International Journal of Innovative
Computing, Information and Control
Volume **16**, Number **1**, February **2020**

 ISSN 1349-4198
pp. 315–329

3D ONLINE PATH PLANNING OF UAV BASED ON IMPROVED DIFFERENTIAL EVOLUTION AND MODEL PREDICTIVE CONTROL

Jia Liu[1,2], Xiaolin Qin[1,2,*], Baolian Qi[1,2] and Xiaoli Cui[3]

[1]Chengdu Institute of Computer Applications
Chinese Academy of Sciences
No. 9, Section 4, Renmin South Road, Chengdu 610041, P. R. China
*Corresponding author: qinxl2001@126.com

[2]School of Computer and Control Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

[3]Sichuan Rainbow Consulting & Software Co., Ltd.
No. 199, Tianfu 4th Street, Gaoxin District, Chengdu 610041, P. R. China

Received May 2019; revised September 2019

Abstract. *This paper presents an efficient 3D online path planning algorithm for UAV flying in partially known environment. The algorithm integrates model predictive control (MPC) and differential evolution (DE) as the planning strategy. In the initial stage, the artificial potential field (APF) model is developed to describe the mutual effect between the UAV and the surrounding environments. Afterwards, a novel objective function is proposed to address the optimization problem of multi-objective and multi-constraints, which take into account the path length, the smoothness degree of a path and the safety of a path. In addition, the multiple constraints based on the realistic scenarios are taken into account, including maximum acceleration, maximum velocity, map and threat constraints. Then, the improved differential evolution algorithm based on the theory of MPC, is developed to optimize the objective function to find the optimal path. Finally, to show the high performance of the proposed method, we compare the proposed algorithm with the existing optimization algorithms and several extended algorithms. The results reveal that the proposed algorithm not only produces an optimal plan for UAV in a local known 3D environment, but also has better performances in terms of running time and stability.*

Keywords: Path planning, Model predictive control, Artificial potential field, Differential evolution, Unmanned aerial vehicle

1. **Introduction.** Unmanned aerial vehicle (UAV) has been widely used in both military and civil applications because of its low cost and autonomous flight characteristics. Path planning defined as a multi-objective optimization problem plays an important role in the autonomous navigation process of UAV. Its goal is to find the optimal path from the starting point to the target point while avoiding threat sources [1]. At present, there are quite a few algorithms designed and developed for path planning, such as Voronoi diagram [2], visibility graph [3], A* algorithm [4], particle swarm optimization [5], genetic algorithm [6], and differential evolution (DE) [7,8]. Among them, the DE algorithm is recognized as an efficient search engine due to its simplicity and efficiency in different optimization domains, such as constrained optimization [9], multi-objective optimization [10] and multimodal optimization [11,12]. Different DE variants exhibit different capabilities in solving different optimization problems. To sustain the balance between exploration

DOI: 10.24507/ijicic.16.01.315

and exploitation, a novel DE variant was proposed in [13], which introduced the concept of parallel computing into DE and designed a multiple-deme based mutation operator to enable information exchange among processing units. Adhikari et al. [14] proposed a fuzzy adaptive DE algorithm for 3D UAV path planning. In the optimization process, the fuzzy logic controller was used to find the parameter values of DE, and the mutation operation of DE was modified to balance the DE/rand/1 and DE/best/1 strategies. In [15], the authors used the differential evolution and genetic algorithm methods to obtain the maximum likelihood estimates of generalized half normal distribution parameters in hybrid censoring samples. In this way, smaller bias and mean squared error could be obtained. Zhou et al. [16] presented an improved DE algorithm which integrates DE with chaotic variable to break away from the local optimum and to find the global optimum in a three-dimensional (3D) environment. The importance of self-learning ability of the parameters in the path planning algorithm of underwater mobile robot was mentioned in [17] and a dynamic DE approach was proposed which provided ideal convergence, diversity and robustness based on proper selection and adaptive adjustment of parameters. And in [18], a self-adaptive DE algorithm (SaDE) was proposed. In SaDE, the mutation strategies and the respective control parameter were self-adapted according to the probabilities gradually learned from the experience to generate improved solutions.

This paper mainly presents an improved differential evolution optimization algorithm, named overlap DE, for the problem of UAV path planning. Firstly, since the environment is not known globally, the artificial potential field is used to calculate the threat cost in the detection scope of the UAV because of the succinct representation and small computation capacity. Then, the objective function is established to mathematically formulate the constrained multi-objective optimization problem of the path planning, and an improved DE optimization algorithm of solving the above model is developed. In this algorithm, based on the MPC idea, the overlapping part of the last moment and the next time moment is taken as part of the solution of the next moment. The non-overlapping part is optimized by performing the mutation, recombination operations, then the combined solution is optimized by performing the mutation, recombination, and selecting operations. In this way, the optimal path can be found. Finally, simulation results confirm the effectiveness of our algorithm.

The advantage of the proposed approach is as follows: 1) the simple expression of APF can quickly describe the interaction between UAV and the environment, and further effectively cope with sudden changes in partially known environment; 2) the problem of UAV path planning is formalized as a multi-objective optimization problem with constraints; 3) the overlap DE combined with MPC optimization algorithm can quickly plan the optimal path for UAV and we verify the effectiveness of the algorithm by a series of simulations; 4) the proposed approach also supports an efficient frame for the cooperative task of multiple-UAVs mission in the future.

The paper is structured as follows. In Section 2, the description of UAV path planning problem is presented, including the basic idea of MPC and the UAV dynamic model based on MPC. Subsequently, the principles of the improved artificial potential field, and overlap DE algorithm are described in Section 3. Section 4 defines the objective function and the process of solving the objective function using overlap DE algorithm. Then, in Section 5, the simulations are given to show the feasibility and good performance of our proposed method. Finally, the concluding remarks and future work are contained in Section 6.

2. **Problem Description.**

In the problem of UAV path planning, it is assumed that the local information of the environment is known, and the mission of the UAV is to ensure the safe flight and reach the target at the lowest cost. A three-dimensional coordinate

system is created to simulate the environment. And the starting point and the ending point are S and T respectively.

2.1. **Model predictive control.** Model predictive control (MPC) is a model-based feedback control strategy. Subsequently, the control process can be divided into a series of look-ahead time intervals named rolling window, and then, at each time interval, MPC optimizes the specific performance parameter for the following intervals based on current available information [19]. The basic idea of MPC is assuming that the current moment is k, determining the control inputs of the next time domain by optimizing specific performance index in the finite time domain Hp. Then the first optimal control input is selected as the current input. At the next moment $k+1$, calculate the control input $[k+1,\ldots,k+Hp+1]$. Repeat the above process until the task is completed.

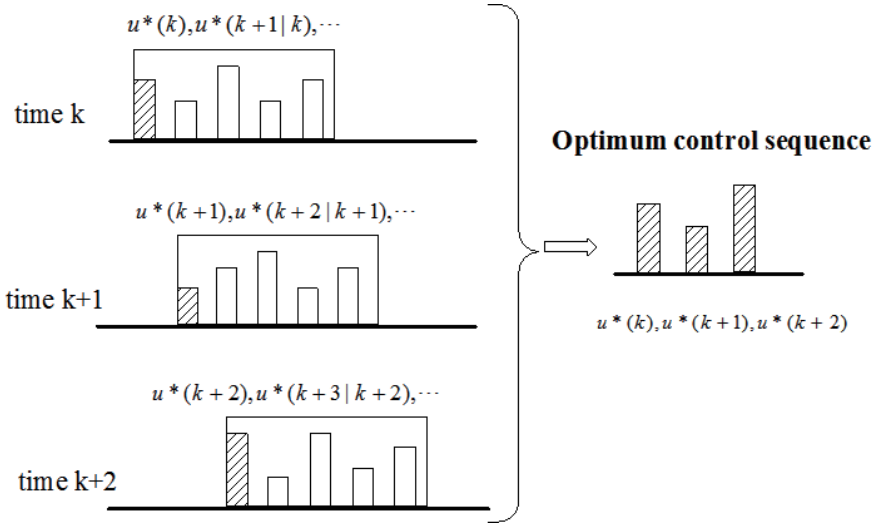


FIGURE 1. Basic ideas of MPC optimization

2.2. **UAV dynamic model based MPC.** General UAV point-mass dynamics model [20] is used in this paper. The UAV dynamic model is

$$\boldsymbol{X}(k+1) = A\boldsymbol{X}(k) + B\boldsymbol{u}(k) \tag{1}$$

where $\boldsymbol{X}(k)$ is the motion state vector of UAV at step k which includes the velocity $\boldsymbol{v}(k)$ and position $\boldsymbol{p}(k)$ of UAV, $\boldsymbol{u}(k)$ is acceleration input.

Then the UAV dynamic model written in MPC formula in this work is formed as

$$\begin{bmatrix} \boldsymbol{p}(k+i+1|k) \\ \boldsymbol{v}(k+i+1|k) \end{bmatrix} = A \begin{bmatrix} \boldsymbol{p}(k+i|k) \\ \boldsymbol{v}(k+i|k) \end{bmatrix} + B\boldsymbol{u}(k+i|k) \tag{2}$$

$$\boldsymbol{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

$$A = \left[\begin{array}{c|c} I_3 & \Delta t \bullet I_3 \\ \hline O_3 & I_3 \end{array}\right], \quad B = \begin{bmatrix} \dfrac{(\Delta t)^2 I_3}{2} & \Delta t I_2 \end{bmatrix}^T$$

where I_3 represents an identity matrix of size 3×3, and O_3 is a zero matrix of size 3×3. Besides, the following constraints limit the magnitude of the acceleration and velocity vectors, provided that the optimization favors minimum time solutions [21].

$$v_{\min} \le \|\boldsymbol{v}\| \le v_{\max} \tag{3}$$

$$\|\boldsymbol{u}\| \leq u_{\max} \tag{4}$$

The initial values $\boldsymbol{p}_o$, $\boldsymbol{v}_o$, $\boldsymbol{u}_o$ are the respective values at the starting point. Once the UAV successfully reaches the target point, the task is completed.

3. Overlap DE Based APF-MPC.

3.1. Artificial potential field.

The APF method is widely used because of its small amount of calculation and fast calculation speed. In this method, the movement of the UAV in the environment is viewed as in an abstract force field. That is to establish the artificial potential field constituted by attractive field of the target and repulsive field of the obstacles. At present, the APF method is mostly used for two-dimensional path planning [22]. This paper extends it to the three-dimensional motion space to describe the interaction between the UAV and surrounding environment.

The attractive potential field function in the three-dimensional environment is

$$U_{att}(\boldsymbol{p}) = \begin{cases} \frac{1}{2}\varepsilon\rho_{goal}^2(\boldsymbol{p}) & \rho_{goal}(\boldsymbol{p}) > r \\ 0 & \rho_{goal}(\boldsymbol{p}) \leq r \end{cases} \tag{5}$$

where ε represent the gain factors. $\rho_{goal}(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{p}_{goal}\|$ is the Euclidean distance between the UAV and target point. When the distance of the UAV from the target is less than r, we think that the UAV reaches the target point.

The repulsion potential field function in the three-dimensional environment is

$$U_{rep}(\boldsymbol{p}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\boldsymbol{p})} - \frac{1}{\rho_o}\right) & \rho(\boldsymbol{p}) \leq \rho_o \\ 0 & \rho(\boldsymbol{p}) > \rho_o \end{cases} \tag{6}$$

where η is positive constant factor, ρ_o is the maximum impact distance of the threat.

In this paper, the environment is three-dimensional and the obstacle cannot be assumed to be regular objects. In order to facilitate our analysis, some simplifications have been made. We assume that the UAV detection area is a sphere with radius r, and the position of the UAV is the center of the sphere $\boldsymbol{p}$. There are six measurement directions on the sphere, as shown in Figure 2, which are $\boldsymbol{pp}_1 = (-r, 0, 0)$, $\boldsymbol{pp}_2 = (r, 0, 0)$, $\boldsymbol{pp}_3 = (0, -r, 0)$, $\boldsymbol{pp}_4 = (0, r, 0)$, $\boldsymbol{pp}_5 = (0, 0, -r)$, $\boldsymbol{pp}_6 = (0, 0, r)$. When the measurement direction is unobstructed, the direction does not generate a repulsive potential field. On the opposite, the repulsive potential field $U_{rep}(\boldsymbol{p})(j)$ generated by the obstacle in the direction is calculated according to Equation (6), where j is the serial number of the detection direction. Then the total potential field acting on the UAV is described as follows.

$$U_{APF} = U_{att}(\boldsymbol{p}) + \sum_{j=1}^{6} U_{rep}(\boldsymbol{p})(j) \tag{7}$$

The advantage of this approximation is that it can quickly perceive the surrounding obstacles and calculate the repulsive potential field. Therefore, it is possible to respond to the changes of the environment promptly.

3.2. Standard DE.

The principle and flow of the DE algorithm are very similar to the genetic algorithm [23], which uses real-number coding, and has three evolutionary operations namely mutation, recombination and selection (Figure 3). Consider an optimization problem formulated as

$$\min_x = f(x) \tag{8}$$

Suppose there are Np candidate solutions in each iteration g, and the total number of iterations is G. The search space is D-dimensional. So, the solution vector in the n-th

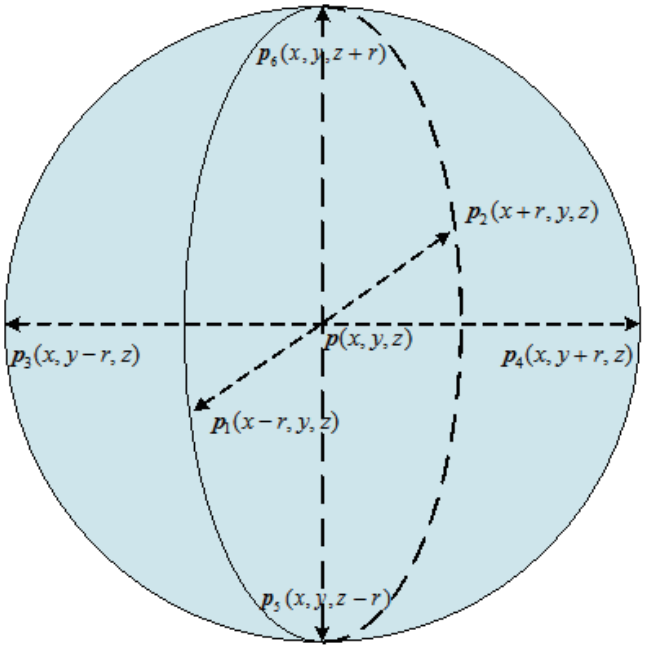


FIGURE 2. Detection area of UAV

iteration can be represented by $\boldsymbol{x}(g) = (\boldsymbol{x}_1(g), \boldsymbol{x}_2(g), \ldots, \boldsymbol{x}_{Np}(g))$, where, $i = 1, \ldots, Np$, $g = 1, \ldots, G$. The basic operators of DE are described as follows.

Definition 3.1. (Coding and initialization): *Initial population is randomly generated:*

$$\boldsymbol{x}_i = \boldsymbol{x}_i^L + rand * \left(\boldsymbol{x}_i^U - \boldsymbol{x}_i^L\right) \tag{9}$$

where $\boldsymbol{x}_i^U$, $\boldsymbol{x}_i^L$ represent the range of values of the i-th component $\boldsymbol{x}_i$ respectively.

Definition 3.2. (Mutation): *The mutation process is to randomly select three different individuals in the population and to generate new individual named mutate vector $\boldsymbol{v}_i$.*

$$\boldsymbol{v}_i(g+1) = \boldsymbol{x}_{r_1} + F * (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) \tag{10}$$

where r_1, r_2, r_3 are randomly chosen indices from the population while $r_1 \neq r_2 \neq r_3 \neq i$, and F, introduced in [10], is a control parameter that is usually chosen from the interval $[0, 2]$. The larger values of F, the higher exploration capability is obtained.

Definition 3.3. (Recombination): *The mutational individual $\boldsymbol{v}_i(g+1)$ and the initial population member $\boldsymbol{x}_i(g)$ are then subjected to the recombination operation.*

$$u_{i,j}(g+1) = \begin{cases} v_{i,j}(g+1) & rand_j \leq CR \text{ or } j = num_i \\ x_{i,j}(g) & rand_j > CR \text{ and } j \neq num_i \end{cases} \quad j = 1, 2, \ldots, D \tag{11}$$

where the random number $rand_j \in [0, 1]$. CR, the probability of the recombination, is a constant in the interval $[0, 1]$ and an integer $num_i \in \{1, 2, \ldots, D\}$.

Definition 3.4. (Selection): *The goal of selection is to choose the better solution for the next generation population with lower fitness value among the reorganized individuals $\boldsymbol{u}_i(g+1)$ and target individuals $\boldsymbol{x}_i(g)$.*

$$\boldsymbol{x}_i(g+1) = \begin{cases} \boldsymbol{u}_i(g+1) & f(\boldsymbol{u}_i(g+1)) \leq f(\boldsymbol{x}_i(g)) \\ \boldsymbol{x}_i(g) & otherwise \end{cases} \quad i = 1, 2, \ldots, Np \tag{12}$$

3.3. **Overlap DE algorithm.** Combined the idea of MPC with the optimal solution at the current moment, the optimal solution at the next moment is similar to the optimal solution at the last moment. As can be seen from Figure 3, the result of the current moment is $\{x(k), x(k+1), \ldots, x(k+Hp-1), x(k+Hp)\}$, the result of the optimization at the next moment is $\{x(k+1), x(k+2), \ldots, x(k+Hp), x(k+Hp+1)\}$, the overlap part is $\{x(k+1), x(k+2), \ldots, x(k+Hp-1), x(k+Hp)\}$ and the different part is $\{x(k+Hp+1)\}$. Therefore, in the optimization process of the next moment, the solution of the individual at the last moment can be used, and then $\{x(k+Hp+1)\}$ is randomly initialized. Figure 4 shows the feasibility of the idea, and the optimal solutions of most overlapping parts are similar in a two-dimensional environment.

$x(k)$	$x(k+1)$	$x(k+2)$	$\ldots$	$x(k+Hp)$	
	$x(k+1)$	$x(k+2)$	$x(k+3)$	$\ldots$	$x(k+Hp+1)$

FIGURE 3. An example diagram of overlap

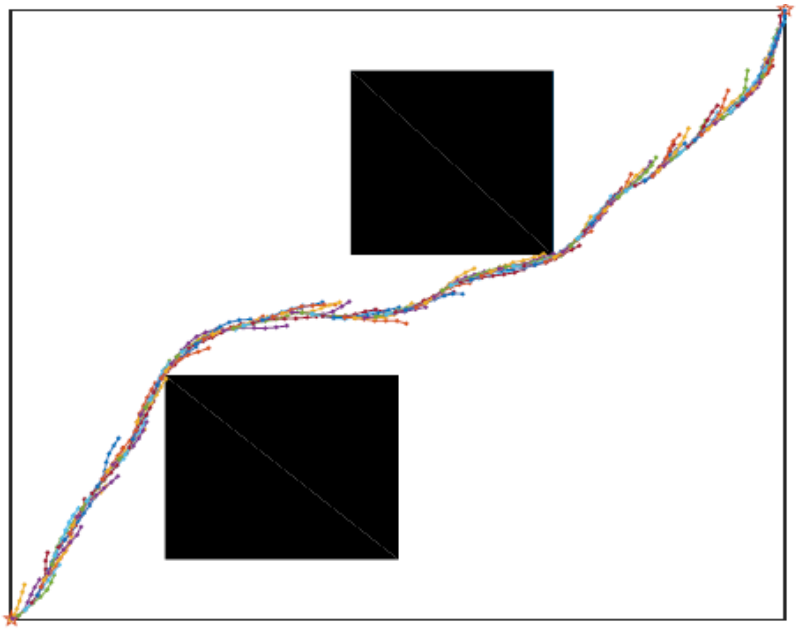


FIGURE 4. Example of path overlap

Taking the result of the solution at the last moment as the initial value of the next moment is equivalent to finding a better value at the next moment. However, the different part is randomly initialized, which may cause the two-part solution imbalance. Therefore, in the optimization process of the next moment, two parts need to be optimized. First, optimize the value of the random initialization, set the optimized length to len, that is to optimize $\{x(k+Hp+2-len), \ldots, x(k+Hp), x(k+1+Hp)\}$. If the result is merged with the overlapping part $\{x_{i,len+1}(k-1), \ldots, x_{i,Hp}(k-1)\}$ of the last moment as a solution of the current moment, it is considered to be a feasible solution that was found. However, this way cannot provide an immediate response to the environmental changes. Because the environment is partially known and the detected environment is different at each moment, the fitness value of each individual at each moment should also be updated in order to respond to the environmental changes. Therefore, the combined

solution continues to be optimized using the DE algorithm to find an optimal solution that can respond quickly to the environment.

Therefore, the optimization process can be simply described as performing the mutation, recombination operations in the non-overlapping part, and then performing the mutation, recombination, and selecting operations on the combined solution. In this way, the solution of the last moment can provide a reference for the next moment.

4. **Objective Function and Optimization Process.**

4.1. **Objective function.** In this paper the objective function is created as following form.

$$\min_{u(k)} J(k) = \sum_{i=1}^{Hp} \begin{pmatrix} \alpha\|U_{APF}(k+i)\|^2 + \beta\|\boldsymbol{p}(k+i) - \boldsymbol{p}(k)\|^2 \\ +\delta\|\boldsymbol{u}(k+i)\|^2 + \gamma\|\boldsymbol{e_v}(k+i)\|^2 \end{pmatrix} \tag{13}$$

where U_{APF} represents the potential field values of the step i within the current control time domain window Hp. The values are added in the objective function to calculate the cost of avoiding obstacles and describe the occurrence of sudden changes. The control trajectory of UAV depends largely on the calculation of the potential function. $\beta\|\boldsymbol{p}(k+i) - \boldsymbol{p}(k)\|^2$ stands for the cost of distance, the shorter path can save more fuel and more time. $\delta\|\boldsymbol{u}(k+i)\|^2$ is the control input which is also important for UAV path planning. $\boldsymbol{e_v}(k+i)$ is the deviation between the predicted velocity and the target speed in the prediction horizon. The purpose of this term is to provide a reference velocity for UAV in barrier-free environment.

Although these items are contradictory, the weighted model can adjust the weights to achieve the goal of optimization. α, β, δ, γ are the adjustment coefficients of the objective function ($\alpha, \beta, \delta, \gamma > 0$). Thus, the goal of path planning is to optimize the multi-objective problem.

4.2. **Optimization process.** The deployment platform is the starting point S for path planning. The end point T is regarded as the end of path planning. The goal of the algorithm is to find a path that minimizes the value of the objective function from the starting point to the end point and satisfy the constraints. The optimization process is as follows.

Step 1. Parameter setting: Initialize the parameters of the system, such as the starting point coordinates and end point coordinates of the UAV. Set the maximum speed $v_{\max}$, maximum acceleration $u_{\max}$, the rolling time domain Hp, the sampling interval Δt. Set the population size ($Np = 50$), the probability of the recombination ($CR = 0.5$), the number of iterations ($G = 150$) and the optimized length len.

Step 2. Initialize the population: Randomly initialize the individual values $x_i(k) = (x_{i,1}(k), x_{i,2}(k), \ldots, x_{i,Hp}(k))$, $k = 0$, $i = 1, 2, \ldots, Np$ according to Equation (9).

Step 3. Optimize the population: First, calculate the fitness function value of each individual according to Equation (13). Then, optimize the population at $k = 0$ according to the DE algorithm. That is, perform mutation, recombination and selection operations in turn and obtain the global optimization $x_{best}(0)$ with the lowest fitness value.

Step 4. Model input: Input the first result $x_{best,1}(0)$ of the optimal result $x_{best}(0) = (x_{best,1}(0), x_{best,2}(0), \ldots, x_{best,Hp}(0))$ into the model (2) and calculate the state of the next moment of the UAV.

Step 5. $k = k + 1$.

Step 6. Each individual i in the population do Steps 7 to 9.

Step 7. Initialize and optimize non-overlapping part: Initialize $\{x_{i,Hp+1-len}(k), \ldots, x_{i,Hp}(k)\}$ according to Equation (9) and then perform mutation, recombination operations to optimize the non-overlapping part.

Step 8. Connect the solutions of the two parts: Copy the overlapping part $\{x_{i,len+1}(k-1), \ldots, x_{i,Hp}(k-1)\}$ of the last moment to $\{x_{i,1}(k), \ldots, x_{i,Hp-len}(k)\}$ of the current moment. The two parts are connected together as the initial solution.

Step 9. Optimize the combination solution: Perform mutation, recombination and selection operations to optimize the entire individual $x_i(k)$ and obtain the global optimization $x_{best}(k)$ with the lowest fitness value.

Step 10. Model input: Input the first result $x_{best,1}(k)$ of the optimal result $x_{best}(k) = (x_{best,1}(k), x_{best,2}(k), \ldots, x_{best,Hp}(k))$ into the model (2). Calculate the state of the next moment of the UAV.

Step 11. Determine whether the UAV reaches the target: If it is reached, the algorithm outputs the optimal solution; otherwise, return to Step 5 to continue the iterative operation.

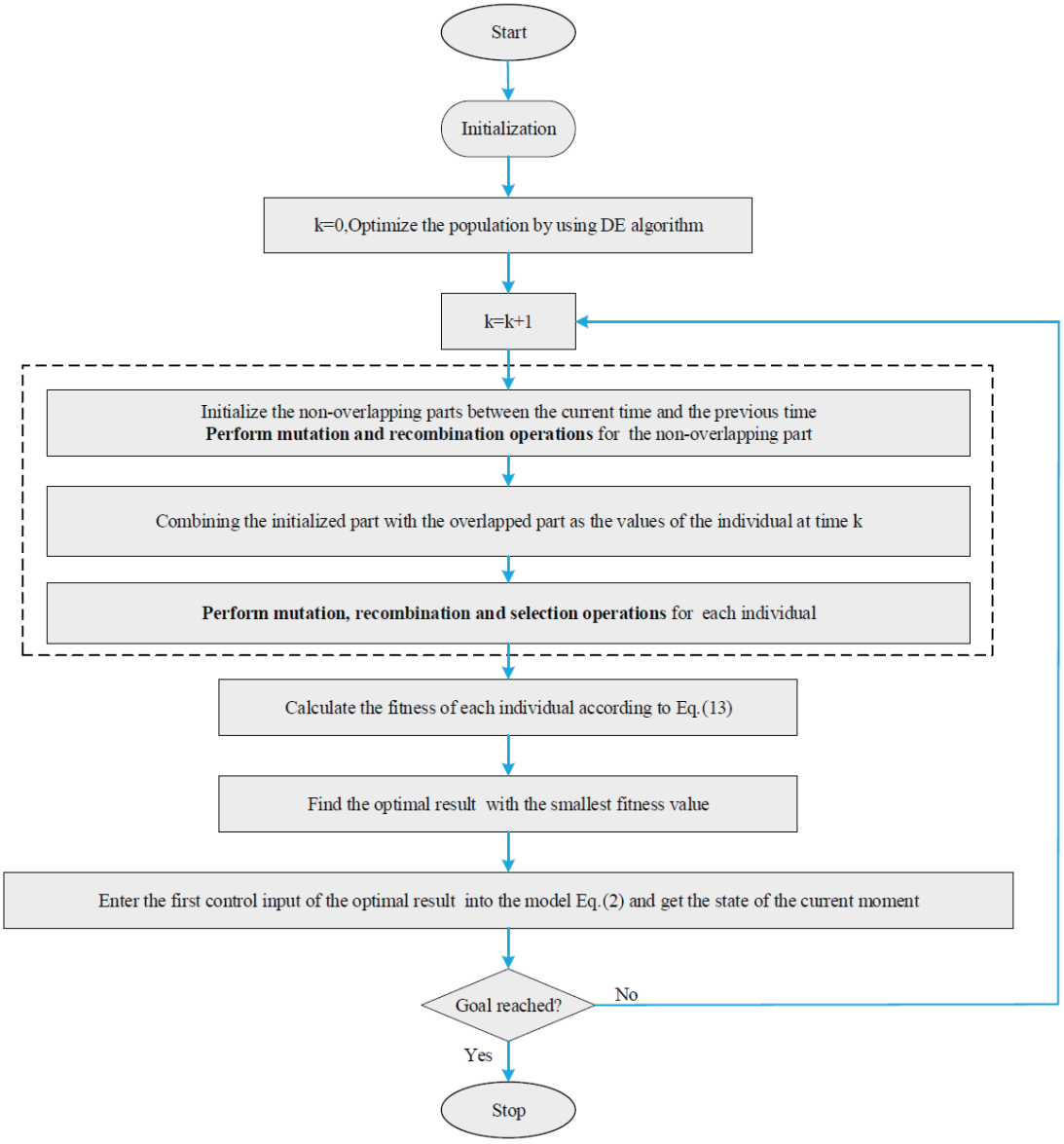


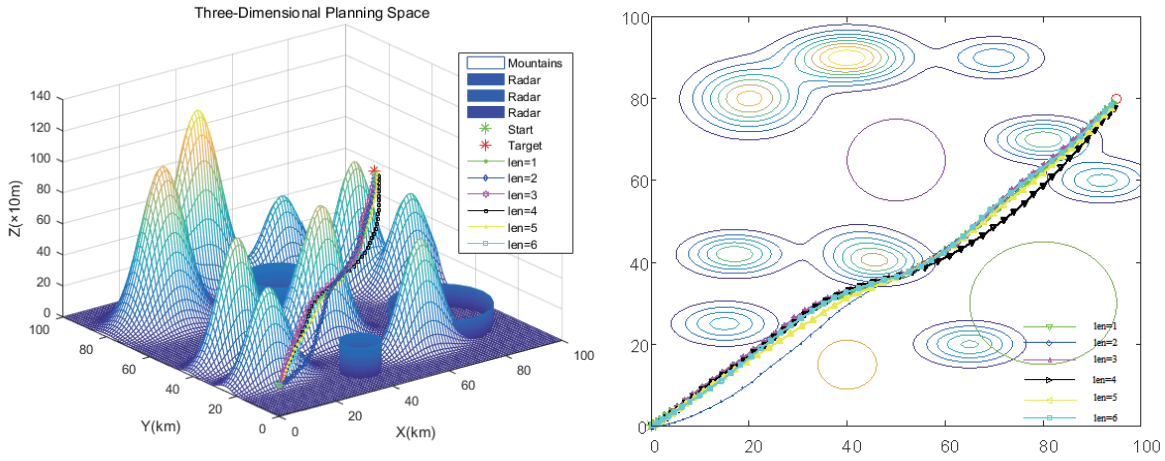
FIGURE 5. Process flow of proposed algorithm

5. **Simulations.** In order to verify the feasibility and effectiveness of the overlap DE-based APF-MPC approach as it is applied to the 3D path planning problems, the simulations are realized by MATLAB 2016a in the windows environment. In the test, a 100km*100km*1.5km region is considered. The specific information of the 3D environment is shown in Table 1.

TABLE 1. The basic information of 3D environment

Information	Value	Radius	Height
Radar threat 1	$(40, 15, 0)$	6	20
Radar threat 2	$(50, 65, 0)$	10	25
Radar threat 3	$(80, 30, 0)$	15	15

5.1. **Simulation 1.** The main idea of the overlap DE optimization algorithm based on the theory of MPC and APF is to take the overlap part between the last moment and the current moment as a part solution of the current moment, and the length of the un-overlapping part is set to *len*. Planning paths with different *len* needs to be discussed to prove the validity of the overlap DE optimization algorithm. In the simulation, the parameters are listed as follows: the probability of the recombination ($CR = 0.5$), the population size ($Np = 50$), the number of iterations ($G = 150$), the sampling interval ($\Delta t = 1$s), the time domain window ($Hp = 6$), the start and target coordinates are (0, 0, 20) and (100, 60, 60). The length *len* is set to 1, 2, 3, 4, 5, 6 respectively. The simulation curves of the paths are shown in Figures 6(a)-6(b). Table 2 presents the average distances and running times obtained in 100 runs for the six cases.



(a) The 3-D view of the UAV paths and the threat zones (b) The 2-D show of (a)

FIGURE 6. Optimal UAV path of the best results obtained in 100 runs with different optimized lengths

From Figures 6(a) and 6(b), it can be seen that when *len* takes different values, the paths planned by the algorithm can all successfully avoid obstacles in the environment and arrive at the target. At the same time, from Table 2, the planned paths length does not change significantly with the change of *len*. Nevertheless, there are two conclusions which can be observed from Table 2.

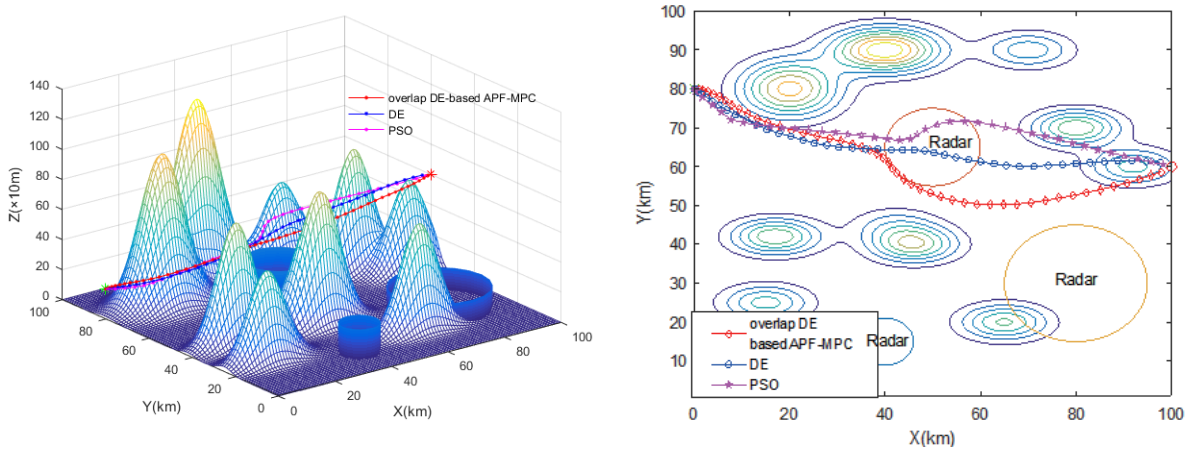
The first is when the optimization length is 6, the average planning time is nearly twice the average planning time of *len*, and is nearly 2.5 times the average running time of

TABLE 2. The best, mean and the worst running times and distances obtained in 100 runs with different optimized length

		$len = 1$	$len = 2$	$len = 3$	$len = 4$	$len = 5$	$len = 6$
Distance	Best	130.3752	130.0128	131.0780	131.1109	130.5242	129.0075
	Worst	131.9254	131.5290	131.5270	131.8510	131.8352	129.9886
	Mean	**131.2452**	**130.8554**	**131.2153**	**131.6486**	**131.2737**	**129.2552**
Running time	Best	33.3320	28.1496	25.0930	26.4990	26.8070	60.9620
	Worst	34.4910	27.5930	26.5660	25.9330	24.1190	59.2700
	Mean	**33.9774**	**27.8535**	**25.8858**	**26.2784**	**25.3920**	**60.2027**

$len = 2$, $len = 3$, $len = 4$, $len = 5$. This is because $len = 6$, which is equivalent to the size of controlling the time domain window Hp, indicates using the standard DE algorithm to optimize the objective function. The optimization is independent at each step. In the previous section we have mentioned that optimal solutions of the overlapping parts are similar, so solving at each moment independently will result in the waste of time. In other cases, the overlap part is utilized, which reduces the planning time to some extent. The second conclusion is that the running time for $len = 1$ is longer than the running time of $len = 2$, $len = 3$, $len = 4$, and $len = 5$. The reason for this result may be that when the optimization length len is 1, most of the solution is from the last moment. It limits the directions of the current time to find a better solution and cannot respond quickly in the face of a sudden threat. Therefore, in order to find the optimal solution, it takes a little longer than the above optimized lengths. From another perspective, the proposed algorithm in this paper can improve the time efficiency in UAV path planning.

5.2. **Simulation 2.** The second simulation is conducted by comparing this algorithm with other traditional algorithms, including particle swarm optimization (PSO) and standard differential evolution algorithm (DE). In the simulation, the parameters are listed as follows: the probability of the recombination ($CR = 0.6$), the population size ($Np = 20$), the number of iterations ($G = 100$), the sampling interval ($\Delta t = 1$s), the time domain ($Hp = 6$). For UAV, the maximum $v_{\max}$ speed is 3m/s and the maximum acceleration $u_{\max}$ is 0.3m/s^2. The start and target coordinates of the UAV are set to $(0, 80, 20)$ and $(100, 60, 60)$. The algorithm runs 100 times and the best results are shown in Figures 7-9. The experimental statistical results are recorded in Table 3.



(a) The 3-D view of the UAV paths and the threat zones (b) The 2-D show of (a)

FIGURE 7. Optimal UAV path in a 3D environment for Simulation 2

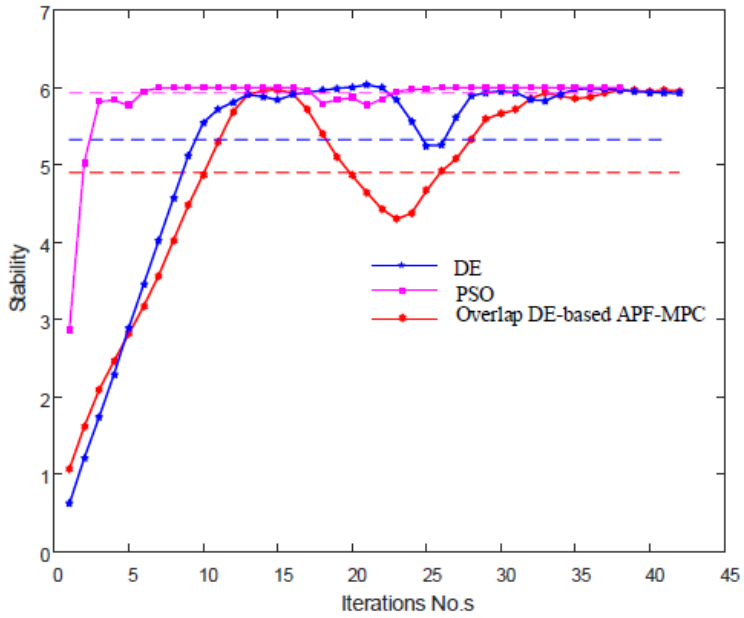


FIGURE 8. Stability comparisons

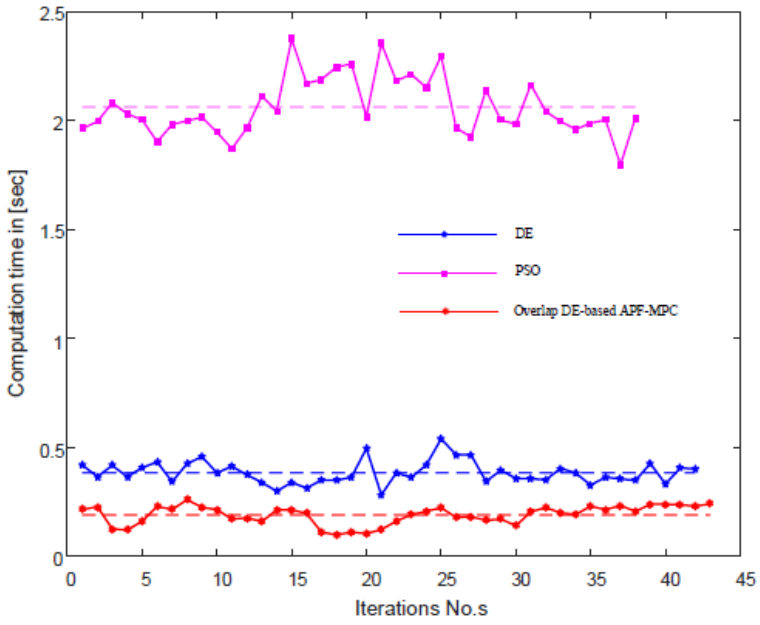


FIGURE 9. Time comparisons

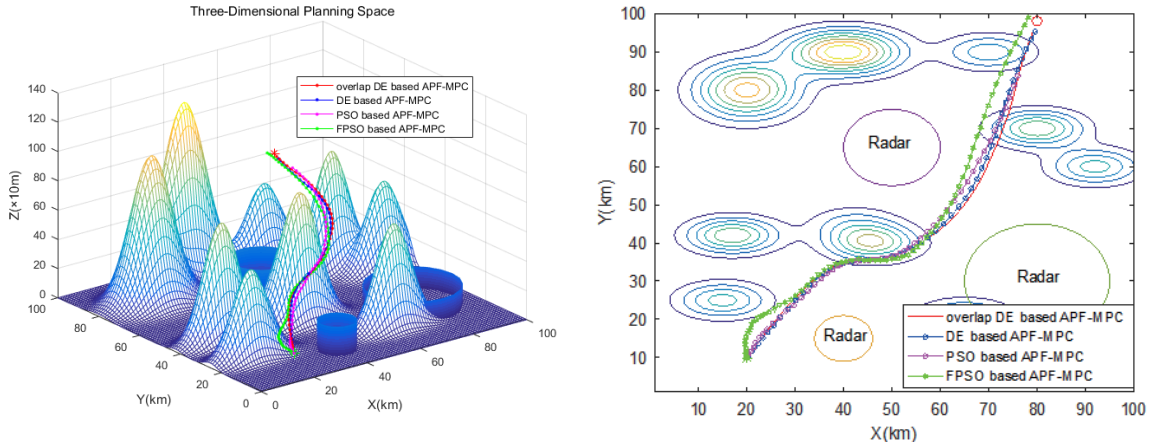
TABLE 3. Performance comparison of the three algorithms

Algorithm	Running time	Distance	Stability	Number of iterations
DE	16.1100	112.0941	5.324	41
PSO	78.3500	116.6758	5.928	38
Overlap DE based APF-MPC	**8.1720**	**119.3494**	**4.901**	**42**

The optimal path comparisons of the three algorithms are shown in Figure 7. It can be seen from the figures that the proposed algorithm can plan a collision-free route under the environment model and reach the target point. Compared with other algorithms, the smoothest route is planned by the algorithm. As can be seen from Figure 8, the algorithm proposed in this paper converges faster and has better stability than the other two algorithms. Also, in Figure 9, the variation of running time at each iteration is displayed to evaluate several algorithms. It is obvious that the running times of overlap DE, which is based on APF and MPC, are smaller than others. Further, it can be seen from Table 3 that the running time is around 50.73% of the DE algorithm and 10.43% of the PSO algorithm. This is because the proposed overlap differential evolution algorithm combined with MPC idea can effectively use the past information, and further predict the future states in the current control time domain as the reference of the next moment in advance. However, the DE and PSO algorithms need to be re-planned at each step in the process of path planning without any reference. This may cause planning time to be long and the fluctuation between the state of the last moment and the state of the current moment to be relatively large. That is, the planning time is long and the stability is relatively poor. Consequently, the stability and runtime route planned by overlap DE based APF-MPC is the lowest, indicating that the algorithm is the best algorithm within these several algorithms for UAV's path planning.

5.3. **Simulation 3.** In this section, we compare the overlap DE based APF-MPC with three improved optimization algorithms in solving the constrained path planning problem, including DE based APF-MPC, PSO based APF-MPC, and fast particle swarm optimization (FPSO) based APF-MPC [24]. These algorithms are based on model control prediction, and their objective functions are the same as the objective function of this paper. The only difference is that the process of optimizing the objective function is different. The population size and maximum iteration are set as $Np = 50$, $G = 100$. For DE based APF-MPC, the objective function is optimized by DE algorithm and its parameter settings are the same as overlap DE; for PSO based APF-MPC, the objective function is optimized by PSO algorithm and the parameter settings are as follows: $w_{\max} = 1$, $w_{\min} = 0$, $c_1 = c_2 = 1.5$; for FPSO based APF-MPC, we use the FPSO algorithm [24] to optimize the objective function, where the parameter is set to $w_{\max}(k,i) = 1 - \frac{F_{k,i}(u)}{\max\limits_{i}(F_{k,i}(u))}$, $F_{k,i}(u)$ is the fitness of the i-th particle of the k-th iteration, $w_{\min} = 0$, $c_1 = c_2 = 1.5$. The start and end points of the UAV are set to $(20, 10, 10)$ and $(80, 98, 60)$. The algorithm runs 100 times and the best results are as follows.

Figure 10(a) and Figure 10(b) are three-dimensional and two-dimensional views of the algorithm's track plan. Obviously, the four algorithms can successfully avoid obstacles in the environment and reach the target. Figures 11 and 12 are time and cost comparisons of the four algorithms. The experimental results are recorded in Table 4. From these simulation results, it can be seen that all of the above algorithms successfully avoid threat sources in the environment and satisfy the constraints of the UAV. However, the PSO based MPC approach takes the most time, followed by FPSO based APF-MPC and DE based APF-MPC, and overlap DE based APF-MPC with the least running time. This further verifies the validity of the concept of overlapping differential evolution, which reduces planning time and enables timely response to environmental changes. At the same time, the algorithm proposed in this paper is lower in cost and has fast convergence rate. Therefore, the algorithm proposed in this study, in general, showed better or relatively similar performance than the above algorithms.



(a) The 3-D view of the UAV paths and the threat zones

(b) The 2-D show of (a)

FIGURE 10. Optimal UAV path in a 3D environment for Simulation 3

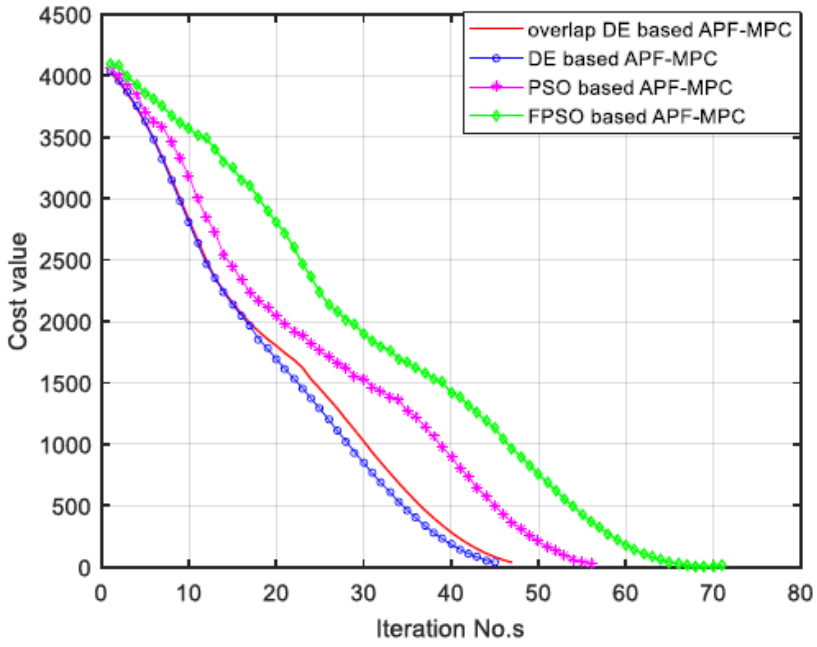


FIGURE 11. The comparison of cost value

TABLE 4. Performance comparison of the four algorithms

Algorithm	Running time	Distance	Number of iterations
DE based APF-MPC	16.7730	128.4707	44
PSO based APF-MPC	34.2260	121.0841	55
FPSO based APF-MPC	23.4540	136.4183	70
Overlap DE based APF-MPC	**8.6600**	**128.5578**	**46**

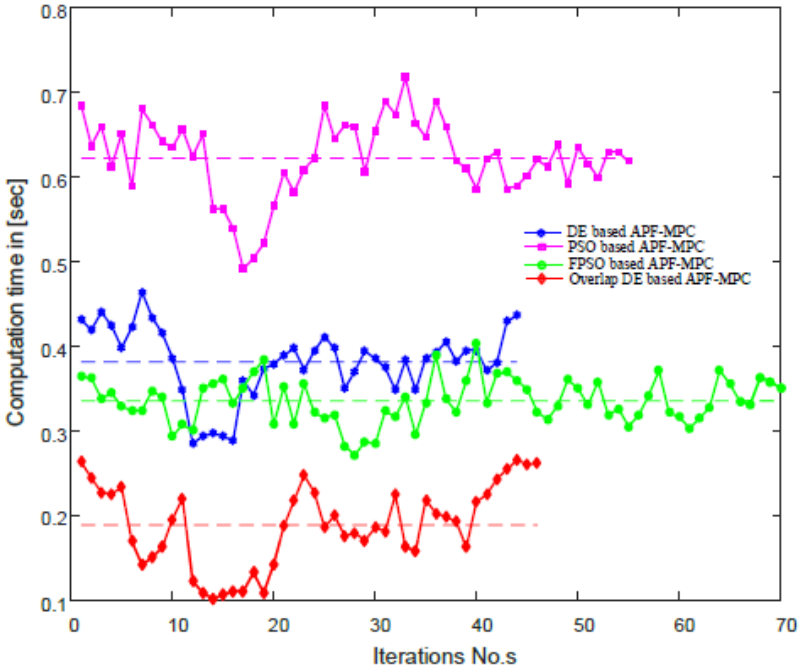


FIGURE 12. The comparison of running time

6. **Conclusion and Future Works.** In this section, a brief summary of the paper is written. To solve the UAV online path planning problem in a 3D local known environment, this paper proposed an overlap DE based on APF-MPC method. With the frame of MPC, the algorithm establishes the objective function to achieve specific performances. At the same time, to describe the relationship between UAV and the environment, the function of APF is added to the objective function. Then, the overlap DE algorithm is adopted to optimize the control actions in MPC to find an optimal path. Simulation results show that the proposed algorithm can solve the optimal control problem of the UAV path planning effectively and has better performance. The shortcoming of the algorithm is that the cooperative task of multiple-UAVs mission is not included. Therefore, on this basis, further improvements should focus on the multiple-UAVs path planning in the future work.

Acknowledgment. This work is partially supported by the Natural Science Foundation of China (No. 61402537), the Sichuan Science and Technology Program (No. 2018GZDZX0041). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] S. A. Bortoff, Path planning for UAVs, *Proc. of the 2000 American Control Conference, ACC (IEEE Cat. No. 00CH36334)*, vol.1, no.6, pp.364-368, 2000.

[2] M. Candeloro, A. M. Lekkas and A. J. Sørensen, A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels, *Control Engineering Practice*, vol.61, pp.41-54, 2017.

[3] N. B. A. Latip, R. Omar and S. K. Debnath, Optimal path planning using equilateral spaces oriented visibility graph method, *International Journal of Electrical and Computer Engineering*, vol.7, no.6, pp.3046-3051, 2017.

[4] C. Wang, L. Wang, J. Qin et al., Path planning of automated guided vehicles based on improved A-Star algorithm, *2015 IEEE International Conference on Information and Automation*, Lijiang, China, pp.2071-2076, 2015.

[5] G. Li and W. Chou, Path planning for mobile robot using self-adaptive learning particle swarm optimization, *Science China Information Sciences*, vol.61, no.5, pp.052204:1-052204:18, 2018.
[6] M. Elhoseny, A. Tharwat and A. E. Hassanien, Bezier curve based path planning in a dynamic field using modified genetic algorithm, *Journal of Computational Science*, vol.25, pp.339-350, 2018.
[7] Y. Fu, M. Ding, C. Zhou et al., Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol.43, no.6, pp.1451-1465, 2013.
[8] A. W. Mohamed, A novel differential evolution algorithm for solving constrained engineering optimization problems, *Journal of Intelligent Manufacturing*, vol.29, no.3, pp.659-692, 2018.
[9] N. M. Hamza, D. L. Essam and R. A. Sarker, Constraint consensus mutation based differential evolution for constrained optimization, *IEEE Transactions on Evolutionary Computation*, vol.20, no.3, pp.447-459, 2015.
[10] X. Qiu, J. X. Xu, K. C. Tan and H. A. Abbass, Adaptive cross-generation differential evolution operators for multiobjective optimization, *IEEE Transactions on Evolutionary Computation*, vol.20, no.2, pp.232-244, 2015.
[11] S. Hui and P. N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, *IEEE Transactions on Cybernetics*, vol.46, no.1, pp.64-74, 2015.
[12] J. Liang, W. Xu, C. Yue et al., Multimodal multiobjective optimization with differential evolution, *Swarm and Evolutionary Computation*, vol.44, pp.1028-1059, 2019.
[13] D. H. Lim, H. N. Luong and C. W. Ahn, A novel differential evolution using multiple-deme based mutation, *International Journal of Innovative Computing, Information and Control*, vol.8, no.5(A), pp.3049-3060, 2012.
[14] D. Adhikari, E. Kim and H. Reza, A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization, *Evolutionary Computation*, 2017.
[15] H. Xin, J. Zhu and T. R. Tsai, Differential evolution and genetic algorithm methods for parameter estimation of the generalized half normal distribution with hybrid censoring, *ICIC Express Letters*, vol.12, no.6, pp.519-527, 2018.
[16] Z. Zhou, H. Duan, P. Li and B. Di, Chaotic differential evolution approach for 3D trajectory planning of unmanned aerial vehicle, *2013 10th IEEE International Conference on Control and Automation (ICCA)*, Hangzhou, China, pp.368-372, 2013.
[17] D. R. Parhi and S. Kundu, Navigational control of underwater mobile robot using dynamic differential evolution approach, *Proc. of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol.231, no.1, pp.284-301, 2017.
[18] A. K. Qin, V. L. Huang and P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Transactions on Evolutionary Computation*, vol.13, no.2, pp.398-417, 2009.
[19] J. Ji, A. Khajepour, W. W. Melek et al., Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints, *IEEE Transactions on Vehicular Technology*, vol.66, no.2, pp.952-964, 2017.
[20] T. Keviczky, P. Falcone, F. Borrelli et al., Predictive control approach to autonomous vehicle steering, *2006 American Control Conference*, p.6, 2006.
[21] Y. Kuwata, A. Richards, T. Schouwenaars et al., Decentralized robust receding horizon control for multi-vehicle guidance, *2006 American Control Conference*, p.6, 2006.
[22] A. Richards and J. P. How, Aircraft trajectory planning with collision avoidance using mixed integer linear programming, *Proc. of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, pp.1936-1941, 2002.
[23] Z. Zhou, J. Wang, Z. Zhu et al., Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field, *Optik*, vol.158, pp.639-651, 2018.
[24] G. C. Onwubolu and B. V. Babu, *New Optimization Techniques in Engineering*, Springer, 2013.
[25] W. B. Wang, X. L. Qin, L. G. Zhang et al., Dynamic UAV trajectory planning based on receding horizon, *CAAI Transactions on Intelligent Systems*, vol.13, no.4, pp.524-533, 2018.