CrossMark

Applied Mathematics and Nonlinear Sciences 1(1) (2016) 65–78

Applied Mathematics and Nonlinear Sciences

http://journals.up4sciences.org

LAV Path Planning by Enhanced Fireworks Algorithm on Prior Knowledge

Li Bing[1] †, You Ning[1] , Du YeHong[2]

1. Science and Technology on Complex Land Systems Simulation laboratory, Beijing Institute of Special Electromechanical Technology, Beijing 100012, China
2. Ordnance Test Center, Baicheng 137001, China

Submission Info

Communicated by Juan L.G. Guirao
Received 7th June 2015
Accepted 9th November 2015
Available online 1st January 2016

Abstract

Path planning plays an extremely important role in the design of LAVs (Loitering Air Vehicles) to accomplish the air combat task fleetly and reliably. The planned path should ensure that LAVs reach the destination along the optimal path with minimum probability of being found and minimal consumed fuel. Traditional methods tend to find local best solutions due to the large search space. So it takes a lot of time and consumes a lot of computing resources. In this paper, a new young intelligent algorithm-fireworks algorithm is introduced, and EFWA (enhanced fireworks algorithm)-its enhanced version is used to find the optimal solution. At the same time, the battlefield prior knowledge is fully utilized to predict the existence space of the potential optimal trajectory. Greatly the search space reduced, plan planning efficiency is significantly improved. Path planning method effectiveness in this paper has further been improved compared with FAC-PSO. Moreover, the EFWA on prior knowledge performs well on the application of dynamic path planning when the threats cruise randomly than FAC-PSO.

Keywords and phrases: FWA; Threat Degree; FAC-PSO; Search Space
2010 Mathematics Subject Classification: 68T42

1 Introduction

LAV is a new kind of aircraft, which can cruise in the airspace over the targets, whose development has benefited from the rapid development of UAV (unmanned aerial vehicle) and missile technology. The path planning of LAV is to generate a space path between an initial safe location and the desired dangerous destination that has an optimal or near-optimal performance under specific constraint conditions. It is always a complex research

†Corresponding author.
Email address: binlin_ lee@163.com.

ISSN 2444-8656 doi:10.21042/AMNS.2016.1.00006

subject, so it is an imperative technology required in the design of LAV. Series of algorithms have been proposed to solve similar complicated multiconstrained optimization problem of UAV and UCAV. Yudong Zhang et al. [1] introduced these methods, and described their advantages and disadvantages, such as GA (Genetic Algorithms) [2, 3], ACO (Artificial ant colony algorithm) [4], ABC (Artificial bee colony algorithm) [5, 6], PSO (Particle swarm optimization) [7]. After that, some scholars still used a modified form of these algorithms to handle the UCAV Path Planning, and the performances had corresponding been increased, such as PSO [8]- [13], ABC [14], ACO [15].

Yudong Zhang et al. [1] improved PSO algorithm by the way of introducing the chaotic random number and adaptive parameters, and Adaptive Chaotic PSO (FAC-PSO) algorithm is proposed, path planning efficiency had improved significantly, and reduced the time consuming. The authors in [16]– [18] improved A * algorithm, making it more suitable for handling the problem in their own papers. Some academics have also introduced newer and younger intelligent algorithms for route planning. For example, the authors in [19, 20] used firefly algorithm for unmanned vehicles path planning. Compared with other intelligent algorithms such as GA, ACO, PSO, this algorithm has a better performance in searching the optimal solution.

In our pre-work [21]– [23], we modeled, analyzed and researched the related content of LAV swarm cooperative combat. The LAVs cooperative combat needs real-time or near real-time path planning. This article is based on this purpose, and strive to further improve the efficiency of route planning for LAVs cooperative combat to provide support.

For now, one way to solve the problem of slow route planning is to improve the performance of intelligent optimization algorithms. Therefore, this paper uses a younger FWA which has more excellent optimization performance. In 2010, Y TAN et al. [24] put forward a algorithm called FWA(fireworks algorithm), received widespread attention in the field of swarm intelligence optimization. They improved FWA in the subsequent years continuously [25]– [30], and performance of FWA was continuously improved. Therefore, this paper introduces FWA, expecting to further reduce the time-consuming in path planning.

Actually, computing speeds of the most intelligent optimization algorithms are closely related to the optimization space of the population, and large optimization space results in large consumption of computing resources, while significantly increasing the time consumption. So is FWA. Within a known space to find the optimal or suboptimal path, therefore, making full use of priori knowledge of a known space to limit the optimization space, will further reduce the waste of time. In this paper, by using a priori battlefield information to generate the optimization space of fireworks algorithm, the computational efficiency of path planning is improved significantly.

2 Fireworks Algorithm

2.1 Conventional Fireworks Algorithm

As shown in the original FWA paper [24], FWA outperformed SPSO and CPSO significantly and converged in most cases towards the function optimum already after a few iterations. However, when applying FWA on shifted functions the results worsen progressively with increasing distance between function optimum and origin of the search space. By investigating the operators of FWA, it is found that conventional FWA has the following drawbacks:

(1)For functions which have their optimum at FWA will find the optimal solution very fast. However, not due to the intelligence of the algorithm but due to the specific mapping and Gaussian mutation operators which map/create sparks close to the origin;

(2) For functions which have their optimum far away from the origin, FWA has to face the two drawbacks that the mapping operator rebounds most solutions which are out of the search space to locations which are far away from the function optimum, and that the mutation operator creates many sparks at locations close to the origin (for example, again far away from the optimum).

(3) FWA has a high computational cost per iteration.

2.2 Enhanced Fireworks Algorithm

As shown in [25], Y Tan et al. made 5 major changes in FWA and got the EFWA. The 5 changes are shown as follows.
(1) A new Minimal Explosion Amplitude Check;
(2) A new Operator for Generating Explosion Sparks;
(3) A new Mapping Operator;
(4) A New Operator for Generating Gaussian Sparks;
(5) A new Selection Operator;
EFWA is a significant improvement of the recently developed FWA. And it eliminates the drawbacks of conventional FWA. As is shown in the experimental evaluation, with the exception of one benchmark function, the results of EFWA are very stable and remain almost unaffected even if the optimum of the function is shifted towards the edges of the search space. EFWA shows significant improvements over conventional FWA. Compared to SPSO, which turned out to be rather sensitive to increasing shift values, EFWA achieves very stable results, and has the advantage that its results do not deteriorate even for large shift values. In terms of computational cost, the new selection operator is faster by a factor of 6 compared to the distance based selection operator of conventional FWA.
TAN also proposed AFWA (an adaptive version of FWA) [29] and dynFWA(dynamic version of FWA) [30], AFWA improvement is similar with FAC-PSO, and dynFWA improvement is also similar with other intelligent algorithm. Although adaptive algorithm can improve computational efficiency to some extent, but to significantly improve the efficiency of path planning, takes full advantage of a priori knowledge of the battlefield environment, directly reduce the optimization EFWA space. Therefore, this article uses EFWA, and apply the prior knowledge of the battlefield environment to determine its optimization space.

3 Optimization Space Determination

3.1 Path and Threats Coding

We use the same path coding in the literature [1], as shown in Figure 1. In this way, the path from the starting node P to the target node P_f can be described as follows:

$$Path = \{P_s, P_1, P_2, \cdots, P_n, P_f\} \tag{1}$$

Therefore, each point$P_i(i = 1, 2, \cdots, n)$can be expressed using only 1 parameter,its distance to the straight line $\overline{P_sP_f}$.The benefits of this kind of path coding is that two-dimensional problem can be represented with a one-dimensional variable to facilitate the problem model solving.

Optimization space involves determining the threat weight calculations. To facilitate the presentation, Let T_i denote the threat i, and set its data structure as:

$$T_i = \{T_i^P, T_i^R, T_i^D, T_i^V, T_i^{TP_sP_f}, T_i^{EIP}, T_i^{EDTP_sP_f}\} \tag{2}$$

Where T_i^P denotes the position of T_i, TR_i , denotes the radius of T_i, T_i^D , denotes the threat degree of T_i, T_i^V , denotes the velocity of $T_i, T_i^{TP_sP_f}$, denotes the distance between T_i and $\overline{P_sP_f}$, T_i^{EIP} denotes the ability of influencing the path of T_i , 1 denotes it can, -1 denotes it can not, $T_i^{EDTP_sP_f}$ denotes the effective threat degree of T_i to $\overline{P_sP_f}$, and $\overline{P_sP_f}$ denotes the line between P_f and P_s. $T_i^P, T_i^R, T_i^D, T_i^V$ is the information already known in the

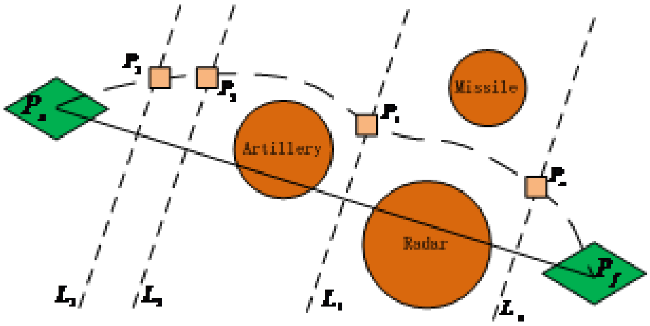


Fig. 1 Typical 2D LAV battlefield model

battlefield, T_i^{EIP} is initialized to -1. $T_i^{TP_sP_f}$ can be calculated with a simple geometric method:

$$T_i^{TP_sP_f} = \begin{vmatrix} P_f - P_s \\ TP_i - P_s \end{vmatrix} / \left\| P_f - P_s \right\| \tag{3}$$

The calculation of $T_i^{EDTP_sP_f}$ is as follows:

$$T_i^{EDTP_sP_f} = \left[T_i^D \cdot S_u / S_i \;\; T_i^D \cdot S_d / S_i \right] \tag{4}$$

Here, S_u and S_d denote the effective threat area of T_i up and down $\overline{P_sP_f}$ respectively. S_i is the area of T_i, and the method of calculating $\left[S_u \; S_d \right]$ is as Formula (5).

$$\left[S_u \; S_d \right] = \begin{cases} \left[S_u = S_i - S_d \;\; S_d = \cos^{-1}\left(\left| T_i^{TP_sP_f} / T_i^R \right| \right) \cdot \left(T_i^R\right)^2 - T_i^{TP_sP_f} \cdot \sqrt{\left(T_i^R\right)^2 - \left(T_i^{TP_sP_f}\right)^2} \right], \\ \qquad if \left| T_i^{TP_sP_f} \right| < T_i^R \text{and} T_i^{TP_sP_f} \geq 0 \\ \left[S_u = \cos^{-1}\left(\left| T_i^{TP_sP_f} / T_i^R \right| \right) \cdot \left(T_i^R\right)^2 - T_i^{TP_sP_f} \cdot \sqrt{\left(T_i^R\right)^2 - \left(T_i^{TP_sP_f}\right)^2} \;\; S_d = S_i - S_u \right], \\ \qquad if \left| T_i^{TP_sP_f} \right| < T_i^R \text{and} T_i^{TP_sP_f} < 0 \\ \left[S_i \cdot \left(T_i^R\right)^2 \Big/ \left(T_i^{TP_sP_f}\right)^2 \;\; 0 \right], if\ T_i^R \leq \left| T_i^{TP_sP_f} \right| \leq 3 \cdot T_i^R \text{and} T_i^{TP_sP_f} \geq 0 \\ \left[0 \;\; S_i \cdot \left(T_i^R\right)^2 \Big/ \left(T_i^{TP_sP_f}\right)^2 \right], if\ T_i^R \leq \left| T_i^{TP_sP_f} \right| \leq 3 \cdot T_i^R \text{and} T_i^{TP_sP_f} < 0 \\ \left[0 \; 0 \right], if \left| T_i^{TP_sP_f} \right| \geq 3 \cdot T_i^R \end{cases} \tag{5}$$

3.2 Calculation of the potential side of the optimal path

$T^{TTL}$of the threats of upper and lower sides to $\overline{P_sP_f}$ can calculated as formula(6).

$$T^{TTL} = \sum_{i=1}^{n} T_i^{EDTP_sP_f} \tag{6}$$

Then the potential side P^{PD} of the optimal path is,

$$P^{PD} = sign\left(T^{TTL}(1,1) - T^{TTL}(1,2) \right) \in \{1, -1\} \tag{7}$$

Here, the value of *sign* is 1 when the independent variable is non-negative, and is -1 when the independent variable is negative. 1 denotes that the optimal path exists on the upper side of $\overline{P_sP_f}$, then -1 denotes the optimal path exists on lower side of $\overline{P_sP_f}$. In order to more intuitively describe, with the data in Table 2 in literature [1], $P^{PD} = -1$ can be calculated, that is that the optimal path is on the lower side of $\overline{P_sP_f}$ as shown in Fig.2.

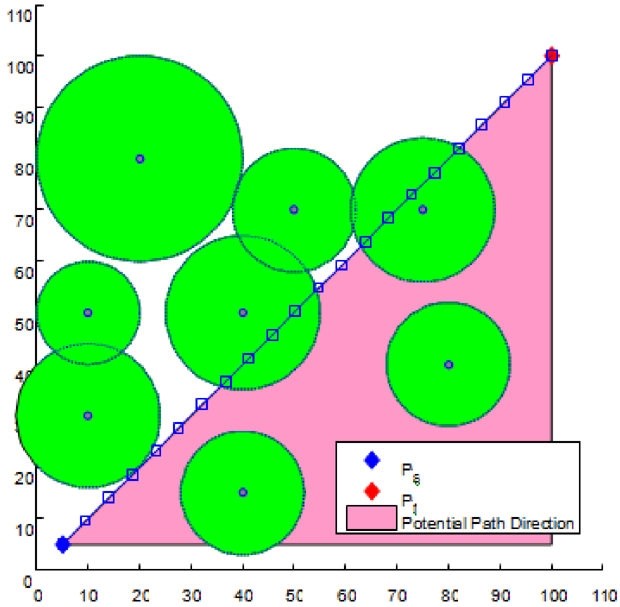


Fig. 2 The potential side of the optimal path

3.3 Upper and lower initial limits of the population

Known the potential side of optimal path, the upper and lower limits of the initial population can be calculated. Let B^I be the upper and lower limits of the initial population, the data structure is as follows,

$$B^I = \left\{ \left(B_i^{IU}, B_i^{ID}, B_i^{IP}\right) | 1 \leq i < n \right\} \tag{8}$$

Where, B_i^{IU} denotes the upper limit of seed i,B_i^{ID} denotes the lower limit of seed I, B_i^{ID} denotes the position locating the upper and lower limit, and $\dim B^I < n$ .The method of calculating B^I is as the pseudo-codes in Fig.3.

Timer=1
Begin Loop in $\{T_i\}$
  If $\left|T_i^{TP_sP_f}\right| < T_i^R$
  $T_i^{EIP} = 1;$
  $B_{Timer}^{ID} = T_i^{TP_sP_f} + P^{PD} \cdot T_i^R;$
  $B_{Timer}^{IU} = B_{Timer}^{ID} + P^{PD} \cdot \Delta L$
  $B_{Timer}^{IP} = \sqrt{\left\|T_i^P - P_s\right\|^2 - \left(T_i^{TP_sP_f}\right)^2}$
Timer++;
Else If $T_i^R \leq \left|T_i^{TP_sP_f}\right| \leq 3 \cdot T_i^R$ and $T_i^{TP_sP_f} \cdot P^{PD} > 0$
  $T_i^{EIP} = 1;$
  $B_i^{IU} = T_i^{TP_sP_f} - P^{PD} \cdot T_i^R;$
  $B_i^{ID} = B_i^{IU} - P^{PD} \cdot \Delta L;$

$B^{IP}_{Timer} = \sqrt{\left\|T_i^P - P_s\right\|^2 - \left(T_i^{TP_sP_f}\right)^2}$
Timer++;
End If
End If
End Loop

Fig. 3 method of calculating B^I

ΔL in Fig.3 can be calculated as follows,

$$\Delta L = \left\|\overline{P_f P_s}\right\| / (n+1) \tag{9}$$

Based on the results calculated in section 3.2, the upper and lower initial limits of the population can be retained as shown in Fig.4. To facilitate the analysis, it is rotated to the horizontal axis as shown in Fig.5.

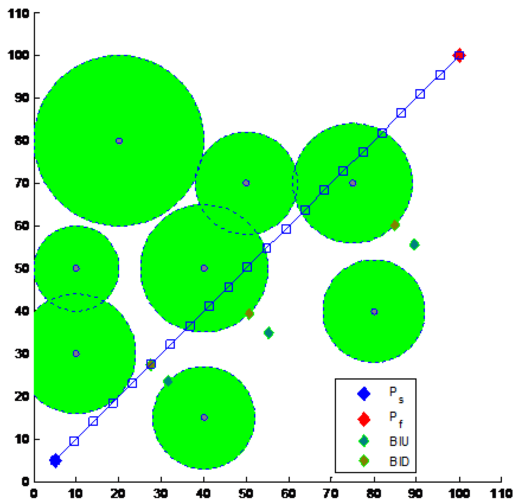


Fig. 4 The upper and lower initial limits of the population

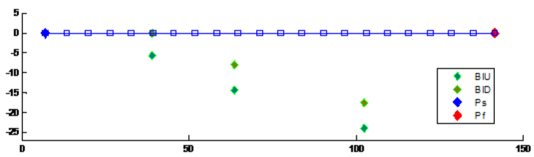


Fig. 5 The upper and lower initial limits of the population shown on the horizontal axis

3.4 The upper and lower limits of the population

After the upper and lower initial limits of the population retained, the path still cannot be optimized with EFWA. Upper and lower limits of the population on each dimension B_i need to be calculated according to the initial limits. The structure of B_i is set as follows.

$$B_i = \left\{B_i^U, B_i^D\right\}, i = 1, 2, \cdots, n \tag{10}$$

Here, B^U denotes upper limit, B^Ddenotes lower limit, $dimB = n$. The method of calculating B is as pseudo-codes in Fig.6.

(1)Declare variables Timer1=1;
Declare temporary matrix $B_Timer^T = []$
Declare the upper and lower limits matrix $B = zeros(2,n)$;
(2)According to the third row of B^{IP} values in ascending matrix B^I ,
that is $B^I = sortrows(B^I,3)$;
(3) Begin Loop
Calculating the pointP_j on $\overline{P_sP_f}$ close to the every element of B^I ;
Set B_j as the values on position corresponding of $B_i{}^I$; ;
$B^T_{Timer1} = \left(B_i^{IU}, B_i^{ID}, j\right)$
$B_j = \left(B_i^{IU}, B_i^{ID}\right)$
Timer++;
End Loop
(4)After linear interpolation ofB^T , the linearized upper and lower limits
of the population B are generate.

Fig. 6 Calculation of B

Based on the retained results in section 3.3, the transition results are got from step 1 to step 3 in Fig.6, and are shown in Fig.7. Then linear interpolation completed by step 4, the final results of upper and lower limits are retained, and are shown in Fig.8.

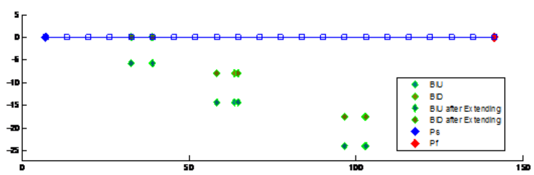


Fig. 7 The transition results of upper and lower limits

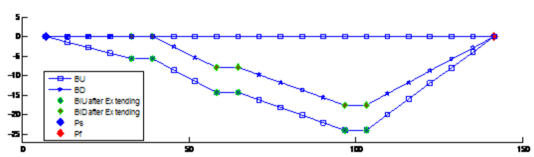


Fig. 8 The final results of upper and lower limits between two path points

3.5 Iterative calculation of upper and lower limits of the population between start point and target point

o increase the probability that final upper and lower limits cover the optimal path, numbers of iterative calculations need to be done between start point and target point, and the final upper and lower limits are retained in which contains the optimal path the probability is highest. That is, optimization space is got finally. Iterative calculation process is as shown in Fig.9.

Table 1 Information of 2D threatening objects

Index	Position	Radius
1	(10,30)	14
2	(10,50)	10
3	(20,80)	20
4	(40,50)	12
5	(45,50)	15
6	(50,70)	12
7	(75,70)	14
8	(80,40)	12

Table 2 Average computation time(s)

n	FAC-PSO	EFWA on Prior Knowledge
10	10.2	0.8
15	11.3	1.2
20	13.7	1.5

(1) Calculating the optimization space B^0 between start point and target point, combining the mean of upper and lower limits on each dimension to form a seed, calculating its fitness value Fit_0 , and recording the optimization space.
(2)Look for m segments of longer intervals paths on $\overline{P_sP_f}$where there is no threat to cover. Set the middle point of every segment as the temporary target point and temporary start point, calculation as section 3.1 to section 3.3 in the $(m+1)$ segments, and combine the results, record the fitness value and upper and lower limits correspondingly. The selection of m and length of path are due to calculation time and dimensions of problem.
(3)Select the optimization space where there is the minimum fitness value as the upper and lower limits between start point and target point.

Fig. 9 The iterative calculation pseudo-codes of optimization space between the start point and target point

4 Experiment

4.1 Comparative Experiment

Because of use the path coding in [1], in order to facilitate comparison, we use the same objective function and simulation settings in [1] too. Programming Environment is MATLAB 2014b. CPU in our computer is P4, its frequency is 2.8GHz, and memory is 2G.

The results of path planning are as shown in 10, and the data of consuming time are as shown in 4.1.

4.2 Experiment under random static threats condition

The number of threats is still 8, set the threat positions as $T_i^P \in [10,80]$and the threat radius as $T_i^P \in [10,15]$, and set the dimension n of path as 30. Six groups of representative results are shown in Figure 11.

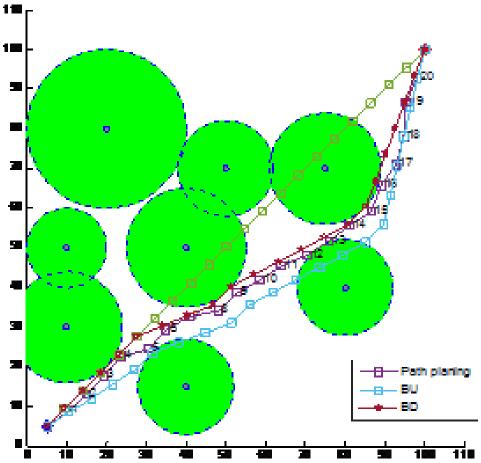
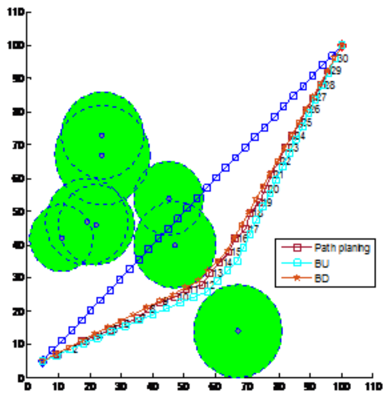
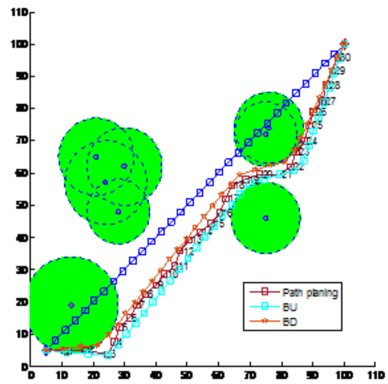


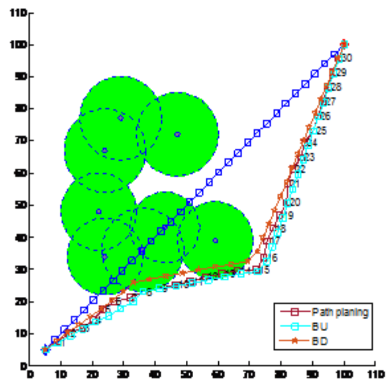
Fig. 10 Path planning by EFWA when n=20



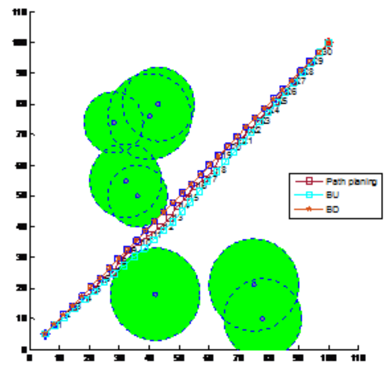
(a)



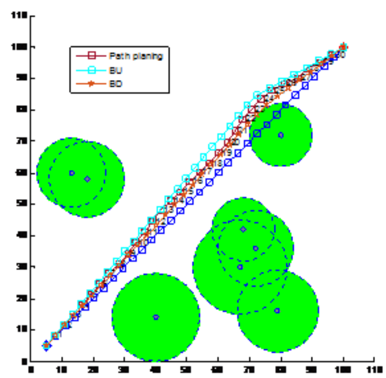
(b)



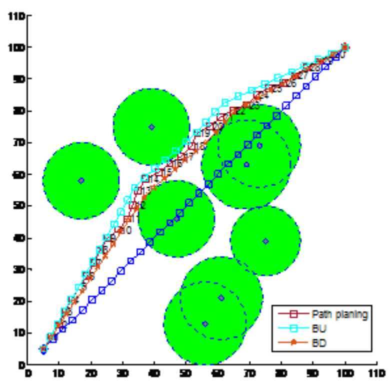
(c)



(d)



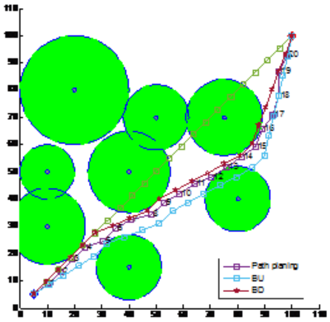
(e)



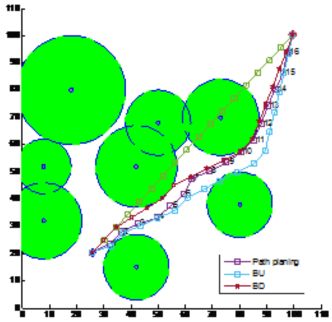
(f)

4.3 Dynamic Path Planning

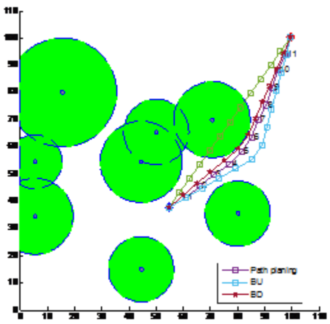
The setting of dynamic path planning is still the same as [1], and set the dimension as 20. To improve the performance of the path planning, the iterative number is increased to 300. The LAV paths by the EFWA on Prior Knowledge at steps 0, 5, 10, and 15 are shown in Fig.12, which implies the feasibility of EFWA on Prior Knowledge under moving threatening conditions. The time consuming of each step is shown in Table 4.



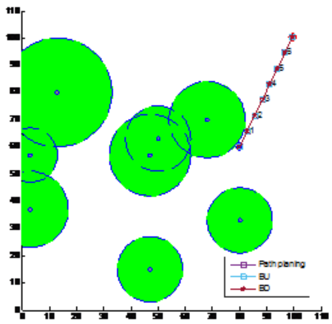
(a)



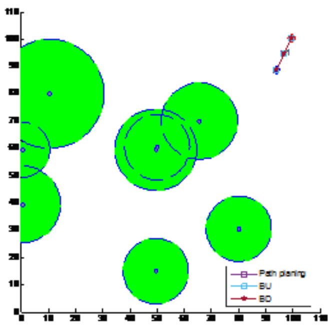
(b)



(c)



(d)



(e)

Fig. 12 All threatening obstacles are moving for dynamic path planning: (a) step 0; (b) step 5; (c) step 10; and (d) step 15.

Table 3 time consuming of each step(s)

Step	Time	Step	Time
1	3.0	11	1.5
2	2.7	12	1.4
3	2.9	13	1.3
4	2.7	14	1.6
5	2.6	15	1.4
6	2.3	16	1.4
7	2.1	17	1.0
8	1.9	18	0.8
9	1.8	19	0.6
10	1.7	20	0.5

5 Discussion

As shown in Table 2, when the settings are the same, the path by EFWA on prior knowledge is close to the one in [1], but the time consuming is an order of magnitude less than that in [1]. The method in this article improve the efficiency of path planning significantly.

In section 4.2, we set the positions and radiuses of threats randomly. The EFWA on prior knowledge is still able to plan out a more reasonable path. That implies the method in this article has good adaptability.

In section 4.3, we give each threat a random initial velocity, and increase the number of iterations appropriately. Figure 12 implies the feasibility of EFWA on Prior Knowledge under moving threatening conditions. Table 4 implies the planning speed is still fast in dynamic environment, and our method has a high real-time.

6 Conclusion

In this article, we introduce a younger FWA which has more excellent optimization performance to deal with the problem of path planning for LAVs. Firstly, we introduce the researches of FWA in recent years briefly. After compare the improved versions of FWA, we choose EFWA.

To improve the efficiency of the path planning for LAVs, we use the prior knowledge of the battle environment sufficiently with the methods in section 3, then retain the optimization space between the start point and target point.

The simulation results show that the proposed Enhanced Fireworks Algorithm on Prior Knowledge excels FAC-PSO algorithm with computation time obviously. We extended our experiment to 2D dynamic path planning, and the results show that the speed of EFWA on Prior Knowledge is faster than FAC-PSO algorithm too. All prove the superiority of Enhanced Fireworks Algorithm on Prior Knowledge. Therefore, it is a feasible and effective way for LAV path planning, and is highly possible to provide support for cooperative combat of LAVs. But, in order to retain better performance of path planning, we still need to continue research in the following areas:(1)To improve the efficiency of path planning and fault tolerance, we need to introduce more rules to calculate the optimization space;(2)Considered the computational efficiency and computational complexity, we use linear interpolation to generate the upper and lower limits, but using non-linear interpolation method to improve the performance of path planning requires further study.

Acknowledgement

This paper is supported by The National Defense Pre-Research Foundation of China (Grant no. B222014XXXX).

References

[1] Zhang Y, Wu L, Wang S. (2013), UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization. Mathematical Problems in Engineering. doi 10.1155/2013/705238

[2] F. C. J. Allaire, M. Tarbouchi, G. Labonte, and G. Fusina, (2009) FPGA implementation of genetic algorithm for UAV real-time path planning, Journal of Intelligent and Robotic Systems, 54, 1-3, 495-510. doi 10.1007/s10846-008-9276-8

[3] Giovanni Giardini and Tamas Kalmar-Nagy.(2011), Genetic Algorithm for Combinatorial Path Planning: The Subtour Problem. Mathematical Problems in Engineering. doi 10.1155/2011/483643

[4] H. Duan, Y. Yu, X. Zhang, and S. Shao, (2010),Three-dimension path planning for UCAV using hybridmeta-heuristic ACO-DE algorithm, Simulation Modelling Practice and Theory, 18(8), 1104-1115. doi 10.1016/j.simpat.2009.10.006

[5] Y. Zhang, L.Wu, and S.Wang,(2011), UCAV path planning based on FSCABC, Information,14(3), 687-692.

[6] Bhattacharjee P, Rakshit P, Goswami I, et al.(2011), Multi-robot path-planning using artificial bee colony optimization algorithm//Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on. IEEE,219-224. doi 10.1109/NaBIC.2011.6089601

[7] Foo J L, Knutzon J S, Oliver J H, et al.(2007), Three Dimensional Multi-Objective Path Planning of Unmanned Aerial Vehicles Using Particle Swarm Optimization[J]. AIAA,(April), 1-10. doi 10.2514/6.2007-1881

[8] Roberge V, Tarbouchi M, Labonte G.(2013), Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, Industrial Informatics, IEEE Transactions on Industrial Informatics publication information, 9(1), 132-141. doi 10.1109/TII.2012.2198665

[9] Holub J, FO L J, Kalivarapu V. (2012),Three dimensional multi-objective UAV path planning using digital pheromone particle swarm optimization//Proc of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 1-10. doi 10.2514/6.2012-1525

[10] Swartzentruber L, Foo J L, Winer E.(2010), Multi-Objective UAV Path Planning with Refined Reconnaissance and Threat Formulations[C]//Proceedings of 6th AIAA Multidisciplinary Design Optimization Specialist Conference, 1-14. doi 10.2514/6.2010-2758

[11] Kenefic R J. (2008), Finding good Dubins tours for UAVs using particle swarm optimization. Journal of Aerospace Computing, Information and Communication, 5(2), 47-56. doi 10.2514/1.35658

[12] Liang J J, Song H, Qu B Y, et al. (2014),Comparison of three different curves used in path planning problems based on particle swarm optimizer. Mathematical Problems in Engineering. doi 10.1155/2014/623156

[13] Ma Y, Zamirian M, Yang Y, et al. (2013),Path planning for mobile objects in four-dimension based on particle swarm optimization method with penalty function. Mathematical Problems in Engineering, 2013, 2013. doi 10.1155/2013/613964

[14] Mansury E, Nikookar A, Salehi M E. (2013),Artificial Bee Colony optimization of ferguson splines for soccer robot path planning[C]//Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on. IEEE, 2013: 85-89. doi ICRoM.2013.6510086

[15] Cekmez U, Ozsiginan M, Sahingoz O K.(2014), A uav path planning with parallel aco algorithm on cuda platform//Unmanned Aircraft Systems (ICUAS), 2014 International Conference on. IEEE, 2014: 347-354. doi 10.1109/ICUAS.2014.6842273

[16] Zhan W, Wang W, Chen N, et al.(2014), Efficient UAV Path Planning with Multiconstraints in a 3D Large Battlefield Environment, Mathematical Problems in Engineering,Volume 2014, Article ID 623156, 15 pages. doi 10.1155/2014/597092

[17] Tseng F H, Liang T T, Lee C H, et al.(2014), A Star Search Algorithm for Civil UAV Path Planning with 3G Communication//Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on. IEEE, 942-945. doi 10.1109/IIH-MSP.2014.236

[18] Mohammadi A, Rahimi M, Suratgar A A.(2014), A new path planning and obstacle avoidance algorithm in dynamic environment//Electrical Engineering (ICEE), 2014 22nd Iranian Conference on. IEEE, 1301-1306.

[19] Brand M, Yu X H. (2013), Autonomous robot path optimization using firefly algorithm//Machine Learning and Cybernetics (ICMLC), 2013 International Conference on. IEEE, 3, 1028-1032. doi 10.1109/ICMLC.2013.6890747

[20] C Liu, Y Zhao, F Gao, L Liu.(2014),Three-Dimensional Path Planning Method for Autonomous Underwater Vehicle Based on Modified Firefly Algorithm, Mathematical Problems in Engineering Volume 2015, Article ID 561394, 10 pages. doi 10.1155/2015/561394

[21] Bing L, Jie L, KeWei H.(2013), Modeling and flocking consensus analysis for large-scale UAV swarms, Mathematical Problems in Engineering, Volume 2013, Article ID 368369, 9 pages. doi 10.1155/2013/368369

[22] Bing L, Jie L, Guanglin H, et al. (2014),Research on Cooperative Combat for Integrated Reconnaissance-Attack-BDA of Group LAVs, Mathematical Problems in Engineering, Volume 2014, Article ID 123142, 6 pages. doi 10.1155/2014/123142

[23] Li Jie, You Ning, Li Bing et al.(2015), Cloud Ammunition System Acta Armanentarii, 36(2), 250-254.
[24] Tan, Ying, and Yuanchun Zhu. (2010), Fireworks algorithm for optimization. Advances in Swarm Intelligence. Springer Berlin Heidelberg, 2010, 355-364. doi 10.1007/978-3-642-13495-1_44
[25] Pei Y, Zheng S, Tan Y, et al.(2012), An empirical study on influence of approximation approaches on enhancing fireworks algorithm//Systems, Man, and Cybernetics (SMC), IEEE International Conference on. IEEE, 1322-1327. doi 10.1109/ICSMC.2012.6377916
[26] Liu J, Zheng S, Tan Y. (2013), The improvement on controlling exploration and exploitation of firework algorithm//Advances in swarm intelligence. Springer Berlin Heidelberg,11-23. doi 10.1007/978-3-642-38703-6_2
[27] Zheng S, Janecek A, Tan Y.(2013) Enhanced fireworks algorithm[C]//Evolutionary Computation (CEC), 2013 IEEE Congress on. IEEE, 2069-2077. doi 10.1109/CEC.2013.6557813
[28] Liu J, Zheng S, Tan Y. (2014),Analysis on global convergence and time complexity of fireworks algorithm//Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 3207-3213. doi 10.1109/CEC.2014.6900652
[29] Li J, Zheng S, Tan Y.(2014), Adaptive Fireworks Algorithm//Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 3214-3221. doi 10.1109/CEC.2014.6900418
[30] Shaoqiu Zheng, Andreas Janecek, Junzhi Li and Ying Tan .Dynamic Search in Fireworks Algorithm// http://www.cil.pku.edu.cn/research/fwa/publication/DynamicSearchInFireworksAlgorithm.pdf. doi 10.1109/CEC.2014.6900485