LARGE-SCALE AIR-COMBAT FORMATION OPTIMIZATION USING SIMULATED-ANNEALING GA (GENETIC ALGORITHM)

Zhang Keshi*, Li Weiji* , Wang Zhengping*
*** School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, P.R.China**

Keywords: *optimization, simulated-annealing GA, large-scale air combat, formation tactics*

Abstract

A method of optimizing large-scale air-combat formation based on simulated-annealing GA (SAGA) was developed and investigated. In this study, a weighting method was used to model the problem of large-scale air-combat formation optimization and a "hierarchical coding method" was developed to encode formation. To avoid the premature convergence of GA, the simulated annealing algorithm was embedded into GA, and self-fitted Markov chain was introduced to improve the performance of SAGA. The optimization of 16-fighter formation was implemented, and the optimal formations were presented and showed that SAGA was much better than simple GA. To explore an efficient and robust SAGA, the comparisons of different Markov chains and different state functions were made, it was shown that the combination of self-fitted Markov chain and the state function of equation (2) was more appropriate for large-scale air-combat formation optimization using SAGA.

1 Introduction

Large-scale air-combat formation tactics plays an important role in modern air combat of BVR (Beyond Visual Range). The Large-scale air-combat formation (LACF) is superposed from some small-scale air-combat formations (SACF), and inherits and develops the formation tactics of theirs. Exploration on the optimization method for LACF is practically significant. It may provide formation references to the pilots, so that they can quickly adjust the formation to acquire the initiative in the air combat. A good formation can make up weakness of single-fighter performance and achieve better outcome of combat.

Mulgund et al utilized stochastic GA to optimize LACF and developed correlative software [1,2]. This method is usually used in the case that long genetic code exists. Yet Krishnakumar indicated that it couldn't solve the problem of premature convergence [3]. We proposed simulated-annealing GA to overcome such premature convergence and obtain optimal formations.

2 The Hierarchical Formation Tactics

The term, formation tactics encompasses two concepts, including [1]:

- Formation tactics that specify how small groups of aircraft can work cooperatively
- Principles for constructing division tactics

The large-scale formation tactics employing a basic fighting unit of small-scale formation is most effective. The formation tactics for large groups can be developed using a hierarchical structure consisting of small units or divisions [2]. Basic fighting units of two aircrafts includes Double-Attack (DA) and Fighting-Wing (FW), as shown in Fig.1:

△ △ △ △

Fig.1. Formation of DA (left) and FW (right)

The top vertex of triangles denotes heads of aircrafts. DA refers to the formation that the wingman is flying in the region between 15 degree ahead of and 20 degree behind the lead man. Such a formation is often adopted to gain attacking and defensive advantage in high altitude. FW, by which the wingman is flying in the region of 30～60 degree behind the lead man, is fit for a low altitude because of its fine maneuverability. While these small formation units are integrated hierarchically, their tactics are also inherited and developed. Giving fighting units of two aircrafts as blocks, the potential four-aircraft formations are shown as follows:

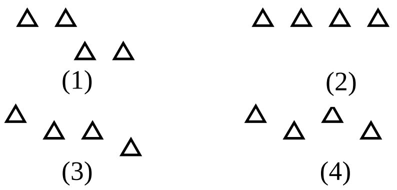

Fig.2. Potential four-aircraft formations

In Fig.2, formations are superposed from two same fighting units. Taking formation (4) in Fig.2 as example, two FW formations form a DA-like formation (In this case, DA was called the "root formation" in this study). One FW of formation (4) is the leading-man unit in this DA-like formation and another FW takes the duty of wingman unit. In the same way, larger-scale formation can be developed from some four-aircraft or three-aircraft fighting units. The next section would discuss how these units could be integrated to form large fighting groups by using SAGA for optimizing the overall air-combat effectiveness.

3 Formation Optimization

3.1 Modeling of Formation Optimization

Taking the air-combat formation as the design variable, fighter casualty statistics was employed to evaluate BVR (beyond visual range) combat outcome and the effectiveness of large-scale air combat formation. The mathematical model could be written as:

$$\max \quad R=\sum_{i=1}^{4} C_i R_i(X) \qquad (1)$$

$$s.t. \quad 0 \le C_i \le 1 \ , \quad i=1,\cdots,4$$

$$0 \le R_i \le 1 \ , \quad i=1,\cdots,4$$

Where, C_i denoted weighting coefficients; X denoted the formation of blue team; R_i was percent of undestroyed blue-team fighters, percent of destroyed red team fighters, average living probability of remained blue-team fighters, average loss of living probability of remained red-team fighters, respectively.

To evaluate the result of air-combat simulation, the simulation of radars and air-to-air missiles should be considered. The probability that radar detects the target relies on azimuth and distance of the target. The models of detecting probability (P_d) and killing probability (P_k) were provided by Ref.4. The latter was utilized to calculate the living probability of the aircraft after being attacked. $P_{k/d}$ denotes the conditional probability that the aircraft is killed after being detected. Assuming that $P_{k/d}$ was equal to P_k, therefore the final living probability was calculated by multiplying P_k by P_d.

How to assign target aircrafts to the attacker, was a key point in many-vs-many engagements. Making use of the method referred in last paragraph, one could calculate the one-to-one relative advantage of blue team to read team. All the data formed the advantage matrix. The rule of assigning targets proposed by Ref.5 and 6 was average, non-repeated and wholly advantageous. Then the target aircrafts were assigned according to the value of advantage, from large to small. If one attacker was selected to attack some target, its threatening power was weakened. That was to say, its referred advantageous value should be cut down. Afterward the advantage matrix was updated. When the living probability of some aircraft became smaller than the set threshold, it deemed to be destroyed. Besides, the initial distance between the two teams was set to 80 kilometers, and it was becoming shorter, for they were all forward with a velocity while fighting. While the distance was as short as 10 km, the combat ended. Finally R_i was evaluated according to the result of combat. Weighting coefficients were decided by one's balance in killing and destroying.

3.2 Hierarchical Coding Method

The formation of blue team was adopted as the design variable in the optimization model. Conventional algorithms cannot deal with such discrete problem, while GA can easily do it. In GA optimization, appropriate coding and decoding method must firstly be ensured. Here the binary code was defined to encode some usual small-scale air-combat formation units, such as “00” for wall-formation, “01” for finger-four formation, and the code of large-scale air-combat formation was obtained by hierarchically superposing the codes of small-scale formations. All the other codes for formation units were shown in table 1.

Table 1. Units of binary codes

Binary codes	Formation	Binary codes	Formation
00	△ △ △ △	10	△ △ △ △
01	△ △ △ △	11	△ △ △ △

The developed “hierarchical coding method” easily translated the formation into binary code with a one-to-one mode. Taking the formation of 16 aircrafts illustrated in Fig.3 as an example, it was explained how to translate a formation into a string of binary codes.

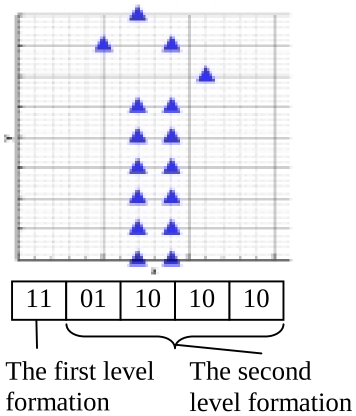

Fig.3. A formation of 16 aircrafts and its corresponding binary codes

This formation could be denoted by ten digits of binary code. The first two digits indicated that the root formation was Lead Trail 4 formation, while every two digits in remained part denoted the formation of units, respectively.

An effective genetic code should be easily one-to-one decoded. The hierarchical method conformed to this rule. Supposing the whole formation was flying in a same altitude, the space between each aircraft was 2 km in x and y direction, respectively. Then the comparative coordinates for each aircraft could be concluded. Given an absolute coordinate to one aircraft in the formation, coordinates for all aircrafts could be gained.

3.3 Optimization based on SAGA

Simple GA owns fine ability for global search. But it is not so well in local search that premature often happened. So simple GA is usually combined with other algorithms of best local search ability. Here Simulated Annealing (SA) algorithm was introduced to improve it. In each generation, “sub-optimization” was implemented with SA before the GA operations, i.e. selection, crossover and mutation. This sub-optimization corresponded to an equilibrium procedure in constant temperature. SA operated on every individual. As the generation increased, the temperature reduced, until convergence. The optimization procedure was shown in Fig.4. Note that, module “Evaluation” meant combat simulation and evaluation of the outcome of engagement.

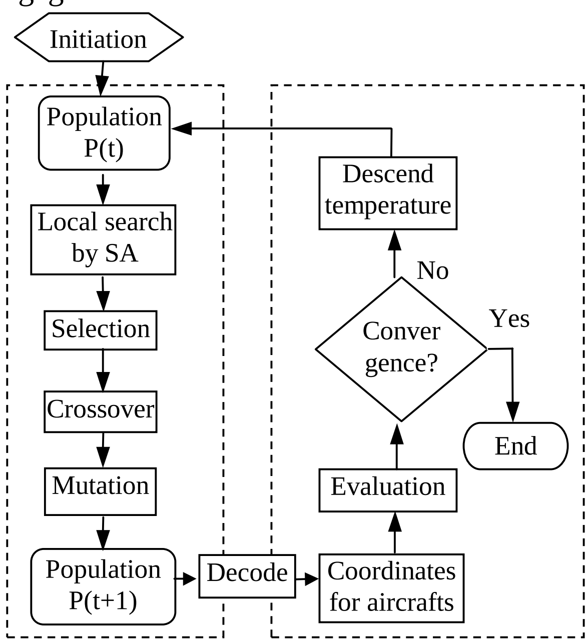

Fig.4. Procedure for formation optimization

In SA algorithm, some key points needed to be emphasized, that was:

- State function

In the equilibrium procedure of constant temperature, a new state is acquired by slightly disturbing the current state. A state function is defined to decide whether this new state is accepted. According to the Metropolis rule in SA, fine points are always accepted and bad points are partly accepted by probability, which was described as [7]:

$$P_t(i \Rightarrow j) = \begin{cases} 1 & , \quad f_j \le f_i \\ \exp(\dfrac{f_i - f_j}{T}) & , \quad f_j > f_i \end{cases} \qquad (2)$$

Where, P_t denotes the probability of replacing state i with state j; f_i and f_j are the function value of target in state i and j, respectively; T denotes the current temperature.

Yet in most cases, the state function proposed by Ref.8, shown in the follows, had higher efficiency and better ability of optimization than the regular one proposed by [7].

$$P_t(i \Rightarrow j) = \begin{cases} 1 & , \quad f_j \le f_i \\ 0 & , \quad f_j > f_i \end{cases} \qquad (3)$$

- Initial temperature

Initial temperature should be not too high and not too low, so that efficiency and optimizing ability can be improved. Here a method suggested by Ref.9 was adopted. One could estimate the best value f_b and worst value f_w of target function, and defined a probability p_a to accept the bad points, and then the initial temperature was as equation (4):

$$T_0 = (f_b - f_w)/\ln p_a \qquad (4)$$

- Function of descending temperature

As the control parameter in SA, Temperature T must be reduced slowly avoiding Markov chain too long. In present work, the following function was adopted:

$$T_{k+1} = \mu \cdot T_k \,, \quad k = 0,1,2,\cdots \qquad (5)$$

Where, μ denotes the descending coefficients, which usually valued with 0.5~0.99. According to this function, temperature would be reduced slower and slower as the optimization going on, which was helpful to gain better stability.

- State producer

In SA algorithm, state producer is used to disturb current state slightly for acquiring a new state while holding the temperature constant. In the case of formation optimization, it meant little change the current formation to get a new one. In GA, this was reflected in the change of codes. Yet was the root formation changed, the combat result must change greatly. In the contrast, change in formation units would only affect combat condition a little. So a unit was randomly chosen from current formation, then replaced with one selected in unit code table.

- Self-fitted Markov chain

As the optimization going on, the optimum was closer. The iteration time needed to achieve balance in constant temperature was shortened gradually. So it need not to always maintain the same number of iterative steps in different temperature. That was to say, the length of Markov chain should not be constant. Firstly we designed manmade Markov chain as following:

$$M = M_0 \cdot \exp\left(-(T_0 - T)/T_0\right) \qquad (6)$$

Where, M and M_0 denotes the length of Markov chain in current temperature and in initial temperature, respectively, while T and T_0 is the current temperature and initial temperature separately. Such an alterable Markov chain was better than a fixed one in optimization ability. Yet it could not timely fit the converging trend.

Therefore, to control the optimization procedure of SA intelligently, a self-fitted Markov chain was used. Given a threshold of chain length M_s, if the optimum in the same temperature had not been changed for M_s times, iteration was interrupted in this condition. By this means, it was realized that the optimization procedure was dominated by the converging condition itself.

4 Results and Analysis

4.1 Formation Optimization

Given a population size of 80 and generation of 40 for GA, the crossover and mutation probability was endowed 0.6 and 0.001 separately. The coefficient μ in equation (5) was evaluated 0.95, and the initial length of Markov chain was 40. Two sides in engagement both owned 16 fighters, which were same in their combat capability. When the red team adopted two formations, the corresponding optimal formations of blue team were shown in Fig.5 and Fig.6, respectively. The outcome of combat was listed in table 2 and 3. From Fig.5 it could be concluded that, when adopting root formation of LT4, the blue team could win the red team 1, which was accord with the results given by Mulgund [1,2]. It showed that better engagement performance of fighter team could be obtained by optimizing large-scale air-combat formation tactics using our SAGA.

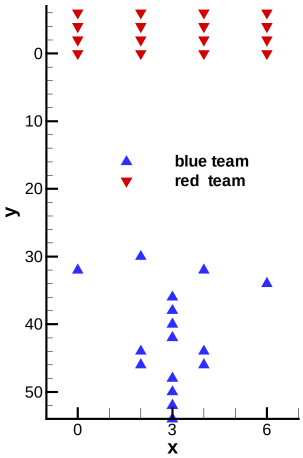

Fig.5. The optimal formation for blue team confronting red team 1

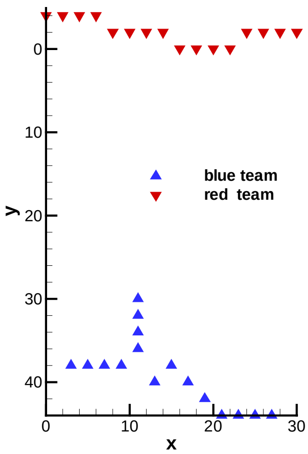

Fig.6. The optimal forma-tion for blue team confronting red team 2

Table 2. The engagement outcome according to Fig.5

Evaluation criterion	Blue	Red
Casualties	2	11
Average living probability	0.37	0.44

Table 3.The engagement outcome according to Fig.6

Evaluation criterion	Blue	Red
Casualties	4	15
Average living probability	0.40	0.80

4.2 Analysis for Computation Ability

Taken formation optimization as example, the computation ability of SAGA was investigated.

4.2.1 Comparison SAGA with Simple GA

Using same parameters with section 3.1, the calculation was performed through Simple GA and SAGA, respectively. Performing trial calculation 50 times, the outcome was listed in table 4:

Table 4. Computation statistics of two algorithms

The algorithm	The simple GA	SAGA
The times converging on the global optimum	8	50
The times converging on the local optimum	42	0
The probability that converging on the local optimum	16%	100%

The investigation above showed preliminarily the effectiveness of our algorithm. It could also be concluded that the convergence performance was markedly improved by using simulated annealing GA.

4.2.2 Investigation of different SAGA

To explore an efficient and robust SAGA, the comparisons of different Markov chains and different state functions were made.

1) Using Fixed Markov Chain (FMC), Manmade Markov Chain (MMC), and Self-fitted Markov Chain (SMC) respectively, state function in formula (2) was compared with that in formula (3). The converging history of average fitness was shown in Fig.7, 8 and 9. Through these three figures above, it was obvious that when FMC and MMC were combined with state function in formula (3), the converging performance of SAGA could be greatly improved. Yet it was not the same to

SMC. State function (3) did not bring SMC obvious improve.

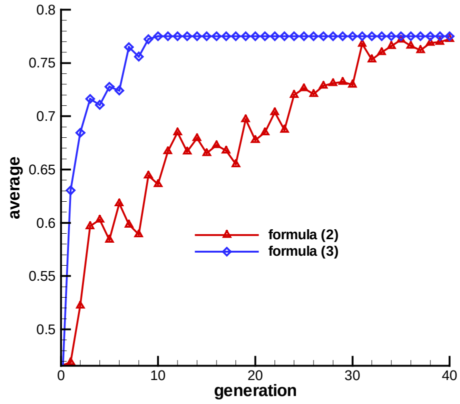

Fig.7. Comparison of two different state functions on condition of FMC

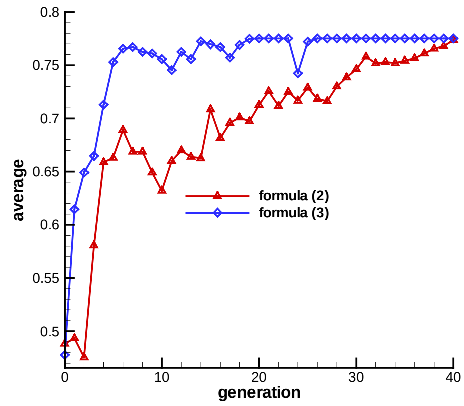

Fig.8. Comparison of two different state functions on condition of MMC

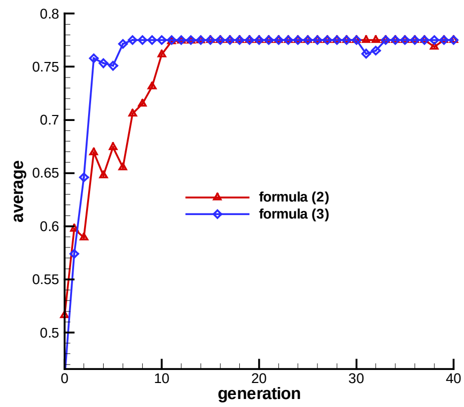

Fig.9. Comparison of two different state functions on condition of SMC

2) Using state function (2) and (3), FMC, MMC and SMC was compared with each other. The converging history of average fitness was shown in Fig.10 and 11.

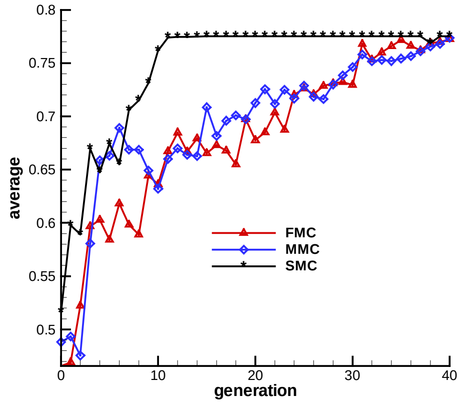

Fig.10. Comparison of three different Markov chains on condition of state function (2)

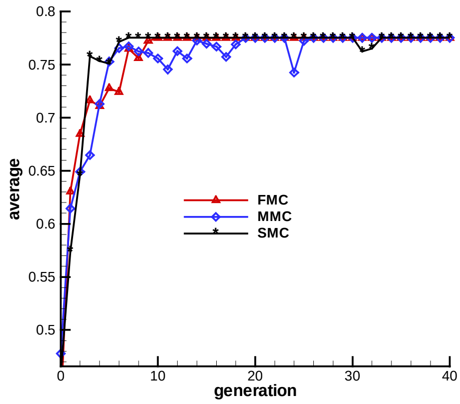

Fig.11. Comparison of three different Markov chains on condition of state function (3)

One could see, from these figures, whether state function (2) or (3) was adopted, SAGA using SMC always won the best converging capability. SAGA, when combined with FMC or MMC, was almost the same in optimization ability. Through the calculation experiments, even FMC was a little better than MMC with state function (3). However the number of iteration steps of MMC was much smaller than that of FMC, which had a proportion of 3620:24820.

5 Conclusions

This paper mainly investigates on two aspects below:

1) Implementation of large-scale air-combat formation optimization
2) Exploration of efficient SAGA

Through these investigations, it can be concluded that:

1) Better engagement performance of fighter team could be obtained by optimizing large-scale air-combat formation tactics using SAGA.
2) The convergence performance was markedly improved by using SAGA.
3) Combination of self-fitted Markov chain and the state function of equation (2) were more appropriate for large-scale air-combat formation optimization using SAGA.

References

[1] Mulgund S, Harper K, Krishnakumar K and Zacharias G. Air Combat Tactics Optimization using Stochastic Genetic Algorithms. *Proc. of 1998 IEEE Intl. Conference on Systems, Man, and Cybernetics*, San Diego, CA, pp 3136-3141, 1998.

[2] Mulgund S, Harper K, Krishnakumar K and Zacharias G. Large-Scale Air Combat Tactics Optimization Using Genetic Algorithms. *AIAA Journal of Guidance, Control and Dynamics*, Vol.24, No.1, pp 140-142, 2001.

[3] Krishnakumar K, Swaminathan R, Garg S and Narayanaswamy S. Solving Large Parameter Optimization Problems using Genetic Algorithms. *Proc. of the Guidance, Navigation, and Control Conference*, Baltimore, MD, 1995.

[4] Zhu B.L, Zhu R.C and Xiong X.F. *Evaluation of combat effectiveness of aircrafts*. Beijing: Aeronautical industry publishing company, 1993.

[5] Li L.S, Tong M.A. Air combat decision of cooperative multi-target attack and its neural net realization. *Acta Aeronautica Et Astronautica Sinica*, Vol.20, No.4, pp 309-312, 1999.

[6] Lan W.H, Zhao C.L. Attacking strategy study on multi-fighters attacking cooperatively on multi-targets in air-to-air combat. *Fire Control & Command Control*, Vol.24, No.1, pp 30-33, 1999.

[7] Kang L.S, et al. *Non-numerical parallel algorithm: simulated annealing algorithm*. Beijing: Scientific publishing company, 1994.

[8] Dueck G, Scheuer T. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing [J]. *J Compute Phys*, Vol.90, pp 161-175, 1990.

[9] Wang L, Zheng D.Z. Unified framework for neighbor search algorithms and hybrid optimization strategies. *Journal of Tsinghua University*, Vol.40, No.9, pp 109-112, 2000.