An Effective and Scalable Approach for Swarm-on-Swarm Air Combat Decision

Qian-yi Fu, Xiao Zhang, Bo Han, and Wen-jie Zhao(✉)

School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China
zhaowenjie8@zju.edu.cn

Abstract. We present an approach simultaneously solving both swarm target assignment and optimal motion control for large-scale swarms to achieve autonomous air combat decision making. The swarm target assignment is solved by using a modified k-means clustering algorithm with balance degree, and our motion control adopts a particle swarm optimization framework. Our approach scale well with a large number of drones and is able to find policies in continuous action spaces. Our experiments test our algorithm on clustering and decision-making, respectively. We find that it is much simple to implement, scalable and outperforms other algorithms we compare against.

Keywords: Swarm · Air combat decision · Autonomous system · Balance degree · Particle swarm optimization framework · K-means clustering

1 Introduction

In recent years, one of the things that has attracted enough attention in the domain of military is the ability to gain an advantage in the swarm-on-swarm air combat, and one primary challenge is to develop autonomous algorithms that can operate in high-dynamic, unpredictable swarm air combat environment.

Numerous algorithms about swarm-on-swarm air combat decision are discussed in the previous literature. These algorithms can be classified into two categories: target assignment and decision-making technology. Target assignment is centered on the idea of allocation auction [1,2], cost minimization [3,4] and optimal transition [5,6]. If these approach are used for large-scale swarms, the computational complexity of allocating the optimal target to each agent increases at least quadratically with the number of agents. Furthermore, some algorithms are relatively complicated, and are not compatible with a variety of environments. Decision-making technology such as matrix game [7], differential game [8,9], expert system [10] and deep reinforcement learning [11] are

Z. Ren et al. (Eds.): *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, LNEE 934, pp. 483–494, 2023.
https://doi.org/10.1007/978-981-19-3998-3_46

preferred on one-to-one air combat. However, the matrix game is suitable to low-dimensional action spaces; the differential game is hard to be applied in practice because of its complexity and enormous amount of calculation; the expert system is not versatile; and the deep reinforcement learning is difficult to train.

This paper seeks to improve the current state of affairs by introducing an approach that is effective and scalable for large-scale swarm air combat decision. We divide the whole problem into two parts: target assignment and air combat decision-making (motion control). We propose a novel objective with balance degree for k-means clustering, which guarantees that target assignment is relatively balanced. We also present a decision-making algorithm based on particle swarm optimization (PSO) which is easy to implement and have good performance.

Our experiments compare the performance of various different versions of k-means clustering, and find that the version with balance degree performs best. We also compare the decision-making algorithm based on PSO to several algorithms. It performs better than almost all the algorithms we compare against, no matter in terms of time or performance.

2 Preliminaries

In this section, we first present the problem statement. We then present the motion model of unmanned aerial vehicle (UAV) and the relative position relationship between two fighter aircrafts.

2.1 Problem Statement

Given the situation on the battlefield as follows: There are a population of n aircrafts trying to destroy m target aircrafts, our aircrafts are numbered as $u_i(i = 1, 2, \cdots, n)$, and enemy aircrafts numbered as $t_j(j = 1, 2, \cdots, m)$. Let x_{ij} be a decision variable. If the aircraft u_i is assigned to attack the target t_j, then $x_{ij} = 1$, otherwise $x_{ij} = 0$. We use p_{ij} to denote the probability that agent u_i destroys target t_j successfully, which is related to the relative position relationship between our aircraft and target aircraft. Any of the advantage and cost for our aircrafts to targets and the total cost can be evaluated. Let v_{ij}, c_{ij} denote the evaluated value and cost of our aircraft u_i attacking the target t_j, respectively.

The objectives of swarm-on-swarm air combat are as follows:

1) Determine the target assignment, so that the swarm of our drones destroys as many enemy targets as possible and attaining best total benefit. The mathematical model of the target assignment is that:

$$\begin{aligned} \underset{x_{ij}}{\text{maximize}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}x_{ij}(v_{ij}-c_{ij}) \\ \text{subject to} \quad & \sum_{j=1}^{m} x_{ij}=1, \quad i=1,2,\ldots,n, \\ & \sum_{i=1}^{n} x_{ij}=1, \quad j=1,2,\ldots,m, \\ & x_{ij}=0,1. \end{aligned} \tag{1}$$

2) After determining the target, the drone needs to make appropriate decisions and take effective actions autonomously in complex air combat fighting environment.

2.2 Aircraft Motion Model

Consider a two-dimensional continuous air combat environment, each aircraft is assumed to be a mass point. The motion model of aircraft can be expressed as:

$$\begin{cases} \dot{p_x} = v\cos\psi \\ \dot{p_y} = v\sin\psi \\ \dot{v} = a \\ \dot{\psi} = \omega \\ \omega = \frac{g}{v}n \end{cases} \tag{2}$$

where $p=[p_x,p_y]^T$ is the position of the aircraft. ψ is the yaw angle. v and ω are the linear velocity and the angular velocity, respectively. g is the acceleration of gravity. a is the forward acceleration. n is the side load. Among them, a and n are the control variables of the aircraft.

To ensure that the dynamic model is accurate, we added some constraints to the state variables and control variables via the following inequalities:

$$\begin{cases} v_{min} \le v \le v_{max} \\ |a| \le a_{max} \\ |n| \le n_{max} \end{cases} \tag{3}$$

where the subscript "max" and "min" represent the lower and upper bounds of corresponding variables.

2.3 Air Combat Situation

As we mentioned above, the relative position relationship between our aircraft and enemy aircraft has a vital influence on air-to-air combat. As shown in Fig. 1, u is our aircraft, t is a target aircraft. $\vec{\mathbf{d}}$ is relative distance vector between the two aircrafts, and α is called the line-of-sight (LOS) angle. $\vec{\mathbf{v_t}}$ and $\vec{\mathbf{v_u}}$ are the velocity vector of our aircraft and the target aircraft, respectively. λ_u and λ_t are the lead angles of u and t, which are defined by the angle between velocity vector and line of target sight. Based on Fig. 1, there are:

$$\begin{cases} \vec{\mathbf{d}} = [x_t - x_u, y_t - y_u]^T \\ \alpha = \arctan \frac{y_t - y_u}{x_t - x_u}, \alpha \in (-\pi, \pi] \\ \lambda_u = \arccos \frac{\vec{\mathbf{d}} \cdot \vec{\mathbf{v_t}}}{dv_u}, \lambda_u \in [0, \pi] \\ \lambda_t = \arccos \frac{\vec{\mathbf{d}} \cdot \vec{\mathbf{v_u}}}{dv_t}, \lambda_t \in [0, \pi] \end{cases} \tag{4}$$

3 Swarm Target Assignment

It is not possible to straightforwardly maximize the expected total profit of the assignment, because the computation cost for generating all the possible combinations of allocations one by one would be prohibitively expensive for a large-scale swarm of autonomous agents. Instead, here we propose an modified k-means clustering algorithm.

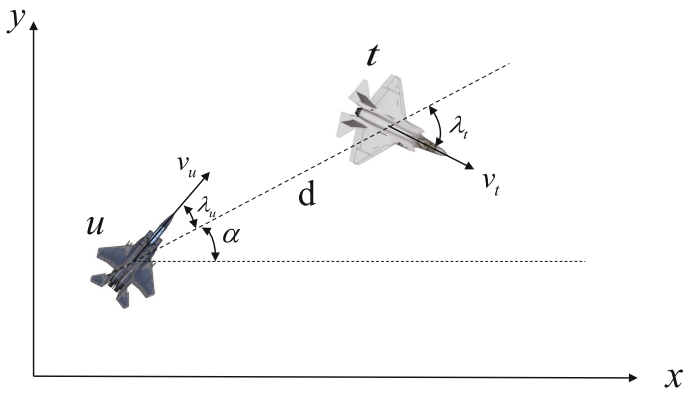


Fig. 1. Relative position diagram of the target aircraft and our aircraft.

Our strategy is to break the intractable task into smaller, more manageable pieces with our algorithm. In the modified k-means clustering algorithm, the swarms of our aircrafts as well as target aircrafts are both classified into k different clusters, depending on their respective states. Moreover, we use the balance degree to guarantee that every cluster has about the same number of members. After that, it is feasible to go ahead and solve this large-scale problem by using a two-step target assignment method, or, to be more specific, we can first implement inter-cluster allocation and then implement intra-cluster allocation. Considering that the latter part is essentially just a simple sorting problem, we will focus on how to achieve balanced clustering.

In the standard k-means algorithm, the objective is to divide the given data objects into k partitions in such a way that the sum of the squared deviations from the cluster focal points is minimal. Mathematically, this corresponds to the optimization of the criterion function

$$E^s = \sum_{i=1}^{k} \sum_{x_j \in C_j} ||x_j - \mu_i||^2 \tag{5}$$

where x_j is the data point and μ_i is the centroid of the cluster C_i. In addition, $||x_j - \mu_i||^2$ is the squared Euclidean distance between x_j and μ_i.

To address that the standard k-means does not consider the issue of "balance", we define a new criterion function, which draw an anology from thermodynamic system. Let p_i denotes the population ratio of the cluster C_i to the entire swarm, then the entropy of the clusters is defined as:

$$H(p) = -\sum_{i=1}^{k} p_i \log p_i \tag{6}$$

In classical thermodynamics, entropy is a measurable physical property that is most commonly associated with a state of disorder, randomness, or uncertainty. Similarly, the entropy of the clusters is a measure of equilibrium in the division of clusters. The lower the entropy, the more disequilibrium the swarm is divided. Since the number of clusters k has an influence on the entropy $H(p)$, we use balance degree $B(p,k)$ instead of entropy $H(p)$ as the final evaluation index:

$$B(p,k) = \frac{H(p)}{\log k} = \frac{-\sum_{i=1}^{k} p_i \log p_i}{\log k}, B(p,k) \in [0,1] \tag{7}$$

We therefore propose a new objective function of our modified k-means algorithm. Specifically,

$$\begin{aligned} E^m &= E^s - \beta B(p,k) \\ &= \sum_{i=1}^{k} \sum_{x_j \in C_j} ||x_j - \mu_i||^2 + \beta \frac{\sum_{i=1}^{k} p_i \log p_i}{\log k} \end{aligned} \tag{8}$$

where β is the weighting coefficient.

When dividing the swarm by the given data of UAVs' states, the different components of the states may have different physical units (for example, positions versus attitude angles) and the range may vary across environments. This can make it difficult to accommodate clusters of variable size. One approach to this problem is to manually scale the features so they are in similar range across environments and units, and use the weighted distance as the similarity measure. To rescale the features, we normalize each dimension across all data objects to have unit mean and variance

$$\begin{aligned} \overline{x} &\leftarrow \frac{1}{n} \sum_{j=1}^{n} x_j \\ \sigma_x^2 &\leftarrow \frac{1}{n} \sum_{j=1}^{n} (x_j - \overline{x})^2 \\ \hat{x}_j &\leftarrow \frac{x_j - \overline{x}}{\sigma_x} \end{aligned} \tag{9}$$

then, we use the weighted Euclidean distance for the preprocessed data $\hat{x}$

$$\mathrm{dist}(\hat{x_j}, \mu_i) = \sqrt{\omega_1 \cdot |\hat{x_{j1}} - \mu_{i1}|^2 + \ldots + \omega_d \cdot |\hat{x_{jd}} - \mu_{id}|^2} \tag{10}$$

where $\omega_1, \ldots, \omega_d$ are weight coefficients, which represent the importance of different features, typically satisfying that $\omega_i \geq 0 (i = 1, \ldots, d)$ and $\sum_{i=1}^{d} \omega_i = 1$. With the techniques of normalization and weighted distance measurement, we are able to make our algorithm more versatile and robust, without needing to pay attention to different types of units.

However, to minimize Eq. 5 is proved NP-hard [12]. In the standard k-means algorithm, the greedy strategy is adopted to approximate the Eq. 5 by iterative optimization. We continued with this strategy, moreover, we use the balance degree to decide whether to accept the clusters formation or choose to improve the clusters formation locally. Since a method of centroid initialization has an important role to play in the clustering result, we adopt technologies of re-initialization as well. The full algorithm, which we call the modified k-means clustering for swarm target assignment, is presented in Algorithm 1.

Algorithm 1. The modified k-means clustering for swarm target assignment

Input: the data points $D = \{x_1, \ldots, x_n\}$; the number of clusters k; the threshold ϵ
Output: A set of k clusters

Normalize all the data points x to $\hat{x}$
Choose the center μ_1 uniformly at random among the data points $\hat{x}$
for $i = 2, \ldots, k$ **do**
    for each data point $\hat{x}$ not chosen yet **do**
        Compute the distance between x and the nearest center: $d(\hat{x})$
        Set the probability of $\hat{x}$ to be chosen proportional to $d(\hat{x})^2$
    end for
    Choose one new data point $\hat{x}$ as the new center μ_i
end for
repeat
    Set $C_i = \emptyset$ $(1 \leq i \leq k)$
    for $j = 1, \ldots, n$ **do**
        Compute the distance between $\hat{x_j}$ and $\mu_i (1 \leq i \leq k)$: $d_{ji} = \mathrm{dist}(\hat{x_j}, \mu_i)$
        Assign each object $\hat{x_j}$ to the closest clusters C_{λ_j}: $\lambda_j = \underset{i \in \{1,\ldots,k\}}{\arg\min}\, d_{ji}$
    end for
    for $i = 1, \ldots, k$ **do**
        Updata center: $\mu_i = \frac{1}{|C_i|} \sum_{\hat{x} \in C_i} \hat{x}$
    end for
until the centers don't change
Compute the balance degree $B(p, k)$
while $B(p, k) < \epsilon$ **do**
    Choose a point $\hat{x}$ farthest from the center of the largest cluster
    Reassign the point to the second closest clusters
    Compute the balance degree $B(p, k)$
end while

4 Autonomous Air Combat Decision

Now consider the decision-making on one-to-one air combat. Most techniques for UAV air combat make use of differential game, which is a conservative strategy and has difficulties in practical application and solution. Furthermore, expert system and matrix game are also common on autonomous air combat decision, while they can only handle discrete action spaces, such as maneuver decision.

To overcome the above challenges, we present an particle swarm optimization based, intelligent air combat decision-making algorithm that can operate in continuous action spaces. A key feature of the approach is its simplicity: it requires only a straightforward particle swarm optimization architecture with very few "moving parts", making it easy to implement and scale to all kinds of air combat environments. Interestingly, our algorithm performs significantly better than any other swarm intelligence algorithm.

In the previous section, we mentioned the importance of the relative position relationship between the enemy and us. We now seeks to describe the air combat situation with the advantage functions. Based on the operational application of aerial combat, we designed three advantage functions, including angle advantage function, distance advantage function and energy advantage function.

Among them, angle is the most critical factor in air combat. By optimizing the relative angle relationship, UAV can gain a dominant position of stern attack, and avoid being attacked by opponent aircraft in the tail. When the opponent aircraft is within the angle of off-boresight launch, the UAV can launch an attack from the rear of enemy. Based on the above analysis, the advantage function of angle factor is built as follows:

$$\eta_a = \begin{cases} (1 - \frac{\lambda_u}{\pi})(1 - \frac{\lambda_t}{\pi}), \lambda_u \leq \lambda_{limit} \\ 0, otherwise \end{cases} \tag{11}$$

where ϕ_{limit} is the off-boresight launch angle of the UAV and $\eta_a \in [0, 1]$.

To ensure the distance advantage and improve attack probability, the distance factor is considered. If the target is within the attack area, the UAV can maintain an distance advantage. Therefore, the advantage function of distance factor is built as follows:

$$\eta_d = \begin{cases} e^{-\frac{2d}{d_{limit}}}, d \leq d_{limit} \\ 0, otherwise \end{cases} \tag{12}$$

where d_{limit} is the maximum range of UAV's weapon and $\eta_d \in [0, 1]$.

The goal of building the energy advantage function is to guide UAV entering the advantage area in the shortest time. When UAV is far from the optimal attack area, it should speed up its velocity to shorten the distance. Based on the above analysis, the energy advantage function is designed as follows:

$$\begin{aligned} &\eta_e = \frac{e^{r_e}}{e^{r_e} + e^{1-r_e}}, \\ &where \quad r_e = \frac{v_u^2}{v_u^2 + v_t^2} \end{aligned} \tag{13}$$

here r_e indicates the relative magnitude of energy, and $\eta_e \in [0, 1]$.

We now describe the procedure for autonomous air combat decision based on particle swarm optimization. In our particle swarm, each particle represents a point in the solution space, i.e., the UAV's forward acceleration a and side load n. And our goal is to find a pretty "good" solution as the UAV's control input so that the UAV can get the advantage over its opponent. To show how "good" a solution is, we design a fitness function f based on our advantage functions

$$f = \alpha_1 \cdot \eta_a + \alpha_2 \cdot \eta_d + \alpha_3 \cdot \eta_e \tag{14}$$

where α_1, α_2 and α_3 are weight coefficients. Considering that the motion model is second order, we use the predicted fitness value after the next three time steps to evaluate the current control input. With regard to a given measure of quality, that is, fitness function, each particles moves in the search-space according to simple mathematical formula over the particle's position and velocity [13]

$$v_{t+1} = cv_t + c_1 r_1 (p_{i,t} - x_t) + c_2 r_2 (p_{g,t} - x_t) \tag{15}$$

$$x_{t+1} = x_t + v_{t+1} \tag{16}$$

In the above, v_t and x_t are the velocity and position of particle i at time t, respectively. $p_{i,t}$ is the best experienced position of particle i, and $p_{g,t}$ represents the best experienced position of the swarm. c, c_1, c_2 are the respective cognition coefficients, and r_1, r_2 are two random functions in the range $[0, 1]$. Equations 15 and 16 imply that each particle's movement is influenced by its local best known position, but is also guided toward the best known position. By iteratively trying to improve the candidate solution, it is expected to move the swarm toward the best solutions, among which the best solution experienced by the particle swarm would be the UAV's control input. The pseudocode of the intelligent air combat decision-making algorithm is given in Algorithm 2.

5 Experiments

5.1 Comparison to Other Versions in Clustering Algorithms

First, we compare our algorithm with the standard version (baseline) and its two natural variations, including k-means algorithm with re-initialization, k-means algorithm with normalization and weighted distance measurement.

We run each algorithm 10 times. In the experiments, the input was 1000 sets of data, each containing five dimensions including p_x, p_y, ψ, v and ω. We consider the following two metrics: (1) the sum of squared errors E^s, and (2) the balance degree $B(p, k)$. Since the two types of results varies greatly, we shifted and scaled the scores for each scoring metric so that the worst result was set to 0 and the best result was set to 1, and averaged over these scores to produce a final score for each algorithm setting.

As is shown in Table 1, our algorithm outperforms the previous versions on all the evaluation indicators and achieved the highest score in the final grade.

Algorithm 2. Air combat decision-making algorithm based on PSO

(One iteration during k^{th} time instant for the aircraft u_i with its target t_j)
Set $t = 0$
for each particle i in the swarm **do**
    Randomly initialize $x_{i,t}$ according to Eq. 3
    Set $v_{i,t} = 0$
    Calculate fitness value f_i of particle i according to Eq. 11–14
end for
Initialize $p_{i,t} = f_i$, $p_{g,t} = \max_i f_i$
for $t = 1, \ldots, T$ **do**
    for each particle i in the swarm **do**
        Calculate $v_{i,t}$ according to Eq. 15
        Calculate $x_{i,t}$ according to Eq. 16
        Calculate fitness value f_i of particle i according to Eq. 11–14
        if $p_{i,t-1} > f_i$ **then**
            Set $p_{i,t} = p_{i,t-1}$
        else
            Set $p_{i,t} = f_i$
        end if
        Set $p_{g,t} = \max_i p_{i,t}$
    end for
end for
Find the best solution: $best = \arg\max_i p_{i,T}$
return $x_{best,T}$

5.2 Comparison to Other Algorithms in One-to-One Air Combat Domain

Next, we compare our algorithm to other algorithms based on different methods: matrix game (MG), genetic algorithm (GA), simulated annealing (SA) and differential evolution (DE). For matrix game method, we discretized the continuous solution space in order to apply it to our problem. For all five algorithms, we used the same algorithm framework.

Each algorithm was run 10 times. See Table 2 for the average time per step of each algorithm. And the specific results are shown in Fig. 2. These results shows that our algorithm can be implemented in real time to achieve autonomous air combat decision-making. No matter in terms of effectiveness or efficiency, our algorithm is superior to other algorithms.

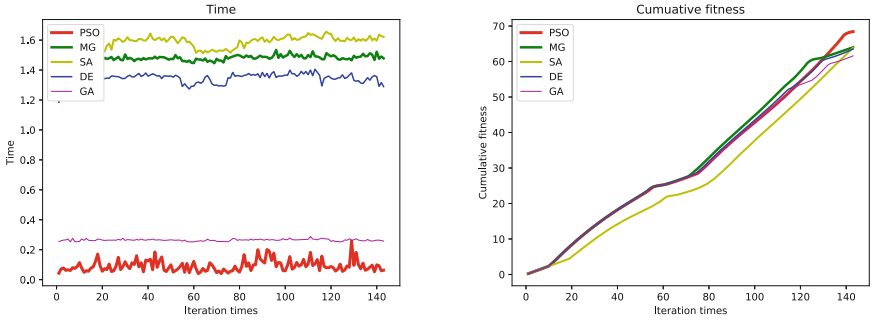
We also run these autonomous decision-making algorithms in an adversarial environment, where two drones using different algorithms tried to attack each other. And the results are shown in Fig. 3. We can conclude that our algorithm outperforms most algorithms in adversarial environments, only inferior to SA in terms of fitness.

Table 1. Average scores of 10 runs. We report the sum of squared errors (SSE) scores, the balance degree (BD) scores and the final scores for the 4 algorithms. Note that all score is normalized so that the worst result (across 10 runs) receives 0 and the best 1.

	The SSE score	The BD score	The final score
Baseline	0.0852	0.6289	0.3571
Reinitialize	0.1117	0.7838	0.4477
Rescale + weighted	0.7224	0.6437	0.6830
Our algorithm	0.7778	0.9201	0.8489

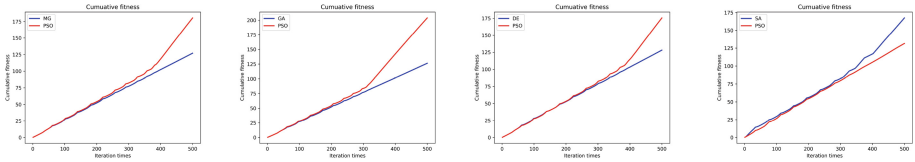
Table 2. Average time per step (over 10 runs of the same algorithm framework, on the same environment) for each algorithm.

Algorithm	PSO	MG	SA	DE	GA
Average time	0.0910	1.4939	1.6273	1.3640	0.2706



(a) Time (b) Cumulative fitness

Fig. 2. Comparison of several algorithms on air combat environment. The left plot shows the running time of each algorithm. The right plot shows the fitness of each algorithm in air combat.



(a) MG vs PSO (b) GA vs PSO (c) DE vs PSO (d) SA vs PSO

Fig. 3. Comparison with other algorithms on adversarial environment. The first three plots show that our algorithm is better than GM, GA and DE, while the last plot shows that our algorithm is sightly inferior to SA in terms of fitness.

6 Conclusion

This paper introduced an effective and scalable approach for swarm-on-swarm air combat decision, this approach combines target assignment based on the modified k-means algorithm and one-to-one air combat decision-making based on particle swarm optimation. Our approach is simple to implement, applicable in high-dynamic, unpredictable swarm air combat environment and have better overall performance.

Acknowledgements. The authors would like to thank Lifang Zeng for insightful discussions on topics related to this work. This work is supported by Defence Industrial Technology Development Programme (No. JCKY2019205A006).

References

1. Bertsekas, D.P., Castanon, D.A.: Parallel synchronous and asynchronous implementations of the auction algorithm. Parallel Comput. **17**(6–7), 707–732 (1991)
2. Morgan, D., Subramanian, G.P., Chung, S.J., Hadaegh, F.Y.: Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. Int. J. Robot. Res. **35**(10), 1261–1285 (2016)
3. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q. **2**(1–2), 83–97 (1955)
4. Qiu, X., et al.: On dynamic target assignment method of UAV swarms based on cost minimization. In: 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), pp. 830–835. IEEE (2021)
5. Berman, S., Halász, A., Hsieh, M.A., Kumar, V.: Optimized stochastic policies for task allocation in swarms of robots. IEEE Trans. Rob. **25**(4), 927–937 (2009)
6. Bandyopadhyay, S., Chung, S.J., Hadaegh, F.Y.: Probabilistic swarm guidance using optimal transport. In: 2014 IEEE Conference on Control Applications (CCA), pp. 498–505. IEEE (2014)
7. Austin, F., Carbone, G., Falco, M., Hinz, H., Lewis, M.: Game theory for automated maneuvering during air-to-air combat. J. Guid. Control. Dyn. **13**(6), 1143–1149 (1990)
8. Isaacs, R.: Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization. Courier Corporation (1999)
9. Fisac, J.F., Sastry, S.S.: The pursuit-evasion-defense differential game in dynamic constrained environments. In: 2015 54th IEEE Conference on Decision and Control (CDC), pp. 4549–4556. IEEE (2015)
10. Platts, J., Howell, S., Peeling, E., Thie, C., Lock, Z., Smith, P.: Increasing UAV intelligence through learning. In: AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit, p. 6413 (2004)
11. Chen, Y., Zhang, J., Yang, Q., Zhou, Y., Shi, G., Wu, Y.: Design and verification of UAV maneuver decision simulation system based on deep q-learning network. In: 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 817–823. IEEE (2020)
12. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: NP-hardness of Euclidean sum-of-squares clustering. Mach. Learn. **75**(2), 245–248 (2009)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)