Solving the Dynamic Programming Problem for Training Airspace Based on Modified PSO algorithm

Yanming Wei[1,a], Xusheng Gan[2,b,*], Rui Yang[2,c], Jingjuan Sun[2,d]

[1]Xijing College, Xi'an, Shaanxi, 710123, China

[2] Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an, Shaanxi, 710051, China

[a]ganxusheng123@163.com, [b]gxsh15934896556@qq.com, [c]yangr18@163.com, [d]sunjin21@163.com

Keywords: Training Airspace, Dynamic Programming, Particle Swarm Optimization, Genetic Algorithm, Gantt Chart

Abstract: The dynamic programming model is established for training airspace, and a modified Particle Swarm Optimization (PSO) algorithm is used to solve the dynamic programming problem. Through introducing the crossover and mutation ideas in genetic algorithm, PSO algorithm ability to jump out of local optimal solution is improved with good convergence and accuracy. Simulation shows that the modified PSO algorithm can get a satisfactory application effects in dynamic programming problem for training airspace.

1. Mathematical expression of Dynamic planning Problem for training airspace

In recent years, with the upgrading of weapons and the combat transformation of the air force, the demand of daily training for airspace is increasing rapidly. At the same time, the airspace demand of civil aviation is also sharply increased. The conflict between military and civil aviation is becoming more and more prominent [1][2]. Therefore, the dynamic programming of training airspace plays a positive role in improving the utilization rate of airspace resources and the flexible use of airspace. At the same time, it can also alleviate the conflict between military and civil aviation. The training airspace is the range designed to ensure the flying team's regular training. The size of the airspace required is not the same because of the different methods of the implementation of each subject. The dynamic planning of tactical training airspace is to arrange the training tasks of each batch properly within the airspace of the airport, shorten the total training time required while ensuring space utilization [3]. The constraint is:

a) Once each training task begins, it must continue to be the end of the service.

b) All training tasks must be completed.

c) All training tasks must be completed within the airspace of the airport and shall not exceed the airspace boundary.

d) The task of the training task is 10 minutes per unit.

e) The airspace of each training mission shall not overlap.

f) In case of actual situation, when the required airspace is arranged for each task, it needs to be parallel to the coordinate axis, but it can rotate 90 degrees.

For each stage of the airspace plan, it refers to the arrangement of training subjects as much as possible on the basis of ensuring the utilization of the airspace. That is, in the process of implementation, the utilization of airspace in the entire period is the largest. Its objective function is:

$$\max \quad E = n + \sum_{i=1}^{n} (l_i \times w_i) / S \tag{1}$$

In the form, S is the area of the operational area. l_i is the length of the airspace required for a

  DOI: 10.25236/isaiee.2020.003

training course; w_i is the width of the airspace required for a training course; n refers to the total number of subjects in the course of the period, and $1 \le i \le n$.

2. Algorithm for solving the dynamic programming problem of training space

2.1 Genetic particle swarm algorithm

The dynamic planning for training airspace refers to the reasonable arrangement of training subjects in the airspace of the airport, allowing it to shorten the time as much as possible while ensuring space utilization. Due to the limitation of the airspace, the training of all subjects cannot be completed at the same time, so this problem is regarded as a double constraint problem of timing and space. In this paper, the problem of the sequence constraint is solved by the method of phase processing, and the spatial constraint problem is solved by PSO algorithm [4][5] modified by introducing the crossover and mutation operator in genetic algorithm [6][7].

Suppose the number of particles in the population is m, the maximum number of iterations is N, P_{id} is the optimal position for an individual. P_{gd} is the globally optimal location.

STEP 1: Initialize the population, assign each location an initial velocity and give the corresponding parameter values;

STEP 2: Cross-operated the position of each particle with the individual optimal and global optimal positions to obtain new chromosomes, and then mutated the new chromosomes to obtain the final individual, and updated the crossover rate and mutation rate at the same time;

STEP 3: Use the improved algorithm to decode the particle and get its fitness value, regard it as the initial individual optimal position of the particle P_{id}, choose the position with the greatest fitness as the global optimal position P_{gd};

STEP 4: Update the particle position according to the particle update method in the algorithm;

STEP 5: Performed two variation operations on each particle and found the fitness value of each new particle. If the function value of the new position is greater than the original function value, update the optimal position of the individual .For the whole particle swarm, the maximum fitness value of each new generation is selected to compare with that of the previous generation. If it is greater, take this position as the new global optimal position.

STEP 6: When the number of iterations reaches the set maximum number of iterations or the fitness no longer changes, terminate the algorithm, and drawn the airspace planning diagram according to the optimal planning scheme.

2.2 Algorithm complexity analysis

The complexity of the algorithm not only determines the execution efficiency of the algorithm, but also affects the solving ability of the algorithm to a great extent. Compared with the basic discrete PSO algorithm, the improved algorithm spends more time on the crossover and mutation of particles. Assuming that the size of the problem is *D*, the population size is N, the number of iterations is *G*. For the basic PSO algorithm, it is necessary to update the position and velocity of particles, calculate the corresponding fitness value and retain the optimal individual, and the computational time complexity is

$$O(N,G,D) = 3G \times O(N \times D) + G \times O(N) \approx G \times O(N \times D) \tag{2}$$

The improved algorithm complexity is

$$O(N,G,D) = 3G \times O(N \times D) + G \times O(N) + (P_c + P_m) G \times O(N \times D) \approx G \times O(N \times D) \tag{3}$$

Therefore, it can be seen that the computational time complexity of the algorithm is only related to the solving problem scale *D*, population scale *N* and iteration times *G*. The improved algorithm does not increase the computational time complexity obviously.

2.3 Gantt chart

Gantt charts, also known as bar charts, are primarily designed to show how the work progresses over time. In the chart, the horizontal axis represents time and the vertical axis represents jobs. Lines represent the completion of each task. A Gantt chart visually shows the start and end times of a task, and evaluates how the task is progressing. Gantt chart is a tool to control work schedule with clear and definite characteristics Its drawing method is also very flexible, the specific method can be seen in the references. In this paper, gantt chart is used to represent the progress of each training task over time, and it is drawn by Matlab software.

3. Simulation

The following simulation is carried out based on the daily training of a flight force. The operational responsibility area of the Department and its discretized airspace are shown in Fig. 1 and Fig. 2.

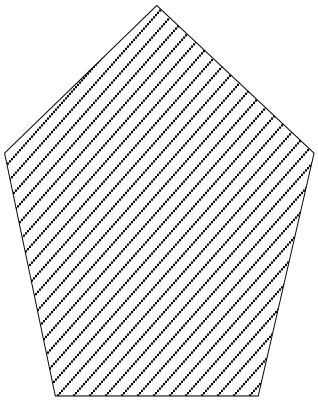

Fig1 Training airspace figure

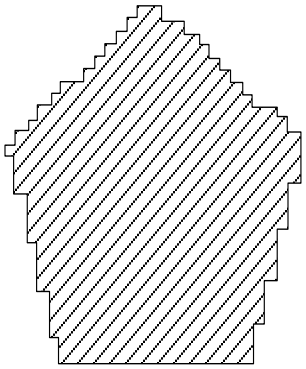

Fig. 2 Discrete training airspace figure

With the traditional PSO algorithm to solve, assuming the same parameters, Through simulation it is found that when the iteration to the 90th generation, basic stable, while the fitness value of 10.7899, the last train diagram and generated the airspace in each generation in the process of iterative optimal fitness value and the average fitness value of change process are shown in Fig. 3 and Fig. 4 below.

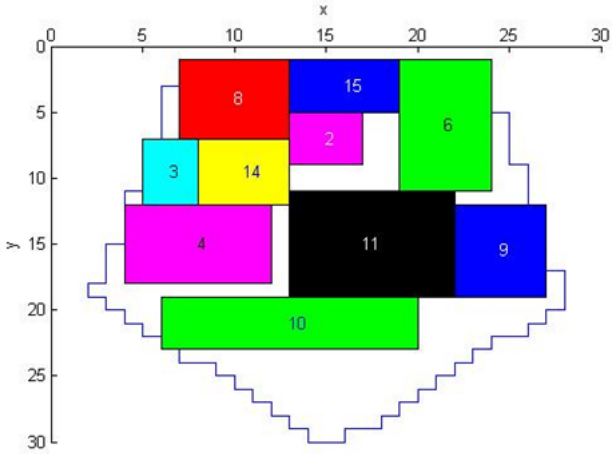

Fig. 3 Tactical training airspace planning diagram

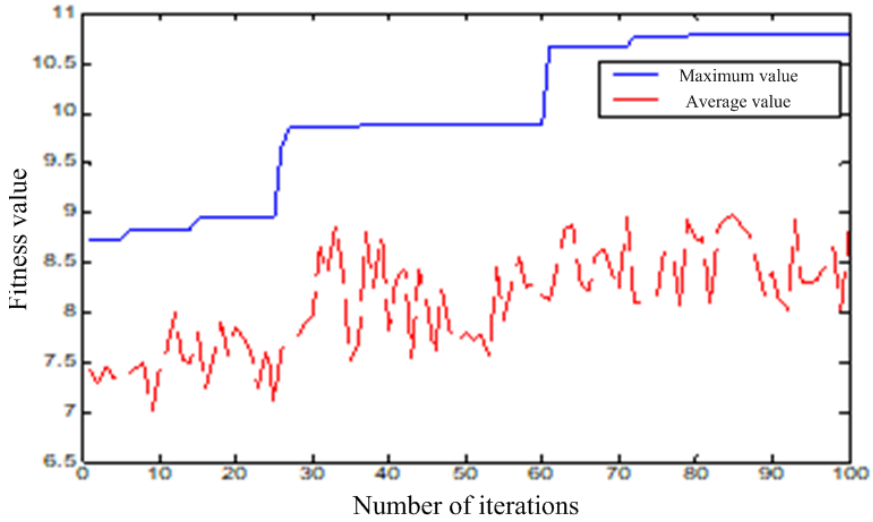

Fig. 4 Curve of fitness value change

By comparison, it can be found that:

1) The training airspace planning scheme obtained by using the genetic-discrete PSO algorithm has a higher utilization rate of the airspace, and more training subjects are arranged in the operational responsibility area, which improves the global searching ability of the algorithm.

2) By comparing the two sets of algebra when the data tends to be stable, it is found that the search performance of the traditional particle swarm optimization algorithm is significantly weaker than that of the genetic-discrete PSO algorithm. When solving the problem by using the genetic-discrete PSO algorithm, the search performance of the algorithm tends to be stable at the 73rd generation, and the convergence ability of the improved algorithm is stronger.

To sum up, both algorithms can be used to solve the dynamic programming problem for training airspace, but the discrete-time PSO algorithm is superior to the traditional PSO algorithm.

It is assumed that a certain part needs to complete 3 tactical training tasks on a certain day, and each task has 4 batches (m=12). Gantt chart shows the progress of training progress over time, as shown in Fig. 5.

As can be seen from the gantt chart, training task 1,2,3,6 is performed first, mission 4,9,12 will be executed at 40 minutes, mission 7 starts at 50 minutes, mission 10 begins at 90 minutes, mission 10 starts at 90 miniutes, mission 5 starts at 100 miniutes, and mission 8,11 starts at 140 miniutes. At present, the planning of tactical training airspace mainly relies on human experience, which can only be carried out in sequence according to the types of training tasks. The change of training schedule over time is shown in Fig. 6 below.

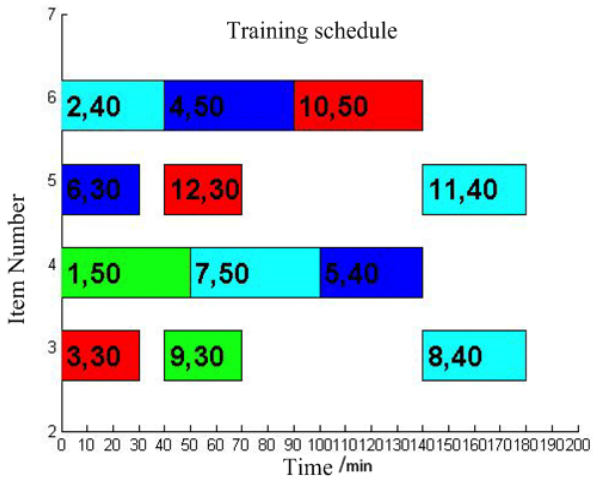

Fig. 5 Training Schedule

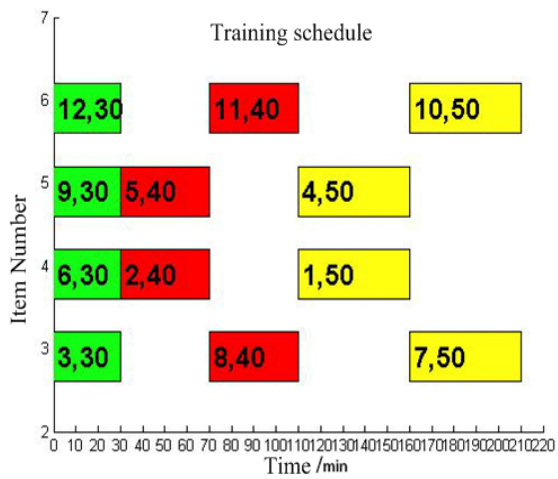

Fig. 6 Current Training Schedule

It can be seen from the above results that, compared with the manual experience setting, all

training tasks can be completed more quickly by using the above algorithm. The training time is shortened from 210 minutes to 180 minutes, saving 30 minutes. It can be seen from the above airspace planning diagram that while realizing dynamic task planning, the utilization rate of airspace and the safety of airspace can also be guaranteed by using the method, which provides certain reference significance for the flexible use of airspace. By using computer to assist design, the time needed for airspace planning is greatly shortened and the planning efficiency is improved.

4. Conclusions

The dynamic programming problem is dealt with in stages for training airspace. A modified PSO algorithm with crossover and mutation operator is applied to solve the dynamic programming problem. Simulation shows that the proposed method can shorten the airspace occupancy time as much as possible while ensuring the airspace utilization. Compared with the current artificial dynamic programming method, the use of the method can greatly reduce the time required for training. This not only can improve flight efficiency, but also can effectively alleviate the flight conflict between civil and military aviation.

References

[1] K. J. Pavithran Kumar. ART1 neural networks for air space sectoring. International Journal of Computer Applications (0975-8887). 37, (2012), 20-24

[2] X. H. Xu Xiaohao, L. L.Wang. Suggestions to improve utility policy of national airspace resource. Nanjing University of Aeronautics & Astronautics, 10(2), (2008), 23-27

[3] C. J. Liang, S. B. Zhang. Integrating scheduling of loading/unloading operation equipment in container terminals. Journal of Liaoning Technical University, 34(2), (2015), 262-266

[4] J. Kennedy, R. C. Eberhart. Particle swarm optimization. Proceedings of 4th IEEE International Conference on Neural Networks, Perth, Australia, Dec. (1995), 1942-1948

[5] F. van den Bergh, A.P. Engelbrecht. A study of particle swarm optimization particle trajectories. Information Sciences, 176(8), (2006), 937-971

[6] M. Mitchell. An Introduction to genetic algorithms. Cambridge, MA: MIT Press, (1999)

[7] M. Srinvas, L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, 24(4), (1994), 656-667