

# Estimation of Distribution Algorithms based Unmanned Aerial Vehicle Path Planner Using a New Coordinate System

Peng Yang

USTC-Birmingham Joint Research  
Institute in Intelligent Computation  
and Its Applications (UBRI)  
School of Computer Science and  
Technology of USTC  
Hefei, China  
Email: trevor@mail.ustc.edu.cn

Ke Tang

USTC-Birmingham Joint Research  
Institute in Intelligent Computation  
and Its Applications (UBRI)  
School of Computer Science and  
Technology of USTC  
Hefei, China  
Email: ketang@ustc.edu.cn

Jose A. Lozano

Intelligent Systems Group  
University of the Basque Country  
Gipuzkoa, Spain  
Email: ja.lozano@ehu.es

**Abstract**—Path planning technique is vital to Unmanned Aerial Vehicle (UAV). Evolutionary Algorithms (EAs) have been widely used in planning path for UAV. In these EA-based path planners, Cartesian coordinate system and polar coordinate system are commonly used to codify the path. However, either of them has its drawback: Cartesian coordinate systems result in an enormous search space, whilst polar coordinate systems are unfit for local modifications resulting e.g., from mutation and/or crossover. In order to overcome these two drawbacks, we solve the UAV path planning in a new coordinate system. As the new coordinate system is only a rotation of Cartesian coordinate system, it is inherently easy for local modification. Besides, this new coordinate system has successfully reduced the search space by explicitly dividing the mission space into several subspaces. Within this new coordinate system, an Estimation of Distribution Algorithms (EDAs) based path planner is proposed in this paper. Some experiments have been designed to test different aspects of the new path planner. The results show the effectiveness of this planner.

**Keywords**—Unmanned Aerial Vehicle; off-line path planning; rotated coordinate system; Estimation of Distribution Algorithms

## I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are aircrafts without onboard pilots [1]. So far, UAVs have been used in various fields, including civil markets [2] and military [3], and draw increasingly research interests. Usually, UAVs execute those missions with being remotely controlled by humans on ground or autonomously flying by following the onboard preprogrammed path planner [4], [5]. As the missions get complex, remote controllers can hardly offer sufficiently accurate and quick reactions due to physiologically limitations of human. In this case, path planning technique becomes very important for UAVs. [3]. The path planning for UAVs is to generate an optimal or near-optimal path that has shortest length and largest survival probability to guide the UAV to reach the desired destination satisfying various

constraints [4]. Finding such a path has been proved a NP-Complete problem [5].

So far, various methods, e.g., A\* [5], [6], Mixed-Integer Linear Programming [7], [8], Nonlinear Programming [9] and Voronoi Diagram [10], [11], have been proposed to construct path planners. These planners build a search map and search a path in it. However, this approach requires the calculation of the configuration space (C-space) for looking up the performance of any point in the search map, which is computationally expensive [12]. Besides, when the mission varies, e.g., an unknown threat pop-ups, the C-space must be explicitly updated [12], [13], [15], which is also very time-consuming. Given these considerations, Evolutionary

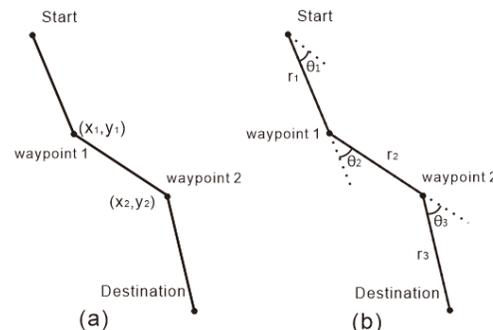


Fig. 1. A 2-D path codified with absolute Cartesian coordinate is shown in (a). A 2-D path with in the relative polar coordinate is shown in (b).

Algorithms (EAs) have been adopted into UAVs path planning [12]-[20] since the individuals (candidate paths) of EAs are evaluated in the search space, while the C-space is no longer required [12].

The search space of an EA-based path planner depends heavily on the representation of the path. In the existing EA-based path planners, a path is commonly represented as a sequence of 3-D waypoints, each of which indicates a place somewhere in the mission space that the UAV will pass-by (Fig.1). Thus, the goal of EA-based planners is to find a

sequence of locations, i.e., waypoints, that makes the whole path optimal. Under this representation, two kinds of coordinate systems, i.e., Cartesian coordinate system [12], [14]-[17], [19], denoted as  $(x, y, z)$ , and polar coordinate system [13], [15], [18], [20], denoted as  $(r, \theta, z)$ , have been adopted in EA-based path planners. However, both of them have inherent drawbacks [4], [12], [15]: the Cartesian coordinate system based planner generates enormous search space while the path represented in polar coordinate system is difficult for local modifications, e.g., mutation and crossover.

To overcome these drawbacks simultaneously, in this paper, a new coordinate system is used to represent the waypoints (Fig.2). This new coordinate system, denoted as  $(x', y', z')$ , makes its  $x'$  axis along the horizontal direction from the start to the destination of the mission and keeps  $y'$  axis being orthogonal to the  $x'$  axis, and leaves  $z'$  axis the same with  $z$ . Within this new coordinate system, the  $x'$  coordinate of successive waypoints should be always larger than the  $x'$  coordinate of their previous ones. To benefit from this, the  $x'$  axis is explicitly divided into  $N_w-2$  equal intervals, where  $N_w$  is the number of waypoints of a path. In this way, the whole mission space can be divided into  $N_w-2$  equal subspaces and each intermediate waypoint, i.e., ignoring the start and destination, will be sampled in its corresponding interval. Thus, the search space generated by the new coordinate system has been significantly reduced. Furthermore, as  $(x', y', z')$  is a rotation of  $(x, y, z)$ , the waypoints are mathematically independent of each other, thus they suit local modification easily. In a word, this new coordinate system has solved the drawbacks of the previous two coordinate systems simultaneously.

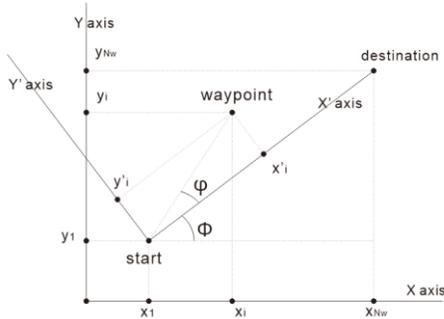


Fig. 2. A 2-D example of the relation between the new coordinate and the absolute Cartesian coordinate.

As the mission space has been divided into  $N_w-2$  subspaces, to find a possible path with  $N_w-2$  intermediate waypoints is actually to find one optimal waypoint in each subspace, so that the line segments from itself to its two adjacent waypoints satisfy all the constraints and objectives. For the sake of finding optimal waypoints in each subspace, Estimation of Distribution Algorithms (EDAs) are adopted. EDAs are a new branch of EAs and have been proved efficient and reliable in many real-world applications [21]. An EDA generates new individuals by sampling from a probability distribution estimated from the promising individuals of previous generations rather than by crossover and mutation that classical EAs (e.g., GA) do. The motivation of using EDAs are two-fold: first, traditional operators like mutation and

crossover are discarded, which simplifies the design of the path planners. Second, the problem structure can be revealed by learning the probability distribution, which has the ability of explicitly guiding the search for optimal waypoints. With this motivation, UMDA<sub>c</sub> [22] is employed to construct a new path planner. UMDA<sub>c</sub> is one of the simplest EDAs and commonly regarded as the baseline of EDAs. The reason of choosing UMDA<sub>c</sub> rather than other more complex and efficient EDAs is that we want to simply test the basal effectiveness of the how EDAs perform in path planning without any specific or complex assumed relations among variables.

Experimental studies consist of two parts. First, 4 problems in different difficulty levels are designed to test the effectiveness of the new proposed path planner. Second, in order to show how the new coordinate system makes sense, another two UMDA<sub>c</sub> based path planners are constructed with the Cartesian coordinate system and the polar coordinate system, and tested. The results show that new coordinate system based path planner is competent on complex problems and outperforms the other two.

The rest of this paper is organized as follows: Section II, the problem of UAV path planning is described. Section III introduces the new coordinate system in detail. After that, the UMDA<sub>c</sub> based path planner are constructed in Section IV. In Section V, the experimental studies are shown.

## II. PROBLEM DESCRIPTIONS

The path planning for UAVs is to generate an optimal or near-optimal path in the mission space to guide the UAV to reach the desired destination. The path should satisfy various constraints and objectives. In this section, 4 constraints and 4 objectives that are commonly used are described as follows.

### A. Constraints

Constraints usually depict the limitations of UAV's physical ability and mission space. Different constraints are required by different UAVs and missions. Here 4 commonly used constraints [12], [14], [15] are described:

1) *Maximal Turning Angle*: The path at each waypoint should be smooth enough, so that the UAV can follow on it easily. Thus, the turning angle, i.e., the angle that included by two adjacent segments of a waypoint, at each intermediate waypoint should be smaller than a threshold. For each intermediate waypoint  $(x'_i, y'_i, z'_i)$ ,  $i = 2, 3, \dots, N_w - 1$ , its turning angle can be calculated as:

$$\theta_i = \arccos \left( \frac{(x'_i - x'_{i-1}, y'_i - y'_{i-1}) \cdot (x'_{i+1} - x'_i, y'_{i+1} - y'_i)^T}{|(x'_i - x'_{i-1}, y'_i - y'_{i-1})| \cdot |(x'_{i+1} - x'_i, y'_{i+1} - y'_i)|} \right) \quad (1)$$

Suppose  $\Delta$  is an arbitrary vector, then  $|\Delta|$  means the norm of vector  $\Delta$ . The number of violations of a path on this constraint can be written as:

$$\sum_{i=2}^{N_w-1} c_i^1 \text{ with } c_i^1 = \begin{cases} 1 & \text{if } \theta_i > \theta_{max} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where  $\theta_{max}$  is the predefined maximal turning angle.

2) *Limited UAV Slope*: The UAV slope indicates that the

angle at the waypoint where a UAV changes its altitude. For each waypoint  $(x'_i, y'_i, z'_i)$ ,  $i = 1, 2, \dots, N_w-1$ , its slope can be calculated as:

$$k = \frac{z'_{i+1} - z'_i}{|(x'_{i+1} - x'_i, y'_{i+1} - y'_i)|} \quad (3)$$

Assuming the maximal diving angle is  $\alpha$  and the maximal climbing angle is  $\beta$ , the number of violations of a path on this constraint can be written as:

$$\sum_{i=1}^{N_w-1} c_i^2 \text{ with } c_i^2 = \begin{cases} 0 & \text{if } \alpha \leq k \leq \beta \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

3) *Terrain Limited*: A feasible path should not go through the terrain and has to avoid collisions with mountains. In order to check whether the path (segment) between two adjacent waypoints is within the terrain or not, that segment is divided equally into 6 parts. And each of the dividing points will be checked. If  $\text{map}(x'_i, y'_i)$  is the function that returns the altitude of the horizontal location  $(x'_i, y'_i)$ , the numbers that the path go through the terrain can be calculated as:

$$\sum_{i=1}^{N_w-1} \sum_{j=1}^6 c_{ij}^3 \text{ with } c_{ij}^3 = \begin{cases} 1 & dz'_{ij} \leq \text{map}(dx'_{ij}, dy'_{ij}) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where  $(dx'_{ij}, dy'_{ij}, dz'_{ij})$  indicates the  $j^{\text{th}}$  dividing point in the path between the  $i^{\text{th}}$  and  $(i+1)^{\text{th}}$  waypoint, it can be calculated as:

$$(dx'_{ij}, dy'_{ij}, dz'_{ij}) = (x'_i, y'_i, z'_i) + (j-1) \cdot \left( (x'_{i+1}, y'_{i+1}, z'_{i+1}) - (x'_i, y'_i, z'_i) \right) / 6 \quad (6)$$

where  $j = 1, 2, \dots, 6$ .

4) *Maps Limited*: To restrict the path inside the given mission range, this constraint is used [15].

$$\sum_{i=1}^{N_w-1} c_i^4 \text{ with } c_i^4 = \begin{cases} 0 & \text{InRange}(x'_i, y'_i) \\ 1 & \text{otherwise} \end{cases} \\ \text{InRange}(x'_i, y'_i) = (l_x \leq x'_i \leq h_x) \wedge (l_y \leq y'_i \leq h_y) \quad (7)$$

where  $l_x$  and  $h_x$  are the lower and higher bounds for  $x'$  coordinate system, and  $l_y$  and  $h_y$  are the lower and higher bounds for  $y'$  coordinate system.

## B. Objectives

1) *Minimal Path Length*: For military missions, shorter paths are better than longer ones, if all the objectives are equally regarded. This is reasonable since those shorter paths require less time of flight and are very likely to have lower chance of encountering some unexpected threat in a real uncertain mission space. The normalized path length is to be minimized rather than the real one since they are equivalent and the normalized one is considered admissible [15]. The path length ratio (PLR) is calculated as follows:

$$\text{PLR} = \frac{\sum_{i=1}^{N_w-1} \sqrt{(x'_{i+1} - x'_i)^2 + (y'_{i+1} - y'_i)^2 + (z'_{i+1} - z'_i)^2}}{\sqrt{(x'_{N_w} - x'_1)^2 + (y'_{N_w} - y'_1)^2 + (z'_{N_w} - z'_1)^2}} \quad (8)$$

2) *Minimal Probability of Kill*: Paths with lower probability of kill are safer than those with higher ones. The probability of kill (PKill) [15] depends on the enemy Air Defense Units (ADUs) that are groups of radars and missiles. In order to calculate the approximate PKill, each dividing point is checked (seen in Fig.3). For each dividing point, the  $k^{\text{th}}$  ADU has a certain probability to destroy the UAV only if that point is inside the region defined by the ADU's maximal risk distance, denoted as  $R_{PKmax}^k$ . And the PKill of a dividing point caused by an ADU is fourth power to the distance from the dividing point to that ADU [12]. Thus, given the location of the  $k^{\text{th}}$  ADU  $(ax'_k, ay'_k, az'_k)$ , the PKill of the whole path can be calculated as:

$$\text{PKill} = \sum_{i=1}^{N_w-1} \sum_{j=1}^6 \sum_{k=1}^A \text{PK}_{ij}^k \\ \text{with } \text{PK}_{ij}^k = \begin{cases} \left( \frac{\gamma}{\text{dis}_{ij}^k} \right)^4 & \text{if } \text{dis}_{ij}^k \leq R_{PKmax}^k \\ 0 & \text{otherwise} \end{cases}$$

$$\text{dis}_{ij}^k = \sqrt{(dx'_{ij} - ax'_k)^2 + (dy'_{ij} - ay'_k)^2 + (dz'_{ij} - az'_k)^2} \quad (9)$$

where  $\gamma$  is a scale which reflects the intensity of ADUs,  $A$  is the number of ADUs.

3) *Minimal Probability of Radar Detection*:

Only if the UAV is detected by radars, it can be destroyed

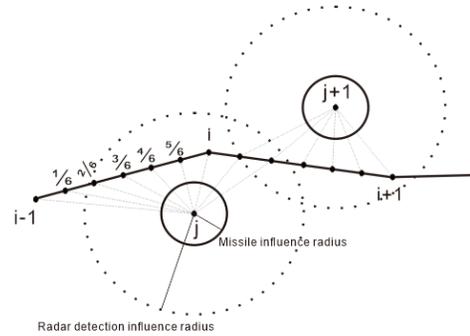


Fig. 3. Check the path between  $(i-1)^{\text{th}}$  waypoint and  $(i+1)^{\text{th}}$  waypoint with  $j^{\text{th}}$  and  $(j+1)^{\text{th}}$  ADUs.

by ADUs. Otherwise, the UAV will keep stealthy. The calculation of the probability of radar detection (PRD) is similar to that of PKill, only if the detective region of the  $k^{\text{th}}$  radar is larger than  $R_{PKmax}^k$ , denoted as  $R_{PRDmax}^k$ . Thus, given the location of the  $k^{\text{th}}$  radar  $(rx'_k, ry'_k, rz'_k)$ , the PRD of the whole path can be calculated as:

$$\text{PRD} = \sum_{i=1}^{N_w-1} \sum_{j=1}^6 \sum_{k=1}^R \text{PD}_{ij}^k \\ \text{with } \text{PD}_{ij}^k = \begin{cases} \left( \frac{\delta}{\text{dis}_{ij}^k} \right)^4 & \text{if } \text{dis}_{ij}^k \leq R_{PRDmax}^k \\ 0 & \text{otherwise} \end{cases}$$

$$\text{dis}_{ij}^k = \sqrt{(dx'_{ij} - rx'_k)^2 + (dy'_{ij} - ry'_k)^2 + (dz'_{ij} - rz'_k)^2}$$

(10)

where  $\delta$  is a scale which reflects the intensity of the radars, and  $R$  is the number of radars.

4) *Minimal Flight Altitude*: UAVs flying at a low altitude can take the advantages of the terrain-mask effect, which helps them to avoid radars. To approximate the Flight Altitude (FA), the average relative height of each dividing point is calculated. The FA is calculated as follows:

$$FA = \sum_{i=1}^{N_w} \sum_{j=1}^6 FA_{ij} \text{ with } FA_{ij} = \begin{cases} 0 & \text{if } dz'_{ij} \leq \text{map}(dx'_{ij}, dy'_{ij}) \\ (dz'_{ij} - \text{map}(dx'_{ij}, dy'_{ij}))/6 & \text{otherwise} \end{cases} \quad (11)$$

### III. THE NEW COORDINATE SYSTEM

#### A. The Commonly Used Coordinate Systems

In UAV path planning problems, EA-based path planners have been extensively studied [12]-[20] for the candidate paths are evaluated within the search space rather than the C-space, which significantly reduces the computational costs. The search space heavily depends on the representation of the path. In previous work, the path of UAV is usually represented in a sequence of 3-D waypoints. Under this representation, the Cartesian coordinate system and polar coordinate system are commonly used to codify the waypoints.

The Cartesian coordinate system is the most direct and natural representation of a spatial point that uses traditional orthogonal  $(x, y, z)$  to address the 3-D waypoints, where  $x$  and  $y$  indicate the location of a waypoint in the horizontal mission space and  $z$  denotes the height of a waypoint. A 2-D example of the Cartesian coordinate system based path is shown in Fig.1 (a). In this coordinate system, the waypoints are easy for local modifications, e.g., mutation and crossover, since the waypoints are independent of each other. Nevertheless, it generates enormous search space since the domain for generating each waypoint has to be always the whole mission space.

The polar coordinate system uses  $(r, \theta, z)$  to indirectly define the location of a waypoint, where  $r$  is the distance between two adjacent waypoints,  $\theta$  is the turning angle at each waypoint, and  $z$  is the height of a way point. A 2-D polar coordinate system based path is shown in Fig.1 (b). The waypoints in the polar coordinate system are just relative locations. This diminishes the search space since the domain for  $(r, \theta, z)$  is smaller than the whole mission space. However, changing any intermediate waypoint can heavily influence all its successive ones since the absolute location in the mission space of a waypoint depends heavily on all its previous ones. Thus, it turns out to be unfit for local modifications, e.g., mutation and crossover.

#### B. The New Coordinate System

In order to solve these two drawbacks simultaneously, a new coordinate system, denoted as  $(x', y', z')$ , is used in this paper. This coordinate system makes its  $x'$  axis along the horizontal direction from the start to the destination of the

mission and keeps  $y'$  axis being orthogonal to the  $x'$  axis, and leaves  $z'$  axis the same with  $z$ . Assume the start and destination in Cartesian coordinate system as  $(x_1, y_1, z_1)$  and  $(x_{N_w}, y_{N_w}, z_{N_w})$  are fixed and already known before path planning. For any point  $(x'_i, y'_i, z'_i)$  in new coordinate system, its location in Cartesian coordinate system,  $(x_i, y_i, z_i)$ , is defined mathematically as follows:

$$\begin{cases} x_i = x_1 + \cos(\varphi + \phi) \cdot \sqrt{x_i'^2 + y_i'^2} \\ y_i = y_1 + \sin(\varphi + \phi) \cdot \sqrt{x_i'^2 + y_i'^2} \\ z_i = z_i' \end{cases} \quad (12)$$

where  $\varphi$  is the angle included by the direction from start to waypoint and  $x'$  axis, and  $\phi$  is the angle between  $x'$  axis and  $x$  axis. The linear transformation from the new coordinate system to the Cartesian coordinate system can be seen in Fig.2. For example, the codifications of the start and destination in new coordinate system are  $(0, 0, z_1)$  and  $(\sqrt{(x_{N_w} - x_1)^2 + (y_{N_w} - y_1)^2}, 0, z_{N_w})$ .

The successive waypoints should be always closer to the destination than their previous ones. Otherwise, these two adjacent waypoints will induce a meaningless larger path length. This is an implicit **demand** of shorter path length. A general example is given in Fig.4 to illustrate why the path that violates the demand above will be the longer. In Fig.4, A, B, P, Q are four waypoints and P is closer than Q to Destination. To fly from A to B, there are four possible path for UAV, i.e., A-P-B, A-Q-B, A-Q-P-B and A-P-Q-B. Among them, only A-P-Q-B violates the demand that Q is successive to P while P is closer to B. Apparently, A-P-B is the shortest path and A-Q-B is the second shortest path according to the axiom that the two sides of a triangle is greater than the third side. For A-Q-P-B and A-P-Q-B, as Q is closer to A and P is closer to B, which indicates AQ is smaller than AP and PB is smaller than QB, thus A-Q-P-B is shorter than A-P-Q-B. Thus, A-P-Q-B is the longest path.

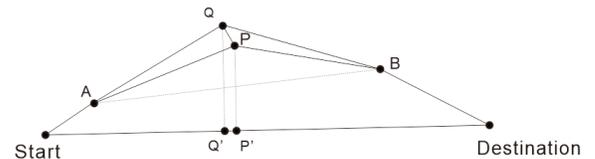


Fig. 4. The path A-P-Q-B violates the demand and is the longest path.

In the new coordinate system, to smooth the path length, the differences of  $y'$  or  $z'$  between two adjacent waypoints should be small, relative to the differences of  $x'$ . Otherwise, the turning angle will be large, which the UAV is difficult to follow on. In that way, the path length between two adjacent waypoints is mainly determined by the distance between the  $x'$  of these two waypoints. Hence, for simplicity, we can modify the **demand** as: the  $x'$  of successive waypoints should be always larger than the  $x'$  of their previous ones. According to this modified demand, the waypoints of a path will be always in an ascending order along the  $x'$  axis. In this case, if the  $x'$  coordinate system of the  $(i - 1)^{\text{th}}$  waypoint of one candidate path will never larger than that of the  $i^{\text{th}}$  waypoint of all the other candidate paths, where  $i = 2, 3, \dots, N_w - 1$ , all

the intermediate waypoints will form  $N_w - 2$  clusters and anyone in the  $i^{\text{th}}$  cluster, along the  $x'$  axis, is the  $x'$  waypoint of its path.

Inspired by this thought, the  $x'$  axis is explicitly divided into  $N_w - 2$  equal intervals, by which the mission space is divided into  $N_w - 2$  equal subspaces, and the  $i^{\text{th}}$  intermediate waypoints will be sampled in the  $i^{\text{th}}$  subspace. In this case, the domain for generating each waypoint becomes  $N_w - 2$  smaller than the original one. As a path consists of  $N_w - 2$  waypoints, the whole search space becomes  $(N_w - 2)^{N_w - 2}$  times smaller than that of the Cartesian coordinate system. Besides, the absolute location of each waypoint in the new coordinate system is only based on the start and destination, which are predefined. Thus, the absolute location of each waypoint is independent of the others, which means changing any of the waypoint will not influence the others. Hence, the drawbacks of the Cartesian coordinate system and the polar coordinate system simultaneously have been solved.

#### IV. THE EDA BASED PATH PLANNERS

In this paper, as the  $x'$  axis of the new coordinate system has been divided into  $N_w - 2$  subspaces, to find an optimal path with  $N_w - 2$  intermediate waypoints is actually to find one optimal waypoint in each subspace, so that the line segments from itself to its two adjacent waypoints satisfy all the constraints and objectives. In order to locate optimal waypoints in each subspace, Estimation of Distribution Algorithms (EDAs) [22] are adopted.

##### A. Estimation of Distribution Algorithms

EDAs are a new branch of EAs that have been adopted in many real-world applications [21]. An EDA estimates a probability distribution from selected individuals to replace the design of crossover and mutation in classical EAs. The framework of EDAs can be described as follows:

- 1)  $g_0 \leftarrow$  Initialize  $N_p$  individuals uniformly.
- 2) Evaluate the  $N_p$  individuals.
- 3)  $g_{i+1/2} \leftarrow$  Select  $N_p/2$  individuals with better fitness.
- 4) Estimate a probability distribution  $p_i$  with  $g_{i+1/2}$ .
- 5)  $g'_{i+1} \leftarrow$  Sample  $N_p/2$  new individuals from  $p_i$ .
- 6)  $g_{i+1} \leftarrow$  Form new population with  $g_{i+1/2}$  and  $g'_{i+1}$ .
- 7) Stop if some stop criterion meets. Otherwise  $i = i + 1$ , go to step 2).

The motivation of introducing EDAs is two-fold: first, EDAs do not require the design of reproductive operators, i.e., mutation and crossover operators. Besides, the parameters are reduced, which take lots of time to fine-tune. Second, EDAs can help guide the path planning by explicitly learning the probability distribution of individuals.

##### B. UMDA<sub>c</sub> Based Path Planner

The framework of EDAs based path planners is essentially the extension of the framework of EDAs. It first uniformly initializes a population with  $N_p$  paths.

Then all the  $N_p$  paths are evaluated. Some of the previous work evaluated individuals with a single objective function, which combines constraints and objectives with predefined parameters [12], [16]-[19]. However, these parameters are difficult to fine-tune and may not be able to reflect the real intention of users. With that consideration, other researchers [13]-[15], [20] adopt multi-objective methods to evaluate the individuals. In this paper, an efficient multi-objective and multi-constraints handling method [23] based on goals, priorities, and Pareto sets is adopted to evaluate the candidate paths. This method has been successfully used in [15] and its ability of dealing with many-objective problems has been shown. Besides, this method does not require any parameters. According to this method, all these criteria are placed into three priority levels: the 4 constraints that must be satisfied are placed in the highest-priority; the Minimal path length and Minimal probability of kill objectives that should be minimized are placed in the second-priority; the last two objectives are placed in the lowest-priority. With this organization, paths that fulfill the highest-priority are better than those that do not, and the latter are basically organized according to how far the constraints have been optimized. If path A is better than path B, path B is said to be preferred by A.

After evaluation, the better  $N_p/2$  paths which have less preferred counts are selected for estimating probability distribution. So far, quite a few EDAs have been proposed in literature, and various methods have been adopted to estimate probability distribution. In this paper, one of the simplest EDA, i.e., UMDA<sub>c</sub>, is adopted to construct the path planner. UMDA<sub>c</sub> is commonly regarded as the baseline of EDAs. The purpose of choosing UMDA<sub>c</sub> rather than other EDAs is to test the basal effectiveness of EDAs based path planners without any specific or complex assuming relation among variables.

UMDA<sub>c</sub> naively assumes there is no interdependencies among variables. In this paper, the variables are actually two-stages: the waypoints stage and the 3-D coordinate systems stage. That is, given a path with  $N_w - 2$  intermediate waypoints  $(w_2, w_3, \dots, w_{N_w-1})$ , there are  $3(N_w - 2)$  variables  $(x'_i, y'_i, z'_i)$ ,  $i = 2, 3, \dots, N_w - 1$ . Hence, the probability distribution can be decomposed as  $3(N_w - 2)$  marginal distribution. That is,

$$\begin{aligned} p(w_2, w_3, \dots, w_{N_w-1}) &= p(w_2) \cdot p(w_3) \cdots p(w_{N_w-1}) \\ &= p(x'_2) \cdot p(y'_2) \cdot p(z'_2) \cdot p(x'_3) \cdot p(y'_3) \cdot \\ &\quad p(z'_3) \cdots p(x'_{N_w-1}) \cdot p(y'_{N_w-1}) \cdot p(z'_{N_w-1}) \end{aligned} \quad (13)$$

As UMDA<sub>c</sub> is a Gaussian distribution based EDA, the joint probability can be written as:

$$\begin{aligned} &p(w_2, w_3, \dots, w_{N_w-1}) \\ &= \prod_{k=2}^{N_w-1} \mathcal{N}(\mu_{x'_k}^{\Delta}, \sigma_{x'_k}^{\Delta}) \cdot \mathcal{N}(\mu_{y'_k}^{\Delta}, \sigma_{y'_k}^{\Delta}) \cdot \mathcal{N}(\mu_{z'_k}^{\Delta}, \sigma_{z'_k}^{\Delta}) \end{aligned} \quad (14)$$

Where  $\mathcal{N}(\mu_{\Delta_k}^{\Delta}, \sigma_{\Delta_k}^{\Delta})$  represents a marginal Gaussian density function with mean  $\mu_{\Delta_k}^{\Delta}$  and standard deviation  $\sigma_{\Delta_k}^{\Delta}$ . The values of  $(\mu_{\Delta_k}^{\Delta}, \sigma_{\Delta_k}^{\Delta})$  can be calculated by maximum likelihood estimation with the  $\Delta$  ( $\Delta = x', y', z'$ ) coordinate

system of the  $k^{\text{th}}$  ( $k = 2, 3, \dots, N_w - 1$ ) waypoint of all the selected paths.

After the probability distribution is estimated, new individuals will be sampled with the probability distribution by the Probabilistic Logic Sampling (PLS) method [22], which can be summarized, in the situation of UAV path planning, as follows:

- 1) For  $i = 1 : S$
- 2)  $(x'_{i1}, y'_{i1}, z'_{i1}) \leftarrow (x'_1, y'_1, z'_1)$
- 3)  $(x'_{iN_w}, y'_{iN_w}, z'_{iN_w}) \leftarrow (x'_{N_w}, y'_{N_w}, z'_{N_w})$
- 4) For  $j = 2 : N_w - 1$
- 5)  $(x'_{ij}, y'_{ij}, z'_{ij}) \leftarrow (p(x'_j), p(y'_j), p(z'_j))$

where  $S$  is the sample size,  $(x'_{ij}, y'_{ij}, z'_{ij})$  is the  $j^{\text{th}}$  waypoint of the  $i^{\text{th}}$  path,  $(x'_1, y'_1, z'_1)$  and  $(x'_{N_w}, y'_{N_w}, z'_{N_w})$  are the fixed start and destination.

After the new  $N_p/2$  individuals are generated, they form the new population for the next generation with the  $N_p/2$  truncatedly selected individuals at the previous generation.

Notice that although EDAs are used to search optimal waypoint in each subspace within this new coordinate system, the traditional EAs can also be adopted to utilize the new coordinate system, inherently.

## V. EXPERIMENTAL STUDIES

### A. Experiments Setup

In this paper, the 3-D off-line single UAV path planning is studied. To illustrate the effectiveness stability of this new proposed path planners, 4 problems with the same terrain but different number of ADUs are adopted, including 7, 15, 30, 60 randomly initialized ADUs. As far as we know, no

previous work has tested its planner with so many ADUs. Most of the previous work [12]-[14], [16]-[19] set at most 7 ADUs in the mission space, and only a few of them have adopted about 15 ADUs [15], [20]. Although in real missions, there might be less ADUs than our experiments have initialized, it is also necessary to have these tests since there might be some other obstacles in mission space which have the similar features to ADUs. For example, if we regard these ADUs as the rugged terrain, then these problems are actually testing the ability of the new planner on terrain following and terrain avoidance.

As the number of ADUs increases, the safe space reduces. Thus, in order to make the optimal path smooth enough for UAV to follow on, the number of waypoints should increase. Some researchers [14]-[17] have noticed the importance of the smoothness of a path, and they adopt the B-spline curves and Bezier curves instead of the line segments used in this paper. By using those curves, the path can be easily smoothed with less waypoints although it costs additional computational times at each generation. In this paper, the reason of not using these curves is that we want to test the scalability of our planner since in some real applications, the mission space can be extremely complex and large.

In this new planners, there are only two parameters, i.e., the population size  $N_p$  and the number of waypoints  $N_w$  of a path. They are set as:  $N_w = 10$ ,  $N_p = 120$  for 7 ADUs problem,  $N_w = 12$ ,  $N_p = 200$  for 15 ADUs problem,  $N_w = 15$ ,  $N_p = 300$  for 30 ADUs problem,  $N_w = 20$ ,  $N_p = 500$  for 60 ADUs problem.

The whole experiment consists of two parts. First, the new planner runs 15 times for each problem to show its effectiveness. Second, in order to test how the new coordinate system proposed helps improve the ability of the new planner,

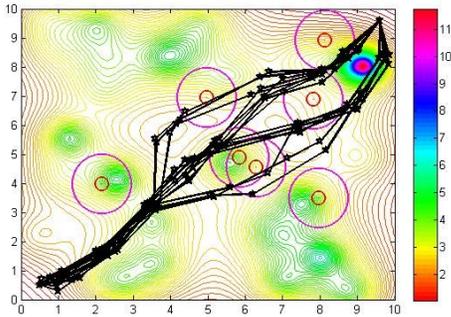


Fig. 5. 15 paths in the 7 ADUs problem.

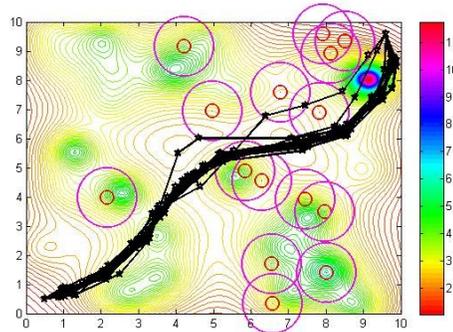


Fig. 6. 15 paths in the 15 ADUs problem.

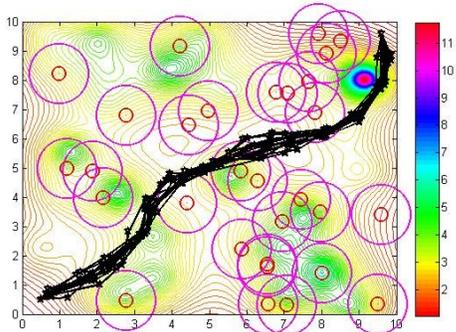


Fig. 7. 15 paths in the 30 ADUs problem.

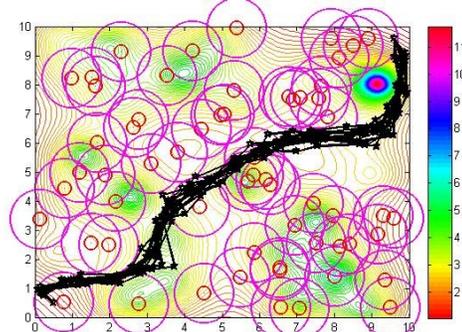


Fig. 8. 15 paths in the 60 ADUs problem.

we construct and test two other UMDA<sub>c</sub> based planners with the Cartesian coordinate system and polar coordinate system, respectively. These three path planners share the same parameters. The only difference among them is the representation of waypoints. All the experiments stop at 100 generations.

### B. The Results and Analyses

In these figures, the color bar indicates the height of areas. There are almost 9 outstanding mountains, shown in dense contour lines. The horizontal mission space is within the range of  $[0,10]^2$ . The start is fixed at (0.5,0.5) and the destination is fixed at (9.6,9.6). The radars detection ranges are shown in bigger circles and the missiles destruction ranges are shown in smaller circles.

1) *The effectiveness of the new planner:* The 15 optimal paths of each problem is shown in Figs.5-8 with black line segments. From these figures we can see the effectiveness of this new planner in the following aspects:

**Stability:** All the paths of the first 3 problems are safe and quite short. As seen in Figs.5-7, all the paths successfully avoid the missiles destruction range and the mountains, where the UAV may be destroyed. To be specific, near the destination is the highest mountain, which the UAV should carefully avoid. Besides, some of the randomly initialized ADUs just stand by that mountain. Thus, there is quite narrow space for safe paths, i.e., zero probability for destruction. As seen that all the paths pass-by the highest mountain from its both sides safely and keep quite far distances to the ADUs. There are two paths in the missile destruction range at almost

(5.0,7.0) in Fig.5, however, the PKill of those two paths are 0. This is because the paths are above the missile destruction range since that ADU stands at a very low height. It can be seen that almost all the paths go through the radars detection range in some places. The reason of such situation is that: as the priority of PLR is higher than PRD, the planner would rather choose the path with smaller PLR than with lower PRD. In this case, all the path length is quite short and the PLR of most paths vary from 1.1 to 1.3. Even that, the PRD is low enough.

In the fourth problem, as the number of ADUs comes up to 60, there are quite few safe space where the optimal paths can happen. As seen in Fig.8, all the paths avoid all the missiles destruction range but one at (5.8, 5.6). Although this single ADU may destroy the UAV, those paths are quite near optimal. We guess the situation in Fig.8 is because 100 generations are not sufficient for the planner to obtain an optimal path, i.e., the waypoints in each subspace have not really converged yet.

**Smoothness:** All the paths in Figs.5-8 are smooth as the maximal turning angle is set to be  $45^\circ$ . Since the paths in Fig.8 are just near optimal, some of them are not smooth. For example, all the paths turns sharply at the second waypoint. And the reason is that one randomly initialized ADU is just in front of the start, which heavily restricts the choice for the second waypoint.

**Scalability:** These 4 problems have different number of ADUs, making the number of waypoints various. In that way, the search space of the 4 problems are quite distinct. As

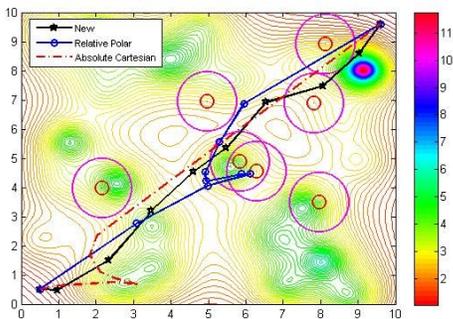


Fig. 9. The paths on 7 ADUs problem produced by the planner with the new coordinate, absolute Cartesian coordinate and relative coordinate are presented in 'start-line', 'dot dash-line' and 'circle-line', respectively.

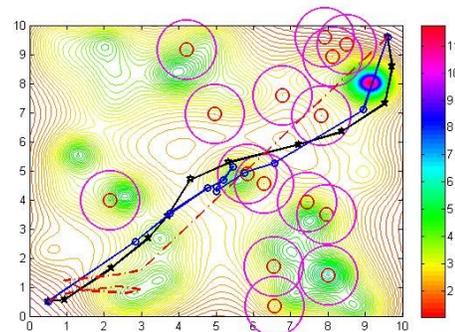


Fig. 10. The paths on 15 ADUs problem produced by the planner with the new coordinate, absolute Cartesian coordinate and relative coordinate are presented in 'start-line', 'dot dash-line' and 'circle-line', respectively.

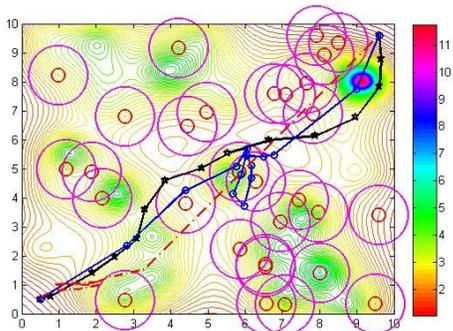


Fig. 11. The paths on 30 ADUs problem produced by the planner with the new coordinate, absolute Cartesian coordinate and relative coordinate are presented in 'start-line', 'dot dash-line' and 'circle-line', respectively.

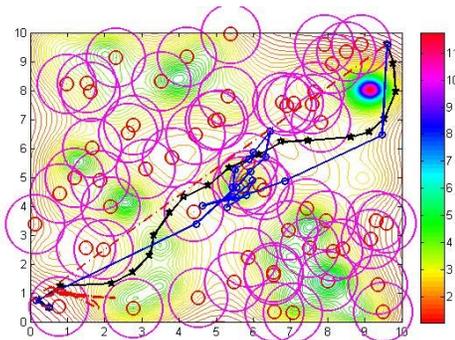


Fig. 12. The paths on 60 ADUs problem produced by the planner with the new coordinate, absolute Cartesian coordinate and relative coordinate are presented in 'start-line', 'dot dash-line' and 'circle-line', respectively.

analyzed above, the new planner performs optimal or near optimal on these 4 problems, which proves the new planner is scalable and are competent on both simple and complex problems.

Notice that the diversity of paths decreases from the simplest problem to the hardest problem. This is reasonable that as the ADUs increase, the safe areas in mission space become narrower, and the local optimal areas in the search space get smaller. On one hand, it gets increasingly difficult to find the local optima in search space, i.e., the optimal paths in mission space. On the other hand, once two optimal paths in mission space are found, they are very likely to belong to the same local optimal area in the search space.

2) *The other coordinate systems based path planners:* Figs.9-12 show the path produced by the UMDA<sub>c</sub> based path planners in the codification of the new coordinate system, Cartesian coordinate system and polar coordinate system, respectively in ‘start-line’, ‘dot dash-line’ and ‘circle-line’. As seen that the previous two coordinate systems based planners perform poorly on all these complex problems. The reason of their poor performance is mainly because the search space becomes quite large as the problems get complex. Oppositely, as the new coordinate system has successfully reduce the search space, the new planner works quite well on these four problems.

## VI. CONCLUSION AND FUTURE WORK

In this paper, the problem of 3-D offline path planning for single UAV is studied. A new coordinate system is used to remedy the drawbacks of two commonly used coordinate systems by rotating the Cartesian coordinate system and dividing the whole mission space into several subspaces, which heavily reduces the search space. Based on this new coordinate system, EDAs are introduced into UAV path planning problem. The motivation of introducing EDAs is to save the fine-tuning time of reproductive parameters and guide the search for optimal waypoint in each subspace. Specifically, one of the simplest EDA, i.e., UMDA<sub>c</sub>, is used to construct a new path planner. Generally speaking, the new coordinate system is also suit for the traditional EAs.

In the future, some other EDAs may be adopted to reveal the interdependencies among variables.

## VII. ACKNOWLEDGMENT

This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grants 61175065 and 61329302, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, the Science and Technological Fund of Anhui Province for Outstanding Youth under Grant 1108085J16, and the European Union Seventh Framework Programme under Grant 247619.

## REFERENCES

- [1] L. R. Newcome, *Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles*, Library of Flight Series, AIAA, Reston, 2004.
- [2] Sarris, Zak, and STN ATLAS. Survey of UAV applications in civil markets (June 2001). *The 9 th IEEE Mediterranean Conference on Control and Automation (MED'01)*. 2001.
- [3] C. Goerzen, Z. Kong, B. Mettler. A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *Journal of Intelligent and Robotic Systems*. 57(1-4). Jan. 2010. Pages 65-100.
- [4] Jia Chenye. UAV Mission Planning: Problem Modeling and Solution Methods. *PhD thesis*, 2011, National University of Singapore.
- [5] R. J. Szczerba, P. Galkowski, I. S. Glicktein, and N. Ternullo. Robust algorithm for real-time route planning, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 869–878, Jul. 2000.
- [6] K. I. Trovato. A\* Planning in Discrete Configuration Spaces of Autonomous Systems. *PhD thesis*, Amsterdam University, 1996.
- [7] A. G. Richards, J. P. How. Aircraft Trajectory Planning with Collision Avoidance Using Mixed-Integer Linear Programming. *In Proceeding of the American Control Conference* (May 2002).
- [8] J. J. Ruz, O. Arévalo, de la Cruz, J.M, and G. Pajares. Using MILP for UAVs Trajectory Optimization under Radar Detection Risk. *In Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation* (Prague, September 2006).
- [9] A.Raghunathan, V.Gopal, D.Subramanian, L.Biegler and T.Samad. Dynamic Optimization Strategies for 3D Conflict Resolution of Multiple Aircraft. *AIAA Journal of Guidance, Control and Dynamics*, 27 (4). (2004), 586-594.
- [10] P. Bhattacharya and M. L. Gavrilova. Voronoi diagram in optimal path planning, *in Proc. IEEE Int. Symp. on Voronoi Diagrams in Science and Engineering*, 2007, pp. 38-47.
- [11] R. W. Beard. Coordinate system target assignment and intercept for unmanned air vehicles, *IEEE Trans. on Robotics and Automation*, vol. 18, no. 6, pp. 911-922, Dec. 2002.
- [12] C. Zheng, L. Li, F. Xu, F. Sun, M. Ding, Evolutionary route planner for unmanned air vehicles, *IEEE Transactions on Robotics* 21 (2005) 609–620.
- [13] J.M. de la Cruz, E. Besada-Portas, L. de la Torre, B. Andres-Toro, J.A. Lopez-Orozco, Evolutionary path planner for UAVs in realistic environments, *in: Proceedings of Genetic Evolutionary Computation Conference*, pp. 1447–1484.
- [14] S. Mittal and K. Deb. Three-dimensional offline path planning for UAVs using multi-objective evolutionary algorithms, *in Proc. IEEE Congr.Evol. Comput.*, 2007, vol. 7, pp. 3195–3202.
- [15] E. Besada-Portas, L. de la Torre, J.M. de la Cruz, B. Andres-Toro, Evolutionary trajectory planner for multiple UAVs in realistic scenarios, *IEEE Transactions on Robotics*. 26 (2010) 619–634.
- [16] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras. Evolutionary algorithm based offline/online path planner for UAV navigation, *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.
- [17] I. Hasircioglu, H. R. Topcuoglu, and M. Ermis. 3-d path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms, *in Proc. Genet. Evol. Comput. Conf.*, 2008, pp. 1499–1506.
- [18] I. K. Nikolos, N. C. Tsourveloudis, and K. P. Valavanis. Evolutionary algorithm based path planning for multiple UAV cooperation, *in Advances in Unmanned Aerial Vehicles*. Berlin, Germany: Springer-Verlag, Jan. 2007, pp. 309–340.
- [19] R. Zhang, C. Zheng, and P. Yan. Route planning for unmanned air vehicles with multiple missions using an evolutionary algorithm, *in Proc. IEEE 3rd Int. Conf. Nat. Comput.*, 2007, pp. 1499–1506.
- [20] De La Cruz, J.M., E. Besada-Portas, L. de la Torre, B. Andres-Toro, and J. A. Lopez-Orozco. Evolutionary path planner for UAVs realistic environments. *in 10th Annual Genetic and Evolutionary Computation Conference, GECCO 2008*, July 12, 2008 - July 16, 2008. 2008. Atlanta, GA, United states: Association for Computing Machinery.
- [21] R. Armananzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. V. de Peer, R. Blanco, V. Robles, C. Bielza, and P. Larranaga, “A review of estimation of distribution algorithms in bioinformatics,” *BioData Min.*, vol. 1, no. 6, pp. 1–12, Sep. 2008.
- [22] P. Larranaga and J. A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Norwell, MA: Kluwer, 2002.
- [23] C. M. Fonseca and P. J. Fleming. Multi-objective optimization and multiple constraint handling with evolutionary algorithms—Part I: Unified formulation, *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 18, no. 1, pp. 26–37, Jan. 1988.