Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization

Songcan Zhang¹,², Jiexin Pu¹, Yanna Si¹ and Lifan Sun¹,³

Abstract
Path planning of mobile robots in complex environments is the most challenging research. A hybrid approach combining the enhanced ant colony system with the local optimization algorithm based on path geometric features, called EACSPGO, has been presented in this study for mobile robot path planning. Firstly, the simplified model of pheromone diffusion, the pheromone initialization strategy of unequal allocation, and the adaptive pheromone update mechanism have been simultaneously introduced to enhance the classical ant colony algorithm, thus providing a significant improvement in the computation efficiency and the quality of the solutions. A local optimization method based on path geometric features has been designed to further optimize the initial path and achieve a good convergence rate. Finally, the performance and advantages of the proposed approach have been verified by a series of tests in the mobile robot path planning. The simulation results demonstrate that the presented EACSPGO approach provides better solutions, adaptability, stability, and faster convergence rate compared to the other tested optimization algorithms.

Keywords
Ant colony optimization, path planning, pheromone diffusion mechanism, pheromone updating strategy, local geometry optimization algorithm

Introduction
With the improvements in automation technology and artificial intelligence, robotics is becoming a necessary part of various areas, such as unmanned aerial vehicles,¹ industrial automation,² surveillance operations,³ agricultural automation,⁴ logistics,⁵ and other applications. As an important branch of robotics, mobile robots are also being widely used. A considerable number of research issues in mobile robotics, such as trajectory tracking, path planning, and simultaneous localization and mapping, have been studied to date.⁶ The most essential research hotspots are the path planning problem among them. Generally speaking, the main purpose of path planning is to figure out a feasible and optimal path so that the robot can reach the target point from the starting position and avoid the scattered obstacles in the environment. Many approaches have been suggested and applied to path planning. The existing methods are mainly classified into classical approaches and heuristic approaches.⁸ The former, such as artificial potential fields, cell decomposition,
mathematical programming, and roadmap approach, dominated in the early days of research. Although these traditional methods have achieved good results, they have some disadvantages, such as expensive computation in high dimensions, lack of adaptation, and the local minima, which make them ineffective in a practical situation. On the other hand, a wide variety of heuristic methodologies, such as neural networks, artificial bee colony, cuckoo optimization algorithm, slime mold algorithm, plant growth mechanism, ant colony algorithm (ACA), and so on, have been introduced for solving the aforementioned disadvantages.

Among the heuristic-based methods, the ant colony optimization (ACO) algorithm, originally designed for solving complex combinatorial optimization problems, has been widely employed for path planning. However, there are still some deficiencies, such as premature stagnation and low convergence rates. Furthermore, these drawbacks become more apparent with the increase of the problem scale. To remedy these weaknesses, a few improvements based on the ant system (AS) were designed to achieve better performances. The ant colony system (ACS) is a well-known improvement that differs from it in three main components: (a) the state transition rule is modified and regulated by the parameter $q_0$ to govern the ratio of exploration and exploitation modes; (b) the global pheromone update rule is only implemented to the global optimal solution; and (c) a local pheromone updating rule is added into AS to allow for a variety of solutions. The MAX-MIN ant system (MMAS) is another AS variant, which employs global updating and pheromone trail limits to address the problem of premature convergence and achieve better exploration.

Furthermore, many enhancement mechanisms, such as state transition rules, pheromone initialization, and update strategy, parameter controlling, and heuristic information, have been introduced to further enhance the ACO algorithm’s optimization capability. Wang et al. suggested an evolving ACS called genetic algorithm (GA)-ACO to solve the trajectory planning problem, in which the initial pheromone values were assigned based on some initial suboptimal and/or optimal solutions from the GA to avoid searching blindness of ACO. Jiao et al. proposed an adaptive polymorphic ACA to escape the local optimum solution by the introduction of the adaptive information updating rule and the adaptive state transition mechanism. Deng et al. introduced a novel method of hybrid optimization with a combination of GA, particle swarm optimization (PSO), and ACO algorithms. The suboptimal solutions (rough searching) obtained by GA and PSO were used for allocating the initial pheromone values of the ACO algorithm. This scheme can evade searching blindness at the early stages and offer a better rate of convergence. However, there are also certain drawbacks, such as the structure is more complex, the effective search scope is smaller, and the local optimum has not been thoroughly solved. To avoid premature stagnant, the pheromone update rules are modified by Luo et al. according to the operating mechanism of MMAS and elite ACA. Mavrovouniotis and Yang proposed a self-adapting pheromone evaporation rate for ACO algorithm according to the number of generations without improving the quality of the solution. Xiong et al. proposed the hybrid Voronoi-based ACO technique that combines ACO and the Voronoi-based approach for solving the adaptive ocean sampling problem.

Additionally, another research trend is the hybridization of the ACO algorithm with other algorithms, aimed at enhancing its global search ability and optimization efficiency. A hybrid particle swarm ACO (PS-ACO) algorithm was designed to modify the pheromone updating mechanisms of the ACO algorithm using the local and global search mechanisms in the PSO algorithm. Zhang et al. proposed a dynamic multirole adaptive collaborative ACO (MRCACO), which combines the adaptive multirole collaborative technology with the heterogeneous multicolonies. Results from the simulation and their practical application demonstrated the effectiveness and superiority of the presented approach. Tuani et al. described an adaptive heterogeneous ACO method (HHACO), where the ACO metaparameters were tuned by an evolutionary algorithm. The simulation results from this work suggested that the proposed HHACO approach can provide a better speed of convergence and achieve better solutions by exploiting the neighboring areas.

Although the above works help the ACS to achieve better performance, some inherent deficiencies, such as the problem of a local optimum, low search efficiency, and insufficient cooperation between the ants, have not been effectively addressed. Therefore, some aspects of the ACO algorithm could be further enhanced by considering the following points: (i) The pheromone initialization methods of uneven distribution may avoid blind searches that occurred in the early stages of the optimization procedure and improve the efficiency of search but may limit the scope of the search and raise the risk of premature convergence. (ii) Although the changes to the state transition rules and the heuristic information may decrease the possibility of trapping into local optima, they require a longer time to achieve the optimal solutions. (iii) In the basic ACO algorithms, the pheromones are distributed on the nodes or edges traveled by ants and the pheromones are effective only when the following ants travel the same position. This leads to some deficiencies of local optimum and insufficient cooperation between the ants. In fact, the pheromones are not only be released along the path visited by ants but also spread to adjacent areas of the traveled path. (iv) When the ACO algorithm is employed to discover the optimal path for mobile robots, some circular, cross, or zigzag path segments could be often found in the candidate path at the beginning of the optimization process. These undesired path segments not only deteriorate the optimization ability
of the algorithm but also reduce the rate of convergence. However, very limited research has been conducted on this aspect.

In this article, we proposed a hybrid approach by combining the enhanced ant colony system (EACS) with local optimization algorithm based on path geometric features (LOAGF) for addressing mobile robot path planning. The proposed approach has three different aspects compared with our previous works\(^\text{32}\): (a) Pheromone initialization mechanism. To further improve exploration ability while saving computation time, we initialized the value of pheromone according to the relative location of the intermediate node to the starting grid-goal grid line, which can be directly attained by heuristic information. However, the initial value of pheromone in previous works is assigned based on the optimal path found by A* algorithm. This pheromone initialization mode not only needs to spend a certain amount of computation time in advance to run A* algorithm but also only a small portion of the search space is considered as promising areas, which may limit the ACS algorithm to explore new parts in the search space. (b) Pheromone updating strategy. In previous works, not only the path quality but also the information entropy of population needs to be evaluated for performing pheromone update strategy. However, after each iteration, we can find global optimal path, iterative optimal path, and worst path by evaluating all the feasible paths, so we designed a new pheromone update strategy based on this information, which can further improve the efficiency of the algorithm. (c) In this article, we also designed a LOAGF, which can further optimize the initial path and achieve a good convergence rate by removing the undesirable path segments. Particularly, it should be noted that the pheromone on the refined path and the initial optimal path is simultaneously updated to speed up convergence rate and increase searching efficiency.

The problem of mobile robot path planning was selected for verifying the superiority of the presented algorithm. The main contributions of this work are outlined as follows:

1. The pheromone diffusion model is designed to reduce the possibility of premature convergence and promote the collaboration between ants.
2. A nonuniform pheromone initialization strategy is introduced into the ACS algorithm to avoid early blindness and improve the convergence rate.
3. A significant feature of the proposed dynamically weighted updating strategy is that the iterative optimal ant and the global optimal ant are simultaneously considered for pheromone updating. Besides, the pheromone increment is dynamically adjusted to achieve a large reinforcement of solution elements belonging to the best path. Furthermore, the additional pheromone evaporation of the worst path is applied to decrease the misguiding effect of the worst path.
4. The local optimization approach based on path geometric features is proposed to further optimize the paths generated by the EACS algorithm.

In the rest of this article, the second section briefly introduces the preliminary of the ACS algorithm and the environment model. The third section expatiates the design details of the EACS, including the nonuniform pheromone initialization mechanism, the pheromone diffusion strategy, and the dynamically weighted pheromone updating scheme. The method of local optimization based on path geometric features is described in the fourth section. The fifth section mainly discusses the realization and application of the presented approach for path planning and the corresponding flowchart is also given. The sixth section reveals the corresponding simulation results and discussions of the developed approach for path planning in different scenarios. Finally, the seventh section concludes this work and gives ideas for future studies.

**Preliminaries**

**Ant colony optimization algorithm**

The ACO algorithm inspired by the foraging processes of actual ants has been widely used for tackling complex discrete and continuous optimization problems. The cooperative behavior of ants depends on the following behavior of ants and the artificial pheromone left by ants. Once the food source is discovered, the ants will drop a certain amount of pheromone trails on the edges they visited. If the tour is good, then the value of pheromone on that path is large. At the same time, if the ants perceive the pheromone trails previously created by other ants, they will follow these edges with a high probability and reinforce these edges with ant-laid pheromone. The pheromone evaporation mechanism is employed to avoid local optimum and explore the new regions of the solution space. Generally speaking, the optimization procedure of the ACO algorithm mainly consists of three parts: initialization, building candidate solutions, and updating pheromones. The phases of building solutions and updating pheromones are iterated to attain better solutions until the defined stopping conditions are met. Once terminated, the algorithm outputs the best-so-far solution.

The solution construction process in the ACO algorithm depends on the state transition probability rule determined by the amount of pheromone on the edges and the desirability information. The probability transition rule and the updating rule of pheromone in the ACS are explicated as follows:

(1) **Probability transition rule**

The selection of the next node is related to the amount of pheromones on edges and the distance between nodes.
When ant $k$ is at node $i$, the probability transition rule is used to pick the next node $j$ and defined as follows:

$$j = \begin{cases} \arg \max_{j \in \text{allowed}_k} \left[ \eta_j(t) \right]^\alpha \left[ \tau_{ij}(t) \right]^\beta & \text{if } q \leq q_0 \\ J & \text{otherwise} \end{cases} \quad (1)$$

$$p_{ij} = \begin{cases} \left[ \frac{\tau_{ij}(t)}{\sum_{s \in \text{allowed}_k} \tau_{is}(t)} \right]^\alpha \left[ \eta_j(t) \right]^\beta & \text{if } s \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $q_0 \in [0, 1]$ is a predetermined design parameter allowing to govern the ratio of the exploration and exploitation modes, $q$ is a random value between 0 and 1, $\eta_j(t)$ indicates the pheromone density on the edge between two nodes in the $t_{th}$ generation, and $\eta_j$ represents a heuristic information on the edge between two nodes and computed by $1/d_{ij}$, where $d_{ij}$ shows the cost of edge between $i$ and $j$, allowed$_k$ denotes the set of nodes not explored by ant $k$ yet. $J$ is a random number generated by the probability transition rule given by equation (2). $\alpha$ and $\beta$ are the parameters regulating the relative weights between the pheromone amount of the heuristic information. The abovementioned process is repeated until the ant arrives at the destination or allowed$_k$ is empty.

(2) Pheromone update mechanism

(a) Local update rule

Local update rule is applied to decrease the pheromone concentration of the traveled paths during the construction of the solutions. Its aim is to encourage the following ants to choose other components and thus increases the diversity of the constructed solutions. The local pheromone updating formula is expressed as follows

$$\tau_j(t+1) = (1 - \zeta) \cdot \tau_j(t) + \zeta \cdot \tau_0 \quad (3)$$

where $\zeta$ represents a parameter of evaporation rate used to regulate the pheromone trails volatilization over time, $\tau_0 = (n + L_{mn})^{-1}$ represents the initial intensity of pheromones, $L_{mn}$ is the cost of solutions produced by the nearest neighbor method, and $n$ is the scale of the problem.

(b) Global update rule

Once all ants accomplish their solution construction, the global update of pheromone is executed to reinforce the pheromone intensity of the best path found so far and increase the likelihood of being selected in the following generations, which can speed the convergence of the algorithm. The global updating rule of pheromone is described as the following

$$\tau_j(t+1) = (1 - \rho) \cdot \tau_j(t) + \rho \cdot \Delta \tau^gb_j(t) \quad (4)$$

$$\Delta \tau^gb_j(t) = \begin{cases} \frac{Q}{L_{gb}} & \text{if } \text{edge}(i, j) \in \text{global best route} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\tau_j$ is the pheromone amount on node $j$, $\rho(0 < \rho < 1)$ denotes the factor of evaporation rate used to regulate how strong the pheromone levels laid by the optimal solution is, $Q$ is a predefined quantity of pheromone laid by ants, $\Delta \tau^gb_j$ is the additional amount of the pheromone on node $j$ and calculated using equation (5), and $L_{gb}$ represents the cost of the global best path and can be expressed by

$$L_{gb} = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (6)$$

where $n$ is the number of the nodes of the global best path, $x$ and $y$ indicate the coordinates of nodes.

**Rasterizing the environment model**

The establishment of a suitable environment model is the first step of path planning. We use the grid method to create a two-dimensional workspace for this work. The robot’s working space is divided into $N \times N$ equal square cells by a certain scale, where white grids are called feasible grids for a user to move, while black grids are represented as obstacles (infeasible grid). The white and black grids are represented by 0 and 1, respectively, to implement the algorithm conveniently. The grid model mentioned above is placed in a two-dimensional coordinate system. The serial number method has been used to assign a unique number for each cell as its address. In an $N \times N$ grid map, the grid labeled $S$ represents the starting position, and the grid labeled $G$ indicates the goal position.

Figure 1 shows an example of the grid environment of scale $10 \times 10$. There are eight passable directions for the robot to move when it is located on the center node $i$. Figure 2 shows the valid directions for robot in the grid environment. The movement of the mobile robot is restricted to the grid only, and therefore, the movements on a boundary node are limited. The nodes of potential directions constitute the adjacency node set of the grid $i$.

**Design of the enhanced ant colony system**

The proposed EACS is similar to the traditional ACS algorithm, but we propose to change three major aspects of the ACS approach, namely, the pheromone diffusion strategy, the pheromone initialization strategy, and the adaptive pheromone updating scheme, to tackle the problem of low computation efficiency and slow convergence.
Among the ACO algorithm and its variants, the pheromones are only laid down on the edges or nodes visited by ants, and the pheromones on the edges or nodes gradually volatilize over time in a certain ratio. This model can only affect the subsequent ants that passed through the same positions, which limits the search capability of the algorithm and the cooperation among ants.

In fact, the pheromones are not only released on the path traveled by ants but also spread to the areas around the traveled paths. In particular, a better solution is more likely to be obtained in the neighborhood around the current optimal path. The main purpose of the pheromone diffusion model proposed in this work is to rectify the defects of the original model and improve the exploration and cooperation capability of ants. In view of the constraints of the occupancy grid maps, a simplified pheromone diffusion model has been developed to increase the pheromone intensity on the surrounding regions of the optimal path.

From Figure 2, we can see that there are eight adjacent cells in the current cell, and thus, the value of pheromone in a specific cell is given as follows

$$f(g, g_0) = \begin{cases} 
1 & d(g, g_0) = 0 \\
0.5 & d(g, g_0) = 1 \\
0.33 & d(g, g_0) = \sqrt{2} 
\end{cases}$$ (7)

where $g$ is a feasible grid around the current grid $g_0$. $g_0$ represents the current grid of ant, $d(g, g_0)$ indicates the distance from $g_0$ to $g$ and is calculated as

$$d(g, g_0) = \sqrt{(x_g - x_{g_0})^2 + (y_g - y_{g_0})^2}$$ (8)

where $(x_g, y_g)$ show the $X$-$Y$ coordinates of the surrounding grid $g$ and $(x_{g_0}, y_{g_0})$ represent the $X$-$Y$ coordinates of the current grid $g_0$. 

In Figure 1, an example of grid-based environment is shown. And in Figure 2, the valid directions of motion are illustrated.
equation (8). Hence, the distance between nodes $S$ and $G$ is initialized using the same value, $t$. Generally, before running, we should assign the initialization intensity of pheromone. In the basic ACO, the initial pheromone intensity is set on the basis of the optimization because good solutions have not yet been obtained. Particularly, this phenomenon is more serious in early stage of the optimization because good solutions have not yet been obtained.

To overcome this deficiency, we designed the pheromone initialization strategy of unequal allocation, where the initial pheromone intensity is set on the basis of the relative location of the intermediate node to the starting grid-goal grid line. The grids at different locations are endowed with different initial pheromone values and can be defined as follows:

$$
\tau_0(j) = \begin{cases} 
\frac{d_{SG}}{d_{Sj} + d_{Gj}} \cdot a_0, & \text{if } j \in \text{feasible grid} \\
0, & \text{otherwise}
\end{cases}
$$

where $\tau_0(j)$ shows the initial intensity of node $j$, $d_{SG}$ represents the distance between $S$ and $G$. $d_{Sj}$ is the distance from the starting grid $S$ to grid $j$, $d_{Gj}$ indicates the distance between grid $j$ and grid $G$. $a_0$ is a constant, we assume that $a_0$ equals 1 in this work for simplicity. The distance between $i$ and $j$ can be computed using equation (8).

From equation (9), we can deduce that the smaller the value of $d_{Sj} + d_{Gj}$, the greater is the initial pheromone on node $j$ and the larger the value of $d_{Sj} + d_{Gj}$, the smaller is the initial pheromone intensity. Thus, the unequal allocation of initial pheromone is achieved based on the designed initialization strategy, which can decrease the blindness at the start of the evolution phase and effectively heighten the algorithm’s optimization capability. An example for the initial pheromone of unequal allocation with a different starting grid and goal grid is shown in Figure 4.

**Adaptive pheromone update rule**

As one of the well-known variants of AS, the ACS has adopted the global rule and the local rule for updating pheromone. During solution construction, ants in the ACS algorithm update the quantity of pheromone trails on the traversed edges using the local pheromone updating rule, which may reduce the probability of selecting the same nodes by subsequent ants. The global strategy for updating pheromone is applied for increasing the concentration of pheromone corresponding to the best path found so far. This ensures that the edges of the best solution have a higher priority to be visited in the subsequent generations. However, this could lead to a local optimum more easily. A dynamically weighted updating strategy is therefore being developed to effectively alleviate this shortcoming and can be described as follows

$$
\tau_j(t+1) = (1 - \rho) \cdot \tau_j(t) + \rho \cdot \left( \Delta \tau_j^{gb}(t) + \Delta \tau_j^{ib}(t) + \Delta \tau_j^{worst}(t) \right)
$$

$$
\Delta \tau_j^{gb}(t) = \begin{cases} 
\frac{Q}{L_{gb}}, & \text{if node } j \in T^{gb} \\
0, & \text{otherwise}
\end{cases}
$$

$$
\Delta \tau_j^{ib}(t) = \begin{cases} 
\frac{L_{gb}}{L_{ib}} \cdot \frac{Q}{L_{ib}}, & \text{if node } j \in T^{ib} \\
0, & \text{otherwise}
\end{cases}
$$

$$
\Delta \tau_j^{worst}(t) = \begin{cases} 
-\frac{Q}{L_{worst}}, & \text{if node } j \in T^{worst} \\
0, & \text{otherwise}
\end{cases}
$$

where $Q$ represents a positive constant, $\rho(0 < \rho < 1)$ indicates the volatilization factor of pheromone, $\Delta \tau_j^{gb}(t)$ denotes the increase in the trail level on node $j$ belonging to the global-optimal path and is expressed by equation (11), $\Delta \tau_j^{ib}(t)$ represents the increment of pheromone on node $j$ belonging to the iteration-optimal path and is defined by equation (12), $L_{gb}$ indicates the cost of the global-best path, $L_{ib}$ indicates the cost of the iteration-best solution, $T^{gb}$ represents the global-best solution, $T^{ib}$
represents the iteration-best path, $\Delta r_{\text{worst}}(t)$ is the additional pheromone evaporation on node $j$ belonging to the worst solution among the current full paths and is defined by equation (13), $L_{\text{worst}}$ is the cost of the worst path, and $T_{\text{worst}}$ is the worst path.

It can be seen from the principle of the improved global strategy that the dynamically weighted global updating strategy not only considers the iterative best ant and the global best ant concurrently but also dynamically regulates the pheromone according to the cost of the iteration-best ant. Through this option, the solution components that frequently occur in the optimal path (the iteration-optimal or the global-optimal path) can achieve a high level of reinforcement and lead to improved performance. The penalty of additional pheromone evaporation for the worst path is included for reducing the misleading effect of the worst solution on the following generations, thus enabling the acceleration of the convergence of the algorithm.

**Local optimization algorithm based on path geometric features algorithm**

During the construction phase of the ACO algorithm, ants construct a feasible solution in a single iteration via the pseudorandom-proportional rule wildly used in almost all advanced ACAs. This approach can speed up the convergence rate and provide better results. The major purpose of the random parts in the state transition rule is to avoid premature stagnation and enhance the exploration capacity. When using ACS algorithm for path planning in a grid workspace, random mode in the probability transition rule may produce some nonoptimized paths, which may contain some undesirable local cross parts, sharp bends, or circular parts during the early phases. These undesirable path segments can reduce the searching ability and optimization efficiency of the algorithm. Figure 5(a) shows one particular example of nonoptimized path. As shown in the figure, $S \rightarrow P_1 \rightarrow \cdots \rightarrow P_{17} \rightarrow G$ is an initial feasible path between $S$ and $G$. Obviously, it can be observed from Figure 5(a) that this path is not the best one. For example, one sharp bend can be found between the nodes $P_{1}$ and $P_{3}$, another sharp bend can also be found between the nodes $P_{11}$ and $P_{13}$. At the same time, there is a cross path between nodes $P_{5}$ and $P_{10}$. As described before earlier, these undesired path segments can affect the solution quality and the convergence rate of the algorithm, and thus, we have introduced the LOAGF algorithm to remove the undesirable path segments and shorten the overall path length.

The main idea of the LOAGF algorithm is that if two nodes on the path selected randomly can be directly connected by a straight line, the path segments or the redundant nodes between them can be removed to further optimize the original path. The condition for a direct connection between the two nodes is that they must be on the same row, column, or diagonal of the grid because the potential heading angles of the mobile robot in grid environments have been artificially constrained to multiples of 45°. In addition, the novel path generated by the LOAGF does not pass through any obstacle grids. Furthermore, when the novel path spans more than one grid, new nodes must be inserted between the two nodes due to the constraint of the direction of motion shown in Figure 2.

We take Figure 5(b) as an example to explicate the principle of the LOAGF algorithm. The red solid line indicates the original path and the blue dotted line shows the path pruned by LOAGF approach. Firstly, we can remove the node $P_2$ because $P_1$ and $P_3$ can be connected directly because the line between them does not pass through any
obstacle grids. Secondly, the initial path between the nodes $P_5$ and $P_{10}$ is a cross path and can be removed to obtain a shorter path because the node $P_5$ can be directly connected to the node $P_{10}$. Further, new waypoints, indicated by $q_1$ and $q_2$ in Figure 5(b), also need to be inserted on the straight line between nodes $P_5$ and $P_{10}$ to accommodate the constraints discussed above. Furthermore, the LOAGF algorithm has been implemented based on a greedy local search strategy for optimizing the initial path, which can improve the path smoothness and shorten the length of the path. This implies that we first try to connect the target node, $G$, from the start node, $S$. If the connection fails, an attempt is made to connect from a closer node until it connects. Once the target node $G$ is connected, we start again with its directly connected node. From Figure 5(b), we can see that the length of the original path and the refined path is 22.9706 and 15.6659, respectively. Meanwhile, the number of nodes corresponding to the original route and the refined route is 19 and 15, respectively. Obviously, the length of the original route and the number of nodes are both reduced effectively. Therefore, we can conclude that the LOAGF algorithm not only can remove the redundant nodes in the initial path but also reduce the path length. Algorithm 1 illustrates the pseudocode of the LOAGF algorithm.

The CollisionCheck function is executed to determine whether or not the straight line between $P(\text{Target index})$ and $P(\text{Current Index})$ passes through any obstacle grids or not. If there is no collision, the flag will be set to false, otherwise, it will be true. The node in line 5 of Algorithm 1 represents the new waypoints to be inserted on the straight line if it spans more than one grid.

In addition, it should be noted that the pheromones on the refined route and the initial optimal route are simultaneously updated to speed up the algorithm’s convergence rate and improve its searching efficiency.

![Figure 5](image.png)

**Figure 5.** Example of a path refined by the LOAGF approach. (a) The original path found by the ant. (b) The refined path generated by LOAGF approach. LOAGF: local optimization algorithm based on path geometric feature.

**Algorithm 1.** Pseudo-code of the LOAGF algorithm

```
Input: initial feasible path (denoted as P), grid environment (denoted as ENV)
Output: refined path (denoted as Q)
1 TargetIndex = length(P), CurrentIndex = 1
2 Q = target node
3 while TargetIndex > 1 do
4     while TargetIndex > CurrentIndex do
5         [flag, node] = CollisionCheck(P(TargetIndex), P(CurrentIndex), ENV)
6         if flag = false then
7             Q = [P(CurrentIndex) node Q]
8             TargetIndex = CurrentIndex
9             Break;
10         else
11             CurrentIndex = CurrentIndex + 1
12         end
13     end
14 end
15 return Q
```

**Application of the proposed EACSPGO for path planning**

To provide better results and convergence rate of the classical ACS, a hybrid approach combining EACS with the LOAGF algorithm, namely, EACSPGO, has been presented in this work. The detailed procedures of the EACSPGO method for path planning are described as following:

Step 1: Loading the environment model for path planning and assigning the starting grid $S$ and the target grid $G$.

Step 2: Initializing parameter. The number of ants $m$, the predefined number of generations, maxIter, the current number of generations, $Nc$, $\alpha$, $\beta$, and $q_0$ are specified.
Step 3: The value of the initial pheromone is assigned using equation (9).

Step 4: Path construction. The ants are placed in the starting position, and equations (1) and (2) are used to determine the next feasible grid, \( j \). Repeat this step until each ant reaches the target location or a path deadlock situation occurs, which means that the newly added node is surrounded by obstacles or visited node. A local pheromone update is performed during solution building.

Step 5: When all ants finish their iteration, all complete routes will be evaluated. Among all the complete routes of this generation, we need to identify the global-best, the iteration-best, and the worst route, respectively. Then, the global pheromone update rule is executed using equation (10), meanwhile, the pheromones on the best route are diffused using equation (7).

Step 6: The LOAGF algorithm is applied to the best route found so far, and the pheromone update rule based on equation (4) is used again for the refined path to enhance the effect of the refined path.

Step 7: Steps 4–6 will be repeated continuously until the stopping conditions are satisfied.

Step 8: The optimal path is outputted.

The flowchart of the EACSPGO for path planning is completely shown in Figure 6.

**Experimental results and discussions**

In this section, we examine the performance of the EACSPGO algorithm in different scenarios. A series of tests have been carried out to demonstrate the performance of the proposed EACSPGO for path planning of mobile robots. In the first part of the tests, we concentrated on testing the effectiveness of the EACS method. The objective of the second part was to assess the effectiveness of the LOAGF approach. Finally, simulations in various scenarios with a variety of complexities and sizes were performed to reveal the adaptability and efficiency of the EACSPGO algorithm in path planning.

**Experimental setup**

To get an unbiased comparison, all simulations were conducted using the same PC with a 3.40 GHz Core CPU and 8
GB RAM. All algorithms were encoded and implemented using the MATLAB(R2016b) programming platform. The stopping criterion for all algorithms was the reaching of the predefined maximum number of generations. All tests were performed independently 30 times with the same parameters to avoid occasional cases.

The results were averaged and compared according to the average performance of the algorithms over 30 independent trials. The best (best), average (ave), and standard deviation (SD) of the optimal solution were employed to evaluate the tests. Best indicates the best result obtained after 30 runs. Average presents the average cost of 30 tests, whereas SD describes the standard deviation of results for 30 independent runs. In addition, the performance metric, fbest, was introduced to indicate the number of generations used for the first time to obtain the best result, which also describes, to some extent, the convergence rate of algorithm. The performance metric of rate was used to evaluate the rate of success of obtaining the best route for all runs. In addition, it should be noted that the length of each grid is 1 unit (1 m). At the same time, in the following experiment, we calculate the length of the path according to the number of grids occupied by the path.

Parameter settings

Since parameter selection may significantly influence the solution quality of intelligent algorithms, the ACO algorithms have several parameters, including the predefined number of generations (maxIter), the amount of ants (m), the global pheromone volatilization variable (ρ), the local pheromone volatilization variable (ξ), the pheromone quantity (Q), (α), (β), and q0 that should be defined before running. The common parameters of the ant colony-based algorithms in all the tests described in the subsequent sections are presented in Table 1. The parameter values of the PS-ACO were equivalent to the recommended settings in the literature. The parameters for the MRCACO were the same as those reported in the work. Parameters of the HHACO were assigned according to the suggested values given by Tuani et al.

Verification for the efficiency of enhanced ant colony system algorithm

To improve its effectiveness and convergence rate of the basic ACS algorithm, we presented an enhanced version of the original ACS method named EACS. The EACS algorithm differs mainly in three aspects: pheromone diffusion strategy, pheromone initialization strategy, and adaptive pheromone updating scheme. The effectiveness of the EACS algorithm has been verified by comparing its results with those of the classical ACS algorithm. Two different grid maps (map1 and map2) were selected for this test, as shown in Figure 7. The results of this test are provided in Table 2, and the best-obtained values are shown in bold. Figure 7 display the optimal paths generated by the EACS.

| Map   | Algorithm | Best   | Average | SD    | Fbest | Rate |
|-------|-----------|--------|---------|-------|-------|------|
| Map1  | ACS       | 38.8701| 40.3259 | 1.05  | 116.5 | 3.33 |
|       | EACS      | 38.8701| 39.5306 | 0.65  | 82.47 | 26.67|
| Map2  | ACS       | 30.9706| 32.0999 | 1.19  | 71.73 | 36.67|
|       | EACS      | 30.9706| 31.4863 | 0.58  | 31.93 | 50   |

EACS: enhanced ant colony system; ACS: ant colony system; SD: standard deviation.

The best result in each case is highlighted in bold.

Verification for the efficiency of enhanced ant colony system algorithm

To improve its effectiveness and convergence rate of the basic ACS algorithm, we presented an enhanced version of the original ACS method named EACS. The EACS algorithm differs mainly in three aspects: pheromone diffusion strategy, pheromone initialization strategy, and adaptive pheromone updating scheme. The effectiveness of the EACS algorithm has been verified by comparing its results with those of the classical ACS algorithm. Two different grid maps (map1 and map2) were selected for this test, as shown in Figure 7. The results of this test are provided in Table 2, and the best-obtained values are shown in bold. Figure 7 display the optimal paths generated by the EACS.

The results in Table 2 present that the ACS and EACS methods obtain the optimal paths in both the maps according to the optimal path length (best). However, with respect to the remaining performance metrics, the proposed EACS method can provide better performance than the original ACS method. As it can be deduced from this table, the EACS algorithm has a much faster convergence speed and
higher success rate compared to ACS. These performance improvements are attributable to the introduction of the designed strategy. Therefore, we can conclude from Table 2 that the optimization capability of the ACS algorithm, the rate of convergence, and the success rate are effectively enhanced by the proposed strategies.

Verification for the effectiveness of the local optimization algorithm based on path geometric feature algorithm

The main goal of the LOAGF method is to remove the undesirable path segments and continue improving the quality of solution and the convergence rate. This subsection provides the details of a comparative test to validate the effectiveness of the LOAGF algorithm in two different grid maps (map3 and map4), as shown in Figure 8. In case I, the ACS algorithm was run without local optimization mechanism, while in case II, the ACS was executed with local optimization mechanism. The results obtained are provided in Table 3, in which the bold results represent the best values. The optimal paths obtained by ACS with LOAGF algorithm are also depicted in Figure 8.

Table 3. Comparative results of the different algorithms in the grid map3 and map4.a

| Env | Algorithm    | Best   | Ave   | SD   | Fbest | Rate |
|-----|--------------|--------|-------|------|-------|------|
| map3 ACS | 44.5269      | 44.5269| 0.15  | 14.1 | 93.33 |
| ACS+LOAGF | 44.5269      | 44.5269| 0     | 10.4 | 100   |
| map4 ACS | 42.1838      | 42.1838| 0     | 4.5  | 100   |
| ACS+LOAGF | 42.1838      | 42.1838| 0     | 2.7  | 100   |

ACS: ant colony system; LOAGF: local optimization algorithm based on path geometric feature; SD: standard deviation.

*aThe best result for each section is highlighted in bold.

Simulations in various environments

To further investigate the adaptability and optimization capability of the presented EACSPGO method for path planning, we compare it with three advanced approaches, namely, PS-ACO, a hybrid of PSO-ACO, HHACO, and a heterogeneous adaptive ACO methodology, and MRCACO, a dynamic MRCACO method, as well as three variants of the AS algorithm, namely, EAS, MMAS, and ACS. The third test was conducted in six grid maps of different sizes, as shown in Figure 9. The statistical results drawn from EACSPGO method and the other algorithms for various environments are illustrated in Tables 4 to 9 by the performance criteria mentioned above, and the best values have been marked in bold. Figure 9 also depicts the best global path generated by the EACSPGO under different scenarios.

From Tables 4 to 9, we can see that all the algorithms provide good results for simple or small-sized maps, such as map5 and map6; however, our algorithm and MRCACO are still able to discover the best path by increasing the scale of the map. In view of the Ave metric, the proposed EACSPGO method exhibits a better performance in four of the six scenarios. In all scenarios, our algorithm achieves the highest success rate except for the case of map6 and map7. Thus, it can be argued that the EACSPGO algorithm
is much more robust and stable than the other algorithms, and has better optimization capabilities. However, when considering the standard deviation of the optimal solution, the ACS algorithm exhibits better performance in five of the six scenarios. As far as the Fbest performance metric is concerned, the PS-ACO algorithm exhibits better performance in five of the six scenarios, but this algorithm shows the worst performance based on the criterion of path length. In other words, our algorithm does not achieve superior performance in terms of these two metrics, which is mainly because the pheromone diffusion mechanism helps the ACS algorithm to search for a larger problem space and to escape the local optimal solution, whereas the other algorithms easily fall into a local optimal state.
Table 4. Statistical results obtained by the different algorithms under grid map5.

| Algorithm       | Best     | Average  | SD   | Fbest   | Rate |
|-----------------|----------|----------|------|---------|------|
| EAS             | 30.3848  | 31.1376  | 1.25 | 59.47   | 60   |
| MMAS            | 30.3848  | 30.7633  | 0.53 | 16.97   | 60   |
| ACS             | 30.3848  | 30.3848  | 0    | 19.5    | 100  |
| PS-ACO          | 30.3848  | 33.3602  | 1.50 | 10.27   | 3.33 |
| HHACO           | 30.3848  | 30.4676  | 0.25 | 23.67   | 90   |
| MRCACO          | 30.3848  | 30.3848  | 0    | 37.4    | 100  |
| EACSPGO         | 30.3848  | 30.3848  | 0    | 6.57    | 100  |

MMAS: MAX-MIN ant system; ACS: ant colony system; PS-ACO: particle swarm-ant colony optimization; HHACO: heterogeneous ant colony optimization; MRCACO: multirole adaptive collaborative ant colony optimization; EACSPGO: enhanced ant colony system with the local optimization algorithm based on path geometric feature; SD: standard deviation.

Table 5. Statistical results obtained by the different algorithms under grid map6.

| Algorithm       | Best     | Average  | SD   | Fbest   | Rate |
|-----------------|----------|----------|------|---------|------|
| EAS             | 43.3553  | 47.3604  | 2.52 | 83.6    | 3.33 |
| MMAS            | 43.3553  | 43.3553  | 0    | 5.9     | 100  |
| ACS             | 43.3553  | 43.3553  | 0    | 4.43    | 100  |
| PS-ACO          | 43.3553  | 46.4368  | 2.44 | 1.07    | 10   |
| HHACO           | 43.3553  | 43.3553  | 0    | 1.13    | 100  |
| MRCACO          | 43.3553  | 43.3553  | 0    | 1.43    | 100  |
| EACSPGO         | 43.3553  | 43.7068  | 0.66 | 18.53   | 66.67|

MMAS: MAX-MIN ant system; ACS: ant colony system; PS-ACO: particle swarm-ant colony optimization; HHACO: heterogeneous ant colony optimization; MRCACO: multirole adaptive collaborative ant colony optimization; EACSPGO: enhanced ant colony system with the local optimization algorithm based on path geometric feature; SD: standard deviation.

Table 6. Statistical results obtained by the different algorithms under grid map7.

| Algorithm       | Best     | Average  | SD   | Fbest   | Rate |
|-----------------|----------|----------|------|---------|------|
| EAS             | 59.255   | 65.2379  | 2.97 | 133.53  | 3.33 |
| MMAS            | 58.669   | 59.1243  | 0.50 | 70.33   | 46.67|
| ACS             | 58.669   | 59.1438  | 0.78 | 29.47   | 70   |
| PS-ACO          | 62.083   | 64.9525  | 1.96 | 18.33   | 3.33 |
| HHACO           | 58.669   | 59.0023  | 0.51 | 47.8    | 63.33|
| MRCACO          | 58.669   | 58.833   | 0.35 | 76.23   | 80   |
| EACSPGO         | 58.669   | 59.8501  | 1.38 | 40.11   | 50   |

MMAS: MAX-MIN ant system; ACS: ant colony system; PS-ACO: particle swarm-ant colony optimization; HHACO: heterogeneous ant colony optimization; MRCACO: multirole adaptive collaborative ant colony optimization; EACSPGO: enhanced ant colony system with the local optimization algorithm based on path geometric feature; SD: standard deviation.

Table 7. Statistical results obtained by the different algorithms under grid map8.

| Algorithm       | Best     | Average  | SD   | Fbest   | Rate |
|-----------------|----------|----------|------|---------|------|
| EAS             | 76.9116  | 85.2492  | 5.10 | 162.67  | 3.33 |
| MMAS            | 73.9828  | 75.4064  | 0.69 | 37.53   | 6.67 |
| ACS             | 73.9828  | 74.7822  | 0.35 | 66.37   | 10   |
| PS-ACO          | 76.468   | 82.4518  | 3.09 | 8.57    | 3.33 |
| HHACO           | 73.9828  | 74.8     | 0.72 | 76.57   | 33.33|
| MRCACO          | 73.9828  | 74.6475  | 0.70 | 73.53   | 40   |
| EACSPGO         | 73.9828  | 74.3147  | 0.63 | 78.2    | 73.33|

MMAS: MAX-MIN ant system; ACS: ant colony system; PS-ACO: particle swarm-ant colony optimization; HHACO: heterogeneous ant colony optimization; MRCACO: multirole adaptive collaborative ant colony optimization; EACSPGO: enhanced ant colony system with the local optimization algorithm based on path geometric feature; SD: standard deviation.

Table 8. Statistical results obtained by the different algorithms under grid map9.

| Algorithm       | Best     | Average  | SD   | Fbest   | Rate |
|-----------------|----------|----------|------|---------|------|
| EAS             | 77.1543  | 87.5287  | 5.83 | 149.27  | 6.67 |
| MMAS            | 73.3970  | 74.3088  | 0.60 | 25.67   | 20.00|
| ACS             | 73.3970  | 73.7727  | 0.50 | 52.60   | 53.33|
| PS-ACO          | 75.6396  | 79.9400  | 2.96 | 5.60    | 3.33 |
| HHACO           | 73.3970  | 73.9936  | 0.49 | 45.30   | 33.33|
| MRCACO          | 73.3970  | 74.1303  | 0.61 | 61.27   | 26.67|
| EACSPGO         | 73.3970  | 73.6703  | 0.63 | 26.67   | 76.67|

MMAS: MAX-MIN ant system; ACS: ant colony system; PS-ACO: particle swarm-ant colony optimization; HHACO: heterogeneous ant colony optimization; MRCACO: multirole adaptive collaborative ant colony optimization; EACSPGO: enhanced ant colony system with the local optimization algorithm based on path geometric feature; SD: standard deviation.

Table 9. Statistical results obtained by the different algorithms under grid map10.

| Algorithm       | Best     | Average  | SD   | Fbest   | Rate |
|-----------------|----------|----------|------|---------|------|
| EAS             | 98.2254  | 109.9084 | 6.75 | 166.63  | 3.33 |
| MMAS            | 91.0538  | 93.1846  | 1.10 | 42.87   | 3.33 |
| ACS             | 91.6396  | 92.8377  | 0.80 | 100.6   | 10   |
| PS-ACO          | 94.7107  | 101.3284 | 3.31 | 8.47    | 3.33 |
| HHACO           | 91.0538  | 92.3804  | 1.06 | 81.1    | 6.67 |
| MRCACO          | 90.468   | 92.328   | 0.89 | 80.9    | 3.33 |
| EACSPGO         | 90.468   | 91.7568  | 1.31 | 88.77   | 23.33|

MMAS: MAX-MIN ant system; ACS: ant colony system; PS-ACO: particle swarm-ant colony optimization; HHACO: heterogeneous ant colony optimization; MRCACO: multirole adaptive collaborative ant colony optimization; EACSPGO: enhanced ant colony system with the local optimization algorithm based on path geometric feature; SD: standard deviation.

Conclusion and future works

In summary, the proposed EACSPGO algorithm performs well overall in all different environments, and it can thus be concluded that it is more stable and robust compared to the other algorithms.
algorithm was developed by combining the pheromone initialization strategy of unequal allocation, a simplified pheromone diffusion model, and the adaptive pheromone update mechanism with the ACS algorithm to improve the optimization capability and efficiency of the ACS algorithm. The LOAGF algorithm has been proposed to further refine the initial path by removing the redundant segments. A number of simulations for mobile robot path planning have been conducted to assess the effectiveness and performance of the proposed methodology. The simulation results suggest that the EACSPGO approach outperforms the other methods and adapts to various scale scenarios.

In future work, the authors intend to further employ the proposed approach to dynamic path planning of robot. Another interesting theme is to apply the proposed EACSPGO approach to more practical applications than just path planning, such as the vehicle routing problem, scheduling problem, feature selection, and so on.

Acknowledgments
The authors would like to thank the associate editor and the anonymous reviewers who contributed their valuable comments to this article.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Thirteen-Five Equipment Pre-Research Foundation of China [Nos 61403120207 and 61402100203], the Aeronautical Science Foundation of China [No. 20185142003], the National Defense Basic Scientific Research Program of China [No. JCKY2018419C001], the Science and Technology Innovative Talents in Universities of Henan Province [No. 21HASTIT030], Young Backbone Teachers in Universities of Henan Province [No. 2020GGJS073], and the Leading Talents of Science and Technology Innovation in the Central Plains of China [No. 194200510012].

ORCID iD
Songcan Zhang https://orcid.org/0000-0002-9040-0408

References
1. Pehlivanoglu YV. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. Aerosp Sci Technol 2012; 16(1): 47–55. DOI: 10.1016/j.ast.2011.02.006.
2. Yen CT and Cheng MF. A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance. Microsysy Technol 2016; 24: 125–135. DOI: 10.1007/s00542-016-3192-9.
3. Lau G and Liu HHT. Real-time path planning algorithm for autonomous border patrol: design, simulation, and experimentation. J Intell Robot Syst 2014; 75(3–4): 517–539. DOI: 10.1007/s10846-013-9841-7.
4. Jia W, Tian Y, Duan H, et al. Autonomous navigation control based on improved adaptive filtering for agricultural robot. Int J Adv Robot Syst 2020; 17. DOI: 10.1177/1729881420925357.
5. Jia W, Mou S, Wang J, et al. Fruit recognition based on pulse coupled neural network and genetic Elman algorithm application in apple harvesting robot. Int J Adv Robot Syst 2020; 17. DOI: 10.1177/1729881419897473.
6. Soler D, Martinez E, and Micó JC. A transformation for the mixed general routing problem with turn penalties. J Oper Res Soc 2008; 59(4): 540–547. DOI: 10.1057/palgrave.jors.2602385.
7. Gao W, Tang Q, Ye B, et al. An enhanced heuristic ant colony optimization for mobile robot path planning. Soft Computing 2020; 24: 6139–6150. DOI: 10.1007/s00500-020-04749-3.
8. Qi H, Xing K, and Alexander T. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing 2013; 120: 509–517. DOI: 10.1016/j.neucom.2013.04.020.
9. Singh MK and Parhi DR. Path optimisation of a mobile robot using an artificial neural network controller. Int J Syst Sci 2011; 42(1): 107–1202. DOI: 10.1080/00207720903470155.
10. Contreras-Cruz MA, Ayala-Ramirez V, and Hernandez-Belmonte UH. Mobile robot path planning using artificial bee colony and evolutionary programming. Appl Soft Comput 2015; 15: 319–328. DOI: 10.1016/j.asoc.2015.01.067.
11. Hosseininejad S and Dadkhah C. Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm. Int J Adv Robot Syst 2019; 16: 1–13. DOI: 10.1177/1729881419839575.
12. Jones J. A morphological adaptation approach to path planning inspired by slime mould. Int J Gen Syst 2015; 44: 279–291. DOI: 10.1080/03081079.2014.997526.
13. Zhou Y, Wang Y, Chen X, et al. A novel path planning algorithm based on plant growth mechanism. Soft Comput 2017; 21(2): 435–445. DOI: 10.1007/s00500-016-2045-x.
14. Viseras A, Losada RO, and Merino L. Planning with ants: efficient path planning with rapidly exploring random trees and ant colony optimization. Int J Adv Robot Syst 2016; 13(5). DOI: 10.1177/172988141664078.
15. Wang L, Luo C, Li M, et al. Trajectory planning of an autonomous mobile robot by evolving ant colony system. Int J Robot Autom 2017; 32(4): 406–413. DOI: 10.2316/Journal. 206.2017.4.206-4917.
16. Akka K and Khaber F. Mobile robot path planning using an improved ant colony optimization. Int J Adv Robot Syst 2018; 15(3). DOI: 10.1177/1729881418774673.
17. Liu J, Yang J, Liu H, et al. Mobile robot path planning using an improved ant colony optimization, Soft Comput 2017; 21(19): 5829–5839. DOI: 10.1007/s00500-016-2161-7.
18. Chen X, Kong Y, Fang X, et al. A fast two-stage ACO algorithm for robotic path planning. Neural Comput Appl 2013; 22(2): 313–319. DOI. 10.1007/s00521-011-0682-7.
19. Zhang C, Li Q, Chen P, et al. Ant colony optimization combined with immunosuppression and parameters
switching strategy for solving path planning problem of landfill inspection robots. Int J Adv Robot Syst 2016; 13(3): 130. DOI: 10.5772/63737.

20. Chen P, Li Q, Zhang C, et al. Hybrid chaos-based particle swarm optimization-ant colony optimization algorithm with asynchronous pheromone updating strategy for path planning of landfill inspection robots. Int J Adv Robot Syst 2019; 16(4): 1–11. DOI: 10.1177/1729881419859083.

21. Mac T T, Copot C, Tran D T, et al. Heuristic approaches in robot path planning: A survey. Robot Auton Syst 2016; 86: 13–28. DOI: 10.1016/j.robot.2016.08.001.

22. Dorigo M and Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1997; 1(1): 53–66. DOI: 10.1109/4235.585892.

23. Stützle T and Hoos H. MAX-MIN ant system. Future Gener Comput Syst 2000; 16(8): 889–914. DOI: 10.1109/4235.585892.

24. Jiao Z, Ma K, Rong Y, et al. A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs. J Comput Sci 2018; 25: 50–57. DOI: 10.1016/j.jocs.2018.02.004.

25. Deng W, Chen R, He B, et al. A novel two-stage hybrid swarm intelligence optimization algorithm and application. Soft Comput 2012; 16(10): 1707–1722. DOI: 10.1007/s00500-012-0855-z.

26. Luo Q, Wang H, Zheng Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm. Neural Comput Appl 2019; 1: 1–12. DOI: 10.1007/s00521-019-04172-2.

27. Mavrovouniotis M and Yang S. Ant colony optimization with self-adaptive evaporation rate in dynamic environments. In: 2014 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE), Orlando, FL, USA, 9–12 December, 2014, pp. 47–54. DOI: 10.1109/CIDUE.2014.7007866.

28. Xiong C, Chen D, Lu D, et al. Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization. Rob Auton Syst 2019; 115: 90–103. DOI: 10.1016/j.robot.2019.02.002.

29. Shuang B, Chen J, and Li Z. Study on hybrid PS-ACO algorithm. Appl Intell 2011; 34(1): 64–73. DOI: 10.1007/s10489-009-0179-6.

30. Zhang D, You X, Liu S, et al. Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning. IEEE Access 2020; 8: 129958–129974. DOI: 10.1109/ACCESS.2020.3009399.

31. Tuani AF, Keedwell E, and Collett M. Heterogenous adaptive ant colony optimization with 3-opt local search for the traveling salesman problem. Appl Soft Comput 2020; 97: 106720. DOI: 10.1016/j.asoc.2020.106720.

32. Zhang S, Pu J, and Si Y. An adaptive improved ant colony system based on population information entropy for path planning of mobile robot. IEEE Access 2021; 9: 24933–24945. DOI: 10.1109/ACCESS.2021.3056651.

33. Hsu CC, Wang WY, Chien YH, et al. FPGA implementation of improved ant colony optimization algorithm based on pheromone diffusion mechanism for path planning. J Mar Sci Technol 2018; 26(2): 170–179. DOI: 10.6119/JMST.2018.04 (2).0004.