2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery

Multi−UAV Task Allocation Based on Improved Algorithm of Multi-Objective Particle Swarm Optimization

Yang Gao, Yingzhou Zhang*, Shurong Zhu, Yi Sun
College of Computer
Nanjing University of Posts and Telecommunications
Nanjing, China
{1016041013, zhangyz, 1016041019, 1216043129}@njupt.edu.cn

Abstract—**With the development of the technology of unmanned aerial vehicle (UAV), the multi-UAV task allocation has become a hot topic in recent years. Recently, many classical intelligent optimization algorithms have been applied to this problem, because the multi-UAV task allocation problem can be formalized as a NP-hard issue. However, most research treat this problem as a single objective optimization problem. In view of this situation, we use an improved algorithm of multi-objective particle swarm optimization (MOPSO) to solve the task allocation problem of multiple UAVs. We will take two stages of SMC resampling to improve the disadvantages in the MOPSO algorithm. In the first stage, resampling is used to improve the slow convergence of the particle swarm optimization in the middle and late stages. In the second stage, resampling is used to expand the search area of the particle swarm optimization algorithm and to prevent the algorithm from falling into the local optimal solution. The simulation results show that the improved algorithm has a good performance in solving the task allocation problem of multiple UAVs.**

Keywords-Multi-UAV Tsak Allocation; Multi-objective Optimization; Sequential Monte Carlo; Particle Swarm Optimization

I. Introduction

With the development of computer technology, sensor technology, wireless communication technology and unmanned aerial vehicles technology, multi-UAV with highly autonomous control ability can carry out cooperative operation [1]. Multi-UAV cooperative operations can improve the operational effectiveness of UAVs, and become the development trend of UAV combat applications [2-3].

The task allocation of multi-UAV has become a hot topic [4] in recent years. Its main research content is to determine which task should be carried out by which UAV in the case of multiple UAV tasks. In this process, we need to consider the benefits, the costs, and the time needed to complete the task. Generally, the task allocation problem of UAV can be regarded as a NP difficult combinatorial optimization problem [5]. The commonly used models include the multi-traveling salesman problem model [6] (MTSP), the vehicle routing problem model [7] (VRP), the mixed-Integer Linear Programming [8] (MILP) and so on. In this paper, the problem of multi-UAV task allocation is modeled as a multi-traveling salesman problem. Each UAV is used as a traveling salesman and the task point is used as a traversal city.

The distribution scheme we are solving should be within a given constraint to obtain as much revenue as possible at the lowest possible cost. In addition, when we solve the optimal allocation scheme, the optimization objectives include the cost of task execution, the total time of task execution and the total revenue of the task, rather than considering only one aspect. Therefore, the task allocation problem of multi UAV is essentially a multi-objective optimization problem [9] (Multi-objective Optimization Problem, MOOP), and it is more appropriate to use multi-objective optimization algorithm to solve it. But, most of the existing studies turn multiple optimization problem into a single objective optimization problem by adding weights to each optimization target. And the constraints are usually added as a penalty item to the back of the optimization function. However, the units, values and importance of all the optimized targets are different in the multi-objective optimization problem. The weights cannot be set accurately when adding together, and the subjective influence is too great. Therefore, the solution of this paper will use the form of Pareto optimal solution.

In recent years, the research on Pareto optimal solution in multi-objective optimization has been increasing. M Khoroshiltseva adopted the multi-objective optimization methodology based on Harmony Search and Pareto front approaches to solve the problem of designing new energy-efficient static daylight devices that would surround the external windows of a residential building in Madrid [10]. Hajipour V (2016) [11]proposed a multi-objective multi-layer facility location-allocation (MLFLA) model with congested facilities using classical queuing systems and used a Pareto-based multi-objective heuristic method to find and analyze the Pareto optimal solutions. M Ali (2018) [12] proposed a modified variant of Differential Evolution (DE) algorithm for solving multi-objective optimization problems and introduced a new selection mechanism for generating a well distributed Pareto optimal front. We will propose a multi-objective particle swarm optimization algorithm based on sequential Monte Carlo resampling to solve the multi-UAV task allocation.

In this paper, we will introduce the second section with the model of multi-UAV task allocation problem, including the

The work was partially funded by the NSF of China under grant No.61300054; the 1311 Talent Program Funding of Nanjing University of Posts and Telecommunications.
* Corresponding author. Email addresses: zhangyz@njupt.edu.cn.

978-1-7281-0974-9/18/$31.00 ©2018 IEEE
DOI 10.1109/CyberC.2018.00086

description of constraints and each optimization objective component; The third section is the improved multi-objective particle swarm algorithm proposed in this paper; And we will show how to use the proposed algorithm to solve the Multi-UAV task allocation problem, and compare them with the original algorithm in the fourth section.

II. Multi-UAV Task Allocation Model

A. Problem description

In this paper, we describe the multi-UAV task allocation scenario as a ternary collection {V、T、C}. The $V = \{V_1, V_2, \dots, V_m\}$ is the set of UAVs, indicating that m UAVs can be used to execute tasks in the battlefield, and each element $V_i(i = 1,2, \dots, m)$ includes the $speed_i$, the position (x_i, y_i) and the farthest voyage D_i of the UAV; The $T = \{T_1, T_2, \dots, T_m\}$ is the set of tasks, indicating that n tasks in the battlefield need to be executed, and each element $T_i(i = 1,2, \dots, n)$ includes the position (x_j, y_j) of the task; C is the set of the constraints in the battlefield.

The task allocation process is similar to the multiple traveling salesman problem. The classic traveling salesman problem can be described as: A traveling salesman travels to several cities. He starts from one city and needs to go through all the cities before returning to the departure city. The question is how to choose the route of travel so that the total itinerary is shortest. Multiple traveling salesman problem [6] is a traveling salesman problem into multiple traveling salesman from a city. Each traveler visits a certain number of cities (at least a non-starting city), and finally returns to its departure city. Each city is visited once and only once by a traveling salesman. The goal of the problem is to have the shortest total itinerary of all traveling salesman.

The problem of multiple traveling salesman is more complicated than the traveling salesman problem, because the problem is not only to determine the city of each traveling salesman to pass, but also to determine the sequence of the cities that each traveling salesman passes through.

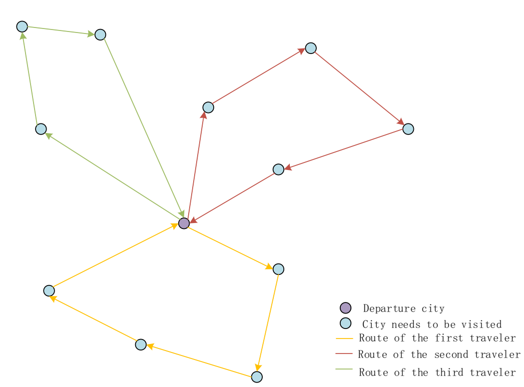

Figure 1. Multi-traveling salesman problem

In the multi UAV task allocation model, the UAV base corresponds to the starting city, the m UAVs correspond to the m traveling salesman, and the n task points correspond the n cities that need to be traversed. Every traveling salesman goes through the sequence of cities and cities, that is, the sequence of tasks and tasks of every UAV.

B. Constraints

The constraints in the multi-drone task assignment model mainly include the following points:

1) Task coordination constraint

In order to prevent a task from being executed many times or not, the task coordination constraint is set up in the model. That is, during the task allocation process, each task must be executed, and each task can only be executed once. Assuming that $t_{i,j} = 1$ indicates that the drone V_i executes the task T_j, $t_{i,j} = 0$ indicates that the drone V_i does not execute the task T_j, and then there are:

$$\sum_{i=1}^{m} t_{i,j} = 1, \quad j = 1,2, \dots, \mathrm{n} \tag{1}$$

2) The UAV's longest range constraint

It is impossible for UAVs to fly indefinitely because the fuel of each UAV is limited, and the UAV's longest range constraint is set in the model. This is, the distance that each UAV has executed after completing all tasks and returning to the starting position shall not exceed the maximum distance that the UAV itself provides. Assuming that the UAV V_i task sequence is $[T_{i1}, T_{i2}, \dots, T_{ik}]$, the distance of all the tasks are executed is d_i, the distance between the task point T_{i1} and the task point T_{i2} is expressed by $\mathrm{D}(T_{i1}, T_{i2})$, and the distance between the UAV V_i and the task T_j is expressed by $\mathrm{D}(V_i, T_j)$, and then there are:

$$d_i = \mathrm{D}(V_i, T_{i1}) + \sum_{j=1}^{k-1} \mathrm{D}\left(T_{ij}, T_{i(j+1)}\right) + \mathrm{D}(V_i, T_{ik}) \tag{2}$$

$$d_i \le D_i, \quad i = 1,2, \dots, \mathrm{m}$$

C. Optimal target components

For how to evaluate the merits and demerits of a task allocation scheme, we set two evaluation indexes in this model. This That is, the targets of this model need to be optimized, which are the total task flight distance and the total task execution time.

1) Total task flight distance

In the process of unmanned aerial vehicle execution, the longer the flight distance is, the more fuel is consumed, and it is more possible that the UAV is found and hit by the enemy. So, we should reduce total task flight distance to decrease the risk of task execution:

$$F_1 = \sum_{i=1}^{m} d_i, \quad i = 1,2, \dots, \mathrm{m} \tag{3}$$

2) Total task flight distance

The longer the total task execution time is, the more easily the UAV is exposed to its location, the greater the threat it is. And in order to reduce the threat, the task allocation scheme should shorten the total task execution time as far as possible:

$$F_2 = \max_{1 \le i \le m} (d_i / speed_i) \tag{4}$$

Based on the above two optimization targets, the optimal objective function of multi UAV task allocation is:

$$\min[F_1, F_2] \tag{5}$$

III. Multi-Objective PSO Based On SMC Resampling

In this paper, we propose a multi-objective particle swarm optimization algorithm based on sequential Monte Carlo resampling. In this section, we will introduce the concept of Pareto optimal solution in multi-objective optimization problems and explain the form of the solution firstly. Then, we will introduce the resampling process in Sequential Monte Carlo algorithm. Finally, the multi-objective particle swarm optimization algorithm is improved by adding the resampling process.

A. Pareto optimal solution

The existence of solutions for multi-objective optimization problems is extremely complex, which is determined by the multiple personality and the complex properties between different objective functions in multi-objective optimization problem. In most cases, the objective functions cannot reach the maximum value or the minimum value at the same time. Therefore, the multi-objective optimization problem seldom has the optimal solution, and the actual problem requires us to make a decision and obtain a better solution. So, we use Pareto optimal solution to measure different optimization objectives.

The Pareto optimal solution [13] refers to an ideal state of optimization. It is impossible to improve a certain optimization target and without damaging any other optimization target. Here are some related concepts of Pareto optimal:

For minimizing multi-objective optimization problems, $F(X) = (F_1(X), F_2(X), \dots, F_n(X))$ represents the target vector, $F_i(i = 1,2, \dots, n)$ represents n target components, H represents the solution set.

1) Pareto dominance

Any given two variables X_1, X_2 if X_1 dominates X_2, represented as $(X_1 \prec X_2)$ then:

$$\forall\, i \in \{1,2,\dots,n\},\ F_i(X_1) \le F_i(X_2)$$
$$\exists j \in \{1,2,\dots,n\},\ F_j(X_1) < F_j(X_2) \tag{6}$$

2) Pareto optimal solution

Pareto optimal solution is also called non-dominated solution. If X_1 is Pareto optimal solution, then:

$$\nexists\, X_i \in H,\ X_i \prec X_2 \tag{7}$$

Where $\nexists$ means non-existent.

From the definition of the Pareto optimal solution, it can be seen that when the Pareto optimal solution is optimized on a target component, at least one of the other target component must be degraded.

3) Pareto optimal set

The set of all Pareto optimal solutions in the multi-objective optimization problem is Pareto optimal set, also called non-dominated solution set. The algorithm proposed in this paper is to find the Pareto optimal set of multi-objective optimization problems.

B. Sequential Monte Carlo Resampling

Sequential Monte Carlo [14] uses particle sets to represent probabilities and can be applied to any form of state space model. Its core idea is to express its distribution by the random state particles extracted from the posterior probability, which is a kind of sequential importance sampling (SIS) method. In simple terms, we select some particle samples according to a certain distribution in the state space and adjusting the weights of these particles to approximate the probability density function. The more the number of samples is selected, the more accurate the approximation will be.

However, there is a degeneracy problem in the sequential importance sampling process. After many iterations, the weight of some particles becomes negligible, and it will reduce the actual effective sample space and affect the accuracy of estimation. To solve this problem, a resampling [15] method is introduced. In the resampling process, particles with larger weights are copied, some particles with smaller weights are abandoned, and it is ensured that the number of particles in the resampling process will not change. The basic diagram of the resampling process is shown as follows:

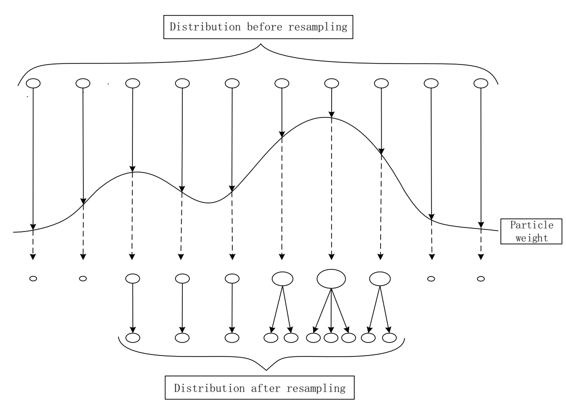

Figure 2. The Resampling process

There are many resampling methods [16], such as Simple random resampling method, Systematic resampling method, Residual resampling method and so on. The following is a method used in this paper—Stratified resampling method:

Algorithm1: Stratified resampling

Input: sample set Z, weight set $\widetilde{\omega}$, sample size k
Output: copy number N

```
for i = 1 to k do
    ω_i = ω̃_i / Σ_{i=1}^{k} ω̃_i
    ã_i = random(1,0)
end for
for i = 1 to k do
    a_i = [(i − 1) + ã_i]/k
    n_i = card{a_i | a_i ∈ (Σ_{s=1}^{i−1} ω_i , Σ_{s=1}^{i} ω_i]}
end for
return N
```

a) Example

There is a sample set include 5 samples $\{Z_1, Z_2, \ldots, Z_5\}$, and the weight of each sample is shown as follows:

sample	Z_1	Z_2	Z_3	Z_4	Z_5
weight	2	3	7	5	3

After the normalization of the weight, the weight of each sample is shown as follows:

sample	Z_1	Z_2	Z_3	Z_4	Z_5
weight	0.1	0.15	0.35	0.25	0.15

The sampling interval for each sample is shown as follows:

sample	Z_1	Z_2	Z_3	Z_4	Z_5
interval	(0,0.1]	(0.1,0.25]	(0.25,0.6]	(0.6,0.85]	(0.85,1]

Assuming that the generated random numbers are: 0.36, 0.86, 0.15, 0.68, 0.42

After stratified the random number are: 0.072, 0.37, 0.43, 0.736, 0.884

Corresponding to the sampling interval of each sample, we can get the number of resample after each sample can be obtained:

sample	Z_1	Z_2	Z_3	Z_4	Z_5
number	1	0	2	2	0

C. Improved multi-objective particle swarm optimization

Particle swarm optimization (PSO) is a kind of population algorithm, first proposed by Kennedly and Eberhart in 1995. It mimics the process of bird searching for food in nature and is suitable for solving complex and nonlinear continuous problems. Through the development and improvement in recent years, the PSO is gradually introduced to discrete problems, and can also solve some combinatorial optimization problems and even multi-objective optimization problems. Liu B et al. (2006) [17] proposed an effective particle swarm optimization, which provided a train of thought for solving the NP problem. Alvarezbenitez J E (2005) [18] applied particle swarm optimization to multi-objective problem solving, and proposed a global guidance method for particle swarm optimization. Norouzi N (2012) [19] applied multi-objective particle swarm optimization to solve the competitive vehicle routing problem and achieve better results.

The particle swarm algorithm has the disadvantages of low solution accuracy and slow search speed in the middle and late stages in the solution process. In this paper, we will take two stages of SMC sampling to improve this shortcoming. In the first stage, we will resample all the particles in the search space. This will improve the convergence speed and accuracy of the algorithm by discarding the particles that are far away from the non-dominant solution, preserving and copying the particles near the non-dominant solution. But, the first stage may lead to loss of diversity of particles. To solve this problem, second stage sampling is carried out in this paper. In this stage, we will resample all the non-dominant particles to discard high-density particles and copy particles with low density. Therefore, we propose a multi-objective particle swarm optimization algorithm based on sequential Monte Carlo resampling (RMOPSO).

The algorithm flow of RMOPSO is listed as follows:

a) Initialization particle swarm:

Setting constant parameters：Particle size N, Maximum number of iterations gen, Inertia weight W, Learning factors c_1, c_2, Resampling interval steps t. Set the initial iteration number $k = 0$, Initialization population P, and the initial position x_i^k, speed v_i^k, Individual optimal position $pbest_i^k = X_i^k$ of each particle.

b) Calculating Pareto optimal set:

We calculate the fitness values of each particle on each target component and contrast the dominate relationship between each particle. Then we will get all the non-dominate solution as the current Pareto optimal set. Deletion: Delete the author and affiliation lines for the second affiliation.

c) determine the population optimal position:

Firstly, we need calculate the density information of each particle in the Pareto optimal set by using the mesh method. The search space is divided into many small regions, and the more particles in the grid, the greater the particle density, the calculation method of particle density is listed as follows:

Algorithm2：Calculation of particle density

Input: Pareto set A, objective function F, Grid number M
Output: Pareto particle density D

```
for each F_i in F do
    maxF_i = max_{a_j∈A} F_i(a_j)
    minF_i = min_{a_j∈A} F_i(a_j)
    ΔF_i = (maxF_i − minF_i)/M
end for
for each a_j in A do
    Gid(a_j) = [Int(F_i^j − minF_i/ΔF_i) + 1],  F_i ∈ F
end for
for each d_j in D do
    d_j = card{a_k | Gid(a_k) == Gid(a_j), a_k ∈ A}
end for
return D
```

Then, each particle p_i selects its own $gbest_i$ according to the density of particle in Pareto optimal and its own dominant relationship. The specific algorithm is shown as follows:

Algorithm3: Calculation of population optimization
Input: Pareto set A, Pareto particle density D, particle p_i
Output: Population optimization of p_i $gbest_i$

1: $A^i = \{a_j | a_j \prec p_i, a_j \in A\}$
2: $d_{min} = \min_{a_j \in A} d_j$
3: $G^i = \{a_j | d_j == d_{min}, a_j \in A^i\}$
4: $gbest_i = rand(G^i).x$
5: **return** $gbest_i$

If the number of iterations is just integer multiples of the resampling interval, then step *d)* is executed, otherwise step *f)* is executed.

d) First resampling:

Calculate the weight of each particle $\widetilde{\omega}_i = \exp(-\|x_i - gbest_i\|_2)$, then calculate the number of resamples per particle according to Algorithm1, and copy or discard the example.

e) Second resampling:

After the first stage resampling, if there is a particle in the Pareto set with a density greater than the density threshold θ, the particles in the Pareto set will be resampling. The density of each particle is $\widetilde{\omega}_i = 1/\sqrt{d_i}$.

f) Adaptively updating:

The learning process is the same as the standard particle swarm optimization (PSO). Each particle adjusts its speed and reaches the next position according to the optimal position of its own history and the position of the global optimal. The process is shown as follows:

$$w_k = W * (gen - k)/gen \tag{8}$$

$$v_i^{k+1} = w_k * v_i^k + c_1 * r_1 * (pbest_i^k - x_i^k) + c_2 * r_2 * (gbest_i^k - x_i^k) \tag{9}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{10}$$

r_1、r_2 are the random number during (0,1).

If the termination condition is satisfied, step *g)* is executed, otherwise, step *b)* is executed.

g) Output Pareto optimal solution:

The non-dominated solution set of the current population is calculated as the output of the algorithm.

The process of RMOPSO's algorithm is shown in Fig3:

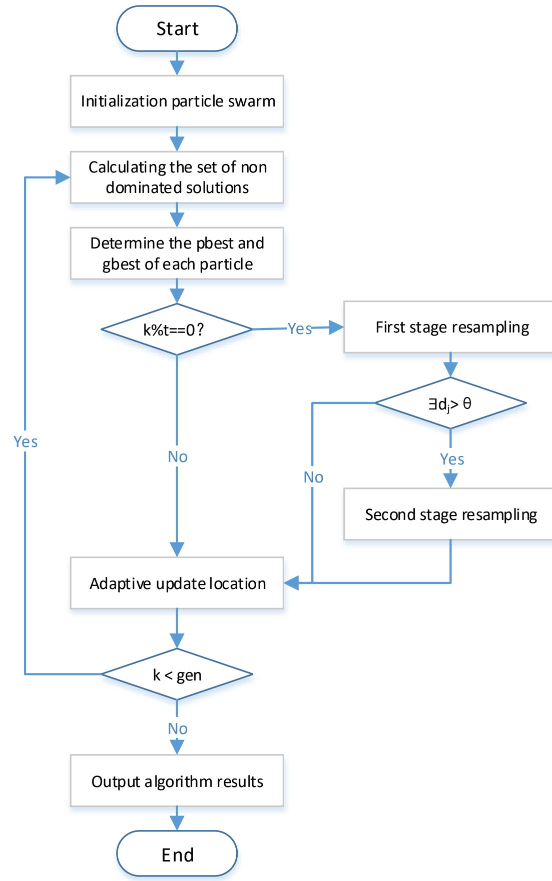

Figure 3. The RMOPSO Algorithm

D. Constraint processing

In single objective optimization problems, constraints are often added to the objective function as penalty items. In the multi-objective optimization problem, there are multiple optimized target components, so constraint is also regarded as an optimized objective component to treat [20].

In this paper, we defines a function L(X) which represents the degree of violation of constraints in particle X. Assuming there are n constraints in the problem model, and we define each constraint as follows:

$$\text{s.t } l_i(\mathrm{X}) \le 0, \mathrm{i} = 1, 2, \dots, \mathrm{n}$$

Then L(X) is defined as follows:

$$\mathrm{L(X)} = \sum_{i=1}^{n} \left|\max(0, l_i(\mathrm{X}))\right|^2 \tag{11}$$

The nature of L(X) is the distance from particle X to the feasible solution area. When X is a feasible solution, the value of L(X) degree is 0. So, the optimization direction of L(X) the as a target component is minimized. In most case, the feasible solution space of a problem occupies a small part of all solution spaces. In order to ensure that most of the areas we search are within the feasible solutions, when we define the Pareto domination relationship, the feasible and infeasible judgment will be a Priority judgment condition. We define X_1 dominate X_2 on constraints:

- The X_1 is a feasible solution, and the X_2 is an infeasible solution;
- The X_1 and X_2 are infeasible solutions, and $\mathrm{L(X_1)} < \mathrm{L(X_2)}$;.
- The X_1 and X_2 are feasible solutions, and $X_1 \prec X_2$.

IV. MULTI-UAV TASK ALLOCATION SIMULATION.

A. Allocation scheme coding

The model of particle swarm optimization has advantages in solving continuous problems. So, in this paper we use a small skill to transform UAV task allocation into semi continuous coding. Assuming that there are m UAVs on the battlefield and n tasks need to be executed, and allocation scheme coding is represented as an array of n bits $[t_1, t_2, \dots, t_n]$. In the array, $t_i (1 \leq i \leq n)$ is the random number during$(1, m+1)$. The integer part of t_i represents the id of UAV that execute the ith task, this means that the task of the same part of the integer is executed by the same UAV. Sorting these tasks according to the decimal part of the corresponding numbers is the order of UAV executing tasks.

For example, there are 3 UAVs in the battlefield and 8 tasks to be executed. The task allocation scheme code is arranged as follows:

Id	1	2	3	4	5	6	7	8
Coding	2.05	3.48	1.54	2.18	1.25	3.05	1.31	2.19

The sequence of UAV execution tasks is listed as follows (0 represents the starting position):

First UAV task sequence: 0, 5, 7, 3, 0

Second UAV task sequence: 0, 1, 4, 8, 0

Third UAV task sequence: 0, 6, 2, 0

B. Comparison and analysis of experiments

We uses TSPlib and VRPlib data sets to conduct comparative experiments. We select one of them as the base of UAV, that is, the starting point and terminal point of all UAVs, and other points as the task points to be executed. We run RMPSO and MOPSO 20 times respectively, and experimental results are shown in Table 1:

TABLE I. COMPARISON BETWEEN RMOPSO AND MOPSO

Test set	RMOPSO				MOPSO			
	Best F_1	AVG F_1	Best F_2	AVG F_2	Best F_1	AVG F_1	Best F_2	AVG F_2
ulysses16	**63.72**	69.87	**2.43**	**2.443**	64.93	**69.68**	**2.43**	2.445
ulysses22	**70.05**	**94.56**	**3.072**	**3.079**	76.36	96.39	**3.072**	3.087
E016-03m	**257.86**	**267.21**	**94.65**	97.48	258.11	267.55	**94.65**	**97.47**
E033-05s	**731.55**	**973.25**	2.423	2.518	748.43	994.05	**2.415**	**2.515**
oliver30	717.53	**965.40**	**2.438**	**2.454**	**683.29**	979.6	**2.438**	2.462
eil51	**652.93**	**720.71**	**1.121**	1.250	659.20	735.56	**1.21**	**1.238**

From the results of the experiment, we can see that the two algorithms have good performance for the minimum value of the total task execution time, and the RMOPSO performs better in the optimization of the task flight task.

C. Example demonstrate

We, taking data ulysses16 as an example, demonstrate the usability of the RMPSO, select point 12 of the original data as the starting point, and the other 15 points as the task point.

1) The scheme of shortest total task flight distance

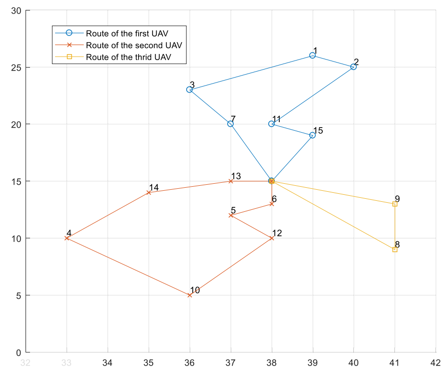

Figure 4. Scheme of shortest total task flight distance

Shortest total task flight distance is 63.72
First UAV task sequence: 0, 7, 3, 1, 2, 11, 15, 0
Second UAV task sequence: 0, 13, 14, 4, 10, 12, 5, 6, 0
Third UAV task sequence: 0, 8, 9, 0

2) The scheme of shortest total task flight distance

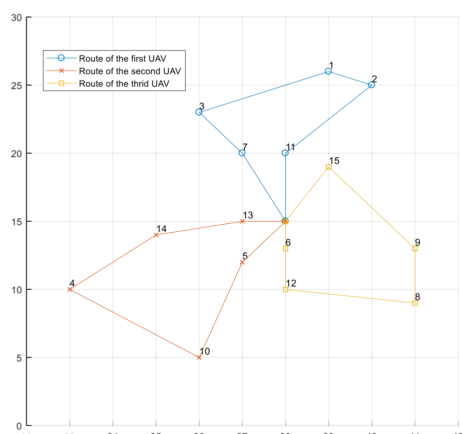

Figure 5. Scheme of shortest total task flight distance

Shortest total task execution time is 2.43
First UAV task sequence: 0, 7, 3, 1, 2, 11, 0
Second UAV task sequence: 0, 13, 14, 4, 10, 5, 0
Third UAV task sequence: 0, 6, 12, 8, 9, 1, 0

V. Conclusion

We clarifies the constraints and optimizes target components in multi-UAV task allocation problem. We model this problem as a constrained multi-objective optimization problem, and use the improved multi-objective particle swarm optimization algorithm to solve it. The sequential Monte Carlo resampling idea is added to the algorithm, which improves the slow convergence rate in the later stage and it is easy to fall into the local optimal solution in the classical particle swarm optimization. Simulation results show that the task allocation method proposed in this paper has certain advantages in some aspects.

References

[1] Shamma J S. Cooperative Control of Dynamical Systems［M］. John Wiley ＆ Sons, 2007.

[2] Shima T, Rasmussen S. UAV Cooperative Decision and Control: Challenges and Practical Approach［M］. Society for Industrial and Applied Mathematics, 2009

[3] Joshua A L. Network level control of collaborative UAVs [D]. Berkeley: University of California, 2011.

[4] Khoroshiltseva M, Slanzi D, Poli I. A Pareto-based multi-objective optimization algorithm to design energy-efficient shading devices[J]. Applied Energy, 2016, 184.

[5] Yang W L, Lei L, Deng J S. Optimization and improvement for multi-UAV cooperative reconnaissance mission planning problem[C]. International Computer Conference on Wavelet Active Media Technology and Information Processing. IEEE, 2015:10-15.

[6] Reviews E M. The multiple traveling salesman problem: an overview of formulations and solution procedures[J]. Omega, 2009, 34(3):209-219.

[7] Laporte G. The vehicle routing problem: An overview of exact and approximate algorithms[J]. European Journal of Operational Research, 2007, 59(2):231-247.

[8] AIAA. UAV Task Assignment with Timing Constraints via Mixed-Integer Linear Programming[J]. 2004.

[9] Li Y, Yue T, Ali S, et al. A multi-objective and cost-aware optimization of requirements assignment for review[C]. Evolutionary Computation. IEEE, 2017:89-96.

[10] Khoroshiltseva M, Slanzi D, Poli I. A Pareto-based multi-objective optimization algorithm to design energy-efficient shading devices[J]. Applied Energy, 2016, 184.

[11] Hajipour V, Fattahi P, Tavana M, et al. Multi-objective multi-layer congested facility location-allocation problem optimization with Pareto-based meta-heuristics[J]. Applied Mathematical Modelling, 2016, 40(7-8):4948-4969.

[12] Ali M, Siarry P, Pant M. An efficient Differential Evolution based algorithm for solving multi-objective optimization problems[J]. European Journal of Operational Research, 2018, 217(2):404-416.

[13] Luc D T. Pareto Optimality[M]. Pareto Optimality, Game Theory And Equilibria. Springer New York, 2008:481-515.

[14] Liu B. Posterior exploration based sequential Monte Carlo for global optimization[J]. Journal of Global Optimization, 2017, 69(4):1-22.

[15] Douc R, Cappe O. Comparison of resampling schemes for particle filtering[C]. International Symposium on Image and Signal Processing and Analysis. 2005:64--69.

[16] Li T, Bolic M, Djuric P M. Resampling Methods for Particle Filtering: Classification, implementation, and strategies[J]. Signal Processing Magazine IEEE, 2015, 32(3):70-86.

[17] Liu B, Wang L, Jin Y, et al. An Effective PSO-Based Memetic Algorithm for TSP[J]. Lecture Notes in Control & Information Sciences, 2006, 345:1151-1156.

[18] Alvarezbenitez J E, Everson R M, Fieldsend J E. A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts[C]. Evolutionary Multi-Criterion Optimization, Third International Conference, EMO

2005, Guanajuato, Mexico, March 9-11, 2005, Proceedings. DBLP, 2005:459-473.

[19] Norouzi N, Tavakkoli-Moghaddam R, Ghazanfari M, et al. A New Multi-objective Competitive Open Vehicle Routing Problem Solved by Particle Swarm Optimization[J]. Networks & Spatial Economics, 2012, 12(4):609-633.

[20] Deb K, Pratap A, Meyarivan T. Constrained Test Problems for Multi-objective Evolutionary Optimization[C]. Evolutionary Multi-Criterion Optimization, First International Conference, EMO 2001, Zurich, Switzerland, March 7-9, 2001, Proceedings. 2001:284--298.