survival probability is uniformly reset as $P_J = 0.2$, the results are shown in Fig. 5 and Tables 5–6.

**Table 5 Comparison of tested algorithms while $P_J = 0.2$**

| Algorithm | $F_1$ | | $F_2$ | | $F_3$ | | $F_4$ | | Fitness value | Probability of target survivability | Cost | $t$/s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ | | | | |
| IAFSA-IHS | $T_1$ | $T_2$ | $T_6$ | $T_3$ | 0 | $T_6$ | $T_5$ | $T_4$ | 0.393 5 | 0.525 2 | 5.4 | 0.479 6 |
| GDHS | $T_4$ | $T_2$ | $T_3$ | $T_1$ | 0 | $T_6$ | $T_5$ | $T_6$ | 0.433 6 | 0.610 1 | 5.4 | 0.671 4 |
| IABHS | $T_1$ | $T_3$ | 0 | $T_5$ | $T_2$ | $T_6$ | $T_6$ | $T_4$ | 0.532 2 | 0.756 1 | 5 | 0.762 3 |
| IBAFSA | $T_1$ | $T_6$ | 0 | $T_5$ | $T_2$ | 0 | $T_3$ | $T_4$ | 0.445 8 | 0.642 4 | 4 | 0.869 9 |
| AFSA | $T_1$ | $T_6$ | 0 | $T_2$ | $T_5$ | $T_2$ | $T_4$ | $T_3$ | 0.586 4 | 0.781 3 | 5 | 1.124 4 |
| HS | $T_2$ | $T_1$ | $T_3$ | $T_5$ | $T_4$ | 0 | $T_5$ | $T_6$ | 0.604 1 | 0.941 7 | 5 | 1.375 1 |
| GA | $T_3$ | $T_1$ | $T_1$ | $T_6$ | $T_5$ | $T_2$ | $T_6$ | $T_4$ | 1.263 3 | 1.762 4 | 6 | 1.967 |
| ACO | 0 | $T_6$ | 0 | $T_3$ | $T_1$ | $T_2$ | $T_5$ | $T_4$ | 0.713 8 | 1.156 5 | 4.4 | 1.996 3 |
| PSO | $T_1$ | $T_2$ | $T_3$ | $T_6$ | $T_4$ | 0 | $T_6$ | $T_5$ | 1.024 4 | 1.642 1 | 5 | 1.035 |
| SA | $T_1$ | $T_6$ | $T_2$ | $T_4$ | $T_3$ | $T_5$ | 0 | $T_5$ | 1.869 8 | 2.554 8 | 5 | 2.061 4 |

**Table 6 Remaining threat value of targets while $P_J = 0.2$**

| Algorithm | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ |
|---|---|---|---|---|---|---|
| IAFSA-IHS | 0.085 3 | 0.035 2 | 0.040 6 | 0.168 2 | 0.049 8 | 0.146 1 |
| GDHS | 0.086 7 | 0.035 2 | 0.097 8 | 0.181 3 | 0.049 8 | 0.159 3 |
| IABHS | 0.085 3 | 0.042 9 | 0.084 3 | 0.168 2 | 0.176 6 | 0.198 8 |
| IBAFSA | 0.085 3 | 0.042 9 | 0.150 9 | 0.168 2 | 0.176 6 | 0.018 5 |
| AFSA | 0.085 3 | 0.015 2 | 0.066 1 | 0.294 0 | 0.305 2 | 0.015 5 |
| HS | 0.094 3 | 0.042 5 | 0.097 8 | 0.344 6 | 0.008 8 | 0.353 7 |
| GA | 0.082 3 | 0.514 7 | 0.578 3 | 0.168 2 | 0.305 2 | 0.113 6 |
| ACO | 0.367 7 | 0.514 7 | 0.040 6 | 0.168 2 | 0.049 8 | 0.015 5 |
| PSO | 0.085 3 | 0.035 2 | 0.097 8 | 0.344 6 | 0.965 6 | 0.113 6 |
| SA | 0.085 3 | 0.842 4 | 0.344 3 | 0.607 8 | 0.659 5 | 0.015 5 |

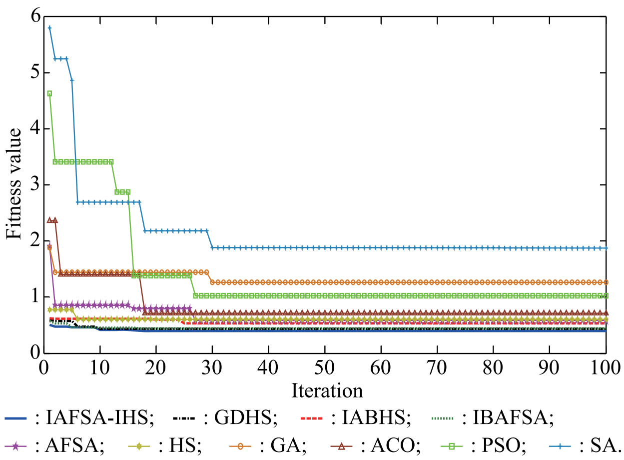



**Fig. 5 Optimization process while $P_J = 0.2$**

The switch happens in the 11th generation and $se = 2.395\,4$, IAFSA-IHS reaches switch condition concerning $se$ regardless of switch condition about $gen_{switch}$. Therefore in the 12nd generation, the algorithm has switched into IHS mechanism for further optimizing.

From Fig. 5 and Tables 5–6 above, it is concluded that all algorithms except GA meet the goal of reducing the consumption of missiles, which indicates the fact that the lower the expected damage to the target is, the less the missiles consumes. Besides, the fitness value and probability of target survivability become higher as the threshold of targets survival probability rises, which demonstrates that the assignment model and the algorithms solves the problem on both sides of damage to the targets and missiles consumption. Among the overall algorithms, SA performs worst in results with the longest span, which can be considered not suitable for the problem. Among other ones, it is easily found that algorithms with improvements outperform the ones without improvements. The proposed IAFSA-IHS achieves a far better performance in all kinds of results, which also meets the demand for threshold of targets survival probability, therefore demonstrating the priority of IAFSA-IHS for WTA.

### 4.2.3 WTA problem in large scale air combat

To evaluate the performance in further step, air combat in large-scale which contains six fighters carrying 12 missiles and 10 targets is studied after consideration of actual conditions. Parameters in algorithms are set identically to the ones in Section 4.1. Cost of each missile is normalized as $\boldsymbol{C} = [0.8, 0.8, 0.8, 0.8, 1, 1, 0.8, 0.8, 0.8, 1, 0.8, 1]$. Threshold of targets survival probability is uniformly set as $P_J = 0.2\ (1 \leqslant J \leqslant 10)$. Kill probability of missiles to targets are given in Table 7.

Through the numerical simulation based on previous parameters settings, the results are shown in Tables 8–9 and Fig. 6.

**Table 7 Kill probability of missiles to targets in large scale air combat**

| $T_j$ | $F_1$ | | $F_2$ | | $F_3$ | | $F_4$ | | $F_5$ | | $F_6$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ | $M_9$ | $M_{10}$ | $M_{11}$ | $M_{12}$ |
| $T_1$ | 0.927 4 | 0.759 3 | 0.437 2 | 0.790 1 | 0.516 1 | 0.322 9 | 0.786 1 | 0.878 6 | 0.961 7 | 0.778 4 | 0.989 4 | 0.345 9 |
| $T_2$ | 0.647 0 | 0.865 6 | 0.735 7 | 0.711 6 | 0.996 2 | 0.341 4 | 0.743 1 | 0.524 5 | 0.924 5 | 0.505 9 | 0.330 1 | 0.542 2 |
| $T_3$ | 0.880 2 | 0.762 9 | 0.612 6 | 0.907 5 | 0.648 9 | 0.870 7 | 0.634 4 | 0.589 9 | 0.669 8 | 0.658 3 | 0.931 6 | 0.582 8 |
| $T_4$ | 0.762 9 | 0.262 1 | 0.887 7 | 0.372 | 0.397 6 | 0.380 7 | 0.273 7 | 0.492 9 | 0.234 2 | 0.273 1 | 0.930 7 | 0.499 9 |
| $T_5$ | 0.382 1 | 0.654 5 | 0.315 7 | 0.919 5 | 0.404 5 | 0.361 7 | 0.577 | 0.400 1 | 0.468 5 | 0.466 3 | 0.735 5 | 0.819 1 |
| $T_6$ | 0.717 9 | 0.330 0 | 0.465 7 | 0.988 2 | 0.995 1 | 0.266 9 | 0.648 5 | 0.237 4 | 0.324 7 | 0.510 1 | 0.209 6 | 0.377 3 |
| $T_7$ | 0.292 4 | 0.725 1 | 0.947 9 | 0.556 1 | 0.376 6 | 0.251 0 | 0.990 0 | 0.594 4 | 0.399 3 | 0.985 9 | 0.417 4 | 0.262 6 |
| $T_8$ | 0.645 4 | 0.878 4 | 0.645 3 | 0.603 0 | 0.572 5 | 0.903 3 | 0.601 9 | 0.590 1 | 0.314 4 | 0.364 5 | 0.365 8 | 0.282 7 |
| $T_9$ | 0.578 7 | 0.904 4 | 0.946 7 | 0.261 4 | 0.747 4 | 0.481 8 | 0.372 3 | 0.817 4 | 0.279 7 | 0.357 9 | 0.446 5 | 0.844 4 |
| $T_{10}$ | 0.821 7 | 0.931 6 | 0.674 9 | 0.475 9 | 0.835 3 | 0.385 7 | 0.704 6 | 0.860 7 | 0.618 4 | 0.422 5 | 0.769 7 | 0.316 3 |

**Table 8 Comparison of tested algorithms in large scale air combat**

| Algorithm | $F_1$ | | $F_2$ | | $F_3$ | | $F_4$ | | $F_5$ | | $F_6$ | | Fitness value | Probability of target survivability | Cost | $t$/s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ | $M_9$ | $M_{10}$ | $M_{11}$ | $M_{12}$ | | | | |
| IAFSA-IHS | 0 | $T_9$ | $T_7$ | $T_3$ | $T_{10}$ | $T_8$ | 0 | $T_1$ | $T_2$ | $T_6$ | $T_4$ | $T_5$ | 0.697 8 | 0.792 0 | 8.8 | 3.217 4 |
| GDHS | $T_3$ | $T_2$ | $T_9$ | 0 | $T_6$ | $T_8$ | $T_7$ | $T_{10}$ | $T_1$ | 0 | $T_4$ | $T_5$ | 0.734 0 | 0.846 9 | 8.6 | 4.033 |
| IABHS | $T_3$ | $T_8$ | $T_4$ | $T_6$ | $T_2$ | 0 | $T_{10}$ | $T_9$ | $T_1$ | $T_7$ | $T_{10}$ | $T_5$ | 0.716 4 | 0.853 3 | 9.4 | 3.857 6 |
| IBAFSA | $T_2$ | $T_2$ | $T_8$ | $T_5$ | $T_{10}$ | $T_5$ | $T_7$ | $T_1$ | $T_9$ | $T_6$ | $T_3$ | $T_4$ | 2.246 1 | 2.528 4 | 10.4 | 3.815 9 |
| AFSA | $T_{10}$ | $T_{10}$ | $T_7$ | $T_8$ | $T_9$ | $T_3$ | $T_6$ | $T_2$ | $T_9$ | $T_5$ | $T_1$ | $T_4$ | 2.345 9 | 2.644 | 10.4 | 4.157 2 |
| HS | $T_6$ | $T_{10}$ | $T_9$ | $T_5$ | $T_3$ | $T_4$ | $T_7$ | $T_6$ | $T_5$ | $T_8$ | $T_1$ | $T_2$ | 2.222 3 | 2.463 7 | 10.4 | 4.038 2 |
| GA | $T_4$ | $T_1$ | $T_6$ | $T_9$ | $T_8$ | $T_5$ | $T_1$ | $T_2$ | $T_5$ | $T_3$ | $T_{10}$ | $T_7$ | 3.583 6 | 4.113 4 | 10.4 | 4.395 4 |
| ACO | $T_2$ | $T_3$ | $T_5$ | $T_{10}$ | $T_6$ | $T_8$ | $T_1$ | $T_4$ | $T_7$ | $T_4$ | $T_9$ | $T_8$ | 2.902 1 | 3.609 4 | 10.4 | 3.935 5 |
| PSO | $T_9$ | $T_3$ | $T_8$ | $T_9$ | $T_6$ | $T_5$ | $T_7$ | $T_4$ | $T_2$ | $T_{10}$ | $T_1$ | $T_3$ | 2.576 3 | 2.588 7 | 10.4 | 4.168 4 |
| SA | $T_4$ | $T_3$ | $T_3$ | $T_5$ | $T_9$ | $T_1$ | $T_8$ | $T_{10}$ | $T_6$ | $T_7$ | $T_6$ | $T_2$ | 2.486 0 | 2.882 2 | 10.4 | 4.507 7 |

**Table 9 Remaining threat value of targets in large scale air combat**

| Algorithm | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| IAFSA-IHS | 0.677 1 | 0.003 8 | 0.387 4 | 0.726‘3 | 0.163 4 | 0.011 8 | 0.737 4 | 0.635 5 | 0.720 3 | 0.009 5 |
| GDHS | 0.038 3 | 0.134 4 | 0.119 8 | 0.069 3 | 0.180 9 | 0.004 9 | 0.01 | 0.096 7 | 0.053 3 | 0.139 3 |
| IABHS | 0.038 3 | 0.003 8 | 0.119 8 | 0.112 3 | 0.180 9 | 0.011 8 | 0.014 1 | 0.121 6 | 0.182 6 | 0.068 0 |
| IBAFSA | 0.121 4 | 0.047 5 | 0.068 4 | 0.500 1 | 0.051 4 | 0.489 9 | 0.010 0 | 0.354 7 | 0.720 3 | 0.164 7 |
| AFSA | 0.010 6 | 0.475 5 | 0.129 3 | 0.500 1 | 0.533 7 | 0.351 5 | 0.052 1 | 0.397 0 | 0.182 0 | 0.012 2 |
| HS | 0.010 6 | 0.457 8 | 0.351 1 | 0.619 3 | 0.042 8 | 0.215 1 | 0.010 0 | 0.635 5 | 0.053 3 | 0.068 4 |
| GA | 0.051 5 | 0.475 5 | 0.341 7 | 0.237 1 | 0.339 3 | 0.534 3 | 0.737 4 | 0.427 5 | 0.738 6 | 0.230 3 |
| ACO | 0.213 9 | 0.353 0 | 0.237 1 | 0.368 6 | 0.684 3 | 0.004 9 | 0.600 7 | 0.069 4 | 0.553 5 | 0.524 1 |
| PSO | 0.010 6 | 0.117 6 | 0.129 3 | 0.765 8 | 0.684 3 | 0.004 9 | 0.010 0 | 0.354 7 | 0.311 2 | 0.577 5 |
| SA | 0.677 1 | 0.457 8 | 0.091 8 | 0.237 1 | 0.080 5 | 0.533 8 | 0.014 1 | 0.398 1 | 0.252 6 | 0.139 3 |

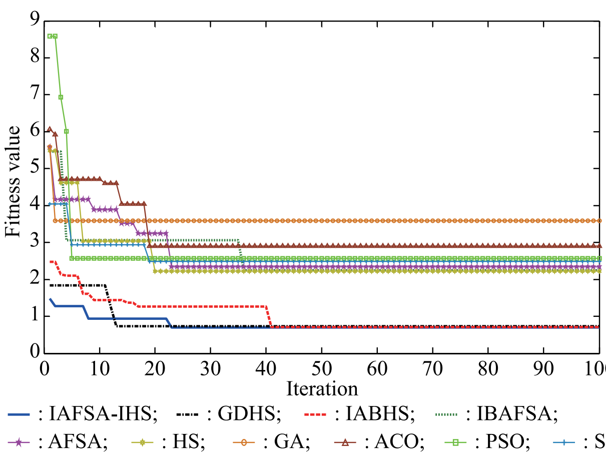



**Fig. 6 Optimization process concerning on large scale air combat**

The switch happens in the 17th generation and $se = 3.261\,6$, although IAFSA-IHS does not meet the switch condition regarding $gen_{switch}$, it satisfies the switch on $se$, which means in the 18th generation, the algorithm turns into IHS mechanism. From the figure and tables above, it can be concluded that the proposed algorithm IAFSA-IHS still outperforms the other ones. It is observed in Fig. 6 that improvement on initialization takes effect in the optimization process and IAFSA-IHS reaches the best outcome with a higher speed, which demonstrates IAFSA-IHS is still adopted in dealing with the WTA problem in large scale air combat. In addition, it is clearly seen that AFSA and HS are more capable and proper to solve the WTA problem in above cases than other standard algorithms through Figs. 4 – 6, which further proves their relative outstanding abilities in convergence speed and quality.

## 5. Conclusions

An approach to WTA in cooperative air combat is proposed. Firstly, the WTA model is constructed with regard to the threshold of targets survival probability and weapons consumption saving. Secondly, IAFSA-IHS, a new algorithm comprised by improved AFSA and improved HS is proposed. Finally, the proposed approach is tested by three concrete WTA problems in multiple aspects, it is easily found that the proposed approach not only obtains the best solution among all algorithms, but also finishes the converge process in overwhelmingly short time which satisfies the requirement for real-time in real combat conditions.

## References


[1] LLOYD S P, WITSENHAUSEN H S. Weapon allocation is NP-complete. Proc. of the Summer Simulation Conference, 1986: 1054 – 1058.

[2] LEE Z J, SU S F, LEE C Y. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Trans. on Systems, Man and Cybernetics, Part B, 2003, 33(1): 113 – 121.

[3] BURKARD R E, RENDL F A. A thermodynamically motivated simulation procedure for combinatorial optimization problems. European Journal of Operational Research, 1984, 17(2): 169 – 174.

[4] ZENG X, ZHU Y, NAN L, et al. Solving weapon-target assignment problem using discrete particle swarm optimization. Proc of the Intelligent Control and Automation Conference, 2006: 3562 – 3565.

[5] XIN B, CHEN J, ZHANG J, et al. Efficient decision making for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. IEEE Trans. on Systems, Man and Cybernetics, Part C, 2010, 40(6): 649 – 662.

[6] LEE Z J, LEE C Y. A hybrid search algorithm with heuristics for resource allocation problem. Information Sciences, 2005, 173(3): 155 – 167.

[7] KARABOGA D, GORKEMLI B. A quick artificial bee colony algorithm and its performance on optimization problems. Applied Soft Computing, 2014, 23: 227 – 238.

[8] QU H, WANG L, LIU R. A constrastive study of the stochastic location-inventor problem with joint replenishment and independent replenishment. Expert Systems with Applications, 2015, 42(4): 2061 – 2072.

[9] WANG S, DA X Y, LI M D. Adaptive backtracking search optimization algorithm with pattern search for numerical optimization. Journal of Systems Engineering and Electronics, 2016, 27(2): 395 – 406.

[10] AHUJA R K, KUMAR A, JHA K, et al. Exact and heuristic algorithms for the weapon-target assignment problem. Operations Research, 2007, 55(6): 1136 – 1146.

[11] SUJIT P B, BEARD R. Multiple MAV task allocation using distributed auctions. Proc of the AIAA Guidance, Navigation, and Control Conference and Exhibit, 2007: AIAA-2007-6452.

[12] SUJIT P B, BEARD R. Distributed sequential auctions for multiple UAV task allocation. Proc. of the American Control Conference, 2007: 3955 – 3960.

[13] DANY D, CAMILLE A R. Multi-UAV decentralized task allocation with intermittent communications: the DTC algorithm. Proc. of the American Control Conference, 2009: 5406 – 5411.

[14] CAI H, LIU J, CHEN Y, et al. Survey of the research on dynamic weapon-target assignment problem. Journal of Systems Engineering and Electronics, 2006, 17(3): 559 – 565.

[15] LI X L, SHAO Z J, QIAN J X. An optimizing method based on autonomous animate: fish swarm algorithm. System Engineering Theory and Practice, 2002, 22(11): 32 – 38. (in Chinese)

[16] MA J W, ZHANG G L, XIE H. Optimization of feed-forward neural networks based on artificial fish-swarm algorithm. Journal of Computer Applications, 2004, 24(10): 21 – 23.

[17] HUANG G Q, LU Q Q. An approach to air leakage points identification in branches of ventilation system based on fish-swarm algorithm. Journal of System Simulation, 2007, 19(12): 2677 – 2682. (in Chinese)

[18] JIANG M Y, WANG Y. Spread spectrum code estimation by artificial Fish swarm algorithm. Proc. of the IEEE International Symposium on Intelligent Signal Processing, 2007: 1 – 6.

[19] GEEM Z W, KIM J H, LOGANATHAN G V. A new heuristic optimization algorithm: harmony search. Simulations, 2001, 76(2): 60 – 68.

[20] MANNE A S. A target-assignment problem. Operations Research, 1958, 6(3): 346 – 351.

[21] ECKLER A R, BURR S A. Mathematical models of target coverage and missile allocation. Alexandria, VA: Military Operations Research Society, 1972.

[22] KARASAKAL O, OZDEMIREL N E, KANDILLER L. Anti-ship missile defence of a naval task group. Naval Research Logistics, 2011, 58(3): 304 – 321.

[23] MURPHEY R A. Target-based weapon target assignment problems. PARDALOS P M, PITSOULIS L S, ed. Nonlinear assignment problems: algorithm and application. Boston, MA: Springer, 2000: 39 – 53.

[24] ROUX J N, VAN VUUREN J H. Threat evaluation and weapon assignment decision support: a review of the state of the art. ORiON, 2007, 23(2): 151 – 187.

[25] DAY R H. Allocating weapons to target complexes by means of non-linear programming. Operations Research, 1966, 26(14): 992 – 1013.

[26] BAYRAK A E, POLAT F. Employment of an evolutionary heuristic to solve the target allocation problem efficiently. Information Sciences, 2013, 22(2): 675 – 695.

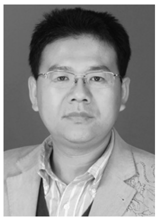
[27] AZAD M A K, ROCHA A M A C. Improved binary artificial fish swarm algorithm for the 0-1 multidimensional knapsack problems. Swarm and Evolutionary Computation, 2014, 14: 66 – 75.

[28] KHALILI M, KHARRAT R, SALAHSHOOR K, et al. Global dynamic harmony search algorithm: GDHS. Applied Mathematics and Computation, 2014, 228: 195 – 219.

[29] WANG L, YANG R X, XU Y, et al. An improved adaptive binary harmony search algorithm. Information Sciences, 2013, 232: 58 – 87.

[30] ZONG W G, SIM K B. Parameter-setting-free harmony search algorithm. Applied Mathematics and Computation, 2010, 217(8): 3881 – 3889.


## Biographies

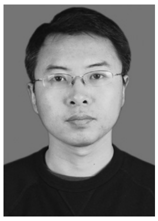


**LI Zhanwu** was born in 1978. He received his B.S. and M.S. degree in fire control engineering from Air Force Engineering University in 2000 and 2007, respectively. Now he is a Ph.D. candidate in Northwestern Polytechnical University. His research interests include aviation fire control theory and system engineering.
E-mail: afeulzw@189.cn

**CHANG Yizhe** was born in 1991. He received his B.S. and M.S. degrees in fire control engineering from Air Force Engineering University in 2013 and 2015, respectively. Now he is a Ph.D. candidate in Air Force Engineering University. His research interests include aviation fire control theory and system engineering.
E-mail: mastercyz@163.com

**KOU Yingxin** was born in 1965. He received his Ph.D degree in fire control engineering from Air Force Engineering University in 2006. Now he is a professor in Air Force Engineering University. His research interests include aviation fire control theory and system engineering.
E-mail: afcekyx@hotmail.com

**YANG Haiyan** was born in 1972. She received her Ph.D. degree in automation from Tsinghua University in 2009. Now she is a lecturer in Air Force Engineering University. Her research interests include situation assessment and Bayesian estimation.
E-mail: yanghy07@yeah.net



**XU An** was born in 1984. He received his Ph.D. degree from Air Force Engineering University in 2012. Now he is a lecturer in Air Force Engineering University. His research interests include aviation fire control theory and system engineering.
E-mail: aspire2010@sohu.com

**LI You** was born in 1989. He received his master degree from Air Force Engineering University in 2014. Now he is a Ph.D. candidate in Air Force Engineering University. His research interests include aviation fire control theory and system engineering.
E-mail: reason891020@qq.com