2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Intelligent Flight for UAV via Integration of Dynamic MOEA, Bayesian Network and Fuzzy Logic

Xingguang Peng, *IEEE Member*, Demin Xu, and Weisheng Yan

Abstract— **Intelligent flight is a key technology for an unmanned aerial vehicle (UAV) to react to the changing environment. Online path planning (OPP) is a basic issue for intelligent flight and is indeed a dynamic multi-objective optimization problem (DMOP). In this paper, we use an OPP scheme in the sense of model predictive control to continuously update the environmental information for the planner. This method is in fact a DMOP. For solving the problem at hand we propose a dynamic multi-objective evolutionary algorithm based on linkage and prediction (LP-DMOEA). Within this algorithm the historical Pareto sets are collected and analyzed to enhance the performance. For intelligently selecting the best path from the output (a set of Pareto solutions obtained by the LP-DMOEA) of the OPP, the Bayesian network and fuzzy logic are used to quantify the bias to each optimal objective. The experimental results show the LP-DMOEA works more effectively for OPP in contrast to the restart method and the intelligent methods for solution selection can automatically assess the changing environment and adapt the path planner.**

I. INTRODUCTION

As an unmanned system, the unmanned aerial vehicle (UAV) has drawn great interest from scientists due to its low cost and danger-endurable characteristics. Depart from some key elements involved in traditional aerial vehicles such as control algorithms, localization and navigation, more emphases have been put on the autonomic and intelligent abilities under uncertain environments. Intelligent flight is the basic issue for UAVs to carry out any complicated mission. When the environment is static and known beforehand the flight path can be well designed offline [1, 2]. However, when the environment is changeable or there is no exact knowledge about the environment, an UAV needs to intelligently plan its path online. To this end, in this paper, we intend to solve the in-telligent online path planning (OPP) problem for UAVs.

In general, path planning involves multiple optimal objectives. For instance, the maximal safety and minimal energy cost are the two common objectives. In some literatures multi-objective optimal problems (MOPs) are transformed to single-objective ones by weighting each objective and sum-ming them up[3, 4]. This requires knowing the bias to objectives of the optimizer beforehand. Unfortunately, sometimes it is difficult to achieve. Therefore, from the nature of the dynamic MOPs (DMOPs), we optimize all the objectives simultaneously. Moreover, considering the fact that no exact information about the environment is known beforehand and the environment may change during the mission, the objectives involved in path planning are time variant. Accordingly, we need a dynamic multi-objective optimal method to deal with the problem at hand.

The evolutionary algorithm (EA) is a stochastic search method. Its problem-independent and global optimal characteristics establish its position in the optimization domain. In the passed decade some methods were proposed to extend EA for solving dynamic optimization problems. For example random immigrants method [5], memory-based method [6, 7], multi-population method [8, 9], etc. Recently, in several works [10-12], the convergence is accelerated by predicting the characteristics of future changes when the behavior of the problem follows a certain trend.

Inspired by these ideas, in this paper, we propose a multi-objective evolutionary algorithm which stores and analyzes the historical information to enhance the perform-ance on DMOPs (i.e. LP-DMOEA). In this algorithm, the historical Pareto solutions are linked to construct several time series, and then a prediction method is employed to anticipate the Pareto set of the next problem. Benefit by such anticipation, the initial population for the new problem could be heuristically generated to accelerate the convergence.

However, the obtainment of Pareto solutions is not the end of a multi-objective optimization problem. One solution should be selected from the Pareto set by decision maker (DM). Since an UAV has no actual DM it should intelligently make such decision without interacting with the human beings. To this end, we employ Bayesian network (BN) to model the inference process of a pilot when he is assessing the dangerous level of environment. In addition, the fuzzy logic is used to quantify the decision bias reference to inference results.

The remaining part of this paper is organized as follows: Section II presents the formulation of the OPP and the details of LP-DMOEA; Section III presents the intelligent decision making methods to choose a Pareto solution. Section IV presents the experimental results and analysis. Conclusions are drawn in Section V.

II. ONLINE PATH PLANNING IN TERM OF DYNAMIC MULTI-OBJECTIVE OPTIMIZATION

Viewing from the practice, an offline global path planning is likely to be invalid when the environment is uncertain and time variant. An UAV has to independently plan its online path referring to the local information detected by its onboard sensors. Pongpunwattana has proposed an OPP scheme in the sense of the model predictive control (MPC) in [13]. As shown in Fig. 1, suppose the UAV has planned a path

This work was supported by National Nature Science Foundation of China under grant 61105068 and 60875071.

The authors are with the School of Marine Engineering, Northwestern Polytechnical University Xi'an 710072, China. pxg0510@gmail.com

starts from point A at t_p. Instead of executing the whole planning result, partial path (i.e. path between A and B) will be executed. While flying from A to B it is planning a new path that starts from B. When it arrives at B (at t_{p+1}) this new path will be used. Of course, part of the new path will be executed. The OPP can be achieved by iteratively executing the steps above and the environmental information can be continuously updated to adapt the planner for changing environment. It is obviously that path between A and B is the executing horizon and the paths planned at A or B are the planning horizon. The optimization problem in each time window could be time variant. Therefore, the online path planning is indeed a type of dynamic optimization problem.

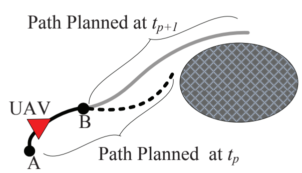


Fig. 1. Illustration of online path planning.

A. Formulation of the OPP Problem

In 2-D case, we formulate the OPP problem as follows.

$$\begin{cases} \min f &= \{f_1(\mathbf{u},t), f_2(\mathbf{u},t)\} \\ f_1(\mathbf{u},t) &= \prod_{i=1}^{n} p_{kill}\left(\mathbf{x}(t) + \sum_{j=1}^{i} g(u_j)\right) \\ f_2(\mathbf{u},t) &= \left\| \mathbf{x}(t) + \sum_{i=1}^{n} g(u_i) - \mathrm{T} \right\|_2 \end{cases} \quad (1)$$

where $u_i \in \mathbf{u}(i = 1, \ldots, n)$, $\mathbf{u}$ is the sequential control input of an UAV from t to $t + \Delta T_s(\Delta T_s = n \times \Delta t)$, $g(u_i)$ denotes the Euclid deviation of the UAV caused by the control input u_i. $\mathbf{x}(t)$ and $\mathbf{T}$ are the vectors of position values of UAV and destination respectively. $p_{kill}(\mathbf{x}(t))$ means the probability of being destroyed. So the first optimal objective is to minimize the destroy probability when the UAV fly along the planning path. The second optimal objective is easy to understand, it aims at minimizing the distance between UAV (at the termination of a path segment) and its goal. For facility of simulation, we use probabilistic threat exposure map [14] to model the battle field. The probability of becoming disabled by the i-th threat is characterized by the multidimensional Gaussian law.

$$p^i_{kill}(t) = \frac{1}{2\pi\sqrt{\det(\mathbf{K}_i)}} \exp\left[-\frac{1}{2}(\mathbf{x}(t) - \mu_\mathbf{i})^T \mathbf{K}_i^{-1} (\mathbf{x}(t) - \mu_\mathbf{i})\right] \quad (2)$$

where $\mu_\mathbf{i} = [\mu_{x,i}, \mu_{y,i}]$, $\mathbf{K}_i = \begin{bmatrix} \sigma^2_{x,i} & 0 \\ 0 & \sigma^2_{y,i} \end{bmatrix}$. $\mu_\mathbf{i}$ denotes the position of a threat and $\mathbf{K}_i$ determines the its acting range.

B. Problem Solving Approach: The LP-DMOEA

In order to deal with the dynamic multi-objective optimization problem at hand, we propose a dynamic multi-objective EA using Pareto set linking and prediction (i.e. LP-DMOEA). The main idea of LP-DMOEA lies in heuristically generating the initial population for new problem by making prediction from the historical information. Historical Pareto solutions are linked to construct several time series, and then a prediction method is employed to anticipate the Pareto set of the next problem. At last, the initial population for the new problem could be heuristically generated to accelerate the convergence.

As shown in Fig. 2, suppose environment changes (new problem arrives) at t, the Pareto set $S_p(t)$ in offspring $O(t)$ is used to generate the next population $P(t+1)$. There are three major steps in the LP-DMOEA. Firstly, LP-DMOEA will select some representative Pareto solutions $S'_p(t)$ from $S_p(t)$ considering both computational complexity of prediction and diversity of Pareto front. Secondly, $S'_p(t)$ and its historical counterparts are used to construct several time series **TS**. At last, an anticipating method will be applied to **TS** to predict the location of the new Pareto set, and the initial population could be generated base on the prediction results.

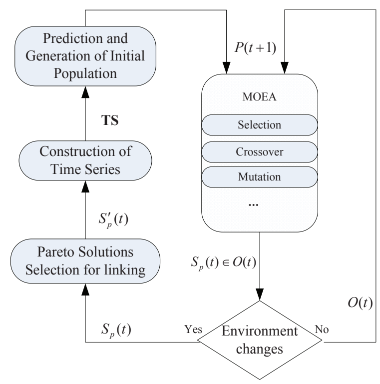


Fig. 2. Follow chart of the LP-DMOEA.

1) Selecting Pareto Solutions for Linking: As for selecting Pareto solutions for linking there are two major ideas in the literature. The first one accomplishes this by taking the feature of each candidate in objective space. In [10] the anchor points and closest-to-ideal point of a Pareto front in objective space are chosen as key feature points, and the corresponding solutions in decision space are used to make anticipation. This approach uses two or three key points to characterize the Pareto front. However, this approach may be invalid when the front is concave or very complex. As for the other idea, the candidates are selected directly in decision space under special principle. In [12] all the Pareto solutions got before an environmental change are used to anticipate. This approach is more direct because the factors involved in time series construction are only in the context of decision space. However each Pareto solution will be linked to a time series, which may lead to a large number of time series and the computational complexity may be enormous.

In this paper, we follow the second idea. The difference to [12] lies in that we use hyper-box based selection (HBS) to construct time series from the Pareto sets. As shown in Fig. 3, suppose there are two decision values and the range of each value is divided into many sections by a preset parameter ε_1

or ε_2. Thus the whole decision space is divided into lots of hyper-boxes. Instead of the whole Pareto set, partial solutions will be selected for time series construction. The HBS allows a hyper-box being occupied by only one Pareto solution in the Pareto set. If there are two or more solutions in a hyper-box, only the one that is the closest to the left-bottom corner of the hyper-box is the winner. For example, P_1 and P_2 are in the same hyper-box, obviously P_2 is the winner and is selected to construct time series.

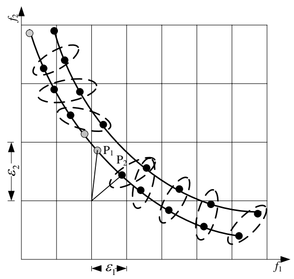


Fig. 3. Diagram of the HBS.

2) Construction of Time Series: As for constructing time series, the key issue is how to identify the relationship between the solutions selected by HBS. In this paper, we use minimal distance principle to identify the relationship between two solutions. Each Pareto solution $x_S(i,t)$ in $S'_p(t)$ will be added to the end of a time series and the number of **TS** is equal to the number of $x_S(i,t)$. Suppose $TS(j) \in \mathbf{TS}$ is a time series constructed previously and x_T is its last element. If the Euclidean distance from $x_S(i,t)$ to x_T is the shortest then $x_S(i,t)$ should be added to $TS(j)$, i.e.

$$TS(i) = \arg \min_{x_s \in S'_p(t), x_T \in TS(i)} \|x_s - x_T\|_2 \tag{3}$$

Considering the limitation of memory and computation resource, we use a preset parameter K to control the maximal order number of a time series. This means there are at most K elements in a time series. If the length of a time series is shorter than K the $x_S(t)$ will be added to the end of the time series directly. Otherwise, the elements in the time series will follow the first-in-first-out principle.

3) Prediction and Generation of Initial Population: Many methods could be used to analyze the time series constructed above. In this work, the following simple linear model is adopted:

$$\tilde{x}_{t+1} = x_t + (x_t - x_{t-1}). \tag{4}$$

Considering the forecasting error caused by the inaccuracy of the forecasting model and the searching algorithm, the prediction results may not be directly used to initialize the new population, the diversity should be maintain to some degree. Here we maintain diversity in two aspects:

a) Only partial initial population is generated based on the prediction results: A preset parameter α is used to control the rate of the individuals which will be initialized refereing to the prediction and the rest individuals will be randomly generated.

b) Variation with a noise: Similar to [12], we bring in a Gaussian noise λ to improve the chance of the initial population to cover the true Pareto set. This noise is added to the predicted result of each decision value. The standard deviation δ of noise is estimated by looking at the changes occurred before:

$$\delta^2 = \frac{1}{4n} \|x_t - x_{t-1}\|_2^2 \tag{5}$$

4) Chromosome Representation: The chromosome is the bridge between the optimization problem and searching space. For a path section, it is straightforward to code it as a series of consecutive line segments. However this coding method is ambiguous for the control system since the control system can not get explicit input signal from such a chromosome. In this work, we use a series of consecutive yaw angle changing values to code a chromosome. In each time step (i.e. Δt) the UCAV flies at a corresponding yaw angle and the resulting path can be got. Suppose the UCAV cruises at a const velocity and only 2-D case is considered. As shown in Fig. 4, the UCAV makes a change of u_1 in its yaw angle at the moment of t, and a line segment from $\mathbf{x(t)}$ to $\mathbf{x}(t+\Delta t)$ could be geometrically calculated using the kinematics of the UCAV.

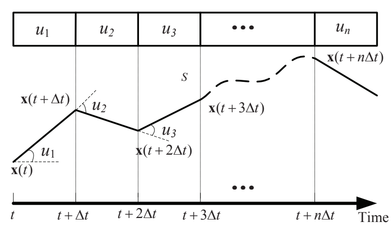


Fig. 4. Diagram of the chromosome representation.

III. INTELLIGENT DECISION MAKING ON SELECTION OF EXECUTIVE SOLUTION

Although a set of Pareto solutions can be dynamically obtained using the LP-DMOEA, the OPP problem is not solved until one feasible solution is selected out for executing. In this section we focus on how to select a solution referring to the bias of the decision maker (DM) and how to intelligently make such decision.

A. Methodology to Select Solutions from Pareto Set

In this work we use the Weighted Stress Function Method (WSFM) proposed in [15] to integrate the DM preferences after the search process has been made. For a problem with M optimal objectives the WSFM is converted to a single objective optimization problem:

$$\mathbf{x} = \arg \min_{\mathbf{x} \in \mathbf{S}} \left(\sum_{1 \le i < j \le M} \left| \gamma_i(f_i(\mathbf{x})) - \gamma_j\left(f_j(\mathbf{x})\right) \right| \right) \tag{6}$$

where $\mathbf{x}$ and $\mathbf{S}$ denote decision value vector and decision space separately, $\gamma_i(f_i(\mathbf{x}))$ denotes the "stress" associated to the corresponding objective according to the weight

(ω_i ($\sum_i^M w_i = 1$), given by DM) contributed to each objective. Please see [15] for more information. Since equation above consider the maximization problem and each objective should be normalized we rewrite the f_1 and f_2 as follows.

$$\begin{aligned} f_1(\mathbf{u},t) &= 1-\prod_{i=1}^{n} p_{kill}\left(\mathbf{x}(t)+\sum_{j=1}^{i} g(u_j)\right) \\ f_2(\mathbf{u},t) &= 1-\frac{\left\|\mathbf{x}(t)+\sum_{i=1}^{n} g(u_i)-\mathrm{T}\right\|_2-\left(\|\mathbf{x}(t)-\mathrm{T}\|_2-nV\Delta T\right)}{2nV\Delta T} \end{aligned} \tag{7}$$

B. Intelligent Situation Assessment via Bayesian Network

Now the problem turns to how to set ω_i intelligently. The weights ω_1 and ω_2 reflect the DM's bias to safety (i.e. f_1) and path length (i.e. f_2) respectively. In this paper the BN is employed to assess the dangerous level of the battle filed. The construction process of a BN, including the structure and parameters, is indeed the integration of the DM's knowledge. The resulting BN will make intelligent inference instead of the DM. In this work, the enemy air defense (i.e. threats for UCAVs) consists of two types of anti-air weapons: the anti-air guns (AAGuns) and the surface-to-air missiles (SAMs). The threat type is written as TT for short. A threat may work in one of the following states: No targets found and the system is inactive (IA), surveillance radar detects the targets (Surv), targets have been intercepted by radars (Intercept), targets are being traced by radars (Trace) and open fire (Fire). There are five environmental dangerous levels (EDLs): very dangerous (VD), dangerous (D), medium (M), safe (S) and very safe (VS). If the working states of the threats (ST) could be known the EDL can be easily inferred.

However, the ST is difficult to be known directly and an UCAV has to infer such information referring to the local information collected by its onboard sensors. Suppose there are two major onboard sensors assembled on an UCAV: the missile launching detector (MLD) and the radar warning receiver (RWR). For these sensors, there are two working states: active (A) and inactive (IA). When the RWR works in state A, the UCAV has been detected or traced by enemy radars and the anti-air weapons may be launched in a short future. The matter is worse when MLD is working in state A which means the UCAV is under attack.

In addition to the sensors' information, the distance between a threat and the UAV is also a key factor impacts the EDL. Suppose there are five range (R) scales: R1 (0 1 km), R2 (1 2 km), R3 (2 4 km), R4 (4 6 km) and R5 (larger than 6 km). The BN structure is shown in Fig 5 and the corresponding conditional probability tables (CPTs) are given in the Appendix.

C. Quantification of Environmental Assessment Results

Since the BN is a qualitative inference tool, we need to quantify the inference results (EDL) to obtain the weight values associated to each optimal objective. In this paper, we use fuzzy logic to set the value of ω_1 and the fuzzy rules are given in Fig 6. The triangular-shaped membership function (as shown in Fig 7) and the center of gravity defuzzification are adopted. After the quantification of ω_1, the can be easily calculated ω_1 since $\omega_1+\omega_2=1$.

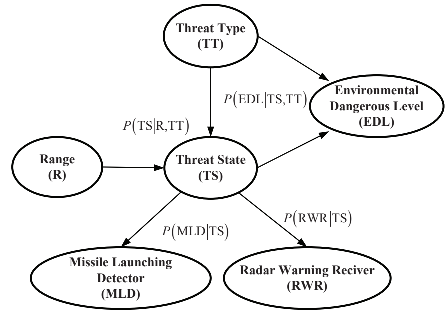


Fig. 5. BN structure of the environmental assessment model.

IF EDL is VD, THEN w1is very high;
IF EDL is D, THEN w1 is high;
IF EDL is M, THEN w1 is medium;
IF EDL is S, THEN w1 is low;
IF EDL is VD, THEN w1 is very low.

Fig. 6. Fuzzy rules between EDL and ω_1

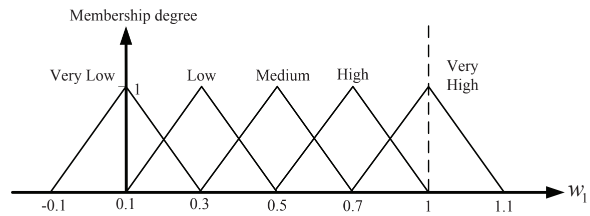


Fig. 7. The triangular-shaped membership function. Suppose the value range of ω_1 is [0.1, 1]

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In the following experiments, we chose the NSGA2 [16] as the basic MOEA. We apply LP-DMOEA and random restart methods to NSGA2 and the resulting dynamic MOEA are written as LP-DNSGA2 and R-DNSGA2 respectively. We firstly compare two OPP algorithms: OPP-A and OPP-B which use LP-DNSGA2 and R-DNSGA2 respectively. Then the intelligent OPP using LP-DNSGA2 and the intelligent decision making methods will be validated.

A. General Algorithmic Setup

Before the simulations and analysis the general algorithmic setup are given as follows:

1) Parameters involved in NSGA2: Population size is set to100, the length of a chromosome is set to 20, the probabilities of simulated binary crossover (SBX) and polynomial mutation (PM) are 0.9 and 1/20 respectively and the special parameters of SBX and PM are 10 and 20 respectively.

2) Parameters involved in LP-DNSGA2: Rate of the heuristically generated individuals is set to 50% (i.e. $\alpha = 0.5$), the maximal order number of a time series is set to 5 (i.e. $K = 5$), both ε_1 and ε_2 are set to 0.1.

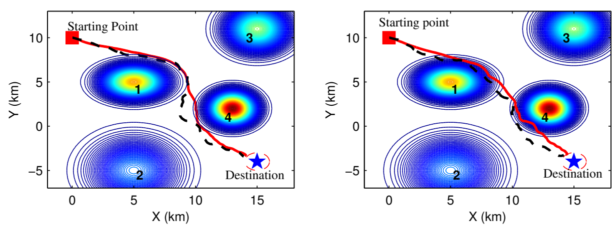
3) Parameters involved in OPP: Time step Δt is set to 1s, executing horizon is set to one time step (i.e. 1s) and the number of time steps (planning horizon) of a sequential control input is 20 (i.e. the chromosome length $n = 20$).

There are four threats in the simulated battlefield among them threat No.1, No. 2 and No. 4 are SAMs and threat No. 3 is AAGun. The simulation terminates when the distance between the goal and the UAV is within 1 km.

B. Comparison of Two OPP Algorithms

Firstly, we compare OPP-A and OPP-B to show the advantage of the proposed dynamic MOEA over the random restart method. Besides, we'd like to test the validity of WSFM. Therefore, in the following experiments, the intelligent decision making method are not used. Two OPP algorithms will be tested in unknown environment (no information about the four threats are known in advance) with fixed weight values ($\omega_1 = 0.7, \omega_2 = 0.3$ or $\omega_1 = 0.3, \omega_2 = 0.7$).

As shown in Fig. 8, the paths obtained by OPP-A (solid line) are more reasonable and smoother than the paths planned by OPP-B (dotted line). Besides, the paths considering the DM bias of $\omega_1 = 0.7$ and $\omega_2 = 0.3$ are more likely to keep away from the threats in contrast to the bias of $\omega_1 = 0.3$ and $\omega_2 = 0.7$. This reveals that the WSFM can effectively integrate the DM bias into the automatic planner.



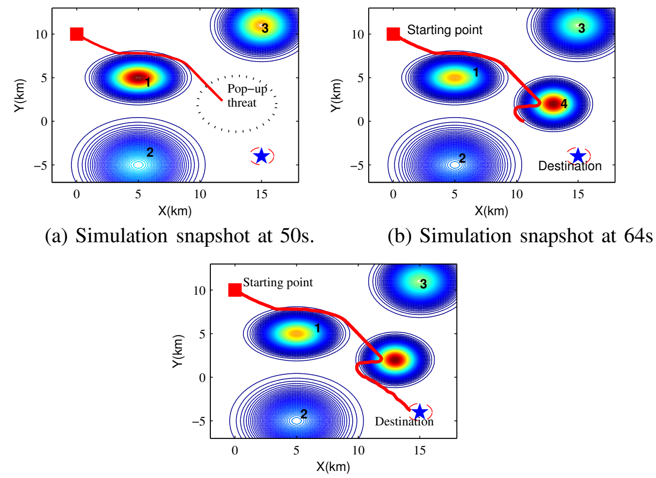
(a) $\omega_1 = 0.7$ and $\omega_2 = 0.3$. (b) $\omega_1 = 0.3$ and $\omega_2 = 0.7$

Fig. 8. Comparison between OPP-A (solid line) and OPP-B (dotted line).

C. Validation of Proposed Intelligent OPP Algorithm

In the following experiments, we will show the validity of the intelligent OPP which is the combination of OPP-A and the intelligent environmental assessment. Here, we let the threat No. 4 does not appear or work until the UAV flies for 50 seconds. In such case, the UAV should assess the environment and react to the pop-up threat intelligently.

As shown in Fig. 9, in the first 50 seconds the UAV has successfully evaded the threats and would straightly fly to-ward to the its goal if the hostile environment would not change. Then a pop-up threat (No.4) suddenly appears at 50s. Fortunately, seen from the dotted path, the UAV can react to this change by flying away from the pop-up threat as quickly as possible. At this moment, the intelligent environmental assessment works effectively to increase the probability of $P(EDL = VD)$ accordingly and the weight value (ω_1) associated to the safety is set to a higher one. Seen from Fig. 10, the probability of $P(EDL = VD)$ raises at 50s and goes down again (at about 58s) when the UAV has flied away from the pop-up threat. According, as shown in Fig. 11, the value of ω_1 is set to 0.9 from 50s to 58s and goes down when the environment seems safe again.



(a) Simulation snapshot at 50s. (b) Simulation snapshot at 64s

(c) Final simulation snapshot.

Fig. 9. Path planned by intelligent OPP algorithm proposed in this paper.

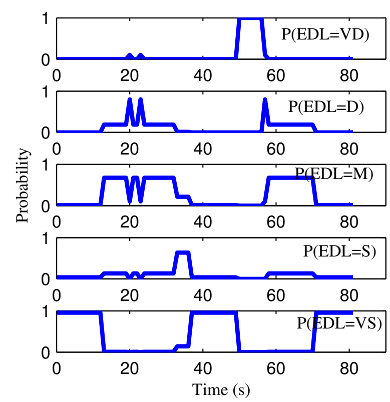


Fig. 10. The probabilities of each EDL versus time.

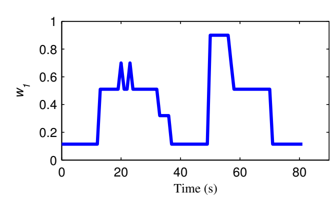


Fig. 11. The value of ω_1 versus time.

V. CONCLUSION

Usually the mission environment for an UAV is unknown and may change arbitrarily. The intelligent flight is a key technology for an UAV to react to the changing environment. The major contribution of this work is to solve the OPP problem which is a basic issue for intelligent flight by integrating dynamic MOEA, BN and fuzzy logic. Considering the fact that an UAV has to collect information via its onboard sensors sometimes, a MPC-like OPP method is employed to continuously update the local environmental information for the planner and this method is in fact a DMOP. For solving this problem, we have proposed the LP-DMOEA and the main idea is to utilize the historical information to enhance the performance. The historical Pareto sets are collected to con-structed several time series and the search process for the new problem could be guided by the prediction of those time series. The WSFM has been introduced to select the best solution referring to the bias of DM. For making use of such posterior method the BN is used to model the environmental assessment accomplished by a pilot and the fuzzy logic is employed to quantify the assessment results so as to obtain the weight value associated to each optimal objective. We have used the NSGA2 as the basic MOEA and the simulation is in a simple military case. The experimental results show that the LP-DMOEA works more effectively for the OPP in contrast to the restart method due to the positive impact on heuristically initializing the population for the new problem. In addition, the intelligent methods for solution selection can automatically assess the changing environment and adapt the path planner.

VI. APPENDIX

Table I~IV show the CPTs of the BN of environmental assessment model in Section III.

TABLE I
P(RWR | TS)

RWR	TS				
	IA	Surv	Intercept	Trace	Fire
A	0	1	1	1	0
IA	1	0	0	0	1

TABLE II
P(MLD | TS)

MLD	TS				
	IA	Surv	Intercept	Trace	Fire
A	0	0	0	0	1
IA	1	1	1	1	0

TABLE III
P(TS | R, TT)

TT	TS	R				
		R1	R2	R3	R4	R5
AAGun	IA	0	0	0.05	0.95	1
	Surv	0	0.05	0.9	0.05	0
	Intercept	0	0.9	0.05	0	0
	Trace	0.6	0.05	0	0	0
	Fire	0.4	0	0	0	0
SAM	IA	0	0	0	0.05	0.95
	Surv	0	0	0.05	0.9	0.05
	Intercept	0	0.05	0.9	0.05	0
	Trace	0.05	0.9	0.05	0	0
	Fire	0.95	0.05	0	0	0

TABLE IV
P(EDL | TS, TT)

TT	EDL	TS				
		IA	Surv	Intercept	Trace	Fire
AAGun	VD	0	0	0	0.95	1
	D	0	0	0.05	0.05	0
	M	0	0.1	0.9	0	0
	S	0	0.8	0.05	0	0
	VS	1	0.1	0	0	0
SAM	VD	0	0	0	0.1	1
	D	0	0	0.2	0.8	0
	M	0	0.2	0.7	0.1	0
	S	0	0.7	0.1	0	0
	VS	1	0.1	0	0	0

REFERENCES

[1] P.R. Chandler, S. Rasmussen and M. Pachter, ”UAV Cooperative Path Planning,” in *proc. AIAA Guidence, Navigation, and Control Conference and Exhibit*, AIAA, 2000.

[2] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis and A.N. Kostaras, ”Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation,” *IEEE Transactions on Systems, Man, and Cybernetics£-Part B: Cybernetics*, vol. 33, no. 6, 2003, pp. 898-912.

[3] D. Rathbun and B. Capozzi, ”Evolutionary Approaches to Path Planning Through Uncertain Environments,” in *proc. of AIAA Unmanned Aerospace Vehicles, Systems, Technologies and Operations Conference and Workshop*, AIAA, 2002.

[4] C. Zheng, L. Li, F. Xu, F. Sun and M. Ding, ”Evolutionary Route Planner for Unmanned Air Vehicles,” *IEEE Transactions on Robotics*, vol. 21, no. 4, 2005, pp. 609-620.

[5] J.J. Grefenstette and J. Fitzpatrick, ”Genetic algorithms for changing environments,” in *proc. the 2nd International Conference on Parallel Problem Solving from Nature (PPSN II)*, 1992, pp. 137-144.

[6] X. Peng, X. Gao and S. Yang, ”Environment identification based memory scheme for estimation of distribution algorithms in dynamic environments,” *Soft Computing*, vol. 15, no. 2, 2011, pp. 311-326.

[7] S. Yang and X. Yao, ”Population-based incremental learning with associative memory for dynamic environments,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, 2008, pp. 542-561.

[8] J. Branke, P. Funes, C. Schmidt and H. Schmeck, ”A multipopulation approach to dynamic optimization problems,” in *proc. Adaptive Computing in Design and Manufacturing (ACDM 2000)*, 2000, pp. 299-308.

[9] R.K. Ursem, ”Multinational GA optimization techniques in dynamic environments,” in *proc. the 2nd Annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2000)*, 2000, pp. 19-26.

[10] I. Hatzakis and D. Wallace, ”Dynamic Multi-Objective Optimization with Evolutionary Algorithms: A Forward-Looking Approach,” in *proc. the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO 2006)*, 2006, pp. 1201-1208.

[11] C. Rossi, M. Abderrahim and J.C. Diaz, ”Tracking Moving Optima Using Kalman-Based Predictions,” *Evolutionary Computation*, vol. 16, no. 1, 2008, pp. 1-30.

[12] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff and E. Tsang, ”Prediction-Based Re-initialization for Evolutionary Dynamic Multi-objective Optimization,” in *proc. the 4th International Conference on Evolutionary Multi-criterion Optimization (EMO2007)*, 2007, pp. 932-846.

[13] A. Pongpunwattana and R. Rysdyk, ”Evolution-based Dynamic Path Planning for Autonomous Vehicles,” *Studies in Computational Intelligence*, vol. 70, 2007, pp. 113-145.

[14] U. Zengin and A. Dogan, ”Dynamic Target Pursuit by UAVs in Prob-abilistic Threat Exposure Map,” in *proc. AIAA 3rd ”Unmanned Unlimited” Technical Conference, Workshop and Exhibit*, 2004.

[15] J.C. Ferreira, C.M. Fonseca and A. Gaspar-Cunha, ”Methodology to Select Solutions from the Pareto-Optimal Set: A Comparative Study,” in *proc. 9th Annual Conference on Genetic and Evolutionary Computation (GECCO 2007)*, 2007.

[16] K. Deb, ”A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002, pp. 182-197.