Journal of
Marine Science and Technology

Volume 28 | Issue 5 Article 14

PATH PLANNING AND OBSTACLE AVOIDANCE OF UAV FOR CAGE CULTURE INSPECTION

Yi-En Cai
Department of Communication, Navigation and Control Engineering National Taiwan Ocean University Keelung, Taiwan, jgjuang@mail.ntou.edu.tw

Jih-Gau Juang
Department of Communication, Navigation and Control Engineering National Taiwan Ocean University Keelung, Taiwan

Recommended Citation
Cai, Yi-En and Juang, Jih-Gau (2020) "PATH PLANNING AND OBSTACLE AVOIDANCE OF UAV FOR CAGE CULTURE INSPECTION," *Journal of Marine Science and Technology*: Vol. 28: Iss. 5, Article 14.
DOI: 10.6119/JMST.202010_28(5).0014
Available at: https://jmstt.ntou.edu.tw/journal/vol28/iss5/14

PATH PLANNING AND OBSTACLE AVOIDANCE OF UAV FOR CAGE CULTURE INSPECTION

Acknowledgements

This research was supported by the MOST AI Biomedical Research Center at NCKU, Taiwan.

Journal of Marine Science and Technology, Vol. 28, No. 5, pp. 444-455 (2020)
DOI: 10.6119/JMST.202010_28(5).0014

PATH PLANNING AND OBSTACLE AVOIDANCE OF UAV FOR CAGE CULTURE INSPECTION

Yi-En Cai and Jih-Gau Juang

Key words: UAV, cage inspection, path planning, collision avoidance.

ABSTRACT

This paper presents path planning and obstacle avoidance of an *unmanned aerial vehicle* (UAV) for cage culture inspection. The proposed system can automatically inspect cage farming environment that can cut manpower and reduce cost. The problem is similar to travelling salesman problem (TSP). Genetic algorithms are useful techniques for TSP and can also be used for cage culture inspection. In addition to path planning, an improved particle swarm optimization (IPSO) is applied to avoid obstacle between different cages. The IPSO is applicable to both stationary and dynamic environments with one or more obstacles. Simulation results show the effectiveness of the proposed method, comparison of other random search path planning methods is given.

I. INTRODUCTION

Developing progress of unmanned aerial vehicles (UAV) is sharply in recent years, and the UAV has been widely used in aerial photography, geographic surveillance and so on. Many companies manufacture folding UAV that make people can carry it to everywhere easily. With the UAV's development and its advantages, the cost of many missions can be reduced and the tasks still can be completed by the use of UAV. The UAV has two categories mainly, fixed-wing and copter, each has its advantages and disadvantage. Takeoff with small space and can hover in the air are the advantages of copter compared to the fixed-wing airplane. Although fixed-wing airplane has longer flight time and higher cruise speed, but in order to maintain sufficient lift, it needs to keep flight speed. And because of its mechanism, it also needs enough runway length to take off. Compared with the fixed-wing airplane, the copter can takeoff vertically and hover at a certain height in any place. It has lower space requirement and can fly in complex terrain. According to these advantages and features, using copter for reconnaissance mission and goods transportation in disaster area is more economical and safer. Besides, the copter can hover in the air so that it can be applied to detect environmental information by the sensors installed on it. Through the sensors we can get desired information to ensure safety and develop more automatic missions. But the sensors may miss some information and cause the control system to make wrong judgment. The way to overcome the problem which is from the inaccurate sensors should be solved. The automation technology is mature gradually whether in vehicle systems or aircraft systems. There are many unexpected conditions should be considered so that the control system of UAV can implement task safely. Through the path planning and obstacle avoidance algorithm with proper sensors, we can reduce the stress of operator and easily reach the place which is predefined.

In this study, the techniques of obstacle avoidance and path planning of UAV are applied to cage culture inspection. UAVs have been used in cage farming recently. In Hu et al.(2016), the authors proposed a drone delivery system that can feed the fishes by a quadcopter instead of labors, and can reduce the working cost. In addition to UAV, reference (Reshma and Kumar, 2016) presents an integrated system with a ZigBee wireless communication network, an Ethernet-based server monitoring platform, and a mobile client monitoring platform to obtain information about the cage culture environment. The available information includes water temperature, salinity, dissolved oxygen, and water velocity. In Zhang et al. (2012), the authors proposed a centralized remote monitoring system for offshore cage farm that can monitor the aquatic environment and fish activities remotely and real-timely. The system can provide users with a variety of culture information such as fish amount, fish growth conditions and culture safety in the offshore fish cage. In Xu and Zhang (2007), an acoustic monitoring system was proposed which can provide fish biomass and the status of the net cage information. The drawback of these studies is that all the cages need to be installed with the hardware system, which will increase the cost. In here, we use only one UAV system that can fly around all the cages and capture data of the cage culture environment. The cost can be reduced drastically.

The quadcopter can be traced back to the early 1920s, Bothezat et al. (2002) produced the first flight table four-rotor

Paper submitted 05/25/20; revised 05/03/20; accepted 07/06/20. *Corresponding Author: Jih-Gau Juang (jgjuang@mail.ntou.edu.tw)*
Department of Communication, Navigation and Control Engineering
National Taiwan Ocean University Keelung, Taiwan

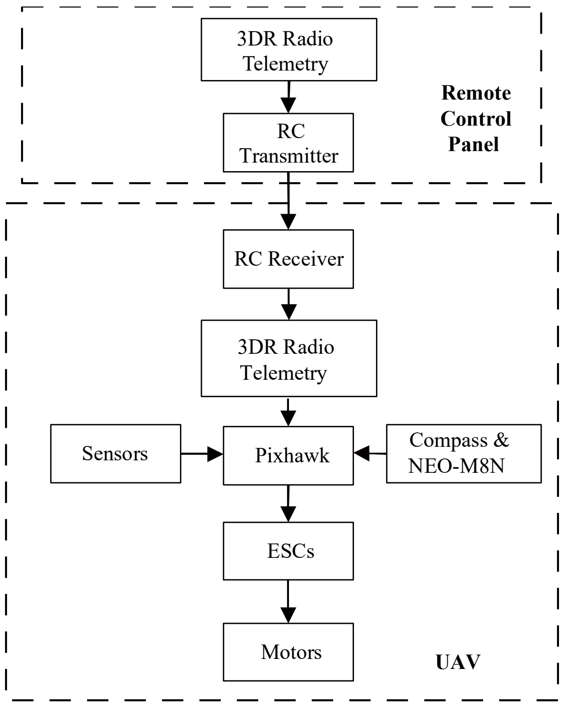

Fig. 1. Control signal of the system.

Fig. 2. Hexacopter main body.

aircraft. The United States takes the lead in development of UAV. The development of UAV mostly focused on theory or the simulation platform, such as Altug et al. used the feedback images for the quadrotor copter research (Altuf et al., 2002; 2003; 2005). The global position system (GPS), the inertia navigation system (INS), and the wireless communication system are mounted on the copter. The copter can know where it is, it also can overcome the terrain factor with the technology well developed. In recent years, many researches of automation on different devices are presented. One important issue about automation of UAV is path planning. There are two conditions of path planning problem: global path planning which allows UAV to move through known obstacles, and local path planning which allows UAV to generate new path when unknown obstacle will cause threats. The purpose of this study is to use UAV to drop a suspensory sensor in the cage nets and get the environment information. Since the cage nets may not be only one, so we consider the traveling salesman problem (TSP) first. In the TSP, the goal is to find the shortest route to visit all cities. By now, TSP has been solved by many methods, such as simulate annealing, ant colony system, neural networks, tabu search, and genetic algorithms (GA) (Wang et al., 2007; Luo et al., 2011; Shi and Jia, 2013; Gupta et al., 2017). GA uses three main operators: selection, crossover and mutation to simulate process in natural for evolution. In this study, we produce better generation using appropriate parameters by the random bias GA (RBGA). And the method in Gupta et al. (2017) is used to plan order of cage nets, then path planning algorithm can generate a path between starting position and ending position.

In obstacle avoidance control, the artificial potential field (APF) is the popular method used (Chen et al., 2013; Chen and Li, 2017). Currently, the traditional random search path planning methods include GA (Shi and Cui, 2010), ant colony optimization (ACO) (Hsiao et al., 2004), rapid exploring random tree (RRT) (Korkmaz and Durdu, 2018; Zhang et al., 2018), particle swarm optimization (PSO) (Yousef et al., 2014) and so on. These algorithms have some advantages in solving the path planning problem. Among these algorithms, PSO is processed through the communication between particles and competition of particles. In PSO, the parameters are few and the convergence speed is fast, and is good for path planning. In (Zhang et al., 2018), the authors proposed an improved particle swarm algorithm (IPSO) whose initialization of particle swarm used beta function. The original PSO is improved from five aspects, initialization of particle swarm, design of the learning factors, design of the inertia weight, updating of particle's velocity, and processing of particle's position trans-boundary. The inertia weight was designed to be exponentially decreasing and to balance global exploitation and local exploration in the optimization. The IPSO was introduced into the path planning for the UAV and simulations proved the effectiveness of the algorithm. In this study, the IPSO is applied, and it is proved that it is better than traditional APF, ACO, and RRT. The improved algorithm is applied to path planning of the UAV. We simulate this algorithm by the environment with obstacles. The simulation used the virtual sensor concept of (Yousef et al., 2014). The IPSO can generate new path which is feasible, safe, and short when UAV encounters obstacles. In this study we assume the obstacle is round and combine the obstacle's tangent line to bypass it. The proposed method decreases the time consumption dramatically. Comparison with APF, ACO, and RRT algorithms is given in the Experiment section.

II. SYSTEM SETUP

The rotorcraft system includes flight control board, motors, remote control RC radio, and the sensors. In this study, we use the Pixhawk as flight control board on the rotorcraft, and use the 3DR Radio Telemetry as communication device to communicate with ground station. A LiDAR is used for detecting obstacles and the other sensors are in the flight control board. The flowchart of the control signal is shown in Fig. 1. The

Fig. 3. Offshore cage nets.

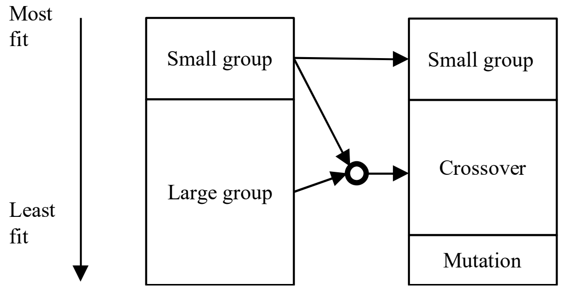

Fig. 4. The way to generate next generation.

3DR Radio Telemetry on the ground station sends control signal via RC Transmitter to the 3DR Radio Telemetry on the UAV through the RC Receiver. The control signal is then sent to the flight control board, Pixhawk, of the UAV. Range, location, and heading information are provided by the sensors, compass, and GPS on the UAV.

In order to make a well experiment and more precise result, we choose a hexacopter as the carrier (Fig. 2). In general, more rotors give better controllability and stability. The body structure of the rotor copter is made of carbon fibers which can provide more strength. For avoiding obstacles, real time computation on the hexacopter is needed. In this study we use the Raspberry Pi3 to take charge of calculations. It can communicate with flight control board. We can use it to calculate the control signal and send the command to flight control board. It is expandable and compatible with Python.

III. PATH PLANNING

In general, there are two types of environment for path planning problems. One is known environment, which allows the device to plan the route bypass the obstacles beforehand, no matter the obstacle is stationary or dynamic. The other one is unknown environment, the control device needs to generate a new route when it encounters unknown obstacles. At first, we should arrange the order of the cage nets because we need to drop the suspensory sensor in the cage nets to get the information by the UAV. The number of cage nets is definitely not only one, as shown in Fig. 3. The problem of this study can be interpreted as the TSP. And then we plan the route between the cage nets to implement the cage inspection mission. Introduction of the algorithms that we used to plan path is given as follows.

1. Random Bias Genetic Algorithm (RBGA)

An improved method was proposed for traveling salesman problem by Gupta et al. (2017). Randomized bias genetic algorithm (RBGA) uses the elitist technique. The population is divided into two groups, small group and large group. Small group keeps the 20% fittest individuals and the next generation will inherit the whole individuals of small group. Then the large group keeps the remaining individuals. The crossover process is that the mating is performed by one parent selected from small group randomly and the other parent selected from large group randomly, it is also the core of this algorithm. And according to the crossover rate, for example the crossover is 0.7, the child inherits the gene with the probability 0.7 from parent one and the remaining gene with the probability 0.3 from the other parent. Additional part is from the mutation, the 10% mutant individuals generated randomly to next generation. It also represents the next generation composes of 20% small group, 70% by crossover, and 10% by mutation. The algorithm keeps the best solution to next generation by elitist technique, it can reduce execution time and its bias selection of parent also can reduce the mean error.

The way to generate the next generation is shown in Fig. 4. The concept of the RBGA is given as follow. At the beginning, numbering the cage's coordinates, then send it as RBGA's input.

1) Initialize the population randomly.
2) Keep small group and large group separately.
3) According to the crossover rate, use crossover operator with 1st parent from small group and with 2nd parent from large group.
4) According to the mutation rate, generate the individual randomly by mutation operator.
5) The new generation includes the individuals by crossover operator, the individual by mutation, and the small group.
6) Repeat the procedure until the termination criterion is true.

This genetic algorithm uses the uniform crossover originally. Because we numbering the cage nets to denote as the city number, we use another crossover operator which is used for the TSP. It works in the following way:

1) One parent is selected from small group and the other parent is from large group, indicated as A and B.
2) Generate the two intersection point randomly, indicate them as i and j, i is smaller than j.
3) Individual P and P' are generated and let P equal to B, then remove the number which belongs to the sequence $\{A_i,\cdots,A_j\}$ become to P'. Finally we use following way to generate the individual X after crossover. The m is the number of order in the individual.

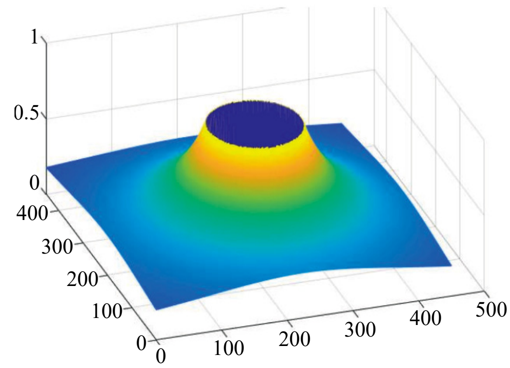

Fig. 5. The schematic diagram of the repulsive force.

$$X_m = \begin{cases} P'_m & m \geq 1 \ \ and \ \ m < j \\ A_{m'} & m \geq i \ \ and \ \ m \leq j \\ P'_{m-j+i-1} & m > j \ \ and \ \ m \leq n \end{cases} \tag{1}$$

4) The other individual Q and Q' are generated and let Q equal to A, then remove the number which belongs to the sequence $\{B_i,\cdots,B_j\}$ become to Q'. Finally we use following way to generate the individual Y after crossover.

$$Y_m = \begin{cases} Q'_m, & m \geq 1 \ \ and \ \ m < j \\ B_m, & m \geq i \ \ and \ \ m \leq j \\ Q'_{m-j+i-1}, & m > j \ \ and \ \ m \leq n \end{cases} \tag{2}$$

For example, assume we have n cities and n = 10, A = (1,2,3,4,5,6,7,8,9,10), B= (2,3,8,9,6,10,4,1,7,5), i= 3, j= 7, then P' = (2,8,9,10,1), Q' = (1,2,3,5,7), finally X = (2,8,3,4,5,6,7,9,10,1) and Y = (1,2,8,9,6,10,4,3,5,7). And the process of mutation is to select two cities in the individual randomly, then change their position. For example, the individual $C = (C_1, \cdot\cdot C_i, \cdot\cdot C_j, \cdot\cdot C_n)$ and the selected point is i and j, therefore the individual after mutation is $C = (C_1, \cdot\cdot C_j, \cdot\cdot C_i, \cdot\cdot C_n)$.

2. Artificial Potential Field Method (APF)

The artificial potential field algorithm was first proposed by Khatib for path planning (Khatib, 1986). The artificial potential field was composed by two forces, the attractive force generated by the goal point and the device, and the repulsive force generated by the obstacles and the device. The concept is that when the device approaches the obstacles then the repulsive force increases, and when the device approaches to the goal the attractive force decreases. In general, the repulsive force is inversely proportional to the distance between the device and obstacles. Assume the position of obstacle is (200,225), then the repulsive force is shown in the Fig. 5. We can see the relationship between the force and the obstacles. When the distance is closer to the obstacles, the repulsive force will be larger. On the other hand, the attractive force is proportional to the distance between goal and the device. The principle of the artificial potential field is utilizing the sum of the repulsive force and the attractive force to plan the route.

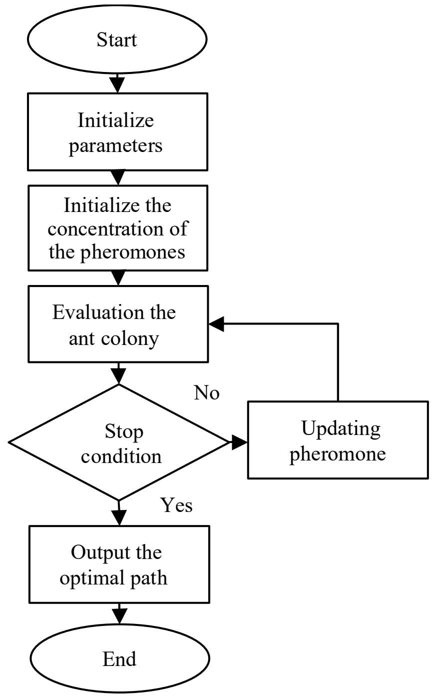

Fig. 6. The flowchart of ant colony algorithm.

3. Ant Colony Optimization (ACO)

Ant colony optimization is used to solve the traveling salesman problem firstly (Zhang et al., 2015). It is a kind of algorithm that imitate the real ants utilize pheromones to communicate with other ants. At the beginning, the ants move randomly in the nature and remain the pheromone. When the other ants encounter the path with pheromones, it will decide whether to follow the path or not. It will make the decision by a probability, as the formula (3). $\tau_{i,j}$ is the pheromones intensity for path i to j and the $\eta_{i,j}$ is the initial intensity of the pheromone between i and j, generally $\eta_{i,j} = 1/d_{i,j}$. α and β are the important parameters to determine the influence of pheromone.

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})} \tag{3}$$

If the ant follows the path, it will remain the pheromones, so the intensity of the pheromones will increase. If not, the pheromones on the path will decrease with the volatilization, so the intensity of pheromones will be updated. ρ is the pheromone evaporation coefficient, L_k is the tour length of the ant_k and the $\Delta\tau_{i,j}^{k}$ is the increment of pheromone on path i to j by ant_k. The ACO procedure is shown in Fig. 6.

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j} \tag{4}$$

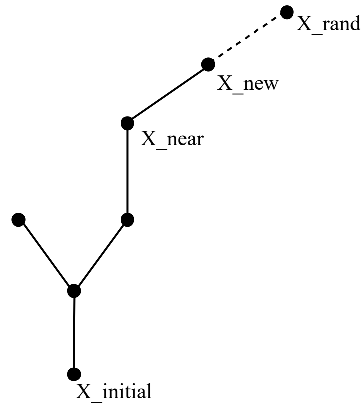

Fig. 7. Node extension.

$$\Delta\tau_{i,j}^{k} = \begin{cases} 1/L_k, & \textit{if } \text{ant}_k \text{ pass the path } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Through this mechanism, more ants pass the path and remain more pheromones on the path. Finally we can get the optimal route. After the wide study by many researchers, many improved algorithms based on the ant system are also proposed. When solving different problems, the ant colony algorithm encountered the parameters that need to be tuned.

4. Rapidly Exploring Random Tree Algorithm (RRT)

The rapidly exploring random tree algorithm (RRT) was proposed by LaValle. The RRT is widely used for solving the path planning problem of robot in high dimensional space or complex environment (Sun et al., 2018). The advantage of the RRT is that it does not need to model the space and does not need to divide the search space. It also has high coverage in the search space, so its area of search is large. It can explore unknown region as much as possible. Although RRT has many good features, it does not consider the cost of the path in the search space. Because its randomness leads to the algorithm contain some disadvantages.

1) When planning for the same mission, it will have different result.
2) The planned path is often just the feasible path not the optimal path.
3) The rapid exploring random tree without any weight or experience towards the goal orientation, so the speed of convergence may be slow.

The basic RRT algorithm assumes the mission region is R, R_{free} represents the free area and the R_{obs} represents the obstacle area. The following description gives the components of R.

$$R = R_{free} \cup R_{obs} \tag{6}$$

$$R_{free} \cap R_{obs} = \varnothing \tag{7}$$

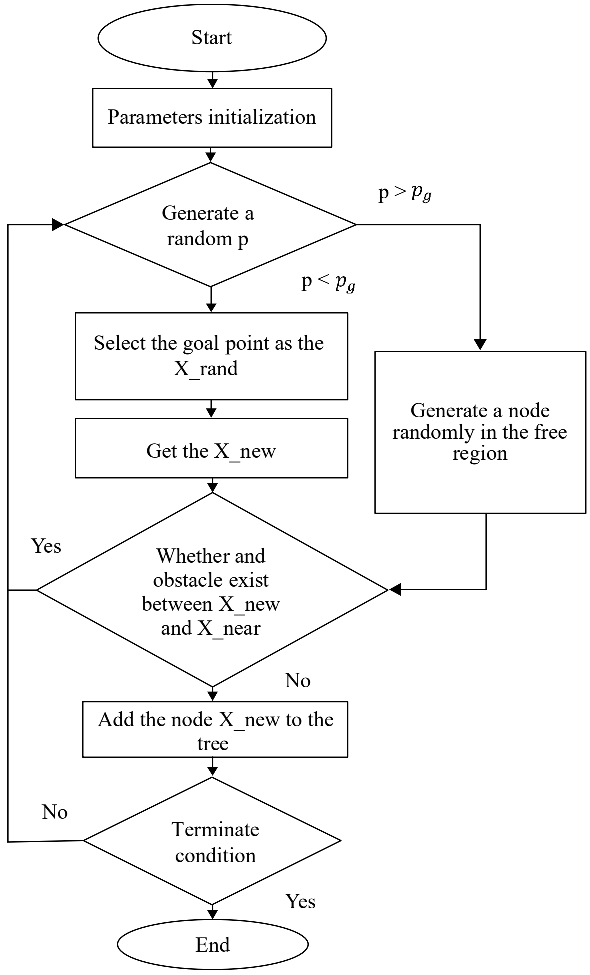

Fig. 8. The flowchart of basic RRT algorithm.

In general, we set the initial position as $X_{initial}$ and the goal as X_{goal}. As long as the goal point can be reached, the RRT algorithm can ensure to find the path from the start point to the goal point after enough time of execution. The extension process is shown in Fig. 7. The new node is generated with probability p_g , when the $p < p_g$, the node select the goal point as X_{rand}, then determine X_{near}, so X_{near} will extend step length forward to the goal point. When the $p > p_g$, the algorithm will generate a node X_{rand} randomly in the free region, determine X_{near}, then extend step length forward to the X_{rand}.

The following is the procedure of basic rapidly exploring random tree algorithm (Fig. 8).

IV. IMPROVED PARTICLE SWARM OPTIMIZATION

The basic PSO algorithm is a concept that imitates the behavior of bird flock. It uses the uniform distribution to generate the initial position of the particle, and exchanges the information of personal and social to update the particle's velocity, then updates the particle's position to achieve optimization.

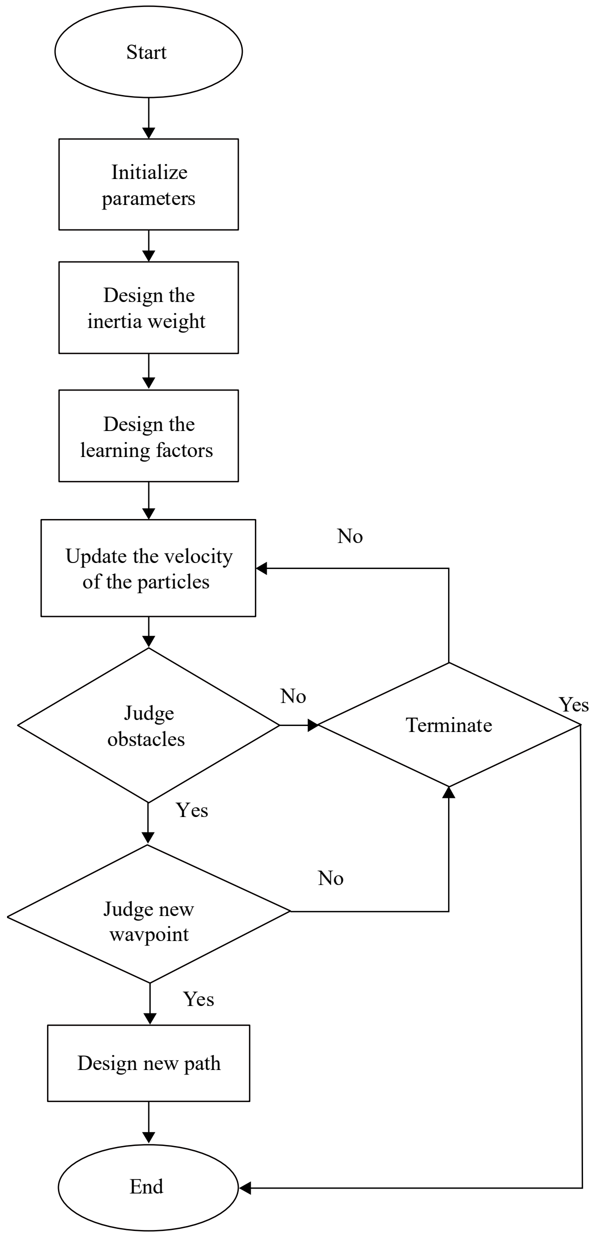

Fig. 9. The flowchart of IPSO.

The procedure of basic PSO algorithm is shown below:
1) Initialize relative parameters.
2) Check the goal, if it is not, move to step 3).
3) Update the particle's velocity and position.
4) Check better personal position then update the particle's best position.
5) If have better global position, then update the swarm's best position and move to step 2) until arrive the goal.

We improved the algorithm based on the IPSO (Zhang et al., 2018). In order to apply this algorithm to UAV, we combine some concept to reduce execution time, make the route shorter, and avoid the local optimal. This study uses some basic parameters setting in original IPSO which can keep its advantage.

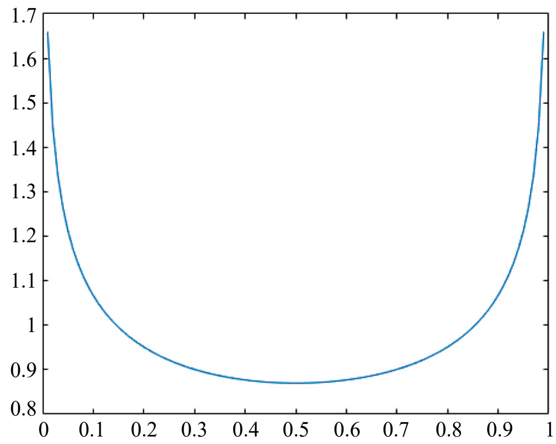

Fig. 10. Beta distribution with $a = b = 0.8$.

The procedures of the proposed IPSO are given as follows. Figure 9 is the flowchart of the IPSO.

1. Initialization of Particle Swarm

As mentioned above, the initial distribution of the particles will affect the efficiency of the algorithm. The traditional method uses the uniform distribution to generate the initial position of the particle swarm usually. The advantage of the uniform distribution is easy execution. But this distribution will not let the particles surround the optimal solution. The literature (Zhang et al., 2018) proposed a beta function distribution to improve the way of the particles' distribution.

$$\beta(x) = \frac{x^{a-1}(1-x)^{b-1}}{\int_0^1 t^{a-1}(1-t)^{b-1}dt}, 0 < x < 1 \tag{8}$$

In the paper, it sets $a = b = 0.8$, the probability density function is shown in Fig. 10. We can find the initial position of particles will exist near the boundary of the search space through the shape of the function. The following shows how the formula initializes particles in each dimension.

$$X_i = min_j + \left(max_j - min_j\right) \times betarand(a,b) \tag{9}$$

The $[min_j, max_j]$ is the range of the j dimension and the candidate solution is X_i. In this formula, $i = 1, 2, \cdots, M$, M is the number of particles, the $betarand$ function can generate random number from the beta distribution.

2 Design of Inertia Weight

The inertia weight is an important parameter in the PSO. The larger inertia weight can let the global search ability better and the smaller inertia weight can have local search ability. We usually focus on the global search ability at the beginning, therefore use larger inertia weight. And we should use smaller inertia weight later to improve the local search ability. The

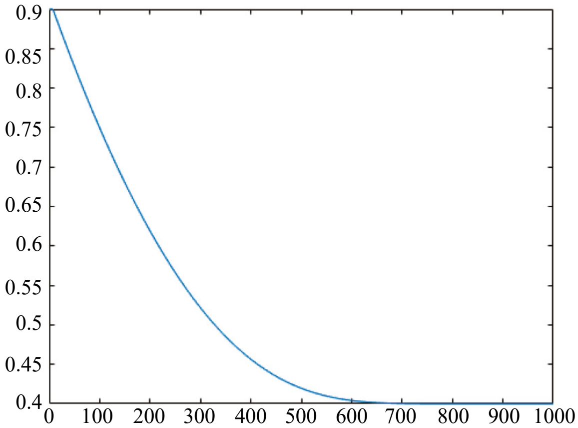

Fig. 11. The relationship between inertia weight and iterations.

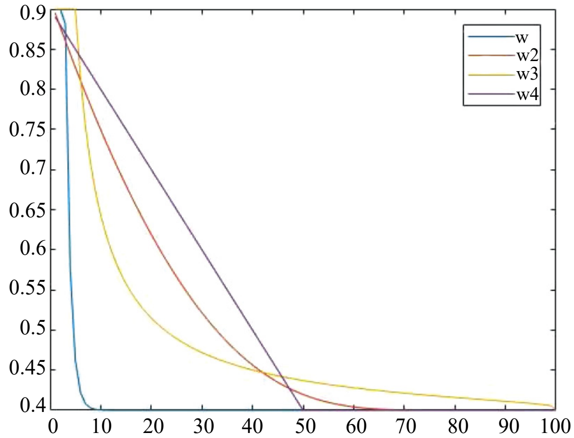

Fig. 12. Inertia weight change curve.

original paper used inverse incomplete function (Zhang et al., 2018), it approached the exponentially decreasing. The formula combines inverse incomplete function γ to generate inertia weight.

$$\gamma(\lambda, a) = \int_0^\lambda e^{-t} t^{a-1} dt \quad (10)$$

$$w(t) = w_{min} + \frac{w_{max} - w_{min}}{\lambda} \times \frac{\gamma(\lambda, 1 - \frac{t}{\max iteration})}{\lambda} \quad (11)$$

t represents the current number of iterations. $\max iteration$ represents the maximum number of the iteration. w_{min} and w_{max} represent the minimum and maximum value of the inertia weight. Usually, the maximum value of the inertia weight is 0.9 and the minimum value of the inertia weight is 0.4. Assume the maximum iteration is 1000, the inertia weight with λ = 0.05 is shown in Fig. 10. We can find the function has decreased in the first half of the iterations. Through this way, the value λ is set to 0.05 to decrease the inertia weight dynamically and it also can avoid the premature convergence phenomenon.

Table 1. Comparisons of execution time (sec).

Time Drop	50	60	70	80	90	100
Figure7	3.009	2.719	2.827	2.704	2.682	2.702
Figure8	8.083	7.968	7.924	7.584	7.753	7.926
Figure9	23.29	22.99	23.03	22.61	22.95	22.71

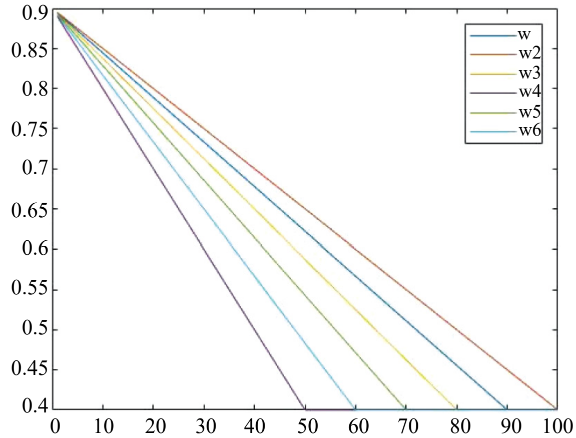

Fig. 13. Inertia weight change curve.

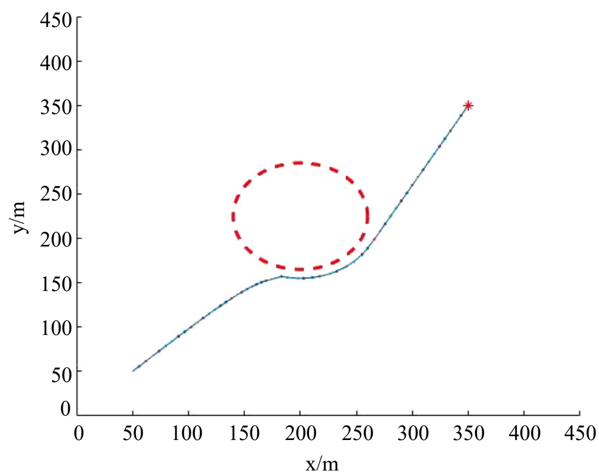

Fig. 14. The known obstacle's radius is 60.

At the beginning, we use different functions, as shown in Fig. 12, to observe their difference. The linear function can address unknown obstacles and is better than others obviously. We attempt to use three conditions to test the effectiveness of different slope functions, as shown in Fig. 13, and ensure they can handle the problem of local minimum.

Figures 14 to 16 show simulations of path planning and obstacle avoidance from starting position to target position under different environments. The target position is the red asterisk. The blue line is the optimal searched and obstacle avoidance path.

Table 1 shows the execution times by using different inertia weight change curves under different environments. The red

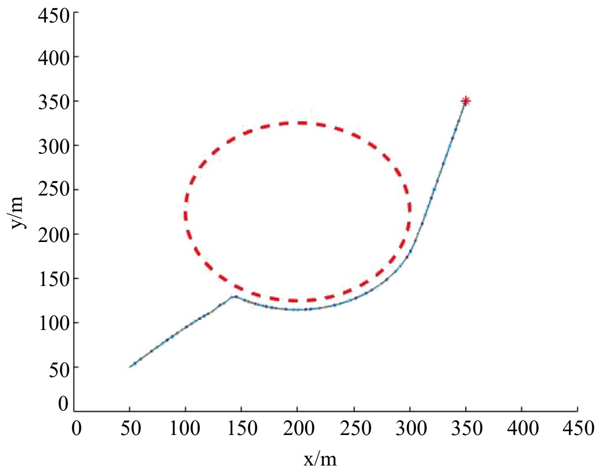

Fig. 15. The known obstacle's radius is 100.

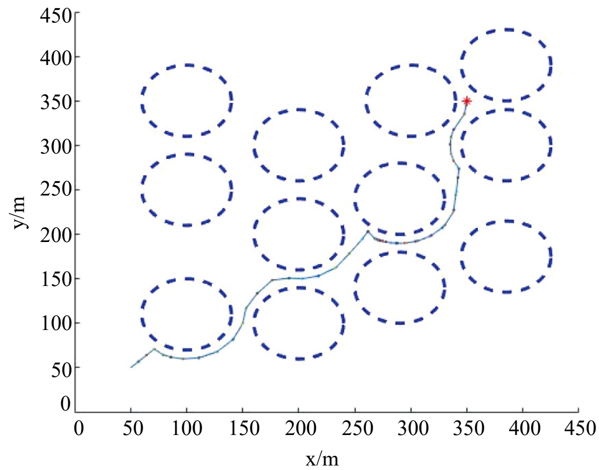

Fig. 16. Multiple obstacles without falling in local minimum.

circle in Fig. 14 and Fig. 15 is the known obstacle. The blue circles in Fig. 16 are the unknown obstacles. The other parameters of IPSO are the same. We can find the better result when the minimum inertia weight linearly drops at the 80 to 90. And the most important is appropriate inertia weight can shorten the convergent time of path planning.

3. Design of the Learning Factors

The learning factors represent the communication between the particles. Usually, we want to focus on the global search ability at the beginning of the PSO, and want to enhance the local search ability gradually later. In (12), the c_1 represents the weight of self-learning, the particle remembers the best location where it had ever been, it will compare with current location. In (13), the c_2 represents weight of social learning, the information from the particle swarm, the swarm record the best location no matter which particle had ever been, and it will compare with current location. In order to achieve the equilibrium of the global exploration and local exploitation, the following formulas are designed to change the value dynamically [21].

$$c_1 = c_{1,initial} + \frac{c_{1,final} - c_{1,initial}}{\max iteration} \times t \tag{12}$$

$$c_2 = c_{2,initial} + \frac{c_{2,final} - c_{2,initial}}{\max iteration} \times t \tag{13}$$

In the initial stage of the particle swarm optimization, we want the self-learning stronger and social learning weaker to realize the better global search ability, and want the self-learning weaker and social learning stronger later to enhance the local search ability. In the formula, the $c_{1,initial}$ and $c_{2,initial}$ represent the initial value of c_1 and c_2. The $c_{1,final}$ and $c_{2,final}$ represent the final value of c_1 and c_2. The t represents the current number of iterations. The max *iteration* represents the maximum number of iteration which is predefined.

4. The Updating Method of Particle's Velocity

This is the core of the whole algorithm, for the balance of the global exploration and the local exploitation, the updating method of particle's velocity is described as below,

$$v(t+1) = w(t)v(t) + c_1 r_1 (P_i - X(t)) + c_2 r_2 (P_g - X(t)) + \delta(t) \tag{14}$$

We keep the δ, because it can help for the global exploration and local exploitation [21]. The w is inertia weight, the c_1, c_2 are learning factors, P_i is personal best, P_g is global best, and the X is current position. And the δ is designed as follow,

$$\delta_i(t) = \left(1 - \frac{t}{\max iteration}\right)^a \left(\frac{1}{N}\sum_{i-1}^{N} P_i^L - X_i\right) \times rand \tag{15}$$

rand is a random number from the interval (0, 1) and it is uniform distribution in the interval. The $\delta_i(t)$ depends on the value of a. The value of $(1\text{-t/ maxiteration})^a$ is larger in the initial stage of iterations, so the value of $\delta_i(t)$ is larger. The $\frac{1}{N}\sum_{i=1}^{N} P_i^L$ is the average value of the particles' best position. From simulations, a = 5 has the best performance. Through this way, we can enhance the global exploration in the initial stage. The δ will decrease gradually later, this condition can enhance the local exploitation. It is beneficial to the balance between global exploration and local exploitation.

5. Judgment of Obstacle

In general, we apply the algorithm when the UAV encounters the obstacle. Since the obstacle is known, we assume the obstacle is round, get the obstacle's tangent line from the current point and check whether it will cross the obstacle or not.

1) Angle between obstacle and target smaller than angle between obstacle and tangent line.
2) Distance of current point to target point is larger than or equal to the distance of current point to obstacle center.

If it is true, then we use a slope to generate the new route. If it

Table 2. Comparisons of RRT, ACO, APF and IPSO.

	RRT	ACO	APF	IPSO
Nodes	43.4	33.1	62.2	33
Length	565.1	563.3	496.8	486.5
Time(s)	16.4	8.5	8.9	2.6
Failure	3	0	0	0

Table 3. Comparison of different algorithms in multiple obstacles.

	RRT	ACO	APF	IPSO
Nodes	67.2	41.5	63.2	40.1
Length	688.4	664.2	522.1	489.4
Time(s)	64.5	14.1	12.7	7.7
Failure	0	0	0	0

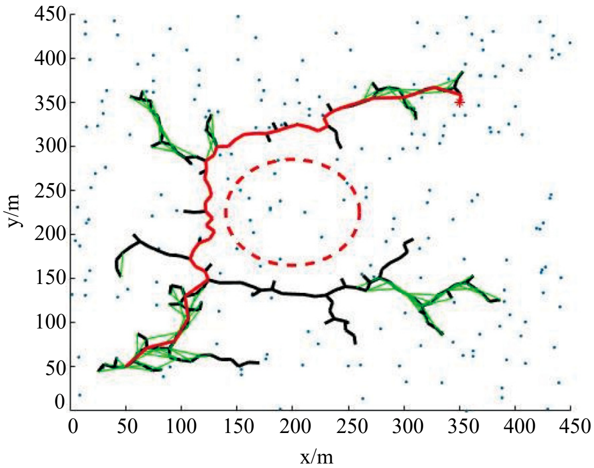

Fig. 17. Simulation of RRT.

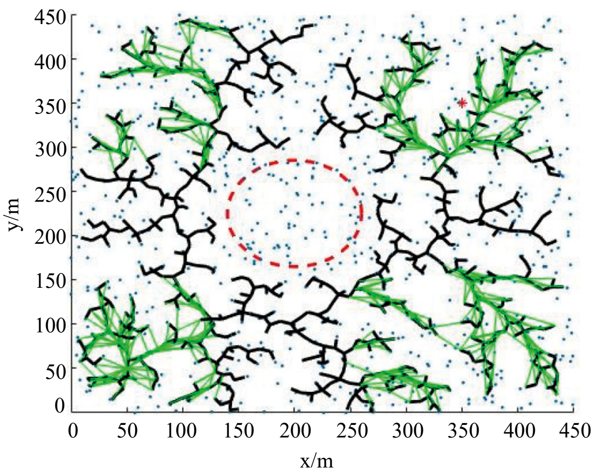

Fig. 18. Failure simulation of the RRT.

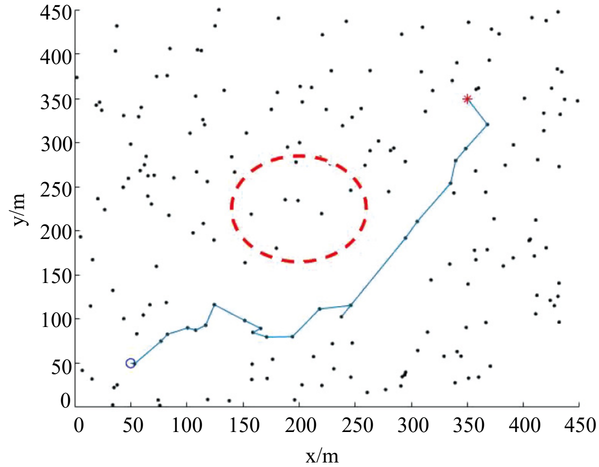

Fig. 19. Simulation of ACO.

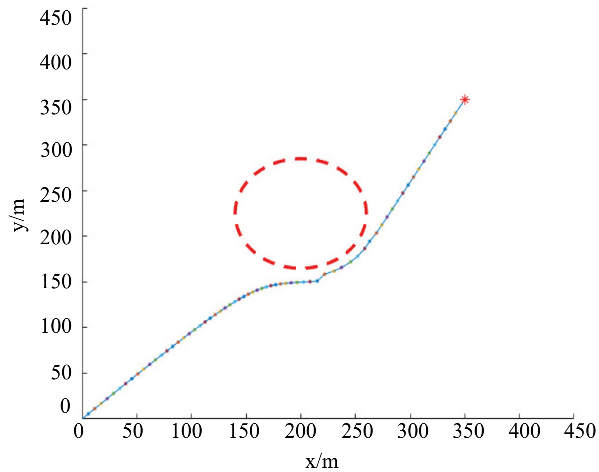

Fig. 20. Artificial potential field.

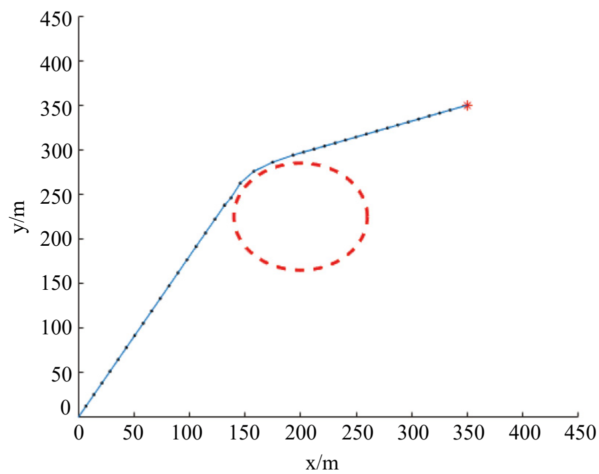

Fig. 21. Simulation of the proposed IPSO.

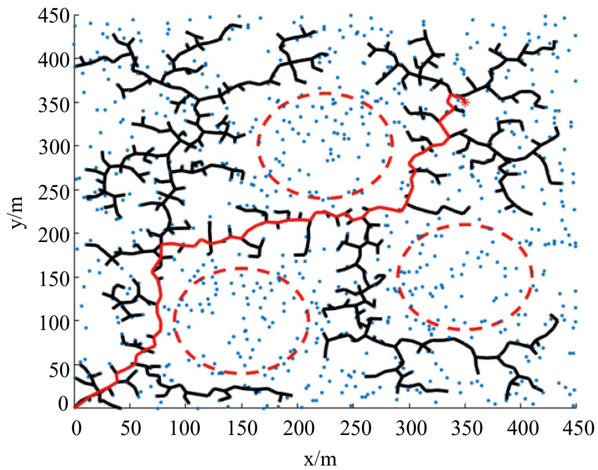

Fig. 22. RRT in multiple obstacles.

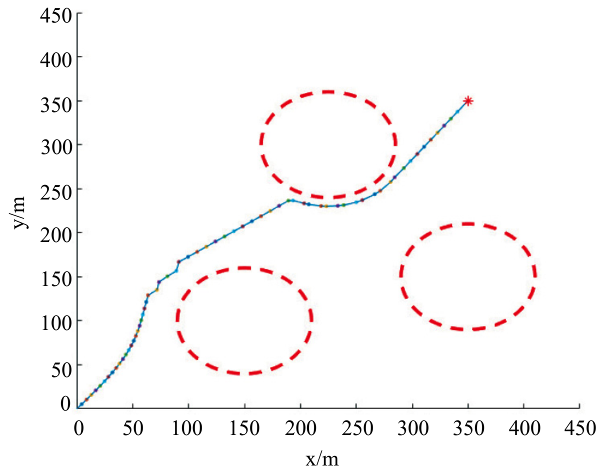

Fig. 24. APF in multiple obstacles.

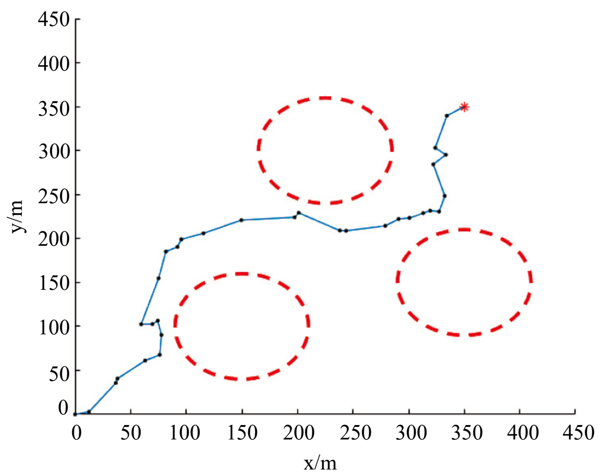

Fig. 23. ACO in multiple obstacles.

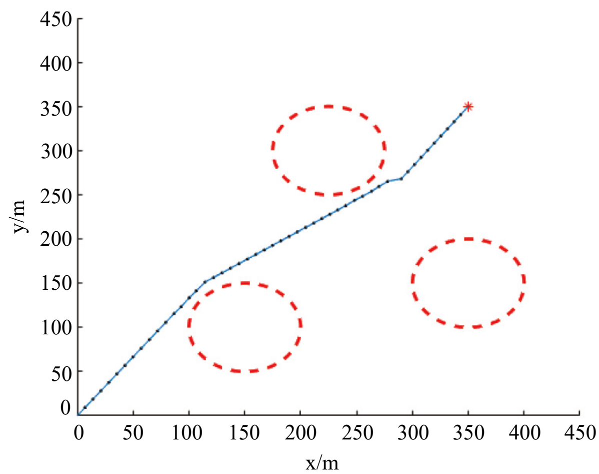

Fig. 25. IPSO in multiple obstacles.

is not, we will move forward to the target point. And we set criterion to ensure bypass the obstacle, when the waypoint is far away from the obstacle, it means bypass the obstacle successfully.

6. Judgment of New Waypoint

We set the step length as the radius of circle with the center of current point. This process can generate a new waypoint in the safe range and shorter route, and assure we go to the goal point step by step. IPSO will generate the random points in the circle and calculate the cost value by cost function. It can find the best solution in the preset round range. And we choose the best solution as the new node.

V. EXPERIMENTS

Procedures of the simulations are given as follows.

1) Set the start point and cages points. Use RBGA to generate the order of waypoints.
2) Check obstacle. In order to improve the obstacle avoidance and the implement time, we use new concept to shorten the route. And check whether the UAV will encounter the obstacle that can be avoided early. Finally, we use the step length to ensure the safety of the route.
3) Initialize the parameters of IPSO. The nodes of path are generated by RBGA. Path planning and obstacle avoidance is executed by using the IPSO, setting the nodes, the number of particles, the dimension, the inertia weight, the learning factors, the maximum iterations, and the step length to ensure the points of route will not cross the obstacle's boundary.
4) Calculate the fitness value of the initial swarm and update the global optimal value.
5) According to the rule which we set to update the particles, when the criterion is satisfied, the optimal route will be generated. And when the obstacle is bypassed, the UAV will move forward to the target.

Comparison of one obstacle with different algorithms is shown

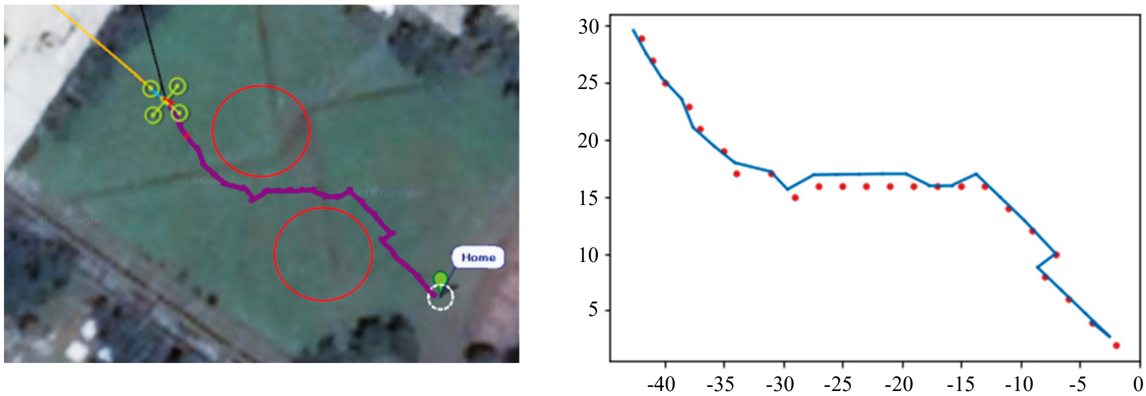

Fig. 26. Actual route shown on Mission Planner.

in Table 2. We set the step length is 10 and the known obstacle's radius is 60 to test the effectiveness of different algorithms.

Figures 17 to 21 show the path planning and obstacle avoidance simulations with respect to different algorithms. In Fig. 17 and Fig. 18, the red line is the simulated path, the black line is the tree generated by the random tree algorithm, and the green line is the searched local path. From Fig. 21 we know that the proposed IPSO has better performance than others. It has shortest distance to the goal point. The path of IPSO also has less heading changes and turning angles.

When the UAV encountered multiple obstacles, IPSO will compare different path length and choose the better one. The results of different algorithms in Table 3 of the simulations are the mean values of ten times of simulations. The simulations of different algorithms and schematic diagrams are shown in Fig. 22 to Fig. 25.

Field test of multiple obstacles using IPSO is shown in Fig. 26.

VI. CONCLUSIONS

This study focuses on the path planning with cage nets on the ocean. We considered the traveling salesman problem first to decide the order of flight path. There are two types of environments to be considered. One is known obstacles, the other is unknown obstacles. The algorithm that we used is based on IPSO. In the environment with unknown obstacles, we use the LiDAR to detect the obstacles. If there is any obstacle on the route, the control scheme will generate a new path to avoid the obstacle. Finally, we use the hexacopter to realize the control system. Experiments show that the proposed method has better performance for path planning and obstacle avoidance.

ACKNOWLEDGEMENT

This research was supported by the MOST AI Biomedical Research Center at NCKU, Taiwan.

REFERENCES

Altug, E., J. P. Ostrowski and C. J. Taylor (2003). Quadrotor control using dual camera visual feedback. Proceedings of the IEEE International Conference on Robotics & Automation, Taipei, Taiwan, September 3, 4294-4299.

Altug, E., J. P. Ostrowski and C. J. Taylor (2005). Control of a quadrotor helicopter using dual camera visual feedback. The International Journal of Robotics Research, 329-341.

Altug, E., J. P. Ostrowski and R. Mahony (2002). Control of a quadrotor helicopter using visual feedback. Proceedings of the IEEE International Conference on Robotics & Automation, Washington, DC. May 1, 72-77.

Chen, L., C. Liu, H. Shi and B. Gao (2013). New robot planning based on improved artificial potential field. Proceeding of the Third International Conference on Instrumentation, Measurement, Computer, Communication and Control.

Chen, Y. and T. Li (2017). Collision avoidance of unmanned ships based on artificial potential field. Proceeding of the Chinese Automation Congress, 2017.

G. D. Bothezat (1920). The General Theory of Blade Screws

Gupta, I. K., A. Choubey and S. Choubey (2017). Randomized bias genetic algorithm to solve traveling salesman problem. Proc. of 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

Hsiao, Y.T., C.L. Chuang and C.C. Chien (2004). Ant colony optimization for best path planning. Proceeding of the IEEE International Symposium on Communications and Information Technologies.

Hu, Y., X. Huang, Q. Tao, S. Wang, H. Liu, G. Guo, and J. Xu (2016). Design of a deep water cage culture environment monitoring system based on ZigBee. Proc. of IEEE International Conference on Computer and Communications, 1545-1549.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research.

Korkmaz, M. and A. Durdu (2018). Comparison of optimal path planning algorithms. IEEE TCSET.

Luo, D., L. Zhang and Z. Xu (2011). Heuristic simulated annealing genetic algorithm for traveling salesman problem. Proc. of 6th International Conference on Computer Science & Education.

Reshma, B. and S. S. Kumar (2016). Precision aquaculture drone algorithm for delivery in sea cages. Proc. of IEEE International Conference on Engineering and Technology.

Shi, P. and S. Jia (2013). A hybrid artificial bee colony algorithm combined with simulated annealing algorithm for traveling salesman problem. Proc. of International Conference on Information Science and Cloud Computing Companion.

Shi, P. and Y. Cui (2010). Dynamic path planning for mobile robot baesd on genetic algorithm in unknown environment. Proceeding of the IEEE Chinese Control and Decision Conference.

Sun, Q., M. Li, T. Wang, and C. Zhao (2018). UAV path planning based on

improved rapidly-exploring random tree. Proceeding of the Chinese Control and Decision Conference.
Wang, L.Y., J. Zhang and H. Li (2007). An improved genetic algorithm for TSP. Proc. of International Conference on Machine Learning and Cybernetics.
Xu, X. and X. Zhang (2007). A remote acoustic monitoring system for offshore aquaculture fish cage. Proc. of International Conference on Mechatronics and Machine Vision in Practice 86-90.
Yousef, M. T., H. E. I. Ali, S. M. Habashy and E. M. Saad (2014). Adaptive controller based PSO with virtual sensor for obstacle avoidance in dynaic environments. Proceeding of the IEEE NRSC.
Yu, F., X. Fu, H. Li and G. Dong (2016). Improved roulette wheel selection-based genetic algorithm for TSP. Proc. of International Conference on Network and Information Systems for Computers.
Zhang, D., Y. Xian, J. Li, G. Lei and Y. Chang (2015). UAV path planning based on chaos ant colony algorithm. Proc. of International Conference on Computer Science and Mechanical Automation.
Zhang, D., Y. Xu and X. Yao (2018). An improved path planning algorithm for unmanned aerial vehicle based on rrt-connect. Proceeding of the IEEE Proceedings of the 37th Chinese Control Conference.
Zhang, J., X. Chen, H. Hu, J. Zhang and L. Wang (2018). Global path planning algorithm based on the IPSO Algorithm for USVs. Proceeding of the IEEE The 30th Chinese Control and Decision Conference.
Zhang, X., X. Xu, Y. Peng, Z. Zou and G. Su (2012). Research on centralized remote monitoring system for offshore cage farm. Proc. of 2012 Oceans.