International Journal of Innovative
Computing, Information and Control
Volume **11**, Number **3**, June **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 833–850

ROBOT PATH PLANNING BASED ON ADAPTIVE INTEGRATING OF GENETIC AND ANT COLONY ALGORITHM

Jiqing Chen, Shaorong Xie, Hengyu Li, Jun Luo and Kai Feng

School of Mechatronic Engineering and Automation
Shanghai University
No. 149, Yanchang Road, Shanghai 200072, P. R. China
{ cjq_shu; lihengyu; srxie; luojun; freshbreeze }@shu.edu.cn

Received July 2014; revised December 2014

Abstract. *Genetic Algorithm (GA) and Ant Colony Algorithm (ACO) are two popular intelligent optimization algorithms for robotic path planning. Genetic Algorithm has strong rapid global search capability, but it cannot use feedback information and do a lot of redundancy iteration, so convergence speed declines. Ant Colony Algorithm has good feedback information. However, due to lack of initial pheromone, the solving is slow. Based on these, an Adaptive Genetic Ant Colony Algorithm (AGAACO) is proposed in this paper. Firstly, a group of solutions generated by Genetic Algorithm is conducted as the initial pheromone of Ant Colony Algorithm. Then a convergence speed parameter is used to combine Genetic Algorithm and Ant Colony Algorithm adaptively. It improves the convergence speed and the problem of local optimum. At last, AGAACO is verified by some simulations of path planning. It shows that the algorithm is better than single Genetic Algorithm or Ant Colony Algorithm.*

Keywords: Genetic Algorithm, Ant Colony Algorithm, Robotic path planning

1. **Introduction.** Navigation is the core of autonomous movement for robot. Path planning is searching a no collision path from the starting point to the target point for mobile robot. According to different levels of environmental information, path planning is generally divided into global path planning and local path planning. Global path planning is used when the environmental information is known, and this method is usually used for indoor service robot. Local path planning is used in environmental information which is unknown or owns dynamic changes, such as the environmental detection.

In the process of exploring environment, the best route of path planning is essentially a shortest path problem. The shortest path problem is not only the shortest distance in geography, but also other factors, such as time, fee, information capacity and safety. The traditional global path planning methods are grids method, topology law, diagram, free space method, etc. These conventional methods are low computational efficiency, and not suitable for high-dimensional optimization problem. With neural networks, genetic algorithms, ant colony algorithm and intelligent bionic optimization algorithm proposed and the rapid development, many scholars applied these algorithms to robot path planning, and achieved good effects.

Ant Colony Algorithm is a random optimization method developed in recent years. It was proposed by M. Dorigo. Ant Colony Algorithm reaches the optimization target mainly through information transmission between the ant colonies. Robot path planning is very similar to the foraging behavior of ants. There are two advantages. The first, its principle is a positive feedback mechanism, and finally converges to the optimal path by constantly updating pheromone. The second, it is a universal method of stochastic optimization with distributed global optimization characteristics, not only for single-objective optimization, but also for solving multi-objective optimization [1-5]. The drawback is the lack of pheromone at the early stage of searching. It makes the accumulation time of pheromones become longer, so solving is slow.

Genetic Algorithm was proposed by John Holland in 1975, University of Michigan. Genetic Algorithm firstly generates a random initial population, then simulates genetic selection and biological evolution natural selection, evolving and generates new populations. The individual population is evaluated by fitness function. According to survival of the fittest principle, the evolution is guided toward the optimal one. At the same time, the best individual of populations is optimized by global parallel search method in order to get the optimal solution satisfied requirements. There are three advantages [6-8]. The first, it has global search capability with a wide range, parallelism and strong robustness. Second, search uses the evaluation function, iteration uses probabilistic mechanism. Process is simple, and with randomness. Third, it is scalability and can be combined with other algorithms. The drawback is that the system's feedback information is not enough used, when solving in a certain range, it tends to do a lot of redundant iterations, so the accurate solutions are less efficient.

Comparing Genetic Algorithm and Ant Colony Algorithm, analysis results are shown as Figure 1. Vertical axis represents the convergence speed, and horizontal axis represents the running time of the algorithm. Genetic Algorithms have a higher convergence speed tending to optimal solution in the early search, and the latter parts tends to do a lot of useless iterations. However, for Ant Colony Algorithm, convergence speed rises fast in latter part, while in the early stage due to pheromone inadequate, search ratios become slower [9-12].

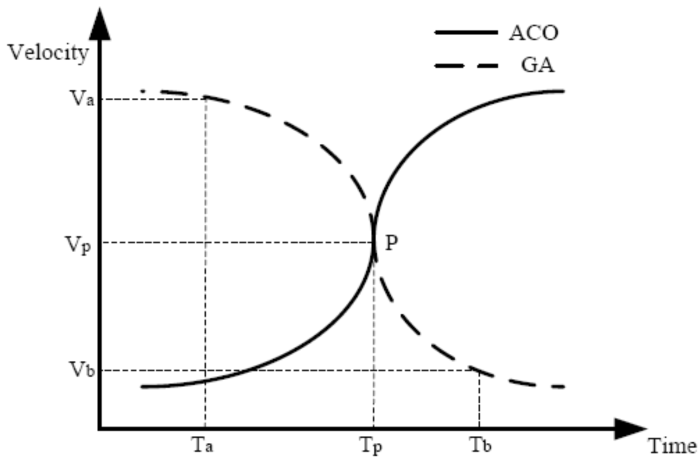


FIGURE 1. Time and velocity curves of two algorithms

So we can combine the advantages of Genetic Algorithm and Ant Colony Algorithm. With the randomness and global search characteristic of Genetic Algorithm, a group of solutions is generated as initial pheromone of Ant Colony Algorithm. Ant Colony Algorithm skips the early stage lack of pheromone, so the convergence rate is significantly higher than the two algorithms alone. That is, the two algorithms combine in point p (see in Figure 1) as close as possible.

According to the above, this paper presents an adaptive combination algorithm based on Adaptive Genetic Ant Colony Algorithm (AGAACO). A genetic operator is integrated adaptively into the basic combination algorithm in the late stage, so it improves the algorithm precocious caused by rapid convergence, the poor performance of randomness

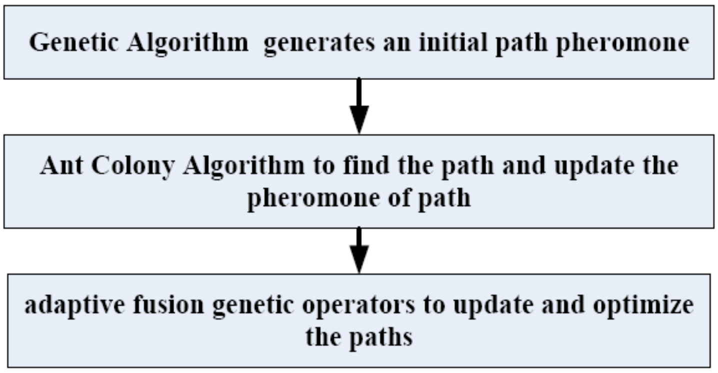


Figure 2. Structure of adaptive Genetic Ant Colony Algorithm

and global search. The algorithm is divided into three steps: Genetic Algorithm to generate an initial path pheromone and distribution; Ant Colony Algorithm to find the path and update the pheromone of path; adaptive fusion genetic operators to update and optimize the paths.

(1) An initial path pheromone distribution is generated by Genetic Algorithm. Setting path planning is a two-dimensional space. Grid method is used to divide the routing space, and then determine the path encoding and fitness function. After the initial population is generated, the population genetic evolution by selection, crossover and mutation genetic operates. When the algorithm terminates, some paths satisfying the requirement were obtained.

(2) Ant Colony Algorithm searches and updates the pheromone of path. The path obtained from Genetic Algorithm needs to be enhanced by Ant Colony Algorithm. Here a variant Max-Min Ant System (MMAS) is taken to set initial parameters. Then path is planning by MMAS and path pheromone is updated at the same time.

(3) Path is updated by adaptive genetic operator in the late stage. Based on the MMAS, the convergence rate is a greater improvement. However, in the latter part, algorithm often converges too fast, and the results fall into local optimum. A convergence speed parameter is introduction into MMAS. When the MMAS converges too fast, the path generated in the latter part is crossover and mutation one time by genetic operator. Repeat until the algorithm satisfies ending conditions and exit.

2. **Problem Statement and Preliminaries.** Firstly, the grid method is used for modeling the robot's working environment [13-15]. The grid method divided the robot two-dimensional working environment into a series of grid cells in some way. In a grid environment, the robot path planning can be expressed by an optimization problem with constraints, described as follows: in a plane environment, environmental reconstruction using the grid method, grid can be divided into two types: barriers and free grid. When a robot is placed in this grid environment, it has a certain memory and perception to acquire and store relevant information in current location and the adjacent range of grid. Given the robot starting point and target point, under the unknown environment, it avoids obstacles grid to find a shortest and best path between the start and the target point. Robot path planning optimization is showed as Figure 3. S represents the starting point, G represents the target point, dark represents barrier grid, and the arrows indicate the direction of robot's route.

This problem can be expressed by a mathematical model. Set that the grid is r lines c columns, the initial grid is (a_0, b_0), and the target grid is (a_r, b_c). The path length of robot walking is calculated by the number of traveled grids. E represents a collection of

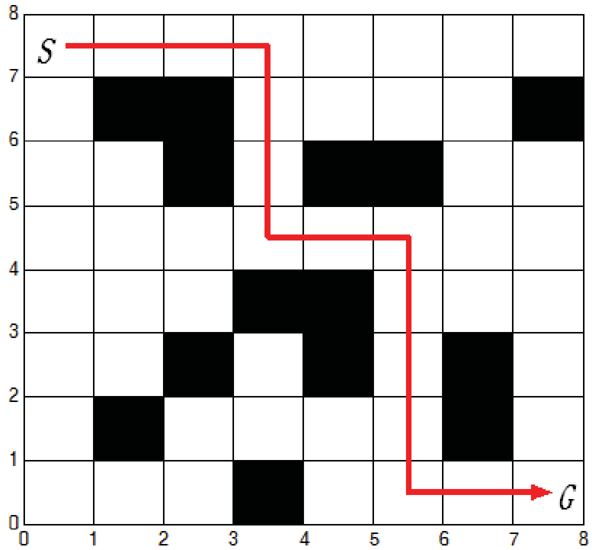


FIGURE 3. Schematic diagram of robot path planning

grids.

$$E = \{(i, j) \,|0 < i \le r,\ 0 < j \le c,\ i, j \in Z\} \tag{1}$$

$block(i, j)$ represents the information of (i, j) grid

$$block(i, j) = \begin{cases} 1 & free - grid \\ 0 & obstacle - grid \end{cases} \tag{2}$$

$D_{(i,j)}$ represents that when robot locates in (i, j) grid, a group of grids may be chosen for next step

$$D_{(i,j)} = \{(i + 1, j), (i, j + 1), (i - 1, j), (i, j - 1) \,|(i, j) \in E\} \tag{3}$$

$R(n)_{(i,j)}$ represents that robot begins at (i, j) grid, the selected grid after n step

$$R(n)_{(i,j)} = \left\{(a, b) \middle| (a, b) \in D_{R(n-1)_{(i,j)}}, block(a, b) = 0\right\} \tag{4}$$

P represents a set of all feasible solutions

$$P = \left\{R(1)_{(a_0,b_0)}, R(2)_{(a_0,b_0)}, \cdots, R(n)_{(a_0,b_0)} \middle| R(k)_{(a_0,b_0)} \in E, R(n)_{(a_0,b_0)} = (a_r, b_r)\right\} \tag{5}$$

p represents one of feasible solutions, $L(p)$ represents the path length of feasible solution P

$$L(p) = \sum_{i=2}^{n_p} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \,|p \in P \tag{6}$$

Robot path planning is to find an optimal path to walk, so that the path length J reaches a minimum

$$J = \min\{L(p), p \in P\} \tag{7}$$

If length of workspace is X, width is Y, so the relationship between the number N and the center of grid coordinates (x, y) is as follows.

$$x = (N \bmod r) \times \frac{X}{r} + \frac{X}{2r} \quad y = \mathrm{int}(N/n) \times \frac{Y}{c} + \frac{Y}{2c} \tag{8}$$

int represents an integer operation, and mod represents remainder operation.

3. **Part of Genetic Algorithm.** One motion path of robot in workspace is regarded as an individual. Sequentially connecting the robot start position S, the traversing grid, and the end position G, a path is obtained. To simplify the path encoding, only the Y-axis coordinate of each grid is recorded in this paper, that is for abscissa, a random number between $0 \sim (N-1)$ is generated by *Rand* function to identify a chromosomal gene. So the specified position abscissa of gene chromosome has been determined. At the same time, the start and the end gene encoding of each chromosome is determined, corresponding to grids of the start and the end points' Y coordinate. For this encoding mode, all individuals in the population are equal length string, which is the number of grid columns N. So the Genetic Algorithm uses sequential store structure, not only saving storage space, but also quickly location to the chromosome in specified position. For example, for a randomly generated chromosome in 5×5 grid, its storage structure and path are shown in Figure 4 and Figure 5.

Chromosome =	5	4	5	3	2

FIGURE 4. Storage structure of chromosomes

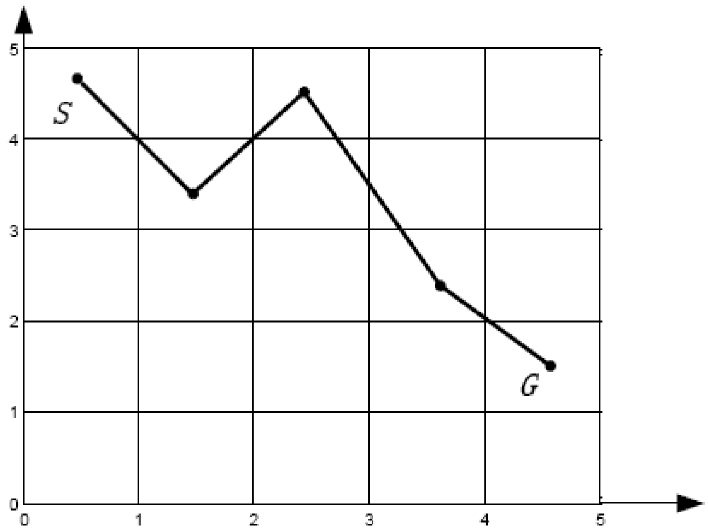


FIGURE 5. Method to generate the initial population

For the size of the initial population, the larger the number, the higher the diversity of individuals, so algorithm global search is better and the possibility of falling into local optimal solution is small. However, computation will sharply rise if the number is excessive. So in general the size of the initial population is 20∼200.

3.1. **Fitness function.** Fitness is an important indicator of the evolution of the Genetic Algorithm. In the complex environment, the fitness function requires satisfying two conditions. One is that paths can be reached within the constraints, and the other is ensuring the shortest path. So a fitness function combining the length and certain hampered is proposed in this paper.

$$F = \frac{K}{\alpha D_l + \beta F_{ob}} \tag{9}$$

K is a constant, D_l is path length, F_{ob} is the sum of barrier grid in the path. Requirement is that path findings robot can only move one grid to adjacent grid in one time, and moving from 45 degree angle is $\sqrt{2}$ times of a horizontal or vertical movement distance.

α, β represent the weighting factor of fitness value, $\alpha + \beta = 1$, in order to ensure the range of fitness values.

$$D_l = \sum_{i=1}^{N-1} d_l(i) \quad F_{ob} = \sum_{i=1}^{N-1} f_{ob}(i) \tag{10}$$

$$d_l(i) = \begin{cases} x_{i+1} - x_i & y_i = y_{i+1} \\ \sqrt{2} \cdot |y_{i+1} - y_i| & y_i \neq y_{i+1} \end{cases} \tag{11}$$

$d_l(i)$: the path length of each grid between the adjacent X coordinates, f_{ob}: obstacles grid in the path.

3.2. **Select factor.** Select factor used to determine the reorganization or cross, and the selected individuals will produce how many offspring individuals. There are various selection strategies, for example roulette method, sorting selection method, random league law, graded choice, steady-state selection method. In order to guarantee the global characteristics of algorithm, the random league law is used in this paper as selection strategy. That is to compare the fitness value of two individuals selected randomly from the population, the larger one will be inherited to the next generation populations. The above process was repeated M times, and M individuals of the next generation population can be obtained.

3.3. **Cross-factor.** Cross is the process that two parent individuals combined in certain probability to generate new individuals. Cross-factor as a major operator of Genetic Algorithm, it is guarantee of global search. The single-point crossover of a random selection is taken in this paper. Two parent chromosomes are selected as follows.

Chromosome1: a b c d e f g h

Chromosome2: a c f g b d m n

If the intersection position randomly selected is the fifth point, then the new chromosome after crossover is as follows.

Chromosome1: a b c d e d m n

Chromosome2: a c f g b f g h

The higher the probability of cross, the faster introduction of new individual populations, and the global search is stronger, but the loss probabilities of strengths individual genetic will grow. A low cross probability will make the search stop. So generally crossover probability $pc = 0.60 \sim 1.00$.

3.4. **Mutation factor.** Mutation operation is an important part to ensure the diversity of solution, actually it is offspring change generated by small perturbations. Usually, there are deleted, substituted, inserted three ways, and this paper uses substituted approach. That is selecting a variation point except for starting point and ending point, another random number generated and replace in this position. For example chromosome (1-2-3-4-5-6) breaks in 2-3 and 5-6, then 3 and 5 reverse the order of substitution, new chromosome becomes (1-2-5-4-3-6). Generally, variation factor $pm = 0.005 \sim 0.01$.

3.5. **Delete operator.** Delete operator removes the same element in the individuals, leaving only one to simplified path. For example two parent chromosomes a, b,

$$\mathrm{a} = 1\,2|3\,4\,5\,6|7\,8\,9, \quad \mathrm{b} = 9\,8|7\,6\,5\,4|3\,2\,1.$$

The b mating area is added before a, also a mating area added before b.

$$\mathrm{a}' = 7\,6\,5\,4|1\,2\,3\,4\,5\,6\,7\,8\,9, \quad \mathrm{b}' = 3\,4\,5\,6|9\,8\,7\,6\,5\,4\,3\,2\,1.$$

Sequentially delete a′, b′ the same elements in mating area, new offspring chromosomes are got.

$$\mathrm{c} = 7\,6\,5\,4\,1\,2\,3\,8\,9, \quad \mathrm{d} = 3\,4\,5\,6\,9\,8\,7\,2\,1.$$

3.6. **Genetic Algorithm end condition.** In order to ensure the time of combination, preset number of iterations and evolutionary rate are used in this paper. If maximum iterations number of Genetic Algorithm is G_{Max}, the minimum number of iterations is G_{Min}, the maximum number of stop iterations is G_{Stop}, the minimum evolution rate is G_{ratio}, and the ending condition can be described as follows.

(1) The number of evolution reached G_{Max}.

(2) The number of evolution is larger than G_{Min}, and evolution rate of successive G_{Stop} is less than G_{ratio}.

If satisfying one of the both rules, it can exit Genetic Algorithm and into the Ant Colony Algorithm.

4. **Part of Colony Algorithm.**

4.1. **Ant's initial pheromone.** The initial distribution of pheromone is crucial to guide the ants search. This paper has appropriately modified the algorithm based on Max-Min Ant System (MMAS) to realize the connection of Ant Colony Algorithm and Genetic Algorithm. In MMAS, the maximum value of pheromone is $\tau_{\max}$, the minimum value of pheromone is $\tau_{\min}$. So pheromone value in a path fluctuates between $[\tau_{\min}, \tau_{\max}]$. When current pheromone is less than the minimum, $\tau = \tau_{\min}$, pheromone is lager than the maximum, $\tau = \tau_{\max}$. Here we set the initial pheromone as follows

$$\tau = \tau_M + \tau_G \tag{12}$$

$$\tau_M = \tau_{\min} + \frac{1}{2}(\tau_{\max} - \tau_{\min}) \tag{13}$$

τ_M is the initial pheromone from Ant Colony Algorithm. τ_G is a constant, and the pheromone comes from the solving results of Genetic Algorithm.

4.2. **Ant transition rules.** In the path planning, if ant k is located in point a at the moment t, the next transfer target point is b. During the ant's movement, the next transfer point is determined based on pheromone on each path and visibility. So P_{ab}^k indicates transfer probability of ant k at time t from the point a to point b.

$$P_{ab}^k = \begin{cases} \frac{[\tau_{ab}(t)]^\alpha [\eta_{ab}(t)]^\beta}{\sum [\tau_{ab}(t)]^\alpha [\eta_{ab}(t)]^\beta} & b \in node \\ 0 & other \end{cases} \tag{14}$$

α and β are weight coefficient of pheromone and visibility. *Node* and *other* denote feasible and infeasible transfer point respectively. $\tau_{ab}(t)$ is the pheromone on path ab at time t. $\eta_{ab}(t)$ is heuristic function on the path ab, that is the visibility. d_{ab} is distance between point a and point b.

$$\eta_{ab} = \frac{1}{d_{ab}} \tag{15}$$

The probability of the current point to the next target point is calculated by Formula (14). According to roulette selection method, the next point will be selected randomly, the larger probability the greater selected opportunity. Then this point is as the current point, and continuous cycle of calculation, until all the points of path are calculated.

4.3. **Pheromone update.** In the max-min ant algorithm, the global update method is used. When one search is completed for all the ants, after it, the pheromone produced by global optimal ant on the path will be updated. This ensures the searched optimal path converging to global optimum. However, it may lose the global optimal solution which contained in iterative solution. Therefore, the algorithm uses an update strategy combined iterative optimal solution and the current global optimum in this paper. The global optimum pheromone updates after optimal iterations update a number of times. This enhanced feedback capability of pheromone on each iteration search optimal path and increase density of pheromone. When pheromone updates in the global optimal path, with increasing number of iterations, dynamically adjust and update the weight factor of the best iteration ant pheromone. It is better for eventually converging to the global optimal solution.

If ρ: volatile factor of pheromone; L_I length of iterative optimal path; L_g length of global optimal path; m_k weight factor of iteration best ant; n_k weights factor of global optimum ant; N_c number of iterations.

$$\tau(t+1) = (1-\rho)\tau(t) + \Delta\tau(t) \tag{16}$$

$$\Delta\tau(t) = \left(\frac{m_k}{L_I} + \frac{n_k}{L_g}\right) \times \frac{1}{N_c} \tag{17}$$

$$m_k + n_k = N_c \tag{18}$$

At the beginning, $m_k = N_c$, with increasing number of iterations, m_k minus 1 every time, thus the proportion of global optimum ants pheromone updating continues to be strengthened, so that the optimal solution is found quickly.

5. **Adaptive Dynamic Integration Phase.** According to the definition of Ant Colony Algorithm, ants tend to choose the path of larger pheromone. When many ants select a same path, the path will be repeatedly strengthened, and the ants quickly gathered, causing the path congestion and stagnation. The performance is the precocious of Ant Colony Algorithm and falls into local optimum. For this case, a convergence speed parameter δ is used in Ant Colony Algorithm. The convergence speed parameter δ represents the optimization ratio of algorithm in the form of ant's path length. Calculation methods are as follows:

$$\delta = \frac{\alpha\delta_s + \beta\delta_b}{\alpha\delta_{sp} + \beta\delta_{bp}} \tag{19}$$

$$\delta_s = \sum_{i=1}^{m} L_i^{n-1} - L_i^n \quad \delta_b = m\sum_{i=1}^{m} L_{best}^{n-1} - L_{best}^n \quad \delta_{sp} = \sum_{i=1}^{m} L_i^{n-1} \quad \delta_{bp} = m\sum_{i=1}^{m} L_{best}^{n-1} \tag{20}$$

L_i^n: path length of ant i in the n cycle, L_{best}^n: the optimal path length in the n cycle, δ_s: the ants total path length difference in two cycles, δ_b: m times of the optimal path length difference in two cycles, δ_{sp}: path length of all the ants in the previous round, δ_{bp}: m times of the optimal path length in the previous round.

Difference of path length may be positive or may be negative. Positive illustrates the path of evolution, and negative represents the path degradation. This may reflect evolution of the population and provide evidence for the speed of convergence.

At the same time, a constant of convergence speed threshold v is defined. When parameter is $\delta > \nu$, it transfers to Genetic Operators in each cycle.

Genetic Operator is to reduce the probability of the algorithm fallen into local optimum, and better combine with MMAS.

6. The Overall Flow of the AGAACO Algorithm.

1. Initialization of environmental information, the starting grid, the target grid, set the maximum rate of evolution of generation and maximum evolution
2. Randomly generated initial population
3. Calculate each individual fitness of the population.
4. Selection operation: Random league selection method
5. Crossover operation: According to the crossover probability to determine, and two individuals generate a random cross position.
6. Mutation operation based on mutation probability
7. Determine whether there is a path overlapped points, if so, perform the removal, until no overlap path points.
8. Determine whether Genetic Algorithms to the termination conditions. If not, turn to 3rd step.
9. The population generated by GA, and settings initial pheromone of Ant Colony Algorithm according to (12), (13). Including number of cycles N, the number of ant m, ant move taboo table initialization.
10. Place m ants on the starting point.
11. Move the m ants sequentially and select the next point according to probability Formula (14). Update taboo table of ant's movement until each ant has reached the end. At the same time record the path of each ant and determine the current optimal path.
12. Calculate convergence speed parameters δ of Ant Colony Algorithm according to (19); if $\delta > \nu$, turn to 13th step; if not, turn to 14th step.
13. The path found in this round was done the single-point crossover and mutation operating, and then determine the optimal path.
14. Update global pheromone.
15. If the end condition is reached, the algorithm terminates, output the path. If not, turn to 10th step and continue.

7. Simulation Experiment.

7.1. Experimental parameters.
In order to verify the feasibility and effectiveness, the algorithm is simulated by MATLAB. The parameters are as follows:

A. Genetic Algorithm parameters. Population size $S = 100$, Crossover probability $Pc = 0.8$, Mutation probability $Pm = 0.07$, Number of iterations of GA algorithm $G = 200$, fitness function $F = \frac{K}{\alpha D_l + \beta F_{ob}}$, $\alpha = 0.5$, $\beta = 0.5$.

B. Ant Colony Algorithm parameters. Inspired factor $\alpha = 1$ expectation factor $\beta = 5$, Pheromone evaporation coefficient $\rho = 0.8$, Pheromone constant $Q = 200$, Number of ants $m = 50$, Initial pheromone of Max-Min Ant System (MMAS) $\tau_M = \tau_{\min} + \frac{1}{2}(\tau_{\max} - \tau_{\min})$, $\tau_M = 100$, the pheromone from Genetic Algorithm $\tau_G = 50$.

C. Adaptive stage parameters. Convergence speed parameter $\delta = \frac{\alpha\delta_s + \beta\delta_b}{\alpha\delta_{sp} + \beta\delta_{bp}}$, $\alpha = 0.7\,\beta = 0.3$, convergence speed threshold $\nu = 0.02$.

7.2. Traveling Salesman Problem (TSP).
Traveling Salesman Problem is one of the famous problems in mathematics [16-18]. Supposed a travel business to visit n cities, he must choose which path to go. The restriction is to visit each city only once and at last return to the original departure city. Selected the destination path is the minimum among all paths. In this paper, the 20 city of TSP is simulated. The 20 points are as follows:

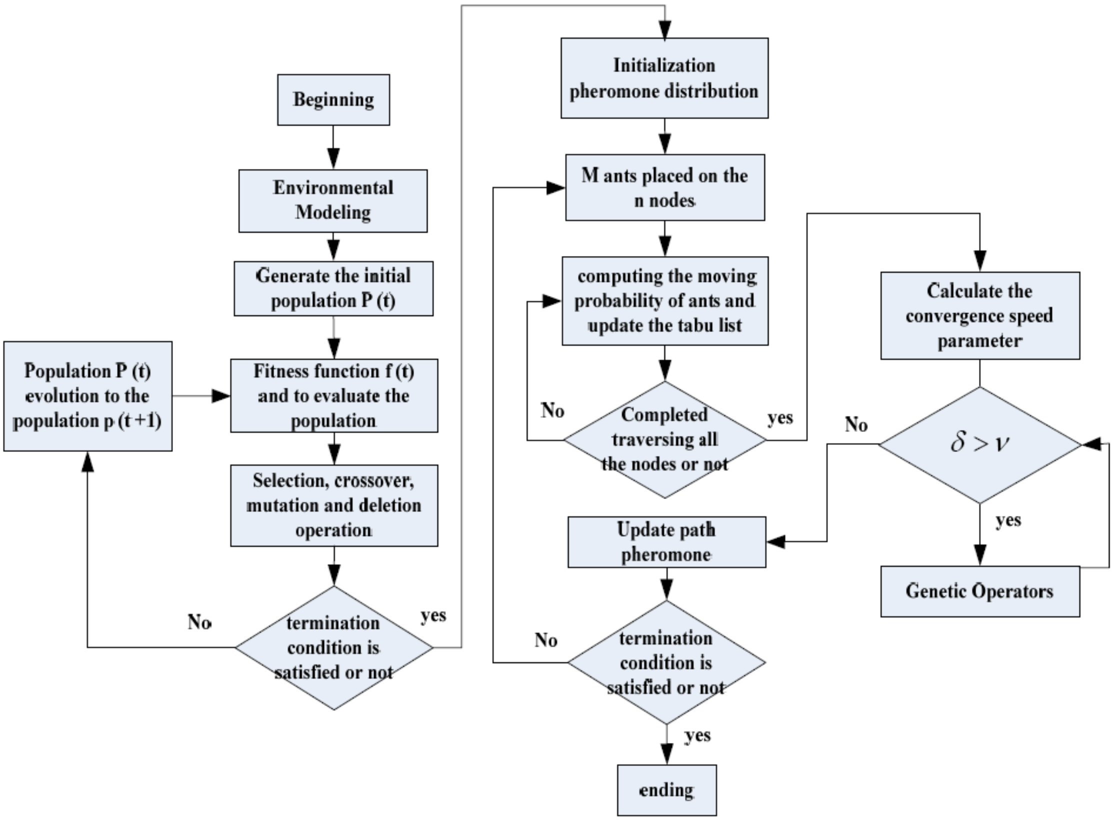


Figure 6. Process of adaptive Genetic Ant Colony Algorithm

[2.4, 1.83; 2.01, 2.183; 2.183, 1.011; 1.833, 1.533; 2.05, 0.9833; 2.25, 2.13; 2.333, 1.416; 1.917, 1.533; 2.25, 2.167; 1.583, 1.000; 0.85, 2.733; 1.7833, 3.2833; 1.9833, 4.3833; 1.200, 1.6167; 1.5167, 1.90; 1.9833, 2.5833; 1.6833, 3.2833; 1.7833, 3.1167; 1.1833, 3.05; 2.90, 2.7833].

In this part, we present the TSP simulation to evaluate the efficiency of AGAACO algorithm. The results compare between AGAACO, ACO, GA and ACO-GA [19]. The ACO-GA made many improvements in the former ant algorithm based on genetic gene to expand search space solutions, to improve its optimization ability and speed. The comparative simulations will help demonstrate the added adaptive of combination ACO and GA. In addition, other metrics were assessed: (1) the path length: it is the shortest path searched by algorithm, (2) the execution time: it is the time to find the best path, (3) the number of iterations: it is the iterations to reach the optimal path by an algorithm. The simulation results are shown as Figures 7-10.

In 20city TSP search, comparison of four algorithms, the shortest path and search time are shown in Table 1.

For the TSP, we get the average of 10 runs results to ensure correct statistical analysis of each algorithm. From the results, due to combining the advantages of Genetic Algorithm

Table 1. Test results of TSP

	The shortest path length	Iterations	Search time
AGAACO	22.608	112	14.2
ACO	26.425	97	23.1
GA	24.412	83	22.5
ACO-GA	24.247	63	20.3

The AGAACO

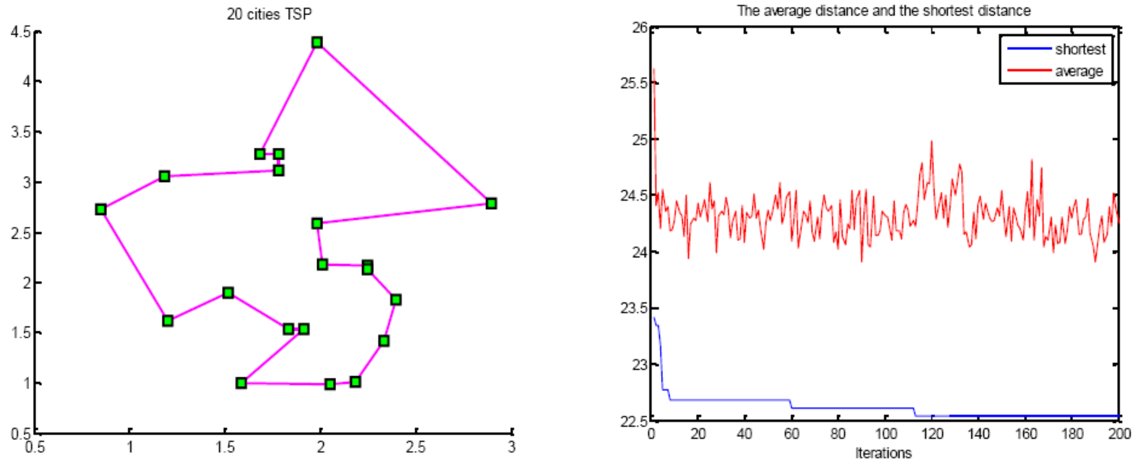


Figure 7. TSP of AGAACO

The ACO

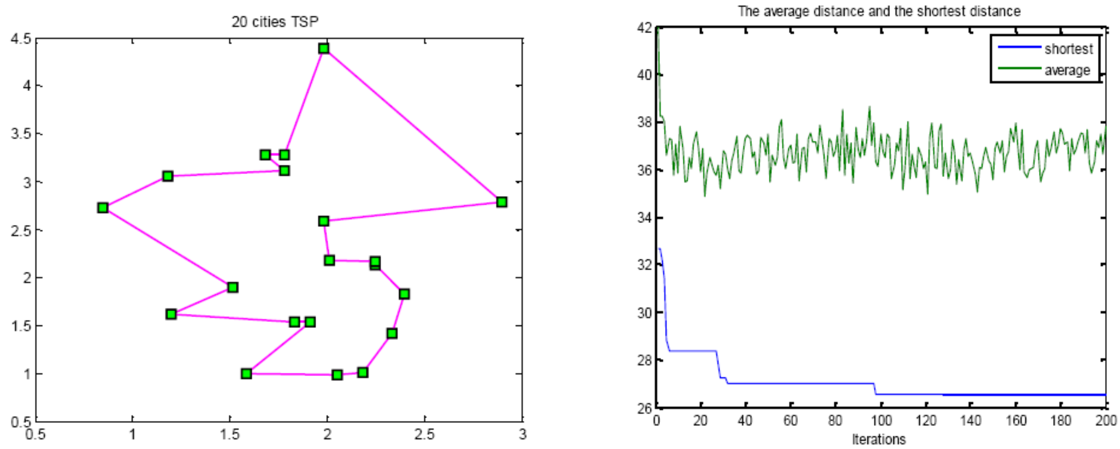


Figure 8. TSP of ACO

The GA

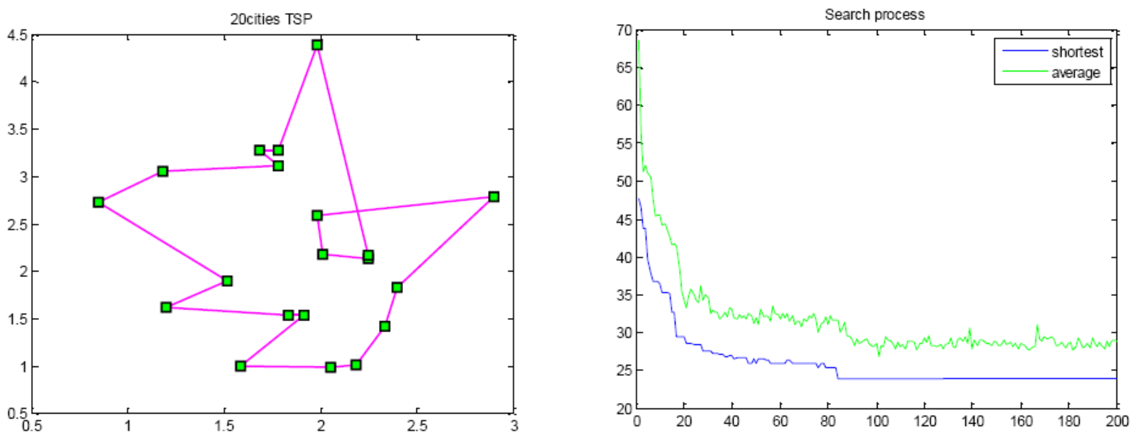


Figure 9. TSP of GA

The ACO-GA

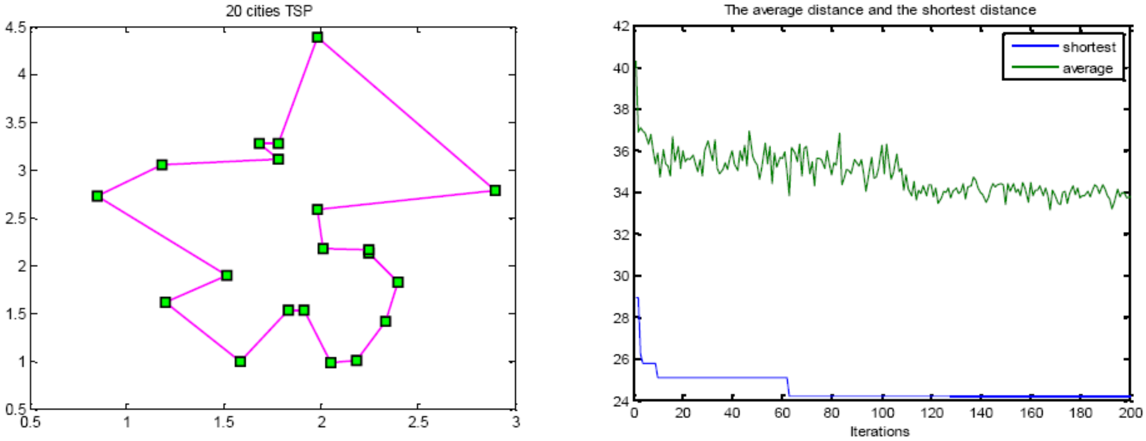


Figure 10. TSP of ACO-GA

and Ant Colony Algorithm, ACO-GA and AGAACO improve apparently in searching the shortest path and the execution time. The convergence of ACO-GA is very fast, but the shortest path length is less than AGAACO. Because in the later stage of AGAACO algorithm, the convergence rate parameter is introduction to prevent convergence too fast and fall into local optimum, it sacrifices the advantages of iterations, but the search speed is better than others. Because the initial pheromone of ACO is relatively small, convergence time and iterations are larger than Genetic Algorithms. The result of GA is a little poorer than ACO-GA, as feedback information of GA is not enough used and redundant iterations. This example demonstrates the superiority of AGAACO.

7.3. **The shortest path.** In path planning for environmental modeling, the common is grid method. This method is simple and effective, strong ability to adapt the obstacles. It can greatly reduce the complexity of modeling, easily for computer to storage and handling, and prevent the loss of some feasible path, so it is currently widely used for environmental modeling. According to the characteristics of robot environment, a 20×20 grid is established. Place the robot's starting grid in the upper left corner, the end grid in the lower right corner, black grid represents an obstacle and white grid represents a feasible path. In order to verify the feasibility and effectiveness of the algorithm, intelligent path planning simulation experiments demonstrated in different obstacles environment.

In this part, we present the obstacle simulation to evaluate the efficiency of AGAACO algorithm. The results compare between AGAACO, ACO, GA and GA-ACO [12]. The GA-ACO [12] is a modified Genetic Algorithm provided for global path planning, and provided the position information negative feedback by the ant colony optimization. Metrics were assessed except the path length and the execution time, but also capability of obstacle avoidance. We simulate the path planning in a simple and a complex environment.

In a simple environment, path planning is shown in Figure 11.

In a relatively complex environment, path planning is shown in Figure 12.

We get the average of 10 runs results to ensure correct statistical analysis of each algorithm. From grid method, comparative data can be found that AGAACO pathfinding algorithm is significantly better than the GA-ACO, ACO, GA in the optimal path, average path length and running time, especially in a complex environment. In addition, from Figure 11, Figure 12, we see that AGAACO is better in the accuracy and stability of pathfinding. In the simple environment, difference of optimal path is not obvious, but in relatively complex environment, classical both ACO and GA algorithm

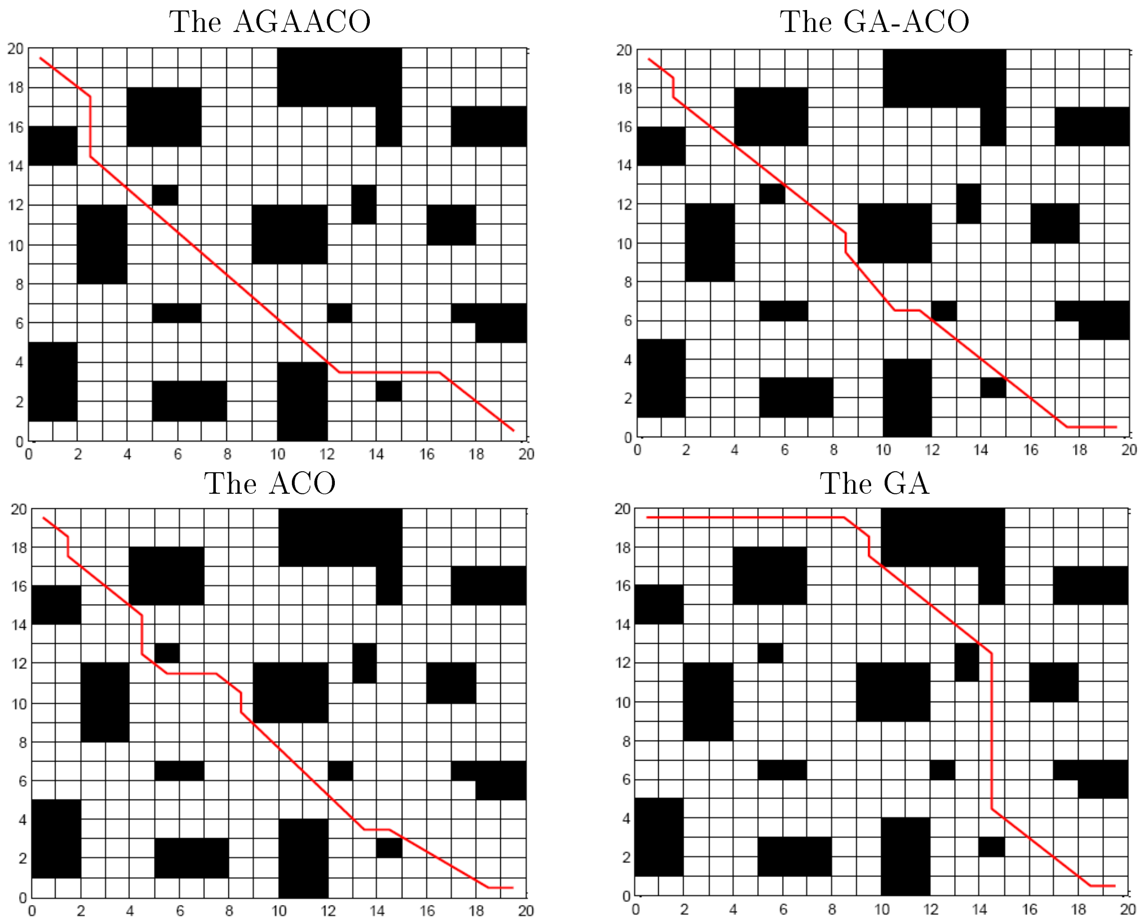


FIGURE 11. AGAACO, GA-ACO, ACO, GA in simple environment

TABLE 2. Test result of the shortest path

		Average path length	The minimum path length	Running time
Simple environments	AGAACO	39.32	27.616	13.59
	GA-ACO	44.67	29.038	14.68
	ACO	50.45	29.524	16.21
	GA	45.26	32.147	17.35
Complex environments	AGAACO	43.78	28.627	14.67
	GA-ACO	45.37	29.796	16.78
	ACO	49.32	32.152	18.54
	GA	49.87	36.036	19.86

fail to search the shortest path. AGAACO is faster and better than GA-ACO, because the convergence speed parameter δ adjusts the part of ACO and GA iterations adaptively. According to this example, we found that AGAACO is suitable for path planning.

8. **Experiment.** In this part, we take an experiment to demonstrate the performance of AGAACO algorithm in real environment. The implementation was taken on a land-yacht. It was driven by wind and used for environment detection. The path planning system includes single-chip microcomputer STM32, controller S3C6410, RF chip nRF24L01 and Gyroscope xsens MTi. This system was developed in Linux. System structure is shown as Figure 13.

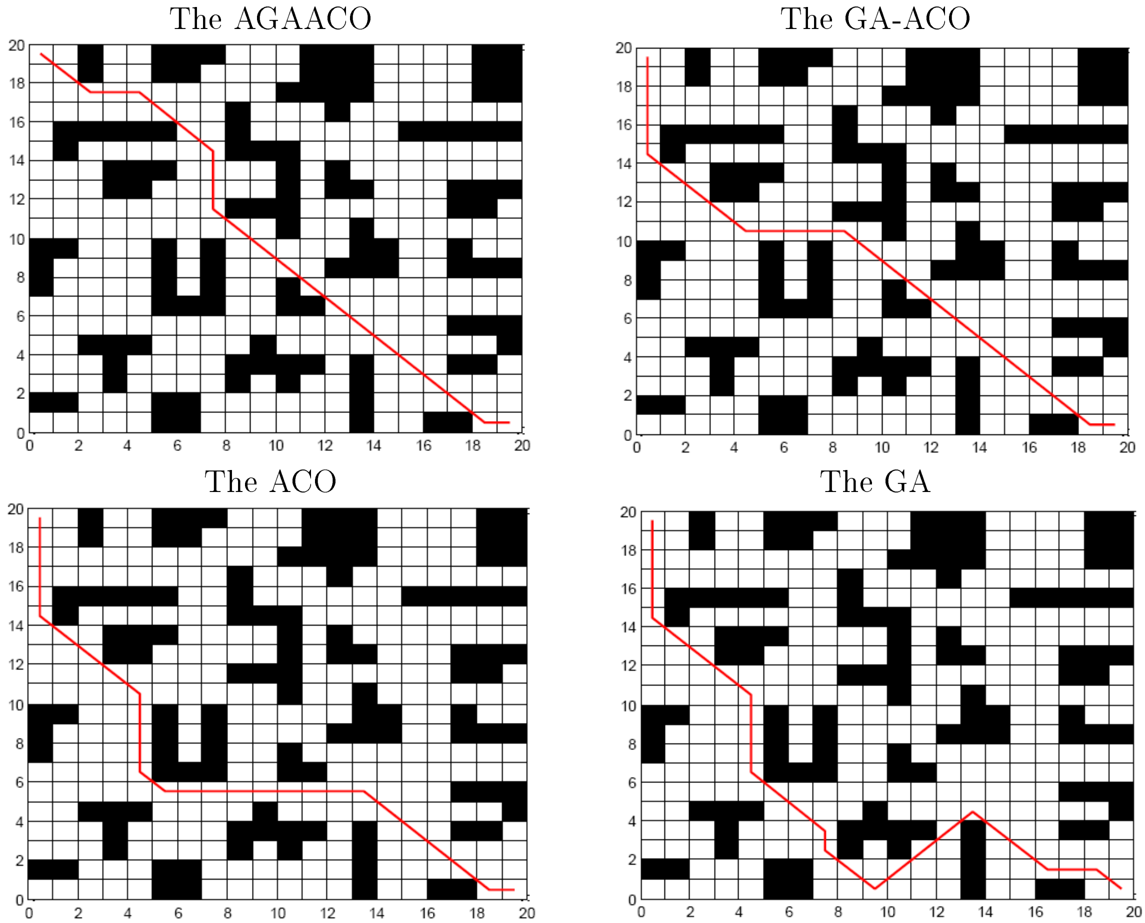


FIGURE 12. AGAACO, GA-ACO, ACO, GA in complex environment

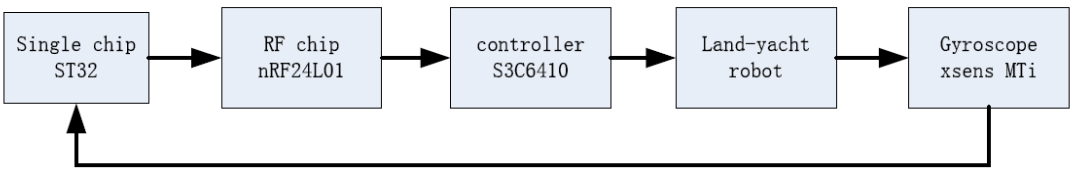
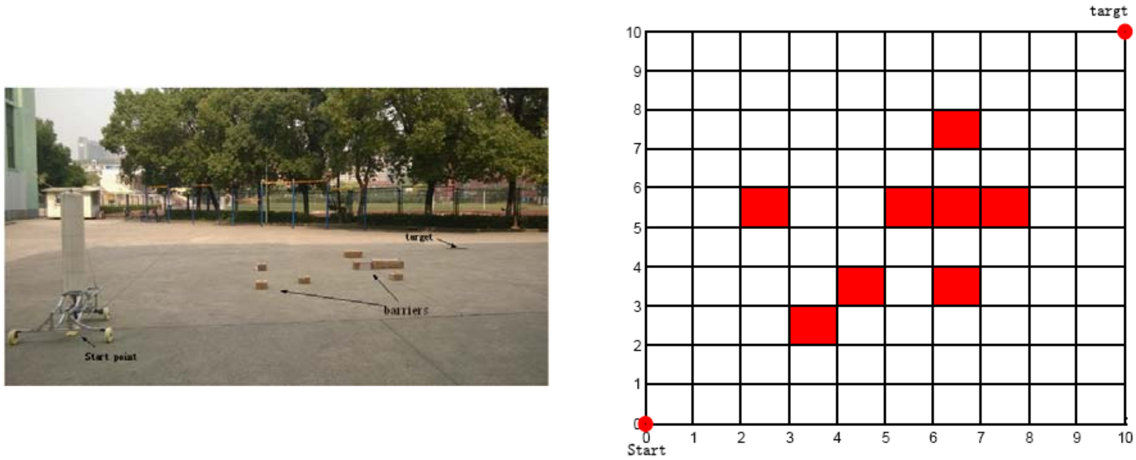


FIGURE 13. Path planning system of land-yacht

We conduct the experiment in outdoor environment of 10*10m. The obstacles are static. The ways of land-yacht are from start point to target point without hitting any obstacle and get the shortest path. At first, the known map is storage in land-yacht as a square matrix denotes input to path planning algorithm. The environment is shown as Figure 14(a). Considering radius of land-yacht, security zone of barriers and avoid collision, etc., environment is divided into 10*10 grids and red areas represent obstacles. So the land-yacht is regarded as a particle during movement. The result is shown as Figure 14(b).

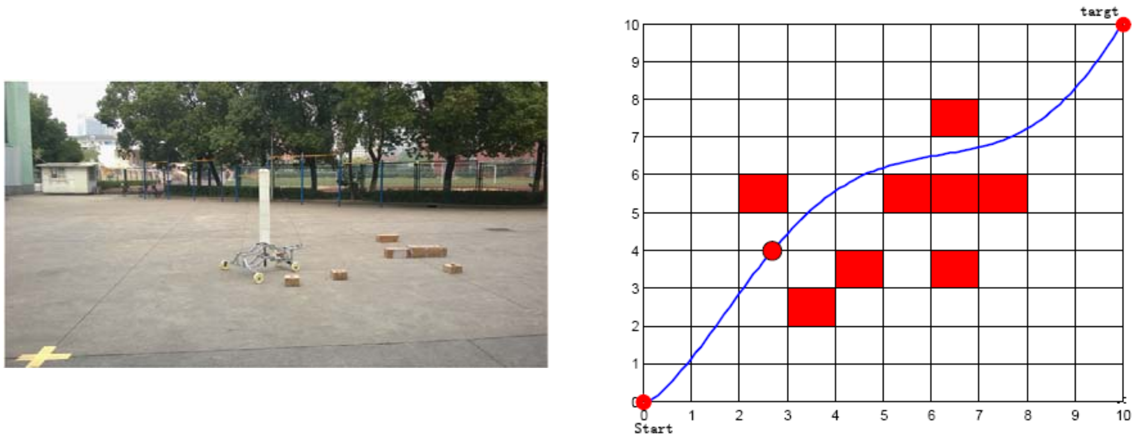
The path planning of land-yacht according to environment and the real-time running gestures are shown as Figure 15.

In this experiment, we conducted five times. Each test land-yacht can reach the target without collision. Then we analyzed the running trajectory tracking error and target point tracking error.

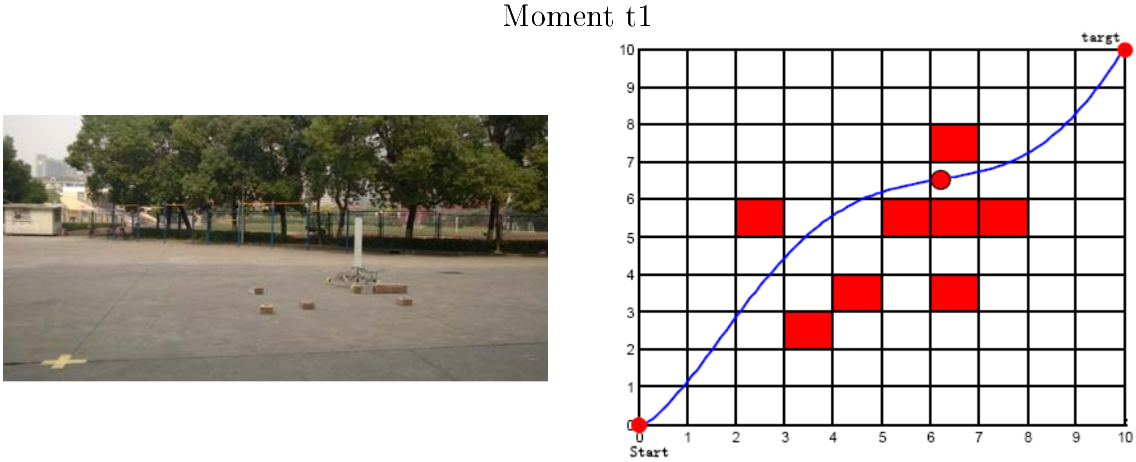


(a) (b)

FIGURE 14. Experiment environment



Moment t1



Moment t2

FIGURE 15. The experimental scene

From the above experimental results, during path planning, we found that the maximum tracking error generally appears at the target point. After analyzing the process of experiment, we inferred that it is error accumulation. It mainly comes from three aspects: firstly, the mechanical and electrical errors of land-yacht; secondly, the measurement error, the measurement device is Gyroscope and it jitters with the running land-yacht and

TABLE 3. Trajectory tracking error result

Tracking Error		1	2	3	4	5
X direction (cm)	Target point	10.3	8.6	4.2	5.7	5.5
	The max	10.3	8.6	4.5	6.1	5.5
Y direction (cm)	Target point	7.5	6.8	3.6	4.1	3.4
	The max	7.5	7.0	3.6	4.5	3.8
Angle (degree)	Target point	6.5	4.3	2.9	5.3	4.5
	The max	6.5	4.5	3.0	5.3	4.5

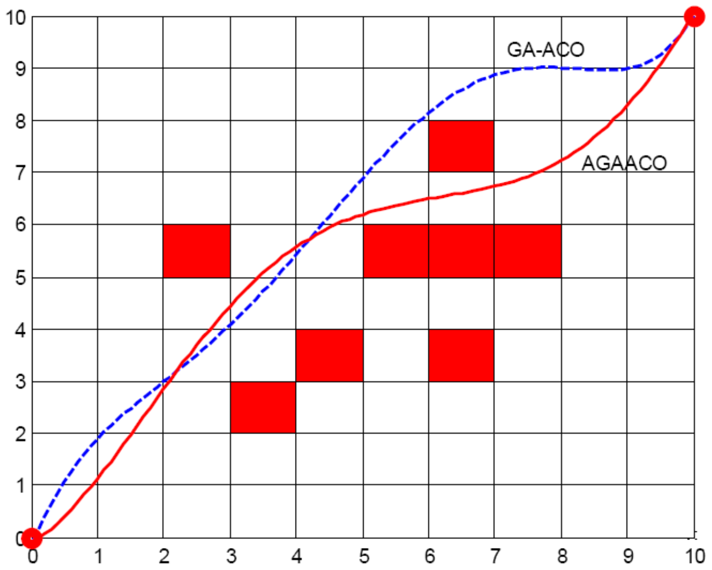


FIGURE 16. Paths planning of AGAACO and GA-ACO

TABLE 4. Result of experiment

Algorithm	Average length	Shortest length	Average time	Minimum time
AGAACO	17.154	16.324	30.18	27.62
GA-ACO	17.893	17.035	30.45	28.96

causes inaccurate measurements; thirdly, the motion errors, kinematic model of land-yacht is modeled in the ideal conditions, for example the pure rolling wheels, but not exist in the body. However, in the land-yacht experiment system, the max error is 10.3cm, the result also satisfied our requirement. The error shows that it is still a higher tracking accuracy for a mobile robot control system. So we say that the AGAACO algorithm was correctly verified by application in this experiment.

In addition, we compare AGAACO algorithm with GA-ACO [12] in this environment. We also conducted five times and take the best one. The result is shown in Figure 16. The red solid line represents AGAACO and the blue dashed line is GA-ACO.

In this section, we assessed the shortest path and the execution time. The result is shown as Table 4.

From the result of experiment, we found that the AGAACO is better than GA-ACO in average path length and average time. So the AGAACO algorithm application in this experiment system is effective, and it improves the ability of mobile robot path planning.

9. **Conclusion.** This paper describes the characteristics of Genetic Algorithm and Ant Colony Algorithm, and the Adaptive Genetic Ant Colony Algorithm (AGAACO) is proposed based on improving integration of Genetic Algorithm and Ant Colony Algorithm. The AGAACO overcomes the respectively defects of Genetic Algorithm and Ant Colony Algorithm. The target is applied in path planning problem, which denotes a basic problem in mobile robot.

Then, the AGAACO is verified by some examples of simulation, the TSP problem, the grid obstacle. According to the examples, it shows significantly better than some hybrid ACO-GA approach, single Genetic Algorithm or Ant Colony Algorithm, not only in time and search performance, but also the accuracy and stability, especially in complex environments with deep traps; it can quickly find the optimal path, improving the solving speed and efficiency of the algorithm. At last, this algorithm is conducted in a land-yacht robot. The experimental result showed that AGAACO can succeed in finding the optimal solution.

However, the algorithm belongs to bionic optimization algorithm, with the drawback of slow response time, so it is not suitable for dynamic path planning problem with a higher response speed. In order to solve it, the algorithm needs to combine with other methods. This is the direction of further study.

Acknowledgment. This work is partially supported by the Key Projects of the National Natural Science Foundation of China (No. 61233010), the National Natural Science Foundation of China (No. 61375093), the Innovation Fund of Shanghai University (No. SDCX201316) and Shanghai Natural Science Foundation (13ZR1454200). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evolutionary Computation*, vol.1, no.1, pp.53-66, 1997.
[2] L. V. Snyder and M. S. Daskin, A random-key genetic algorithm for the generalized traveling salesman problem, *Operations Research*, vol.173, no.1, pp.38-53, 2006.
[3] W. Darrell, H. Doug and H. Adele, A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover, *Lecture Notes in Computer Science*, vol.6238, pp.566-575, 2011.
[4] Y. Gajpal and P. L. Abad, Multi-ant colony system (MACS) for a vehicle routing problem with backhauls, *European Journal of Operational Research*, vol.196, no.1, pp.102-117, 2009.
[5] C. Y. Lee, Z. J. Lee, W. S. Lin and K. C. Ying, An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem, *Applied Intelligence*, vol.32, no.1, pp.88-95, 2010.
[6] G. M. A. Porta, M. Oscar et al., Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Applied Soft Computing*, vol.9, no.3, pp.1102-1110, 2009.
[7] S.-M. Chen and C.-Y. Chien, Parallelized genetic ant colony systems for solving the traveling salesman problem, *Expert Systems with Applications*, vol.38, no.4, pp.3873-3883, 2011.
[8] L. Lin, H. J. Wang and Q. S. Wu, Improved genetic algorithms based path planning of mobile robot under dynamic unknown environment, *Proc. of the 2006 IEEE International Conference on Mechatronics and Automation*, pp.33-45, 2010.
[9] J. Wei and P. Wang, Optimization of fuzzy rule based on adaptive genetic algorithm and ant colony algorithm, *International Conference on Computational and Information Sciences*, pp.21-26, 2010.
[10] D. X. Gong and X. G. Ruan, A hybrid approach of GA and ACO for TSP, *Proc. of the 5th World Congress on Intelligent Control and Automation*, pp.2068-2072, 2004.
[11] J. Pan, X. Wang and Y. Cheng, Improved ant colony algorithm for mobile robot path planning, *Journal of China University of Mining & Technology*, vol.41, no.1, pp.108-113, 2012.
[12] L. Zhu and J. Fan, Global path planning and local obstacle avoidance of searching robot in mine disasters based on grid method, *Journal of Central South University (Science and Technology)*, vol.42, no.11, pp.3421-3428, 2011.

[13] F. Jin and H. Gao, Research on path planning of robot using adaptive ant colony system, *Journal of Harbin Institute of Technology*, vol.42, no.7, pp.1014-1018, 2010.
[14] P. Wang, Z. Feng and X. Huang, An improved ant algorithm for mobile robot path planning, *Robot*, vol.30, no.6, pp.554-560, 2008.
[15] C. W. Ahm and R. S. Ramakrishna, A genetic algorithm for shortest path routing problem and the sizing of population, *IEEE Trans. Evolutionary Computation*, vol.6, no.6, pp.566-579, 2002.
[16] J. Cheng and A. Lin, Review of genetic algorithms for traveling salesman problem, *Journal of Kunming University of Science and Technology (Science and Technology)*, vol.28, no.14, pp.9-13, 2003.
[17] H. Shen, G. Zhang and Z. Li, New algorithm for solving TSP, *Computer Engineering and Applications*, vol.44, no.6, pp.65-67, 2008.
[18] P. Cao, P. Chen and S. Liu, Application of improved particle swarm optimization in TSP, *Computer Engineering*, vol.34, no.11, pp.217-218, 2008.
[19] J. Jiang and F. Pan, Hybrid of ant colony algorithm and genetic algorithm and its application in TSP, *Journal of Jiangnan University (Natural Science Edition)*, vol.11, no.3, pp.253-256, 2012.