

Path Planning of Aircraft Based on Adaptive Multiobjective Estimation of Distribution Algorithm

Tao Lin

School of Astronautics,
Harbin Institute of Technology
Harbin 150001, China
laestsea2005@163.com

Ke Zhang

Beijing Electro-mechanical
Engineering Institute
Beijing 100074, China
zhangkeh16@yahoo.com

Naigang Cui

School of Astronautics,
Harbin Institute of Technology
Harbin 150001, China
cui_naigang@163.com

Zhenbiao Tu and Hu Zhang

Beijing Electro-mechanical
Engineering Institute
Beijing 100074, China
jxzhanghu@126.com

Abstract—Path planning is able to effectively improve the survival probability and operational efficiency of a combat aircraft. The essence of path planning of the aircraft is a multiobjective optimization problem. To deal with this problem efficiently, this paper proposes an adaptive multiobjective estimation of distribution algorithm named as AMEDA. In AMEDA, a novel clustering-based multivariate Gaussian sampling strategy is designed. At each generation, a clustering analysis approach is utilized to discover the distribution structure of the population. Based on the distribution information, with a certain probability, a local or a global multivariate Gaussian model (MGM) is built for each solution to sample a new solution. A covariance sharing strategy is designed in AMEDA to reduce the complexity of building MGMs, and an adaptive update strategy of the probability that controls the contributions of the two types of MGMs is developed to dynamically balance exploration and exploitation. Experiments show that AMEDA is efficient to deal with the path planning model of the aircraft. Meanwhile, it is convenient to provide multiple flight paths with different characteristics for the decision makers.

I. INTRODUCTION

The characteristic of modern warfare has been gradually changed from mass destruction to accurate and effective strike. For a combat aircraft, a good flight path is able to effectively improve its survival probability and operational efficiency. Therefore, path planning is very important for the aircrafts. Typical path planning approaches include [1]: A* search algorithm, dynamic programming algorithm, artificial potential field algorithm and evolutionary algorithms (EAs), etc. Especially in recent years, with the rapid development of EAs and their wide applications in dealing with complicated engineering optimization problems, the scholars have carried out a lot of research works on path planning of aircrafts based on EAs [1], [2]. EA [3] is a type of nature inspired global searching algorithm, its solving operations include crossover, mutation and selection, etc. Since the operations do not depend on the properties of the problems to be solved, thus EA is quite suitable to be applied to deal with the complicated optimization problems which are hard for the traditional deterministic optimization approaches.

Since the path planning of aircrafts usually requires to take multiple conflicting objectives into consideration [1], such as, the shortest total length of flight path, the lowest flight altitude, and the successful avoidance of all the threats, etc., the essence

of aircrafts' path planning is a complicated multiobjective optimization problem (MOP). However, at present, most of the existing works usually use aggregation techniques to transform multiple objectives into a single objective for solving the problems [2], and typical aggregation techniques are weighted sum approach and constraint approach. Actually, the aggregation techniques have their own shortages: (a) the solutions of the MOPs obtained by the aggregation techniques greatly depend on the adopted aggregation parameters, such as the weights for each objective and the constraint parameters, etc.; (b) for an actual path planning model, the aggregation parameters are often hard to be set, especially when the mission information is unknown, the parameter settings are much harder; (c) a suite of aggregation parameters can only yield one flight path, but during decision making, the decision makers often hope to have more paths with different features for their decision. Therefore, to improve the shortages of aggregation techniques, it is necessary to study much more efficient path planning approaches of the aircrafts. Multiobjective evolutionary algorithm (MOEA) [4], [5], [6] is a type of EAs which can deal with MOPs efficiently. Since MOEA is able to obtain a set of solutions for a MOP through a single run, that is to say, it is able to yield multiple flight paths for a path planning model though a single run, consequently, more and more scholars start to perform path planning using the MOEAs [2], [7]. Similarly, this paper proposes to study path planning based a multiobjective estimation of distribution algorithm (MEDA) [8].

Estimation of distribution algorithm (EDA) [9] is a specialized paradigm in EAs. An EDA does not adopt the well-known traditional genetic operations (e.g. crossover and mutation). Instead, it explicitly extracts globally statistical information from the selected solutions, and builds a posterior probability distribution model of promising solutions based on the extracted statistical information. New solutions are sampled from the model thus built. In the genetic-based MOEAs, the genetic operations may disrupt the building of strong schemas of a population, and thus movement towards optimal is extremely difficult to predict. However, MEDAs are able to predict the locations or patterns of the PFs or to predict the favorable movements in the search spaces. By regulating the search to follow the discovered or predicted favorable movement

directions, promising solutions can be generated better [10]. A variety of MEDAs have been proposed and they yielded lots of encouraging results [8].

Although MEDAs have been studied by more and more scholars, they have not completely lived up to their a priori expectations until now. This can be attributed to a number of different causes, and three of them are the incorrect treatment of population outliers; the loss of population diversity, and too much computational effort being spent on finding an optimal population model [11]. To improve the effectiveness of path planning, this paper modifies the sampling strategy for generating new solutions, and proposes an adaptive multiobjective estimation of distribution algorithm (AMEDA) to address the path planning model. The features of AMEDA include:

- A novel clustering-based multivariate Gaussian sampling (CMGS) strategy is designed. In CMGS strategy, a clustering analysis approach is applied to cluster the population. To model the population more accurately and to enhance the searching ability of AMEDA, with a certain probability, a local multivariate Gaussian model (MGM) or a global MGM is built for each solution based on the clusters to sample a new solution. To reduce the modeling complexity, the solutions in the same cluster share the same covariance matrix to build MGMs.
- An adaptive update strategy of the probability that controls the contributions of local and global MGMs is developed. In the strategy, the control probability is adaptively updated at each generation according to the reproduction utility of the two types of MGMs over the last certain generations, so that the balance between exploration and exploitation can be dynamically maintained.

The rest of this paper is organized as follows. Section II introduces the path planning model. Section III describe the proposed AMEDA in detail. Section IV presents the experimental results of applying AMEDA to conduct path planning. The paper is concluded in Section V.

II. PATH PLANNING MODEL OF AIRCRAFT

Path planning model of an aircraft includes representations of terrain and threats, design of optimization objectives, constraint conditions and optimization variables.

A. Representations of Terrain and Threats

1) *Original Digital Terrain*: Eq.(1) is used to simulated the original digital terrain.

$$z_1(x, y) = \sin(y + a) + c \cos\left(d\sqrt{y^2 + x^2}\right) + b \sin(x) + e \sin\left(e\sqrt{y^2 + x^2}\right) + f \cos(y), \quad (1)$$

where x and y represent horizontal and vertical coordinates of a point in the horizontal plane, respectively, and z_1 represents terrain altitude corresponding to point (x, y) . Five constants, including a, b, c, d, e, f are the terrain coefficients. Through adjusting the terrain coefficients, it is able to generate a variety of terrains with different features. The simulated terrains can be utilized as the explored terrain to provide basic data for the aircraft path planning.

2) *Equivalent Terrains of Threats*: During the flight, aircrafts usually encounter a variety of threats including peaks, enemy defense areas, etc. In order to conduct path planning conveniently, all the threats are transferred into peak terrains and contained in the path planning model of this paper. Eq. (2) are adopted to yield peak terrains.

$$z_2(x, y) = \sum_{i=1}^k \mathcal{H}(i) \exp\left(-\frac{(x - \mathcal{X}_c(i))^2}{\mathcal{X}_t(i)} - \frac{(y - \mathcal{Y}_c(i))^2}{\mathcal{Y}_t(i)}\right), \quad (2)$$

where x and y represent horizontal and vertical coordinates of a point in the horizontal plane, respectively, and z_2 represents terrain altitude corresponding to point (x, y) . k is the number of peak terrains and $\mathcal{H}(i)$ is the height of peak i . $\mathcal{X}_c(i)$ and $\mathcal{Y}_c(i)$ are horizontal and vertical coordinates, respectively, corresponding to the center of peak i . $\mathcal{X}_t(i)$ and $\mathcal{Y}_t(i)$ represent profile parameters of peak i . Through changing the parameters in Eq. (2), it is able to simulate various equivalent terrains of threats with different number and shapes.

3) *Equivalent Digital Terrain*: Based on the principle of information fusion of digital terrains, the equivalent terrains of threats and the original digital terrain are fused to generate the equivalent digital terrain by Eq. (3):

$$z(x, y) = \max(z_1(x, y), z_2(x, y)). \quad (3)$$

In Eq. (3), through selecting much higher terrains, it is able to reflect the influence range of the threats more realistically. Digital terrain obtained by Eq. (3) is applied to conduct path planning directly.

B. Optimization Objectives

Total length of flight path, flight altitude and turning angle are selected as the optimization objectives of the path planning.

1) *Total Length of Flight Path*: In order to save fuel and fight time, the total length of flight path of aircraft from the start point to the target point is required as short as possible. Supposing $\mathbf{p}^0, \mathbf{p}^1, \dots, \mathbf{p}^n, \mathbf{p}^{n+1}$ are waypoints, thus $\mathbf{p}^0, \mathbf{p}^{n+1}$ represent start point and target point, respectively, $\mathbf{p}^1, \dots, \mathbf{p}^n$ denote n waypoints obtained by planning, and $\mathbf{p}^i = (x_i, y_i, z_i)^\top$ represents 3D coordinate of waypoint i . As a result, total length of the flight path can be expressed below:

$$f_1 = \sum_{i=1}^{n+1} d_{i,i-1}, \quad (4)$$

where $d_{i,i-1} = \|\mathbf{p}^i - \mathbf{p}^{i-1}\|$ represents distance between waypoints i and $i-1$.

2) *Flight Altitude*: When the aircraft passes through enemy defense area, flying along the terrain can strengthen ground cover for it. Therefore, under the premise of satisfying required lowest flight altitude, the flight altitude of the aircraft is the lower the better. The optimization objective of the flight altitude is expressed as follows:

$$f_2 = \sum_{i=1}^n z_i, \quad (5)$$

where z_i is the flight altitude of waypoint i .

3) *Turning Angle*: Due to the limited maneuverability of the aircraft, thus when flying, under the constraint of allowed maximum turning angle, the turning angle of the aircraft should be as small as possible. The optimization objective of the aircraft's turning angle is expressed as follows:

$$f_3 = \sum_{i=1}^n (1 - \cos(\theta_i)) = \sum_{i=1}^n \left(1 - \frac{\mathbf{a}_i^T \mathbf{a}_{i+1}}{\|\mathbf{a}_i\| \|\mathbf{a}_{i+1}\|} \right), \quad (6)$$

where \mathbf{a}_i represents the projection of path segment i on the horizontal plane, that is, $\mathbf{a}_i = (x_i - x_{i-1}, y_i - y_{i-1})^T$.

C. Constraint Conditions

Due to the limitation of aircrafts design performance and according to the requirements of tactical use, aircraft is required to satisfy various constraints when flying. In this paper, constraints to be considered are as follows:

1) *Allowed Highest Flight Altitude*: In order to guarantee the flight safety, the aircraft must fly below a certain altitude. Supposing the allowed highest flight altitude of the aircraft is H_{\max} , and the flight altitude of path segment i is H_i , so the constraint follows:

$$H_i \leq H_{\max}, \quad i = 1, \dots, n+1. \quad (7)$$

2) *Required Lowest Flight Altitude*: The optimization objective f_2 requires the flight altitude of aircraft to be as low as possible so as to improve penetration effect. Actually, the flight altitude is not the lower the better, since a quite low flight altitude will increase the probability of the aircraft's crash. The flight of the aircraft should be as low as possible under the condition of satisfying required lowest flight altitude. Supposing the required lowest flight altitude of the aircraft is H_{\min} , so the constraint follows:

$$H_i \geq H_{\min}, \quad i = 1, \dots, n+1. \quad (8)$$

3) *Required Shortest Length of Path Segment*: In general, in order to fulfill flight mission effectively, the aircraft is required to fly straightly for a certain distance before and after maneuvering. This distance is defined as the minimum step size L_{\min} . and the length of each path segment should be longer than or equal to L_{\min} , that is to say:

$$d_{i,i-1} \geq L_{\min}, \quad i = 1, \dots, n+1. \quad (9)$$

4) *Allowed Maximum Turning Angle*: The design performance of the aircraft determines that aircraft can only turn within the range of the predefined allowed maximum turning angle. Supposing the allowed maximum turning angle of the aircraft is ϕ , and turning angle i of the aircraft is θ_i , thus the constraint of turning angle is as follows:

$$\begin{aligned} \cos(\theta_i) &\geq \cos(\phi), \quad i = 1, \dots, n \\ \cos(\theta_i) &= \frac{\mathbf{a}_i^T \mathbf{a}_{i+1}}{\|\mathbf{a}_i\| \|\mathbf{a}_{i+1}\|}. \end{aligned} \quad (10)$$

5) *Allowed Maximum Total Length of Flight Path*: There exists a limit of allowed maximum flight distance for each type of aircraft. Supposing the maximum allowable flight distance of aircraft is expressed as L_{\max} . The constraint follows:

$$\sum_{i=1}^{n+1} d_{i,i-1} \leq L_{\max}. \quad (11)$$

6) *Allowed Maximum Climb Angle/Dive Angle*: Since restricted by the restriction of longitudinal overload, the allowed maximum climb/dive ability of the aircraft is limited. The allowed maximum climb angle/ dive angle of the aircraft is supposed to be φ . The constraint follows:

$$\frac{|z_i - z_{i-1}|}{\|\mathbf{a}_i\|} \leq \tan(\varphi), \quad i = 1, \dots, n. \quad (12)$$

D. Optimization Variables

In the path planning model of an aircraft, the variables that need to be optimized are the coordinates (x, y, z) of all the waypoints. In this paper, the coordinates are directly employed to encode for the path planning. In the operations of AMEDA, each solution is represented by $(x_1, \dots, x_n, y_1, \dots, y_n, z_1, \dots, z_n)^T$, that is to say, each solution actually denotes a set of waypoints.

III. PROPOSED ALGORITHMS

A. Framework

The basic idea of AMEDA is that more MGMs are built by using a computationally cheap approach to describe the population more accurately; a number of different MGMs are employed to generate diverse new solutions, and the contributions of different types of MGMs are adaptively controlled to dynamically maintain the balance between exploration and exploitation. Algorithm 1 presents the framework of AMEDA.

Algorithm 1 AMEDA

- 1: Initialize population $\mathcal{P} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$, control probability β .
 - 2: **for** $t = 1$ to T **do**
 - 3: Set an auxiliary archive $\mathcal{A} = \emptyset$.
 - 4: Partition the population into local clusters, $\{\mathcal{LC}^1, \dots, \mathcal{LC}^K\} = \text{AHC}(\mathcal{P}, K)$.
 - 5: Construct a global cluster \mathcal{GC} .
 - 6: Calculate covariances of \mathcal{LC}^k and \mathcal{GC} as $\Sigma_{\mathcal{LC}^k}$ and $\Sigma_{\mathcal{GC}}$, respectively, $k = 1, \dots, K$.
 - 7: **for** each $\mathbf{x}^i \in \mathcal{P}$, $i = 1, \dots, N$ **do**
 - 8: Decide a covariance matrix Σ^i for \mathbf{x}^i , $\Sigma^i = \begin{cases} \Sigma_{\mathcal{LC}^k} & \text{if } \text{rand}() < \beta \\ \Sigma_{\mathcal{GC}} & \text{otherwise} \end{cases}$.
 - 9: Generate a new solution $\mathbf{y}^i = \text{SolGen}(\Sigma^i, \mathbf{x}^i)$.
 - 10: Store the new solution $\mathcal{A} = \mathcal{A} \cup \{\mathbf{y}^i\}$.
 - 11: **end for**
 - 12: Update the population $\mathcal{P} = \text{EnvSel}(\mathcal{A} \cup \mathcal{P})$.
 - 13: Update β based on reproduction utility.
 - 14: **end for**
 - 15: **return** final population \mathcal{P} .
-

In Algorithm 1, N is the population size. K is the maximum number of clusters defined in agglomerative hierarchical clustering (AHC) [12] approach. T is the maximum evolutionary

generations. \mathcal{GC} and \mathcal{LC}^k denote the global cluster and the k -th local cluster, respectively. $\Sigma_{\mathcal{LC}^k}^i$ is the covariance matrix of the k -th local cluster that \mathbf{x}^i locates in. β is a probability to control which covariance matrix is decided for \mathbf{x}^i . $\text{rand}()$ generates a uniformly distributed random number in $[0, 1]$.

At each generation of AMEDA, the population is firstly partitioned into K local clusters by a AHC approach (line 4), and a solution is randomly selected from each local cluster to construct a global cluster (line 5). Afterwards, the covariance matrix of the global cluster and all the covariance matrices of the local clusters are estimated as $\Sigma_{\mathcal{GC}}$ and $\Sigma_{\mathcal{LC}^k}$ ($k = 1, \dots, K$) (line 6). Next, for each solution \mathbf{x}^i , a covariance matrix Σ^i is set using $\Sigma_{\mathcal{LC}^k}^i$ or $\Sigma_{\mathcal{GC}}$ with probability β and $1 - \beta$, respectively (line 8), a new solution \mathbf{y}^i is sampled from the MGM built by \mathbf{x}^i and Σ^i (line 9), and \mathbf{y}^i is preserved in an auxiliary archive \mathcal{A} (line 10). After completing solution generation, an environmental selection operation is conducted to update the population (line 12). Finally, a reproduction utility-based update strategy is applied on β to get a new probability of building different types of MGMs for the next generation (line 13). With respect to the AMEDA framework, there are following comments should be noted.

- A general agglomerative AHC approach [12] is employed to proceed clustering. The group average linkage algorithm is used in the AHC of AMEDA to define the distance between two clusters. Detailed descriptions of the agglomerative AHC refer to [12].
- The MGM built based on a local or a global cluster is called a local or a global MGM. Sampling from the local MGMs emphasizes on exploitation, and sampling solutions from the global MGM is beneficial to exploration. The local and global MGMs work together to maintain the balance between exploitation and exploration.
- The covariance matrix of \mathcal{GC} is a $n \times n$ matrix, i.e., $\Sigma_{\mathcal{GC}} = [\text{Cov}(x_s, x_t)]$, $s, t = 1, \dots, n$. $\text{Cov}(x_s, x_t)$ denotes the covariance between the elements x_s and x_t of a solution \mathbf{x} . $\Sigma_{\mathcal{GC}}$ is estimated by the equation below.

$$\text{Cov}(x_s, x_t) = \frac{1}{|\mathcal{GC}| - 1} \sum_{\mathbf{x} \in \mathcal{GC}} (x_s - x_t)^2$$

for $s, t = 1, \dots, n$. The covariance matrix $\Sigma_{\mathcal{LC}^k}$ ($k = 1, \dots, K$) can also be estimated by the approach for estimating $\Sigma_{\mathcal{GC}}$.

- If any one of the local clusters includes only one solution, the global covariance matrix will be used for the solution to build a MGM.

B. Solution Generation

SolGen operator in line 9 of Algorithm 1 aims to produce new solutions, which is implemented by MGMs sampling and polynomial mutation [13] approaches shown in Algorithm 2. When producing an offspring, a MGM is firstly built to sample a new trial solution (lines 1-3). Afterwards, a repair operation is performed to make the solution feasible (line 4). Next, the trial solution is mutated by a polynomial mutation operator

(line 5). Finally, the mutated solution is repaired again to ensure its feasibility (line 6).

Algorithm 2 SolGen(Σ^k, \mathbf{x}^k)

- 1: Take the Cholesky method to decompose the covariance matrix Σ^k to obtain a lower triangular matrix Λ , and $\Sigma^k = \Lambda\Lambda^\top$.
- 2: Generate a vector $\mathbf{v} = (v_1, \dots, v_n)^\top$, where $v_i \sim N(0, 1)$, $i = 1, \dots, n$, obey the unit Gaussian distribution.
- 3: Produce a trial solution $\mathbf{y}' = \mathbf{x}^k + \Lambda\mathbf{v}$, $\mathbf{y}' = (y'_1, \dots, y'_n)^\top$.
- 4: Repair the solution

$$y_i'' = \begin{cases} x_i^k - \text{rand}() \cdot (x_i^k - a_i) & \text{if } y'_i < a_i \\ x_i^k + \text{rand}() \cdot (b_i - x_i^k) & \text{else if } y'_i > b_i \\ y_i & \text{otherwise} \end{cases},$$

for $i = 1, 2, \dots, n$.

- 5: Mutate the solution

$$y_i''' = \begin{cases} y_i'' + \delta_i \times (b_i - a_i) & \text{if } \text{rand}() < p_m \\ y_i'' & \text{otherwise} \end{cases}$$

with

$$\delta_i = \begin{cases} [2r + (1 - 2r) \left(\frac{b_i - y_i''}{b_i - a_i} \right)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} - 1 & \text{if } r < 0.5 \\ 1 - [2 - 2r + (2r - 1) \left(\frac{y_i'' - a_i}{b_i - a_i} \right)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} & \text{otherwise} \end{cases},$$

where a_i and b_i denote the lower and upper boundaries of the i -th variable, respectively, $i = 1, \dots, n$; p_m is the mutation probability, and η_m denotes the distribution index of mutation; $r = \text{rand}()$.

- 6: Repair the solution

$$y_i^k = \begin{cases} a_i & \text{if } y_i''' < a_i \\ b_i & \text{else if } y_i''' > b_i \\ y_i''' & \text{otherwise} \end{cases}, i = 1, 2, \dots, n.$$

- 7: **return** the new solution $\mathbf{y}^k = (y_1^k, \dots, y_n^k)^\top$.
-

Different from the traditional MEDA [14] that only one MGM is built for a cluster, in AMEDA, each solution \mathbf{x}^i is assumed to be from a different MGM which adopts \mathbf{x}^i as the mean vector, i.e., $\mathbf{x}^i \sim N(\mathbf{x}^i, \Sigma)$. With a probability β , a local MGM is built for \mathbf{x}^i to sample a solution, and the covariance matrix of the local cluster that \mathbf{x}^i locates in is used to build the local MGM. Otherwise, with a probability $1 - \beta$, a global MGM is built for \mathbf{x}^i to sample a solution, and the covariance matrix of the global cluster is employed to build the global MGM. We have the following comments on the solution generation in AMEDA.

- The distribution structure of solutions in a population is usually strongly nonlinear. In AMEDA through building a MGM for each solution, the MGMs can work together to capture the population structure more accurately.
- In AMEDA, a different MGM is build for each solution to sample only one offspring. By this way, more diverse solutions can be produced, and the searching ability for the variable space can be enhanced greatly.
- AMEDA samples offsprings based on each solution considers outliers enough, which is also helpful for strengthening the searching ability of the algorithm.
- The solutions in the same cluster are set to share a same covariance matrix to build MGMs for themselves, the

modeling cost can be saved.

C. Environmental Selection

The $EnvSel(\mathcal{A} \cup \mathcal{P})$ operator in line 12 of Algorithm 1 is to preserve the promising solutions in $\mathcal{A} \cup \mathcal{P}$ survive into the next generation. A hypervolume metric-based environmental selection approach modified from the environmental selection in SMS-EMOA [15] is used in AMEDA. Algorithm 3 presents the details of the environmental selection in AMEDA.

Algorithm 3 $EnvSel(\mathcal{A} \cup \mathcal{P})$

1: Partition $\mathcal{A} \cup \mathcal{P}$ into L fronts by fast nondominated sorting approach

$$\{\mathcal{B}_1, \dots, \mathcal{B}_L\} = FastNondominatedSorting(\mathcal{A} \cup \mathcal{P}).$$

2: Copy better solutions,

$$\mathcal{P}' = \{\mathbf{x}^i | \mathbf{x}^i \in \bigcup_{j=1}^l \mathcal{B}_j, \bigcup_{j=1}^{l-1} \mathcal{B}_j < N \wedge \bigcup_{j=1}^l \mathcal{B}_j \geq N, l < L\}.$$

3: **if** $l > 1$ **then**

4: **while** $|\mathcal{P}'| > N$ **do**

5: $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{P}'} d(\mathbf{x}, \mathcal{P}')$.

6: Set $\mathcal{P}' = \mathcal{P}' \setminus \{\mathbf{x}^*\}$.

7: **end while**

8: **else**

9: **while** $|\mathcal{P}'| > N$ **do**

10: $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{P}'} \Delta_\varphi(\mathbf{x}, \mathcal{P}')$.

11: Set $\mathcal{P}' = \mathcal{P}' \setminus \{\mathbf{x}^*\}$.

12: **end while**

13: **end if**

14: Set $\mathcal{P} = \mathcal{P}'$.

15: **return** \mathcal{P} .

At first, a large population $\mathcal{A} \cup \mathcal{P}$ is constructed by combining the current population \mathcal{P} and the auxiliary archive \mathcal{A} consisted of new solutions, and the fast nondominated sorting approach proposed in NSGA-II [16] is used to partition $\mathcal{A} \cup \mathcal{P}$ into L different nondominated fronts $\{\mathcal{B}_1, \dots, \mathcal{B}_L\}$, where \mathcal{B}_1 and \mathcal{B}_L represent the best and the worst fronts, respectively. Afterwards, all the solutions in $\mathcal{B}_j, j = 1, \dots, l, l < L$ are copied into a transitional archive \mathcal{P}' until $|\mathcal{P}'| \geq N$. if $l > 1$, i.e., the solutions in more than one front are put into \mathcal{P}' , the solutions with the maximum $d(\mathbf{x}, \mathcal{P}')$ are removed one by one until $|\mathcal{P}'| = N$, where $d(\mathbf{x}, \mathcal{P}')$ is the number of solutions in \mathcal{P}' that dominate \mathbf{x} . Otherwise, the solutions with the minimum hypervolume contribution $\Delta_\varphi(\mathbf{x}, \mathcal{P}')$ are deleted one by one until the size of \mathcal{P}' equals to N , where Δ_φ is calculated by the approach in [15]. After $|\mathcal{P}'| = N$, \mathcal{P}' is transferred to \mathcal{P} as the population of next generation.

D. Adaptive Probability Update Strategy

In AMEDA, local and global MGMs are used to sample new solutions for emphasizing on exploitation and exploration, respectively. To balance the exploitation and exploration, a probability β is applied to control the contributions of the different types of MGMs. In practice, different β values need to be given for different MOPs. Even at different evolutionary

stages, β may also require different values. Thus setting an adaptive β is necessary for AMEDA. A reproduction utility-based approach is proposed to adaptively adjust the β value at each generation (line 13 of Algorithm 1), which is shown in Algorithm 4.

Algorithm 4 Probability Update

1: Estimate the reproduction utility of the MGMs built based on local clusters or the global cluster at the t -th generation:

$$u_t^{LC(GC)} = \begin{cases} \frac{\sum_{k=1}^t S_k^{LC(GC)}}{\sum_{k=1}^t Q_k^{LC(GC)}} & \text{if } t < HL \\ \frac{\sum_{k=t-HL+1}^t S_k^{LC(GC)}}{\sum_{k=t-HL+1}^t Q_k^{LC(GC)}} & \text{otherwise} \end{cases},$$

where $u_t^{LC(GC)}$ is the reproduction utility of local MGMs (the global MGM) over the previous HL generations. $Q_k^{LC(GC)}$ represents the number of new solutions sampled from the local MGMs (the global MGM) at the k -th generation. $S_k^{LC(GC)}$ is the number of successful solutions sampled from the local MGMs (the global MGM) at the k -th generation, where successful solutions indicate the solutions that enter the next generation successfully.

2: Calculate the mating restriction probability for the $(t+1)$ -th generation:

$$\beta_{t+1} = \frac{u_t^{GC} + \varepsilon}{u_t^{GC} + u_t^{LC} + \varepsilon},$$

where $\varepsilon = 10^{-10}$ is employed to ensure the legality of the calculation.

3: **return** β_{t+1} .

It should be noted that, at each generation five solutions are forced to be sampled from local and global MGMs, respectively, so that there always exist more than five new solutions generated from each different type of MGMs, which is with the purpose of keeping the update of β to be available all the time.

IV. EXPERIMENTAL STUDIES

A. Performance Metric

It is well-known that a MOP has a set of Pareto optimal solutions called Pareto set (PS). The set of objective points of Pareto optimal solutions is named as Pareto front (PF). A MOEA is able to obtain an approximated solution set of a MOP through a single run. In general, it is hoped that the corresponding front of approximated solution set (approximated front for short) approaches to the PF as close as possible (convergence), the objective points spread along the PF as widely as possible (diversity) and distribute along the PF as uniformly as possible (uniformity).

A commonly used quality indicator called hypervolume (HV) [17] is employed to quantitatively measure the quality of approximated front obtained by the algorithms. Suppose \mathcal{P} is an approximated front. The HV metric is defined as follows:

$$HV(\mathcal{P}, \mathbf{z}) = \text{VOL}(\bigcup_{\mathbf{x} \in \mathcal{P}} [f_1(\mathbf{x}), z_1] \times \dots \times [f_m(\mathbf{x}), z_m]),$$

where $\mathbf{z} = (z_1, \dots, z_m)^\top$ is a reference point in the objective space dominated by any objective points in \mathcal{P} , and $\text{VOL}(\cdot)$ is the Lebesgue measure. HV metric measures the size of the objective space dominated by the objective points in \mathcal{P} and bounded by \mathbf{z} . When calculating the HV metric values of the obtained approximated front of path planning model, the reference point is set as $\mathbf{z} = (500, 5, 15)^\top$.

HV metric can measure the convergence, diversity and uniformity of the obtained approximated fronts. If approximated front \mathcal{P} does not miss any parts of PF, and the objective points approach to PF as close as possible and distribute along the PF as evenly as possible, \mathcal{P} will have a larger HV metric value.

B. Algorithm Parameters

To examine the performance of AMEDA for the path planning model, five state of the art MOEAs, i.e., MOEA/D-DE [18], TMOEA/D [19], RM-MEDA [20], NSGA-II [16] and SMS-EMOA [15] are selected to conduct comparison experiments. The simulated binary crossover operator in original NSGA-II and SMS-EMOA are replaced by the differential evolution (DE) operator to take part in the comparison experiments. The parameters of all the MOEAs are tuned through the preliminary experiments and the best parameter combinations are adopted in the comparison studies. All the parameter settings are as follows:

- *Public parameters:*
 - population size N : $N = 105$ for all the algorithms;
 - variable dimension: $n = 30$ (i.e., 10 waypoints);
 - maximum evolutionary generations: $T = 500$.
- *AMEDA Parameters:*
 - initial control probability: $\beta_0 = 0.9$;
 - history length: $H = 10$;
 - maximum number of clusters: $K = 4$;
 - polynomial mutation: $p_m = 1/n, \eta_m = 20$.
- *MOEA/D-DE Parameters:*
 - neighborhood size: $NS = 5$;
 - probability of neighborhood search: 0.7;
 - maximal number of solution replacement: $n_r = 2$;
 - differential evolution: $F = 0.5, CR = 0.6$;
 - polynomial mutation: $p_m = 1/n, \eta_m = 20$.
- *TMOEA/D Parameters:*
 - neighborhood size: $NS = 30$;
 - first searching phase: $T1 = T/10$;
 - second searching phase: $T2 = \alpha T, \alpha = \{0.01, 0.02, \dots, 0.1, 0.1, 0.1, 0.15\}$;
 - differential evolution: $F = 0.9, CR = 1$;
- *RM-MEDA Parameters:*
 - number of clusters in local PCA: 7;
 - maximal iteration number for local PCA: 50;
 - extension rate of sampling: 0.25.
- *NSGA-II and SMS-EMOA Parameters:*

Two algorithms have the same parameter settings.

 - differential evolution: $F = 0.3, CR = 0.8$;
 - polynomial mutation: $p_m = 1/n, \eta_m = 20$;

To get statistically sound experimental conclusions, each algorithm runs 30 times independently for the path planning model, and the comparisons are performed based on the statistical metric values, i.e., mean values and standard deviations.

C. Model Parameters

In this paper, suppose that an aircraft flies across a combat area with size $200 \times 200 \text{ km}^2$, and there are six threats in the area. According to Eq.(1), Eq.(13) is used to generate the original digital terrain shown in Fig.1(a).

$$z_1(x, y) = \sin(y/90 + 3\pi/2) + \sin(x/15)/10 + 9 \cos\left(\sqrt{(y/18)^2 + (x/15)^2}/2\right)/10 + \sin\left(\sqrt{(y/18)^2 + (x/15)^2}/2\right)/2 + 3 \cos(y/18)/10. \quad (13)$$

Parameters of the five peak terrains of threats are set as: heights $\mathcal{H} = \{0.7, 2.5, 3.2, 2.34, 1.77\}$; horizontal coordinates of the centers $\mathcal{X}_c = \{50, 100, 100, 130, 160\}$, $\mathcal{Y}_c = \{60, 160, 100, 20, 100\}$; parameters of the terrain profiles $\mathcal{X}_t = \{140, 280, 150, 160, 170\}$, $\mathcal{Y}_t = \{20, 220, 280, 190, 230\}$. Based on the parameters, the yielded equivalent terrains of threats are presented in Fig.1(b).

Through fusing the original digital terrain and the equivalent terrains of threats, the generated equivalent digital terrain is shown in Fig.1(c). The path planning in this paper is conducted based on the digital terrain in Fig.1(c). The start and target points of flight are $[0, 0, 0.25]$ and $[200, 200, 0]$, respectively.

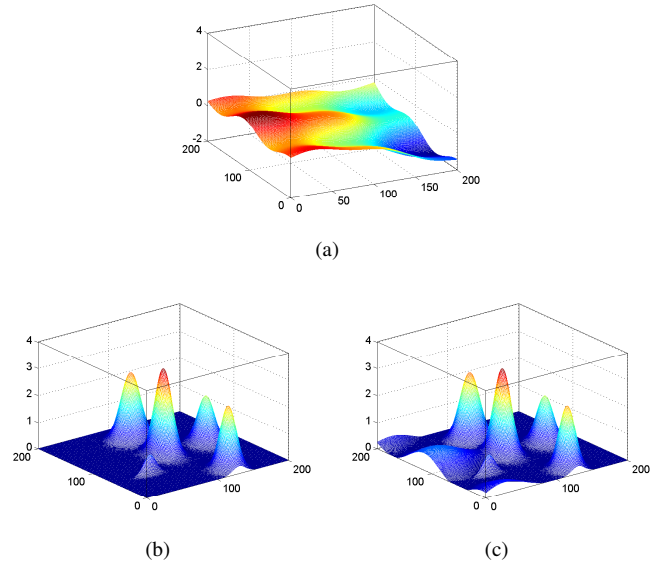


Fig. 1. (a) Original digital terrain; (b) Equivalent terrain of threats; (c) Equivalent digital terrain.

In addition, in the path planning model, the allowed highest altitude, lowest flight altitudes, maximum turning angle, maximum total length of flight path and maximum climb/dive angle are set as $H_{\max} = 1.5\text{km}$, $H_{\min} = 0.05\text{km}$, $\phi = 60^\circ$, $L_{\max} = 500\text{km}$ and $\varphi = 45^\circ$, respectively; the required

shortest length of path segment is set as $L_{\min} = 2\text{km}$. The penalty method is used to handle the constraints when the path planning model is optimized.

D. Comparison Experiments

The error bars of HV metric values obtained by MOEA/D-DE, TMOEA/D, RM-MEDA, NSGA-II, SMS-EMOA and AMEA, respectively, over 30 independent runs on the path planning model are given in Fig.2. It can be seen from the figure that AMEDA obtains the maximum mean HV metric value and the minimum standard deviation value, which denotes that AMEDA is able to stably and effectively find the approximated front with good diversity and excellent convergence for the path planning model.

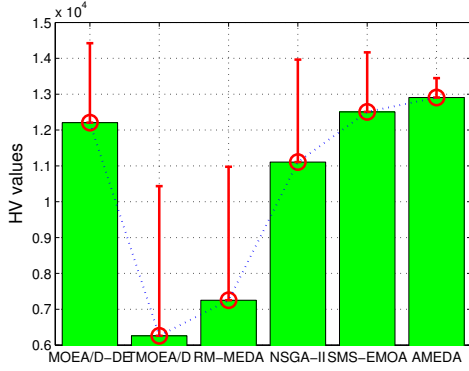


Fig. 2. The error bars of HV metric values obtained by MOEA/D-DE, TMOEA/D, RM-MEDA, NSGA-II, SMS-EMOA and AMEA, respectively, over 30 independent runs on the path planning model.

The evolution of mean values and standard deviations of HV metric values versus generations is plotted in Fig.3. Fig.3 shows that during evolution for 300 generations, AMEDA reaches the maximum mean HV metric values using the least evolutionary generations, and AMEDA always has small standard deviations of HV metric values when evolving. The figure indicates that during evolutions, AMEDA always maintains a population with better convergence, diversity and uniformity.

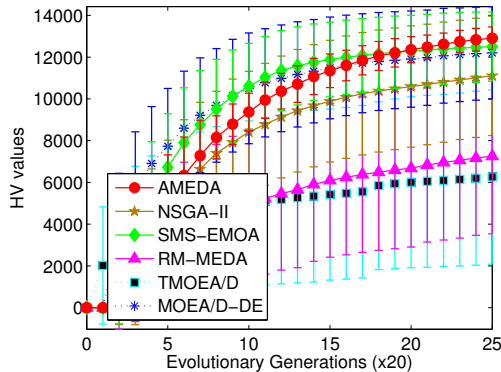


Fig. 3. Evolution of mean values and standard deviations of HV metric values versus generations.

Fig.4 plots the approximated fronts obtained by SMS-EMOA and AMEDA. All the 30 approximated fronts yielded by SMS-EMOA and AMEDA, respectively, are plotted in Fig.4(a). The approximated front with median HV metric value (called representative front) obtained by SMS-EMOA and AMEDA, respectively, over 30 independent runs are plotted in Fig. 4(b). Fig.4(a) shows that compared with SMS-EMOA, all the approximated fronts obtained by AMEDA all stably converge to the region of objective space with smaller objective values, and the distribution of the approximated fronts is much more widely. Part of the approximated fronts achieved by SMS-EMOA have not reached the region with smaller objective values, and the objective points in these fronts are much more concentrated. According to Fig.4(b), it is easy to observe that the representative approximated front obtained by AMEDA spreads much more widely than those yielded by SMS-EMOA.

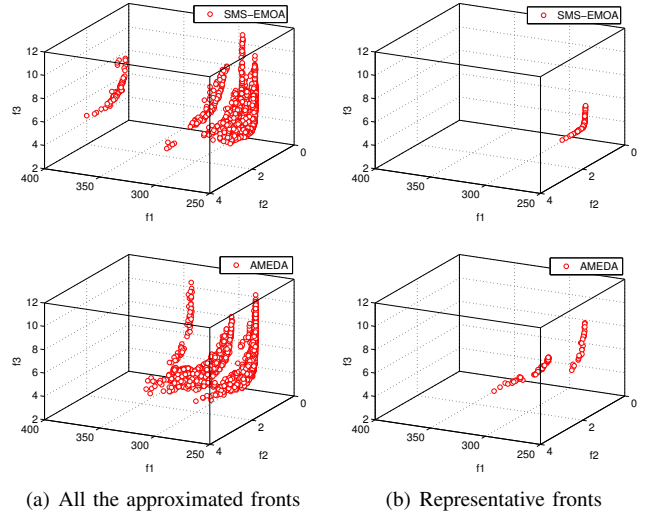


Fig. 4. Approximated fronts obtained by SMS-EMOA and AMEDA.

In the final population with median HV metric value, the corresponding flight paths with the shortest total length, the lowest flight altitude and the smallest turning angle are presented in Fig.5. Fig.5(a) shows that the flight path with the shortest total length almost straightly traverses the combat area. However, the flight altitude is relatively high, which implies a higher risk of being discovered and attacked. Fig.5(b) presents that the flight path with the lowest altitude is quite close to the ground, which is helpful for concealment and penetration. However, the path length is much longer, which will increase fight time. Fig.5(c) shows that the path with the smallest turning angle is very smooth, the turning angles during flight are all very small. Fig.5 illustrates that AMEDA is able to yield multiple flight paths with different characteristics for the decision makers to help their decision making through a single run. This type of path planning approach is quite meaningful and efficient.

The comparisons in Fig.2, Fig.3 and Fig.4 indicate that AMEDA has excellent performance to deal with the path

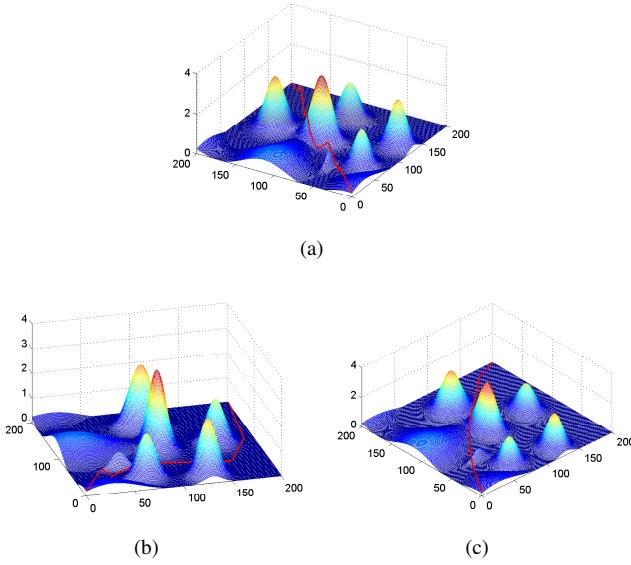


Fig. 5. (a) Path with the shortest total length; (b) Path with the lowest flight altitude; (c) Path with the smallest turning angle.

planning model of this paper. The visualization in Fig.5 illustrates AMEDA is very helpful for the decision makers on planning the paths of the aircraft.

V. CONCLUSION

This paper has proposed a novel adaptive multiobjective estimation of distribution algorithm named as AMEDA to deal with the path planning of aircraft. The characteristic of AMEDA is that a clustering-based multivariate Gaussian sampling strategy is designed. At each generation, AMEDA firstly employs an agglomerative hierarchical clustering approach to partition the population into a number of local clusters, and constructs a global cluster by randomly selecting a solution from each local cluster. Afterwards, for each solution, with a probability a local multivariate Gaussian model (MGM) is built based on the local cluster that the solution locates in to sample an offspring, which is beneficial to exploitation; otherwise, a global MGM is built for the solution based on the global cluster to generate an offspring, which is helpful for exploration. To reduce the modeling complexity, a covariance sharing strategy is also designed for the solutions within the same cluster to build MGMs. To balance the exploration and exploitation better, the probability that controls the contributions of the local and global MGMs is updated at each generation, according to the reproduction utility of the two types of MGMs over the last certain generations.

AMEDA has been compared with five representative MOEAs (i.e., MOEA/D-DE, TMOEA/D, RM-MEDA, NSGA-II, and SMS-EMOA) on the path planning model. The experimental results have shown that AMEDA dramatically outperforms the comparison algorithms on planning the flight path of the aircraft. AMEDA is very meaningful and efficient for the decision makers on planning the paths of the aircraft.

REFERENCES

- [1] P. Yang, K. Tang, J. A. Lozano, and X. Cao, "Path planning for single unmanned aerial vehicle by separately evolving waypoints," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1130–1146, 2015.
- [2] S. Mittal and K. Deb, "Three-dimensional offline path planning for uavs using multiobjective evolutionary algorithms," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*. New York, USA: IEEE, 2007, pp. 3195–3202.
- [3] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. London, UK: Springer-Verlag, 2010.
- [4] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, USA: Kluwer Academic Publishers, 1999.
- [5] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2076–2088, 2015.
- [6] Z. Li and Q. Wang, "An improved multi-objective evolutionary algorithm for multi-objective 0/1 knapsack problem," *International Journal of Multimedia & Ubiquitous Engineering*, vol. 10, no. 5, pp. 383–394, 2015.
- [7] H. Tao, Z. Wang, and J. Li, "Three-dimensional path planning for unmanned aerial vehicles based on multi-objective genetic algorithm," in *Proceedings of the 33rd Chinese Control Conference (CCC 2014)*, July 2014, pp. 8617–8621.
- [8] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective estimation of distribution algorithms," in *Scalable Optimization via Probabilistic Modeling*. Heidelberg, Berlin, Germany: Springer-Verlag, 2006, pp. 223–248.
- [9] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. New York: Springer Science & Business Media, 2002.
- [10] V. A. Shim and K. C. Tan, "Probabilistic graphical approaches for learning, modeling, and sampling in evolutionary multi-objective optimization," in *Advances in Computational Intelligence*. Heidelberg, Berlin, Germany: Springer-Verlag, 2012, pp. 122–144.
- [11] L. Marti, C. Grimme, P. Kerschke, H. Trautmann, and G. Rudolph, "Averaged hausdorff approximations of pareto fronts based on multi-objective estimation of distribution algorithms," in *Proceedings of the 17th Annual Genetic and Evolutionary Computation Conference*. New York, USA: ACM, 2015, pp. 1427–1428.
- [12] R. Xu and D. Wunsch, *Clustering*. Hoboken, New Jersey, USA: John Wiley & Sons, 2008.
- [13] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, New Jersey, USA: L. Erlbaum Associates Inc., 1985, pp. 93–100.
- [14] P. A. Bosman and D. Thierens, "Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 259–289, 2002.
- [15] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [17] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [18] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [19] H.-L. Liu, F. Gu, and Y. Cheung, "T-MOEA/D: MOEA/D with objective transform in multi-objective problems," in *Proceedings of the 2010 International Conference of Information Science and Management Engineering*, vol. 2. New York, USA: IEEE, 2010, pp. 282–285.
- [20] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.