

Improved discrete mapping differential evolution for multi-unmanned aerial vehicles cooperative multi-targets assignment under unified model

Zhao Ming¹ · Zhao Lingling¹ · Su Xiaohong¹ · Ma Peijun¹ · Zhang Yanhang¹

Received: 4 July 2014 / Accepted: 15 April 2015 / Published online: 29 April 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract The cooperative multi-targets assignment for multiple unmanned aerial vehicles (UAV) is a complex combinatorial optimization problem. Multi-UAVs cooperation increases the scale of problems which cause a noticeable increase in task planning time. Moreover, it is difficult to build a unified assignment model because different tasks often require different numbers of UAVs and targets. Besides, the cooperative constraints of multi-UAVs in a three-dimensional environments are more complex than that in a two-dimensional environments, which makes it difficult to obtain an optimal solution. To solve these problems, we present a unified gene coding strategy to handle various models in a consistent framework. Then, a cooperative target assignment algorithm in a three-dimensional environments based on discrete mapping differential evolution is given. First, we use flight path cost to indicate the assignment relationship between the UAV and the target, which turns the optimization problem from discrete space to continuous space, and so the solving process can be simplified. Secondly, in order to obtain reasonable offspring for differential evolution, we map the solution back to the assignment relationship space according to inverse mapping rules. Finally, to avoid falling into a local optimal, a balance between exploration and exploitation is achieved by combining the dynamic crossover rate with the hybrid evolution strategy. The simulation results show that the proposed discrete mapping differential evolution algorithm with the unified gene coding strategy not only effectively

solves the cooperative multi-targets assignment problem, but also improves the accuracy of the multi-targets assignment. It is also suitable for solving the large scale problem of assignment.

Keywords Multi-UAVs · Cooperative target assignment · Combinatorial optimization · Mapping rules · Discrete differential evolution

1 Introduction

Compared with single unmanned aerial vehicles (UAV), multi-unmanned aerial vehicles (Multi-UAVs) system has the advantages of high speed, reliability and flexibility by cooperation and coordination [1]. Many combat tasks such as Wide Area Search, Suppression of Enemy Air Defense (SEAD), and Intelligence Surveillance and Reconnaissance (ISR) need to be executed by a Multi-UAVs cooperative in complex battlefield situations [2, 3]. Thus, the Multi-UAVs system has gradually replaced the single UAV in performing a variety of complex tasks. However, the cooperation of Multi-UAVs increases the problem scale, and thus the computational complexity grows exponentially. Moreover, the cooperation of Multi-UAVs is affected by other issues, such as environmental constraints, equipment degradation, resource competition, cooperative constraints and the conflict of UAVs [4–9] etc. Therefore, it is a challenge to search for the optimal solution for Multi-UAVs cooperation at a minimum cost.

The problem of Multi-UAVs cooperative multi-targets assignment (MUCMTA) plays an important role in the task planning system. How to minimize the total cost or maximize the total benefits is the key to properly assigning a series of UAVs to implement multiple targets [6]. In

✉ Su Xiaohong
sxh@hit.edu.cn

Zhao Ming
sequoia00@163.com

¹ Harbin Institute of Technology, Harbin 150001, China

particular, the target assignment problem is a well-known discrete combinatorial optimization problem with multi-constraints and multi-models. It is a non-deterministic polynomial (NP) and is hard to find an optimal solution [7], especially in a three-dimensional (3D) environment.

The MUCMTA has been tackled with some classical optimization methods, such as mathematical programming and distribution approaches. Mathematical programming approaches have included the Hungarian algorithm [8–10], mixed integer linear programming [11], and Dijkstra algorithm [12]. The assignments based on those techniques work well using simple models in a two-dimensional (2D) scenario, but they are not suitable for the complicated MUCMTA in 3D environments. The distributed approaches have a negotiation scheme based on the contract net [13] and market mechanism [14–19]. These distributed methods balance the conflict by consultation and information sharing. So they can solve the multi-target assignment problem even in a larger scale. However, those methods emphasize avoidance of conflict without taking into account the practical cost between the UAVs and the targets, so it cannot provide an accurate assignment result.

Inspired by the good performance of artificial intelligence, intelligence algorithms have been widely used in the field of MUCMTA. For example, Genetic Algorithm (GA) is extended to seek the optimal solution of an assignment [20–23], because it mimics the process of natural selection and is not restricted by continuity, differentiability, or unimodality in a search domain. But GA usually converges rapidly and falls into local optimization easily. The particle swarm optimization (PSO) algorithm is another commonly used intelligence method. It was improved in [24–28] for solving the assignment problem. This approach finds the best solution by particles swirling around, but it has the potential failure in achieving the global optimization. Moreover, the ant colony optimization (ACO) and the bee colony methods have also been introduced [29–33], which have better performance in MUCMTA. However, it degenerates very fast when its situations become more complicated, especially in 3D. So it is hardly suitable for the assignment problem with a large scale.

The MUCMTA research in a 3D environment has more practical applications than 2D [34]. However, it is difficult to establish an effective representation by using a graph structure in 3D environment because it has a more extensive searching space. On the other hand, the UAVs should fly through the complex uneven terrain and avoid various threatening zones and no fly zones, furthermore the UAVs should satisfy their own and cooperative constraints. The less the 3D environment is considered, the greater damage probability the UAV will be suffered, which will lead to task failure. So the reality task cost should be estimated by the flight distance, execution time and other various

constraints, but not the linear distance between the start location and target position nor the shortest path on a graph. Research has also been proposed for the task assignment in 3D environments [34, 35]. An approach of a self-organizing map (SOM) neural network has been used for task assignment and path planning of multiple autonomous underwater vehicles (AUV) [34] in a 3D workspace. The approach combines task assignment with path planning, and the AUV is similar to a UAV in a 3D environment.

Differential evolution (DE) algorithm is a stochastic, parallel and population based search method, using the distance and direction difference of individuals to guide the search process [36–39]. Characterized by continuous parameters it has a strong search capability and fast convergence. The improved DE has been proved to be quite appropriate for the combinatorial optimization problems with discrete parameters, and it is effective and competitive with other approaches in this domain [39]. In our previous work [40], we have also used a DE algorithm to deal with a uniform assignment model using partially mapped crossover (PMX). This approach has provided better simulation results. But because it performs differential operations with only sequenced targets as well as rounding the differential results directly into integer, physical properties are ignored and the rationality of the problem is reduced.

In this paper, we continue improving the DE algorithm for solving the MUCMTA more elaborately. Although the DE algorithm has a better effect on the combinatorial optimization problems, there are still some major challenges to solve for practical problems. They are summarized as follows:

1. How do we design a unified gene code that indicates the task assignment? A population with unified coding can not only represent various assignment models, but can also be processed by a uniform algorithm framework in 3D environment.
2. How do we define reasonable mapping rules for DE with discrete parameters relating to MUCMTA? They should properly transform the problem from discrete space to continuous space by physical properties, so that it can correctly guide the direction of search.
3. How to dynamically select a differential strategy during the evolutionary process? Because the DE algorithm improves the efficiency by creating a balance between the exploration and the exploitation, another challenge lies in the selection of the differential strategy.

Therefore, this paper presents a novel discrete mapping differential evolution (DMDE) approach based on a unified gene code strategy in 3D, which can handle various assignment models in a uniform algorithm framework. Then

we improved the differential operation by discrete mapping and inverse mapping rules. The map of the flight path cost transforms the problem from discrete space to continuous space, so that the rational optimal result with physical properties can easily be found. Moreover, the inverse mapping with matched strategies inverses the differential result back to the assignment relationship for generating feasible offspring of DE. Furthermore, a hybrid differential strategy with a dynamic crossover rate is used to improve the efficiency of the DMDE algorithm. Finally, we replace the constraint of arriving at the same time with the asynchronous initial waiting time, which further enhances the cooperative capacity of UAVs.

The rest of this paper is organized as follows. Section 2 describes the unified algorithm framework for MUCMTA. The unified encoding strategy based on the cost of the flight path is given in Sect. 3. In Sect. 4 we present the differential mapping rules for discrete parameters. Simulation results are shown in Sect. 5 before the conclusion is given in Sect. 6.

2 The unified algorithm framework

2.1 Unified assignment model

We have previously worked on the unified assignment model for the MUCMTA problem in a 3D environment, and estimated the flight path cost by using the vertical section between a UAV and a target [40]. In the meanwhile the assignment situations have been summarized as these three basic cases: $N > M$, $N < M$ and $N = M$, which are classified by the number of UAV and target.

Based on the previous research, the scenario of MUCMTA is described as a set of UAVs $N = \{U_1, U_2, \dots, U_n\}$ to attack a set of targets $M = \{T_1, T_2, \dots, T_m\}$ in a complex 3D environment. The objective of this problem is to find an assignment sequence, which has the optimal total cost of the flight path and the shortest flying time for all UAVs. Meanwhile, all UAVs fit a variety of constraints and avoid radar threat by cooperating with each other. The typical scenario and simulation environment are shown in Fig. 1.

The objective function of the unified assignment model can be formulated as follows:

$$\begin{aligned} \min(f(x)) = & \sum_{i=1}^n \sum_{j=1}^m w_j d_{(i,j)} x_{(i,j)} \\ & + \alpha \left(\max \left(\sum_{i=1}^n \sum_{j=1}^m t_{(i,j)} x_{(i,j)} \right) \right) + \beta \sum_{k=1}^K c_k \end{aligned} \tag{1}$$

Where $d_{(i,j)}$ is the distance of flight path from UAV to the target. $t_{(i,j)}$ is the flying time of the UAV. c_k is the penalty corresponding to the constraints. w_j is the weight of target

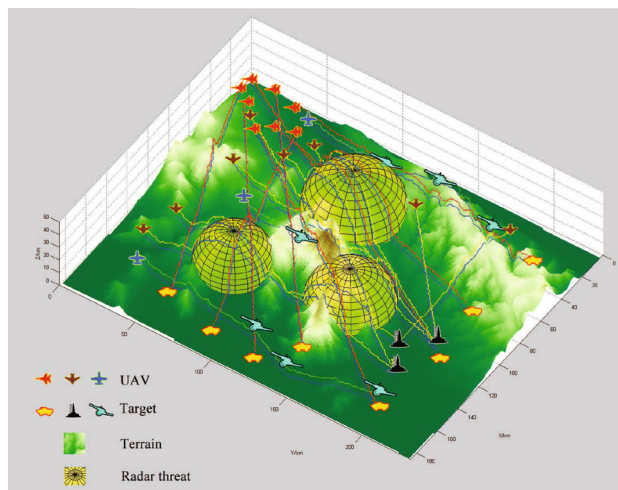


Fig. 1 The scenario of MUCMTA in 3D environment

j , and $0 < w_j \leq 1$, α and β are scale factors for keeping the balance of polynomial in the objective function, which can be empirically set. $x_{(i,j)}$ is the decision variable, which decides the corresponding of UAV and target. The forms of $x_{(i,j)}$ with different assignment cases are as follows.

$$x_{(i,j)} = \begin{cases} x_{(i,j)} & N = M \\ x_{(i',j)}, i' \in [i_s, \dots, i_l], s, l \in N & N > M \\ x_{(i,j')}, j' \in [j_p, \dots, j_q], p, q \in M & N < M \end{cases} \tag{2}$$

In (2), while $N > M$, $x_{(i',j)}$ is the decision between many UAVs and one target. That means many UAVs can attack the same target. While $N < M$, $x_{(i,j')}$ is the decision between one UAV and many targets, and that means a single UAV can attack a group of targets.

According to the discussion in the previous research [40], a variety of constraints can be described as follows.

$$\sum_{i=1}^n X_{ij} = 1, \forall j = 1, \dots, m, \quad n \geq m \tag{3}$$

$$\sum_{j=1}^m X_{ij} = 1, \forall i = 1, \dots, n, \quad n \leq m \tag{4}$$

$$\begin{cases} \tau = T(j)_{\min} - T(i)_{\max} \\ t_j \geq t_i + \tau \quad i, j \in 1, \dots, m \end{cases} \tag{5}$$

$$\begin{cases} [T_{k-1}, T_k] \neq \Phi \\ \max \{t_i, \dots, t_j\} \leq T_k \quad i, j \in 1, \dots, m \\ \min \{t_i, \dots, t_j\} \geq T_{k-1} \quad i, j \in 1, \dots, m \end{cases} \tag{6}$$

$$\sum_{i=1}^n \sum_{j=1}^m d_{(i,j)} x_{(i,j)} \leq \sum_{k=1}^n D_k \quad \forall k = 1, \dots, n \tag{7}$$

$$\max \left\{ \sum_{i=1}^n \sum_{j=1}^m t_{(i,j)} x_{(i,j)} \right\} \leq T_k \quad \forall k = 1, \dots, n \tag{8}$$

$$\bigcap_{i=1}^K ([t(i)_{min}, t(i)_{max}]) \neq \Phi \quad K \in 1, \dots, n \quad (9)$$

Where (3) and (4) indicate the relationship between UAV and target; (5) represents the perform sequence of targets and τ is the minimum time interval. Then, (6) is the constraint of time window, (7) is the constraint of maximum flight path, (8) is the constraint of maximum flying time, and (9) is the constraint of arriving at the same time.

Moreover, in order to ensure the UAVs simultaneous arrival, we introduce the vector of maximum initial waiting time T_{wait} , which allows some UAVs to wait until other UAVs take off in a certain time period. It can be described as:

$$T_{wait} = [t_{wait}(1), t_{wait}(2), \dots, t_{wait}(k)], \quad k \in \{1, \dots, n\} \quad (10)$$

Then the penalty of the simultaneous arrival constraint can be changed as:

$$C_{constrain} = \begin{cases} 0 \bigcap_{i=1}^L ([T_{min}(i), (T_{max}(i) + T_{wait}(i))]) \neq \Phi, & L \in N \\ K_p \bigcap_{i=1}^L ([T_{min}(i), (T_{max}(i) + T_{wait}(i))]) = \Phi, & L \in N \end{cases} \quad (11)$$

Where the $T_{min}(i)$ is the time of the UAV flying at the highest speed, and the $T_{max}(i)$ is the time of the UAV flying at the lowest speed. The K_p is a penalty term, when the cooperative tasks have no intersection.

The objective function presents the optimal direction of search, so it can measure the evolution of individual merits. Therefore, we use the objective function to be the fitness function of DE. The individual with high fitness has more chances to survive and reproduce in the evolutionary process, and the fitness function can make the algorithm search toward the direction of the optimal solution.

2.2 The DE algorithm framework

The DE algorithm developed by Storn and Price in 1995 is similar to other evolution algorithms in the characters of population transfer, crossover and mutation [41–44]. However it significantly differs from other evolution methods in using the vector of distance and direction difference between individuals to guide the search process, which performs quickly and efficiently searching in continuous space. The main performance advantage of the DE algorithm is reflected in the differential strategies [36, 37, 42]. The general strategies used by DE are indicated by the notation of DE/ x (base)/ y (number)/ z (cross), where x represents the base of the differential tendentious, y is the

Table 1 Common formulas of DE strategies

Strategy	Formula
DE/best/1/bin	$v = x_{best}^{(G)} + F \cdot (x_{r2}^{(G)} - x_{r3}^{(G)})$
DE/rand/1/bin	$v = x_{r1}^{(G)} + F \cdot (x_{r2}^{(G)} - x_{r3}^{(G)})$
DE/ rand-to-best/1/bin	$v = x_i^{(G)} + \lambda \cdot (x_{best}^{(G)} - x_i^{(G)}) + F \cdot (x_{r1}^{(G)} - x_{r2}^{(G)})$
DE/best/2/bin	$v = x_{best}^{(G)} + F \cdot (x_{r1}^{(G)} + x_{r2}^{(G)} - x_{r3}^{(G)} - x_{r4}^{(G)})$
DE/rand/2/bin	$v = x_{r5}^{(G)} + F \cdot (x_{r1}^{(G)} + x_{r2}^{(G)} - x_{r3}^{(G)} - x_{r4}^{(G)})$

number of individuals involved in the differential, and z is the type of crossover. The crossover z should usually be chosen between binomial (bin) and exponential (exp), where the binomial crossover is randomly selected from the set of possible crossover points, and the exponential crossover is selected from a sequence of adjacent crossover points in one loop. The research of [37] has stated that the exponential and binomial crossovers yield similar results for discrete optimization problems, so we selected the commonly used type of binomial crossover in our research. The DE strategies most frequently used in the literatures are shown in Table 1.

However, the traditional DE only handles the problem with continuous parameters. If using DE to solve the combinatorial optimization problem with discrete parameters, the discrete parameters should be changed to continuous space by some translation approaches. Through the translation, DE can easily solve the discrete problem with continuous parameters. Therefore, in order to avoid the defects of DE for MUCMTA, we developed a reasonable translation method with map and inverse mapping strategies. The proposed strategy not only takes advantage of the DE algorithm, but also reduces the differential error by involving the physical properties of MUCMTA.

In this paper, we first propose a unified gene encoding for assignment models of $N = M$, $N > M$ and $N < M$, so that different models can be processed in a consistent framework of the DE algorithm. Then we convert the discrete parameters into continuous space by mapping the matrix of the flight path cost, while we obtain the differential result with the matrix. Next we inverse the differential result to that of the relationship of assignment by the inverse mapping rules. Those translation processes can easily produce reasonable offspring of a DE individual with practical properties. Finally, we make the evolution always move toward the optimized direction of MUCMTA by using a cooperative fitness function with various constraints. In addition, we also use a hybrid differential strategy to enhance the efficiency of DMDE. The algorithm framework is shown in Fig. 2.

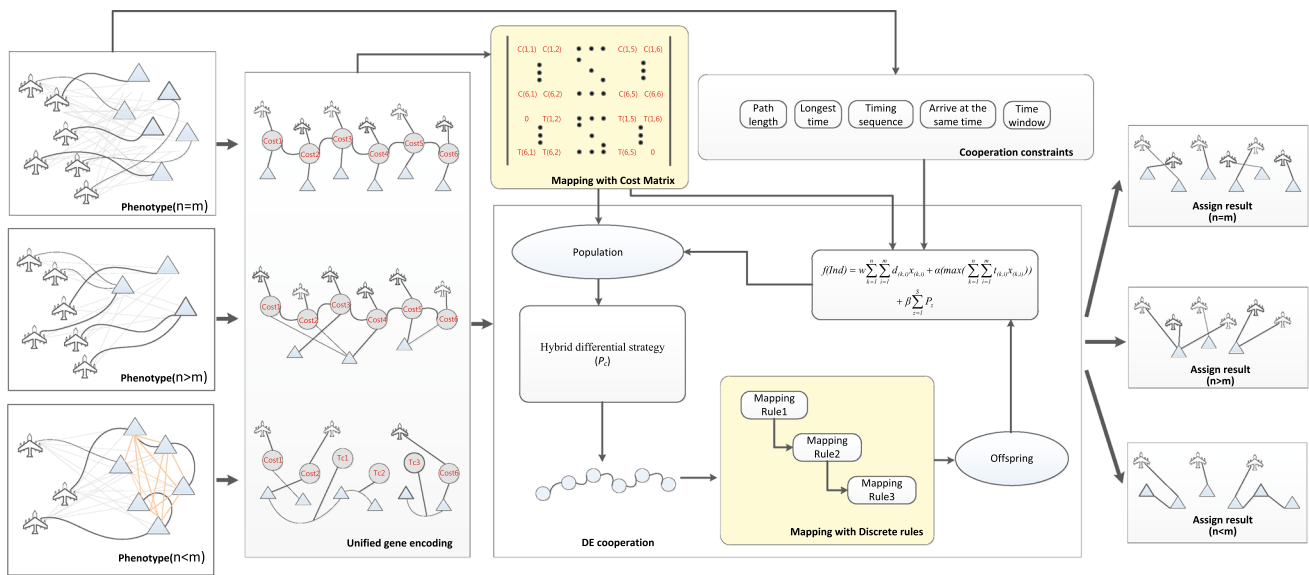


Fig. 2 Framework of unified targets assignment algorithm based on the improved discrete DE

3 The Unified encoding strategies

3.1 Encoding strategies

The efficiency and complexity of a search algorithm relies heavily on the representation scheme [36]. Therefore, the key to an evolution algorithm is to find an efficient representation scheme. For MUCMTA, it usually uses the permutation of targets to represent the individual of evolution. The gene of an individual is a serial number of targets, while the length of individual gene is the total of combat tasks. Moreover, If the number of UAVs is more than that of the targets, a target can be randomly assigned to multiple UAVs. If the number of UAVs is less than that of the targets, many targets can consist of a sequence for UAV cruising. This approach easily represents the MUCMTA by the individual of evolution algorithm, but it is difficult to deal with different models in a uniform framework. Furthermore, the algorithm randomly assigns the UAVs to targets and blindly searches the optimal solution during the evolution process, resulting in the reduction of the algorithm efficiency.

Therefore, we propose a uniform gene encoding for various assignment models ($N = M$, $N > M$ and $N < M$) based on the flight cost of UAVs and targets. In [40], we have introduced that the cost matrix of various models has a similar form, so we can unify the gene encoding by the assignment relationship of UAVs, targets and their flight path cost. Then the uniform gene encoding of DE can be represented by a triple vector $(U_i, T_j, C(i, j))$ or $(T_j, T_{(j+1)}, T_c(j, (j + 1)))$, where U_i denotes the UAV i , T_j denotes the target j , $C(i, j)$ the cost of the UAV i flying to

the target j , and $T_c(j, (j + 1))$ the cost of the target j flying to the next target $(j + 1)$ (target j is treated as a new starting point for a UAV when the UAV cruises a group of targets). Thus, the genes of an individual composes a feasible solution for the MUCMTA. Figure 3 describes an example of the cost matrix and the individual encoding.

In Fig. 3, $\{U_1, \dots, U_6\}$ indicates the UAVs, $\{T_1, \dots, T_6\}$ is the targets, $C(i, j)$ is the flight path cost of the UAV i to the target j , and $T_c(i, j)$ is the flight path cost of the target i to the next target j . In the left sub-region of the graph in Fig. 3, the area enclosed by the black solid lines shows the cost matrix in which the number of UAVs equals to the targets ($N = M$). The area enclosed by the blue dotted lines represents the cost matrix in which the number of UAVs is larger than the targets ($N > M$). And, the area enclosed by the red span lines indicates the cost matrix in which the number of UAVs is less than the targets ($N < M$). In the right sub-region of the graph in Fig. 3, there are the genes encoding of the individuals for the three models.

3.2 The rules of code generation

The unified encoding of various models has a similar form but different meaning, so the rules of code generation are slightly different. They are described as follows:

1. While $N = M$, the relationship of the UAV and the target is one to one. So each gene denotes a set of a UAV U_i , a target T_j and their cost $C(i, j)$. $C(i, j)$ is the flight path cost of U_i to T_j . Because each target is performed by a UAV and a UAV only executes one target, the code of the evolutionary individuals has no

Fig. 3 Uniform code based on cost matrix

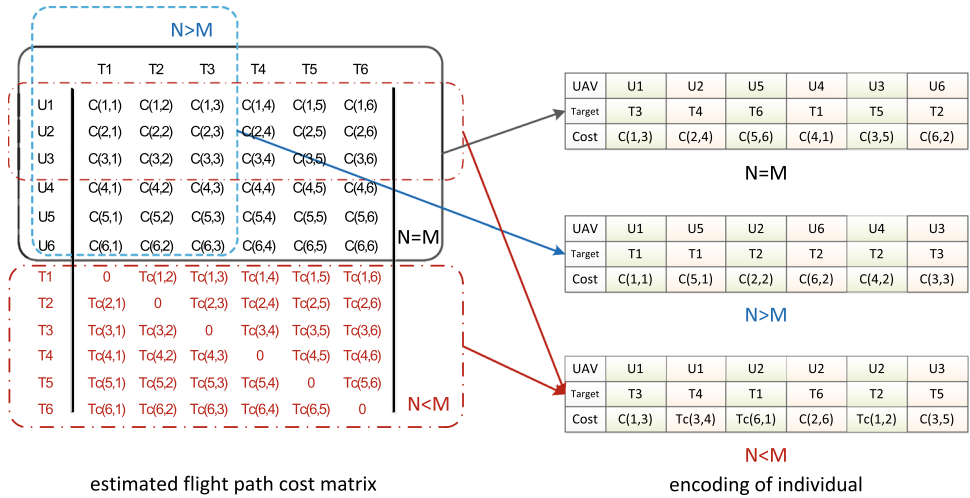
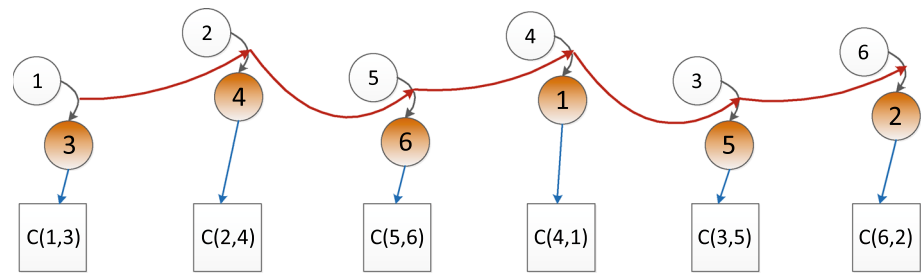


Fig. 4 The gene code sequence of individuals when $N = M$



repetitive U_i and T_j . Figure 4 shows the gene code of the individuals.

- While $N > M$, the relationship of UAV and target is many to one. Each gene has the same meaning as N equals to M . However, because many UAVs can attack the same target, a repetitive U_i does not exist, but T_j could be repetitive in a code of evolutionary individuals. Moreover, each target is at least performed by one UAV, so T_j occurs at least once in an individual. It is shown in Fig. 5.
- While $N < M$, the relationship of UAV and target is one to many. Each gene may have relations with other genes because a UAV can cruises to a group of targets. Those targets assigned to the same UAV, can be grouped into a swarm team. In the swarm team, the UAV implements all those targets sequentially. Therefore, the UAV U_i corresponds to a group of targets $[T_j, \dots, T_k]$. $C(i,j)$ is the flight path cost of U_i to T_j . Then we set the current finished target T_j as a new starting point for the UAV. Then, $T_c(j,k)$ is the flight path cost from the new starting point T_j to next target T_k . The sequence of $[T_j, \dots, T_k]$ is based on the principle of the shortest path, so that the UAV can cruises to all targets in the team with a minimum cost. In the code of evolutionary individuals, a repetitive T_j does not exist but U_i could be repetitive. The U_i occurs

in an individual at least once, because every UAV should be involved. The gene code of the individual is shown in Fig. 6.

The above code rules represent the evolutionary individual of three basic MUCMTA missions. It can unify various assignment models to be dealt with in a consistent algorithm framework. Meanwhile, this approach stores the flight path cost in a gene, so that the distance information between the UAV and a target can be used to guide the direction of search. So it is a more general and more reasonable algorithm.

4 Improved DE by discrete mapping rules

4.1 Discrete mapping rules

The feasible solution of MUCMTA is constituted by the discrete integer set of UAVs, targets and their cost. So the classical DE algorithm can't solve this problem directly. If only the integer part of the float differential result is utilized, the new individual would have invalid duplicate values, zero values, negative values and over bounds values. Such an individual cannot be used in subsequent evolutionary computation, which will inevitably lead to the failure of the searching algorithm.

Fig. 5 The gene code sequence of individuals when $N > M$

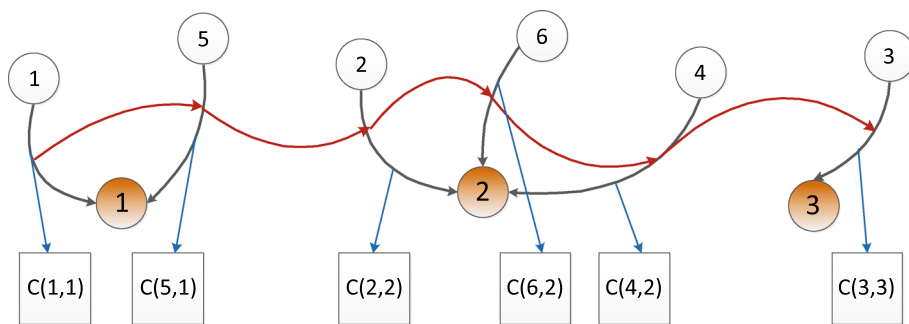
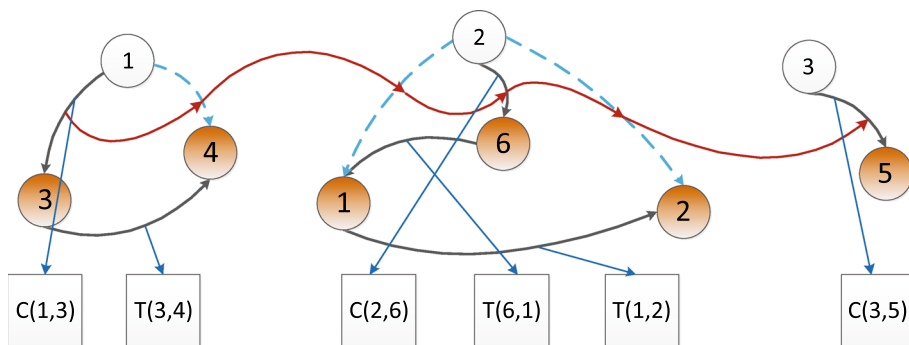


Fig. 6 The gene code sequence of individuals when $N < M$



Some approaches such as the partially matched crossover method [23, 40] or the sorting differential results method [37] have solved the invalidation of DE. But, they are variants of random search, and neglect the actual physical features of MUCMTA. Therefore, it is difficult to achieve optimal results.

To solve this problem, the improved algorithm uses the cost value in the gene code to map the discrete relationship of the UAV and the target, so that the search process makes the differential in continuous space. Then we inverse the search results from the continuous space back to the original discrete space by employing the inverse mapping strategies, while a new feasible individual is obtained. The space conversion with map and inverse mapping strategies can generate new feasible individuals and avoid the failures of the other approaches.

In the physical significance of the MUCMTA problem, the relationship of the UAV and target connects with the flight path cost. The main goal of the search is to find the assignment with a set of minimum cost. Therefore, we calculate the differential of individuals in DMDE with the corresponding cost of the UAV and the target. Because the individual can be encoded as an n_d -dimensional bivariate vector, we set the discrete space as $\{(U_i, T_j) \in Z^{n_d}\}$ and the continuous cost space as R_c^n . So the mapping function for conversion can be interpreted by the following expression:

$$\phi : \{(U_i, T_j) \in Z^{n_d}\} \xrightarrow{C_{cost}} R_c^n \tag{12}$$

The cost value matrix of C_{cost} is the permutation matrix of the mapping function. As we discussed before, the flight path cost has been added in the unified gene encoding, so it is easy to get the cost value of the permutation matrix in the genes of individuals. According to DE strategies, the algorithm calculates the differential temporary solution with the cost value as follows:

$$C_j^{(i)'}(t) = \begin{cases} C_j^{(r_1)}(t) + F \cdot (C_j^{(r_2)}(t) - C_j^{(r_3)}(t)), & \text{rand}[0,1] < CR \\ C_j^{(i)}(t), & \text{Otherwise} \end{cases} \tag{13}$$

In (13), F is the scale factor which controls the amplification of the differential variation rate. CR is the crossover rate. $C_j^{(i)'}(t)$ is a temporary result given by the DE differential strategy, which represents the distance difference between father individuals. It also may be a reasonable value or an invalid duplicate value, zero value, negative value or over bounds value. Therefore, in order to get feasible new offspring individuals, we need inverse mapping the temporary result to correspond to the relationship of the UAV and the target by some matching rules. The inverse mapping function can be defined as:

$$\phi' : R_c^n \xrightarrow{Match_{rules}(1,2,3)} \{(U_i, T_j) \in Z^{n_d}\} \tag{14}$$

In the inverse mapping strategies we adopt three matching rules, which ensure that the temporary result transforms

into a new feasible offspring of an individual. Those match rules can be described as follows:

1. Rule 1 (Nearest neighbor matching): For each reasonable value in a differential temporary result, we find out the nearest value $C(i, j)$ in the cost matrix, and make the corresponding $(U_i, T_j, C(i, j))$ the new gene position of offspring individual. The nearest value is the minimum value compared with all the cost values in the matrix, which is described as follows:

$$C'_{(i,j)} = C_{(i,j)} | \min\{(x_j^{(i)})'(t) - C_{(i,j)}, 0 < i < n, 0 < j < m\} \quad (15)$$

2. Rule 2 (Unique matching): The match should guarantee the reasonable value uniquely corresponding to a relationship of the UAV and the target, and meanwhile avoid the duplicate value of mapping to the same position. So some strategies are required to choose the position of the genes in the cost matrix. The different assignment model need different strategies, which are as follows:

- a. While $N = M$, both of UAVs and targets are not repetitive in a gene sequence. After the match element in the cost matrix is found and added to the new offspring individuals, we delete the row and the column of this element in the matrix, so that the cost values of that row and column are no longer involved in the next matching.
 - b. While $N > M$, the UAVs are not repetitive but the same target can be repetitively matched by different UAVs. When we matched the element in the cost matrix and add it to the new offspring individuals, we delete the row of this element in matrix, which can prevent any UAV duplication.
 - c. While $N < M$, the UAVs may cruises to a group of different targets and the targets can't be repetitive in a gene sequence. So we adopt the iterative replacement method to match the results. First, we match all the UAVs in the upper part of cost matrix, so that each UAV matches at least one target. Then we use the row in the lower part of matrix to replace the row of UAV in upper part of matrix. The replacement uses current visited target point as the UAV's new start point, so that the UAV can visit the next target from this new location. Finally we iteratively execute this replacement process, and match the target in the changed upper part of cost matrix until the rest of targets are matched.
3. Rule 3 (Invalid mutation matching): After matching the reasonable values of the temporary result by Rule 1

and 2, the invalid values and the unmapped matrix are left. Those invalid zero values, negative values and over bounds values in the temporary result can be matched by the local random mutation. Therefore, an element is randomly assigned in the unmapped matrix to an invalid value, and the unmapped matrix is modified by Rule 2 until all the invalid values are matched and the unmapped matrix is null.

During inverse mapping with the match rules, the operation of deleting rows or columns seriously reduces the algorithm's efficiency. Therefore, we use an unlimited large value *Inf* to reset those rows or columns instead of deleting them. The *Inf* can easily improve the efficiency of the algorithm, and ensures that the matching process is implemented smoothly. Figure 7 shows how the match rules work.

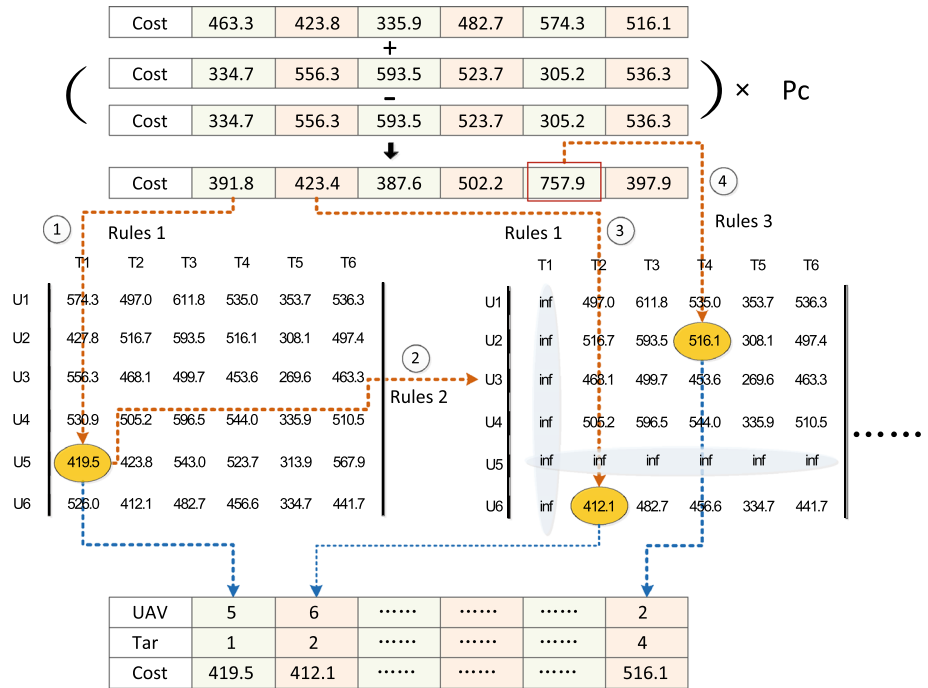
4.2 DMDE with hybrid evolution strategies

In the application of DMDE for the MUCMTA problem, a main issue is to find the trade off between the exploration and the exploitation of the search by setting the parameters and differential strategies. The exploration effectively identifies the optimal solution in the search regions, exploitation accelerates the convergence to the local optimal solution [44]. To improve the performance of DMDE, something is needed to be done to not only enhance the exploration to cover more space, but also to increase the exploitation to refine the search. However, those two aspects interact with each other. Excessive exploration may make the algorithm difficult to converge to the optimal solution, while the over exploitation may cause the algorithm to fall into a local optimum. Consequently, only by keeping a balance between exploration and exploitation, can the performance of the DMDE algorithm rise in the searching process.

The DE algorithm with a single strategy tends towards better exploration (DE/rand/y/z) or faster convergent exploitation (DE/best/y/z), which only focuses on one aspect of optimization. The DE with strategy of DE/rand-to-best/y/z keeps a better balance of both aspects than the above. However, this strategy also lowers the capability of optimization, because the random or best parameter is still a fixed proportion in the evolutionary process.

For the above reasons, we adopt a hybrid strategy to solve this problem, which dynamically mixes two evolutionary strategies. The hybrid strategy uses a dynamic crossover rate to adjust the proportion of DE/rand/1/bin and DE/best/2/bin in the evolutionary process. In the early stages of evolution, the algorithm is dominated by the strategy of DE/rand/1/bin, which maintains population diversity. Then the algorithm gradually turns to the strategy of DE/best/2/bin, employing a more sophisticated search

Fig. 7 The schematic of match action



for the optimal solution. So, the hybrid pattern dynamically adjusts the evolutionary strategies to trade off the exploration and the exploitation in the search process. It can effectively improve the efficiency of the DMDE algorithm. The dynamic crossover rate CR is defined as follows:

$$CR = 1 - \left(\frac{\log(x)}{\log(S)} \right)^\theta, \quad x \in (1, 2, \dots, S), \quad \theta \in N \quad (16)$$

Where x is the current generation, S the total generation, and θ the exponent of the curvature. The curves of CR with different θ are shown in Fig. 8.

From Fig. 8, it can be seen that the value of θ determines the variation of the curves. So the appropriate value of θ can better balance the exploration and the exploitation. For example, when $\theta = 3$, the dynamic crossover rate reduces very quickly at the beginning of the search, so that the proportion of the exploration by random search quickly approaches to the proportion of exploitation by the refined search. Then the crossover rate changes more and more slowly and the proportion of the exploitation is gradually larger than the exploration, which ensures that the refined search is dominant at the end of the search process.

Then, the new individual generated according to the hybrid strategy and the dynamic crossover rate is as follows:

$$C_j^{(i)'}(t) = \begin{cases} C_j^{(r_1)}(t) + F \cdot (C_j^{(r_2)}(t) - C_j^{(r_3)}(t)), & rand[0, 1] \leq CR \\ C_j^{(best)}(t) + F \cdot (C_j^{(r_1)}(t) + C_j^{(r_2)}(t) - C_j^{(r_3)}(t) - C_j^{(r_4)}(t)), & rand[0, 1] > CR \end{cases} \quad (17)$$

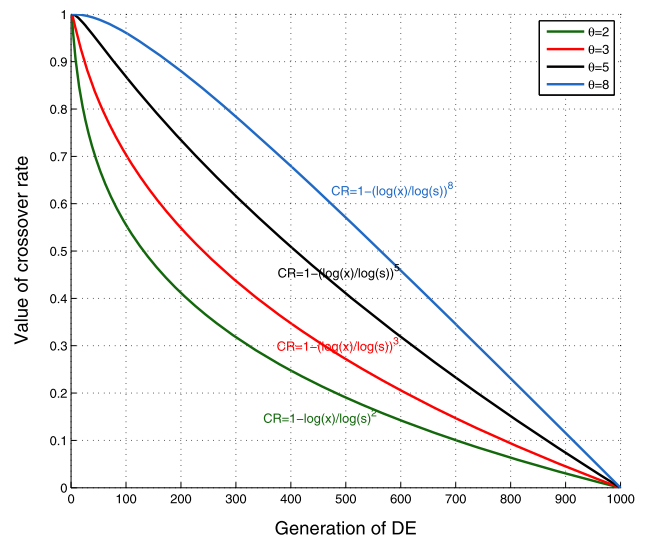


Fig. 8 Crossover rate curve

$$F = \begin{cases} CR, & rand[0, 1] \leq CR \\ (1 - CR)/2, & rand[0, 1] > CR \end{cases} \quad (18)$$

Equation (17) shows that the crossover strategy of DE/ rand/1/bin helps to explore more individuals when CR is a

Table 2 Initial data for experiment 1

Model	Data type	1	2	3	4	5	6	7	8	9	10	
$N = M$	U_{pos}	(78, 20, 15)	(93, 31, 12)	(31, 20, 13)	(112, 32, 15)	(134, 26, 17)	(45, 52, 12)	(36, 63, 13)	(67, 34, 16)	(21, 85, 15)	(32, 62, 13)	
	T_{pos}	(70, 181, 12)	(15, 133, 13)	(19, 151, 12)	(160, 192, 13)	(101, 120, 23)	(160, 113, 12)	(82, 101, 11)	(173, 140, 12)	(182, 65, 14)	(128, 156, 12)	
	Radar	(120, 70, 4, 23)	(110, 140, 4, 26)	(52, 103, 5, 30)								
	V (km/m)	[1.2, 1.8]	[1.2, 2.4]	[1.8, 3.0]	[1.2, 1.8]	[1.8, 2.4]	[1.8, 3.0]	[1.2, 2.4]	[1.2, 2.4]	[1.8, 3.0]	[1.2, 2.4]	[1.8, 3.0]
	D (km)	300	400	500	300	450	500	400	400	500	400	500
	W	1.0	0.4	0.5	0.7	0.9	1.0	0.6	0.6	0.8	1.0	1.0
	T_{sort}	[3, 4]	[5, 2]	[6, 8]	[7, 4]							
	$N > M$	U_{pos}	(78, 20, 15)	(93, 31, 12)	(31, 20, 13)	(112, 32, 15)	(134, 26, 17)	(45, 52, 12)	(36, 63, 13)	(67, 34, 16)	(21, 85, 15)	(32, 62, 13)
		T_{pos}	(70, 181, 12)	(15, 133, 13)	(19, 151, 12)	(160, 192, 13)						
		Radar	(120, 70, 4, 23)	(110, 140, 4, 26)	(52, 103, 5, 30)							
V (km/m)		[1.2, 1.8]	[1.2, 2.4]	[1.8, 3.0]	[1.2, 2.4]	[1.8, 2.4]	[1.8, 3.0]	[1.2, 1.8]	[1.2, 1.8]	[1.2, 2.4]	[1.8, 3.0]	[1.2, 2.4]
D (km)		300	400	500	400	450	500	400	300	400	500	400
W		0.8	1.0	0.6	0.9							
T_{sort}		[3, 2]	[1, 4]	[1, 2]								
T_{wait} (m)		0.3	0.4	0.2	0.9	0.5	0.7	0.3	0.3	0.8	0.9	0.6
$N < M$		U_{pos}	(78, 20, 15)	(93, 31, 12)	(31, 20, 13)	(112, 32, 15)						
		T_{pos}	(70, 131, 12)	(15, 103, 13)	(19, 151, 12)	(160, 192, 13)	(101, 120, 23)	(160, 113, 12)	(82, 101, 11)	(173, 140, 12)	(182, 65, 14)	(45, 176, 12)
	Radar	(120, 70, 4, 23)	(110, 140, 4, 26)	(52, 103, 5, 30)								
	V (km/m)	[1.2, 1.8]	[1.8, 3.0]	[1.2, 2.4]	[1.8, 3.0]							
	D (km)	300	500	400	500							
	W	1.0	0.8	0.6	0.7	0.9	0.8	1.0	1.0	0.7	1.0	0.8
	T_{sort}	[3, 4]	[5, 2]	[6, 8]	[7, 4]							
	T_{win} (m)	[100, 600]										

big value in the beginning. Then the proportion gradually changes with CR . After evolution iterates many generations, solutions are close to the optimization, and CR is near the minimum value. So the crossover strategy of DE/best/2/bin guide the searching process to exploit the optimal solution. Moreover, Eq. (18) shows that the value of the scale factor F is related to CR so that it is also the value of dynamic change.

On the other hand, we also utilize the generational mutation mechanism in DMDE to enlarge the search coverage. The generational mutation saves some excellence individuals, and replaces all other common individuals in the population during generation iteration. So, it significantly changes the tendency of before search. It also enhances the population diversity and random search coverage. We set the generation mutation rate as $GMR = 1 - CR$, which gradually increase with the reduction of CR .

5 Experimental results

To demonstrate the effectiveness of the improved DMDE algorithm, several simulation cases are implemented on the software platform of MATLAB R2011b in Windows 7. The PC had a Core i5 3.20 GHz (quad-core) processor and 8GB of RAM. We also simulated the same 3D terrain, randomly fixed radars coordinates, and different numbers of UAVs and targets as the previous research to compare the improved approach with previous common approach under the unified assignment model.

5.1 The effectiveness and efficiency of DMDE

In order to verify the DMDE algorithm searching the global optimal solution validity, we set two different scales for the experiments. Exp1 has a small number of UAVs and targets to verify the feasibility of the algorithm and in comparison with the result of previous research’s method. Exp2 has a big number of UAVs and targets to prove the algorithm’s effectiveness in complex situations. We set the initial data format as: the velocity matrix of UAVs as

$$V = \begin{pmatrix} V_{min}^1 & V_{max}^1 \\ \vdots & \vdots \\ V_{min}^n & V_{max}^n \end{pmatrix}, \text{ the constraint vector of maximum}$$

flight path cost as $D = [d_{max}^1, \dots, d_{max}^n]$, the constraint vector of maximum flying time as $T = [\frac{d_{max}^1}{v_{min}^1}, \dots, \frac{d_{max}^n}{v_{min}^n}]$, the weight vector of targets as $W = [\omega_1, \dots, \omega_n]$, the constraint matrix of targets order as $T_{sort} = \begin{pmatrix} t^{(i)} & t^{(j)} \\ \vdots & \vdots \end{pmatrix}$, the constraint vector of arriving at the same time as $T_s = [k_1, k_2, \dots, k_l]$ (k is the number of teams by assignment), the vector of maximum initial waiting time as $T_{wait} = [t_{wait}^1, t_{wait}^2, \dots, t_{wait}^k]$, and the constraint of the time window as $T_{win} = [t_{start}, t_{end}]$. The initial data for Exp1 is shown in Table 2. Moreover the scale factors in the objective function are chosen with $\alpha = 2.5, \beta = 1.5(N \geq M)$ and $\alpha = 1.5, \beta = 1(N < M)$. Then we set the dynamic crossover rate as CR from Eq. (16) and assume the exponent as $\theta = 3$. The other evolution parameters of DMDE for each experiment are shown in Table 3.

Where, ‘Pop’ denotes the number of individuals in populations, ‘Gen’ is the number of iterations in generations, and ‘Num’ is the repeated times of the experiments.

The statistical data from the results of experiments is listed in Table 4.

Where ‘Avg time’ is the average running time, ‘Avg cost’ is the average total cost of the flight path, ‘Best cost’ is the optimal total cost of the flight path over many runs of the experiments, ‘Violation’ is cooperative violation rate, which is defined as the sum of the violation percentage of the total cost in the optimal solution, and the ‘Optimal’ is the rate of optimal result times the percentage of experiment times. The ‘Avg cost’ and ‘Best cost’ are all fitness scores, which are composed of optimization goals and various constraints violations. So they are dimensionless values.

We can draw two conclusions from Table 4: (1) the improved DMDE algorithm can get reasonable results of targets assignments for various MUCMTA models; (2) the algorithm can also handle the large scale MUCMTA problem effectively.

For Exp2, if the larger scale experiment increases the complexity and reduces the efficiency of the problem, the DMDE algorithm can efficiently search for an optimal solution. Furthermore, the average running time of both Exp1 and Exp2 are lower and at the same level, which

Table 3 Evolution parameters of each experiment

Model	Exp1					Exp2				
	UAV	Target	Pop	Gen	Num	UAV	Target	Pop	Gen	Num
N = M	10	10	50	1000	20	50	50	50	1000	10
N > M	10	4	50	1000	20	50	15	50	1000	10
N < M	4	10	50	1000	20	15	50	50	1000	10

Table 4 Statistics data by multi-groups assignment result

Exp	Model	Avg time(s)	Avg cost	Best cost	Violation (%)	Optimal (%)
Exp1	N = M	28.67	3879.50	3867.90	7.73	60
	N > M	40.90	4963.80	4948.70	13.3	60
	N < M	50.18	2108.50	2108.30	9.4	90
Exp2	N = M	92.51	25589.00	25183.50	8.36	50
	N > M	81.83	33164.00	32284.30	7.10	60
	N < M	158.71	17843.30	16853.00	6.26	80

Table 5 Experimental assignment results of Exp1

Models	Assignment results									
N = M	U(1)	U(2)	U(3)	U(4)	U(5)	U(6)	U(7)	U(8)	U(9)	U(10)
	T(7)	T(5)	T(4)	T(9)	T(6)	T(8)	T(3)	T(10)	T(2)	T(1)
	458.1	276.2	574.7	137.8	143.8	391.7	307.8	482.0	124.3	447.7
N > M	U(1)	U(2)	U(3)	U(4)	U(5)	U(6)	U(7)	U(8)	U(9)	U(10)
	T(1)	T(4)	T(3)	T(3)	T(2)	T(3)	T(3)	T(3)	T(2)	T(3)
	642.9	564.3	448.6	404.5	388.4	330.2	307.8	521.4	124.3	290.6
N < M	U(1)	U(2)	U(2)	U(3)	U(3)	U(3)	U(4)	U(4)	U(4)	U(4)
	T(7)	T(5)	T(1)	T(2)	T(3)	T(10)	T(9)	T(6)	T(8)	T(4)
	458.1	276.2	164.2	185.0	142.2	201.7	137.8	61.5	39.0	87.8

proves that the unified DMDE algorithm has more efficiency for various assignment models and is suitable for solving larger scale problems. Moreover, the average total cost is very close to the optimal cost, which indicates that the algorithm search is near the global optimal result. Meanwhile, the smaller the cooperative violation rate is, the stronger the cooperative capacity of the solution there is. So the algorithm has less chance to violate constraints. In addition, the higher optimal rate illustrates that experiments have higher probability in finding the optimal solution.

In order to present the assignment results directly, Table 5 illustrates the optimal assignment results of Exp1 for various models, and Fig. 9 shows the corresponding results in the 3D simulation battlefield environment with radar threats and multi-constraints.

In Table 5, ‘U’ is the UAV, ‘T’ is the target, and the value is the flight path cost by the UAV and the target. Then, in Fig 9, the yellow circles indicate the UAVs starting position. The yellow stars indicates the targets position. The red lines show the assignment relationship of the UAV and target, and it also indicates the flight path cost between the UAV and the target.

From Fig. 9, we can see that the DMDE algorithm can effectively deal with the assignment for different models, when the distribution of radars is intensive and the targets are randomly distributed in a 3D environment. Moreover, when the radar is not very intensive in the battlefield, the estimated cost of the flight path will be closer to the real

path, which can provide an important reference for the next phase of the UAVs trajectory planning.

Comparing with the previous study in [40], we list the data of two experiments in Table 6.

From Table 6, the average running time of the experiments with DMDE is shorter than that in the previous research. That is because the DMDE does not only modify the genes generation strategies, but also introduces the differential mapping rules. Thus, the optimal rate shows that DMDE has more capacity to find the optimal result than the previous research.

5.2 Performance of DMDE

The curves of convergence of the three models in Exp1 are shown in Fig. 10, where we can analyze the tendency of DMDE based on the following three aspects:

First, the blue line represents the convergence curve of a single experiment with the optimal point in generation iteration, and the red line shows the average convergence curve by statistical data of the experiments over many running times. The curves of different models have similar convergence trends. All of them can converge near to the optimal solution after about 150 generations, which means that the speed of convergence is high in the beginning of the search. Consequently, the DMDE algorithm has better convergence performance.

Secondly, the red circle in Fig. 10 indicates that the current optimum is obtained by differential mutation, while the blue star represents that the current optimum is

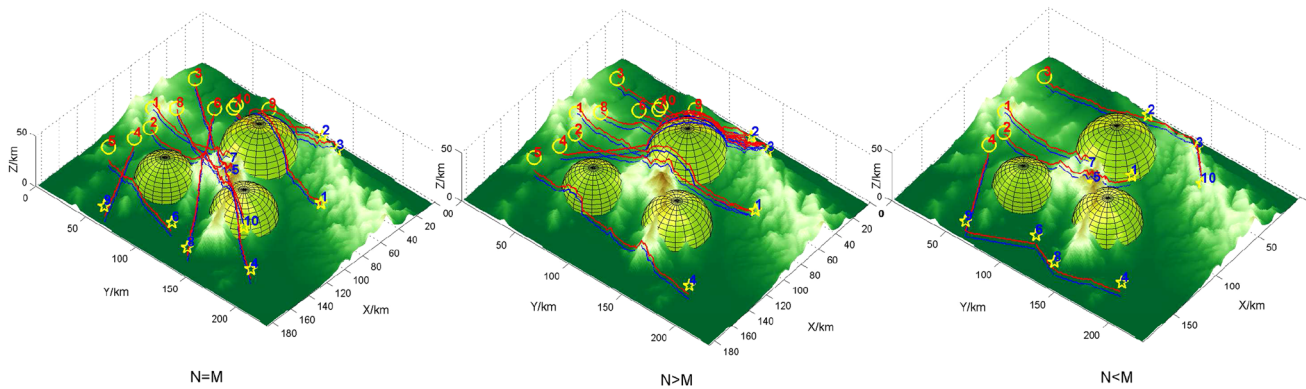


Fig. 9 Simulation results of Exp1 in 3D simulation battlefield environment

Table 6 Comparison with the previous experimental data

Models	Exp	UAV	Target	Pop	Gen	Avg time(s)	Optimal (%)
N = M	New	10	10	50	1000	28.67	80
	Pre	10	10	50	1000	90.39	60
N > M	New	10	4	50	1000	40.96	60
	Pre	10	4	50	1000	244.6	60
N < M	New	4	10	50	1000	50.38	90
	Pre	4	10	50	1000	482.4	80

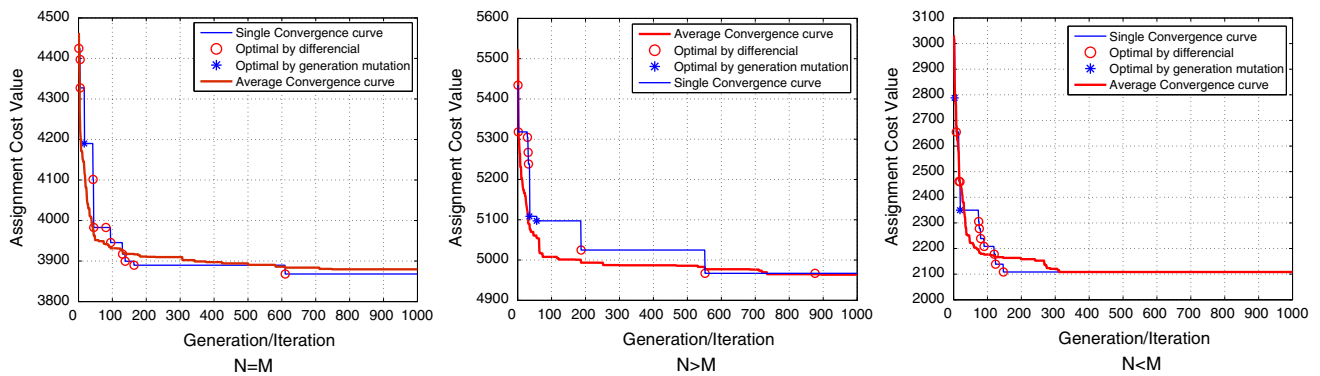


Fig. 10 The Single convergence curve and average convergence curve of assignment result by Exp1

obtained by generation mutation. Then we can see that the generation mutation not only increases the population diversity, but also can find some optimum randomly, while differential mutation still plays a dominant role during the searching process in DMDE. On the other hand, the DMDE algorithm maps physical properties to guide the direction of search, so that it rapidly moves towards the global optimum. Meanwhile, the hybrid strategy reduces the scale of the random search. So the DMDE algorithm has substantially stronger search capabilities.

Finally, in the last phase of the search process, convergence still occurs. It proves that the dynamic crossover rate, generation mutation and hybrid strategies increase the capacity of the algorithm to escape from a local optimum, and ensures the balance between exploration and exploitation.

On the other hand, we compared the different DE crossover strategies for the search. The average convergence curves are shown in Fig. 11. From Fig. 11, it can be seen that the hybrid strategy is better than the single strategy (DE/rand/1/bin or DE/best/2/bin). Moreover, when the hybrid strategy is used in place of DE/rand/2 and DE/best/1, they have very similar search results and convergence curves.

5.3 A comparison between DMDE and other algorithms

We also compared our algorithm with the general GA algorithm, Sort DE algorithm, and PMXDE algorithm. The experiments were set with the same evolution parameters,

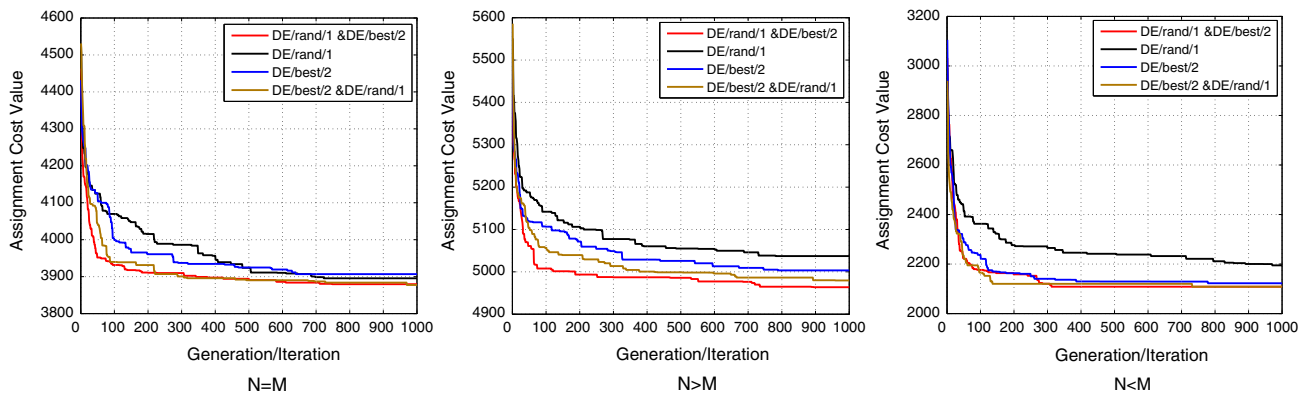


Fig. 11 Average convergence curve of assignment result by different crossover strategies

Table 7 A comparison between DMDE and other algorithms

Model	Method	Num	Pop	Gen	Num	CR	MR	GMR	Best cost	Avg cost	Avg time(s)
$N = M$	DMDE	$N = M = 10$	30	1000	10	CR	Null	1-CR	3867.90	3890.20	17.48
	PMXDE	$N = M = 10$	30	1000	10	0.9	Null	0.3	3869.80	3902.80	77.25
	SortDE	$N = M = 10$	30	1000	10	0.9	Null	0.3	3869.80	3898.60	38.93
	GA	$N = M = 10$	30	1000	10	0.9	0.3	Null	3892.00	3916.60	91.60
$N > M$	DMDE	$N = 10 M = 4$	30	1000	10	CR	Null	1-CR	4954.20	4991.30	29.19
	PMXDE	$N = 10 M = 4$	30	1000	10	0.9	Null	0.3	4993.10	5012.50	184.63
	SortDE	$N = 10 M = 4$	30	1000	10	0.9	Null	0.3	4993.10	5005.90	68.90
	GA	$N = 10 M = 4$	30	1000	10	0.9	0.3	Null	4993.10	5031.10	194.75
$N < M$	DMDE	$N = 4 M = 10$	30	1000	10	CR	Null	1-CR	2108.30	2108.50	45.08
	PMXDE	$N = 4 M = 10$	30	1000	10	0.9	Null	0.3	2274.10	2303.30	478.48
	SortDE	$N = 4 M = 10$	30	1000	10	0.9	Null	0.3	2258.00	2297.00	471.13
	GA	$N = 4 M = 10$	30	1000	10	0.9	0.3	Null	2258.00	2285.60	549.68

population size, and iteration times for the four algorithms. The results are shown in Table 7. In Table 7, the variable CR is crossover rate, the variable MR is mutation rate and the variable GMR is generation mutation rate for DE. The average convergence curve are shown in Fig. 12.

From Table 7 and Fig. 12, it can be seen that DMDE algorithm obtains the optimal solution at a minimum cost, the fastest speed of convergence and with the shortest running time compared with the other algorithms. The Sort DE algorithm has a shorter running time than GA and PMX DE, when $N = M$ and $N > M$. Although those three algorithms can all converge near to the optimal solution, they are worse than DMDE and need more running time.

Furthermore, we analyze the time complexity of the DMDE algorithm. We set n, m to be the scale of individuals (the number of UAV and target), and d, l the scale of DE (population size and generation iterators). Then the time complexity of the DMDE algorithm is mainly affected by the initialization operation $(n \cdot m + m)$, differential operation $(d \cdot l)$, mutation operation $((n \cdot m + m) \cdot l)$, fitness

and constraints calculation $(6n)$, and inverse mapping matching $(m \cdot d \cdot l)$. So the time complexity is as follows:

$$\begin{aligned}
 f(n) &= (n \cdot m + m) + (d \cdot l) + ((n \cdot m + m) \cdot l) \\
 &\quad + (6n) + (m \cdot d \cdot l) \\
 &= 2n^3 + 3n^2 + 2n
 \end{aligned}
 \tag{19}$$

Thus the inverse mapping matched operation and mutation operation mostly lead to the loss of algorithm performance, but the overall efficiency of the algorithm does not reduce because of directly calculating the differential by flight path cost. So the worst-case of time complexity is $O(n^3)$.

In contrast, GA and PMX DE handle the cross and mutation with the discrete sequence of the individual. So the time complexity of evolution is mainly described as $(d \cdot l \cdot (n \cdot m))$, which has more loss of performance than DMDE. The worst-case of time complexity is $O(n^4)$. The other algorithm of Sort DE is simpler than GA and PMX DE in differential operation. It has similar time complexity as DMDE, whose worst-case of time complexity is also $O(n^3)$.

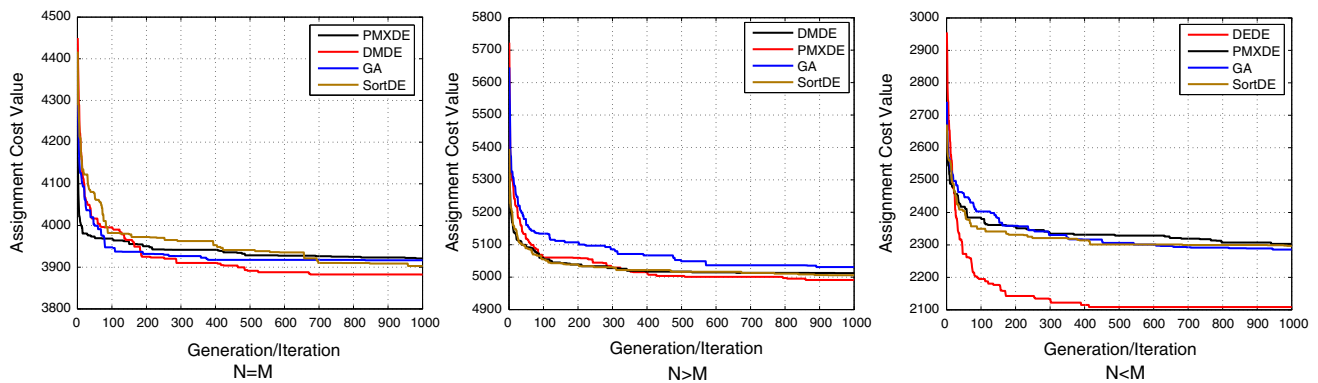


Fig. 12 Average convergence curve of assignment result by different algorithm

These experimental results indicate that the DMDE algorithm has better performance than other algorithms in solving the MUCMTA problem. Because other algorithms still are variants of random search. A blind random search not only leads to the reduction of efficiency, but also rapidly declines the performance when the problem scale increases. On the contrary, the DMDE algorithm is to some degree free from the restriction of random searching, so it is adaptable to solve a large-scale MUCMTA problem.

6 Conclusions

A novel method for UAVs cooperative multi-targets assignment based on a unified model is proposed in this paper. The main contributions of our work include:

1. A unified gene coding strategy is proposed to unify various target assignment models in a consistent framework by analyzing the sequence of targets and UAVs and their flight path cost. The flight path cost is added into the gene code to make it easy to transform the discrete space into continuous space.
2. Mapping and inverse mapping rules are proposed to handle the combinational optimization problem in DE. The improved DMDE algorithm converts the discrete space between UAVs and targets into a continuous space of flight path cost, and does differential operations by using the cost value of flight cost. In order to get rational offspring individuals, the temporary differential results are mapped back to the space of the relationship between the UAV and the target according to the inverse mapping rules. Besides, the distance information is utilized to guide the direction for searching for the global optimum and this could help speed up the searching process.
3. In order to avoid falling into a local optimum, a dynamic crossover rate and hybrid evolution strategies

are applied to balance the exploration and exploitation during the searching process.

Compared with the current algorithms, the improved DMDE algorithm is more efficient. Simulation results show that feasible optimal solutions can be obtained in a reasonable running time.

Future work is to apply the results of MUCMTA to guide the multi-UAVs cooperative trajectory planning.

Acknowledgments The authors would like to thank the national natural science funds of China: 61175027, 61305013, and the fundamental research funds for the central universities HIT.NSRIF.2015069.

References

1. Barlow GJ, Oh CK, Smith SF (2008) Evolving cooperative control on sparsely distributed tasks for UAV teams without global communication [C]. In: Proceedings of the 10th annual conference on genetic and evolutionary computation. ACM, pp 177–184
2. Vandermeersch BRR, Chu QP, Mulder JA et al (2005) Design and Implementation of a mission planner for multiple UCAVs in a SEAD mission [C]. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, vol 6480. AIAA, San Francisco
3. Eun Y, Bang H (2009) Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithms [J]. *J Aircr* 46(1):338–343
4. Bethke B, Valenti M, How JP (2008) UAV task assignment [J]. *IEEE Robot Autom Mag* 15(1):39–44
5. Yi Liu, Weimin Li, Qinghua Xing et al (2010) Cooperative mission assignment optimization of unmanned combat aerial vehicles based on bi-level programming [J]. *Syst Eng Electron* 32(3):579–583
6. Humphrey L, Cohen K (2010) Application of proper orthogonal decomposition and artificial neural networks to multiple UAV task assignment. In: Invited Paper, AIAA Guidance, Navigation, and Control Conference, AIAA, vol 8439. Toronto, Ontario Canada, 2–5 Aug 2010
7. Hoai ALT, Duc MN, Tao PD (2012) Globally solving a nonlinear UAV task assignment problem by stochastic and deterministic optimization approaches [J]. *Optim Lett* 6(2):315–329
8. Kuhn HW (1956) Variants of the Hungarian method for assignment problems [J]. *Nav Res Logist Q* 3(4):253–258

9. Yi L, MingAn Tong (2002) An application of Hungarian algorithm to the multi-target assignment [J]. *Fire Control and Command Control* 27(4):34–37
10. Kuncheva LI (2010) Full-class set classification using the Hungarian algorithm[J]. *Int J Mach Learn Cybern* 1(1–4):53–61
11. Bellingham J, Tillerson M, Richards A et al (2003) Multi-task allocation and path planning for cooperating UAVs [M]. *Cooperative Control: Models, Applications and Algorithms*. Springer, US, pp 23–41
12. Maddula T, Minai AA, Polycarpou MM (2004) Multi-Target assignment and path planning for groups of UAVs [J]. *Recent Dev Coop Control Optim* 3:261–272
13. Lemaire T, Alami R, Lacroix S (2004) A distributed tasks allocation scheme in multi-UAV context [C] *Robotics and Automation, 2004*. In: *Proceedings of ICRA'04*. IEEE International Conference on IEEE 2004, vol 4, pp 3622–3627
14. Sujit PB, Sinha A, Ghose D (2006) Multiple UAV task allocation using negotiation [C]. In: *Proceedings of the fifth International Joint Conference on Autonomous agents and multiagent systems*. ACM, pp 471–478
15. Tao L, Lincheng S, Huayong Z, Yifeng Niu (2007) Distributed task allocation & coordination technique of multiple UCAVs for cooperative tasks [J]. *Acta Autom Sin* 33(7):731–737
16. Yuan L (2011) Research on resources allocation and formation trajectories optimization for multiple UAVs cooperation mission [D]. National University of Defense Technology, Changsha
17. Chen J, Sun D (2012) Coalition-based approach to task allocation of multiple robots with resource constraints [J]. *IEEE Trans Autom Sci Eng* 9(3):516–528
18. Zhao PS, Jiang JG, Liang CY (2011) A distributed algorithm for parallel multi-task allocation based on profit sharing learning [J]. *Acta Autom Sin* 37(7):865–872
19. Alidaee B, Wang H, Landram F (2011) On the flexible demand assignment problems: case of unmanned aerial vehicles [J]. *IEEE Trans Autom Sci Eng* 8(4):865–868
20. Tian J, Zheng Y, Zhu H et al (2005) A MPC and genetic algorithm based approach for multiple UAVs cooperative search[M] *Computational Intelligence and Security*. Springer, Berlin
21. Gupta P, Mehlatat MK, Mittal G (2013) A fuzzy approach to multicriteria assignment problem using exponential membership functions[J]. *Int J Mach Learn Cybern* 4(6):647–657
22. Shima T, Schumacher C (2009) Assignment of cooperating UAVs to simultaneous tasks using genetic algorithm [J]. *J Oper Res Soc* 2008(60):973–982
23. Mingyue D, Changwen Z, Chengping Z (2009) *Unmanned aerial vehicle trajectory planning*. Publishing House of Electronics Industry, Beijing
24. Salman A, Ahmad I, AI-Madani S (2002) Particle swarm optimization for task assignment problem [J]. *Microprocess Microsyst* 26(8):363–371
25. Sujit PB, George JM, Beard R (2008) Multiple UAV Task Allocation Using Particle Swarm Optimization [C]. In: *AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA*, pp 18–21
26. Ho SY, Lin HS, Liauh WH et al (2008) OPSO: Orthogonal particle swarm optimization and its application to task assignment problems [J]. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(2):288–298
27. Bo G, Shewei W, Jun T (2009) Cooperative task allocation for unmanned combat aerial vehicles using improved particle colony algorithm [J]. *Comput Simul* 26(7):62–64
28. Souravlias D, Parsopoulos KE (2014) Particle swarm optimization with neighborhood-based budget allocation. *Int J Mach Learn Cybern*, pp 1–27. doi:10.1007/s13042-014-0308-3
29. Dasgupta P (2008) A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles [J]. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(3):549–563
30. Fei S, Yan C, Lincheng S (2008) UAV cooperative multi-task assignment based on ant colony algorithm [J]. *Acta Aeronaut Astronaut Sin* 29(S1):184–191
31. Chang WL, Zeng D, Chen RC et al (2013) An artificial bee colony algorithm for data collection path planning in sparse wireless sensor networks. *Int J Mach Learn Cybern*, pp 1–9. doi:10.1007/s13042-013-0195-z
32. Jevtic A, Andina D, Jaimes A et al (2010) Unmanned aerial vehicle route optimization using ant system algorithm [C]. In: *IEEE 2010 5th International Conference on System of Systems Engineering (SoSE)*, pp 1–6
33. Zhong L, Luo Q, Wen D et al (2013) A task assignment algorithm for multiple aerial Vehicles to attack targets with dynamic values [J]. *IEEE Trans Intell Transp Syst* 14(1):236–248
34. Zhu D, Huang H, Yang DX (2013) Dynamic task assignment and path planning of Multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace [J]. *IEEE Trans Cybern* 43(2):504–514
35. Gao Q, Zou H, Zhang X et al (2013) New relaxation algorithm for three-dimensional assignment problem [c]. In: *IEEE Conference Anthology*. IEEE, China, pp 1–4
36. Engelbrecht AP (2010) *Computational intelligence: an introduction*[M]. Tsinghua University Press, Beijing Tan Ying translation
37. Onwubolu G, Davendra D (2009) *Differential Evolution for Permutation-Based Combinatorial Problems* [M]. Springer, Berlin
38. Mezura ME, Velquez RJ, Coello CA (2006) A comparative study of differential evolution variants for global optimization[C]. In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, pp 485–492
39. Min S, Ruixuan W, Zhiming F (2010) Cooperative task assignment for heterogeneous multi-uavs based on differential evolution algorithm [J]. *J Syst Simul* 22(7):1706–1710
40. Zhao MS, Xiaohong MP, Lingling Z (2012) A unified modeling method of UAVs cooperative target assignment by complex multi-constraint conditions [J]. *Acta Autom Sin* 38(12):2038–2048
41. Storn R, Price K (1997) Differential evolution A simple and efficient heuristic for global optimization over continuous spaces [J]. *J Glob Optim* 11(4):341–359
42. Otani T, Suzuki R, Arita T (2013) DE/isolated/I: a new mutation operator for multimodal optimization with differential evolution[J]. *Int J Mach Learn Cybern* 4(2):99–105
43. Epitropakis MG, Tasoulis DK, Pavlidis NG et al (2011) Enhancing differential evolution utilizing proximity-based mutation operators [J]. *IEEE Trans Evol Comput* 15(1):99–119
44. Zamuda A, Brest J, Boskovic B et al (2008) Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution [C]. In: *IEEE Congress on Evolutionary Computation CEC 2008*. (IEEE World Congress on Computational Intelligence), IEEE pp 3718–3725