A Robot Path Planning Method Based on Synergy Behavior of Cockroach Colony

Le Cheng
College of Computer Science and Communication, Jiangsu Vocational College of Electronics and Information, China
cl211282@163.com

Lyu Chang
College of Computer Science and Communication, Jiangsu Vocational College of Electronics and Information, China
464813038@qq.com

Yanhong Song
College of Computer Science and Communication, Jiangsu Vocational College of Electronics and Information, China
4793173@qq.com

Haibo Wang
Jiangsu Industrial Cloud Edge Collaborative Technology Engineering Research Center, China
5328702 @qq.com

Yuetang Bian
School of Business, Nanjing Normal University, China
93448437 @qq.com

Abstract: *By studying the biological behavior of cockroaches, a bionic algorithm, Cooperative Learning Cockroach Colony Optimization (CLCCO), is presented in this paper. The aim of CLCCO is to provide an efficient method to solve Robot Path Planning (RPP) problems. The CLCCO algorithm is based on the idea of synergy behavior of cockroach colony and machine learning. With pheromone, the cockroach colony achieves population synergy, which includes the follow and diversion behaviors. The strategy of Fibonacci transformation is used for the cockroach individual to choose the next feasible cell. The technologies of λ-geometry and multi-objective search make the paths searched smoother and greatly improve the algorithm search efficiency. In particular, the CLCCO algorithm requires only two parameters to be set. When CLCCO is applied to real robots, a path compression technique is designed. The simulation results show that the CLCCO algorithm demonstrates high efficiency in mostly tests.*

Keywords: *Cooperative learning, robot path planning, fibonacci transformation, controlling parameters, path compression technique.*

Received August 20, 2021; accepted December 1, 2022
https://doi.org/10.34028/iajit/20/5/4

1. Introduction

Synergy exists widely in the natural world. The cooperation and competition are important factors for survival of populations, such as ant colonies, beehives, termite mounts, flocks of birds, schools of fish etc., The individual is smarter when working in teams, the sum much greater than the parts. Synergy based on the cooperation provides the inspiration for intelligent computation. The Swarm Intelligence (SI) algorithm has implemented synergy by bionic approach, such as Genetic Algorithm (GA) [5, 18, 21], Particle Swarm Optimization (PSO) [8, 10, 11, 17], Ant Colony Optimization (ACO) [6, 16, 22] etc. In recent years, the cockroach-inspired algorithms are proposed and developed. In 2008, literature [9] firstly presented the Cockroach Swarm Optimization (CSO) algorithm for the Travelling Salesman Problem (TSP). Later, ZhaoHui and HaiYan [20] applied the CSO to global optimization problems. And then, some new cockroach-inspired algorithms are proposed and applied to some practical problems [4, 12, 19]. In recent five years, the cockroach-inspired algorithms more and more attracted the attention of scholars, which is mainly developed in the field of numerical optimization and Robot Path Planning (RPP). For example, by improving Roach Infestation Optimization (RIO) [4], Tsai proposed a Center RIO (CRIO) algorithm [14]. In CRIO, each roach agent moves toward its friendship center rather than oscillate around the swarm center. Based on the literature [13], Obagbuwa proposes an Adaptive Cockroach Swarm algorithm (ACSO) for global optimization, which executes an adaptive search. Cheng presents the Cockroach Colony Optimization (CCO) algorithm [3]. The logistic multi-peak map and the margin control strategies are introduced in CCO. Literature [2] presents the CCO algorithm for RPP problem. The improved grid map and non-probabilistic search strategy are used for the CCO algorithm. However, the CCO algorithm needs too many controlling parameters. This problem increases the complexity of CCO.

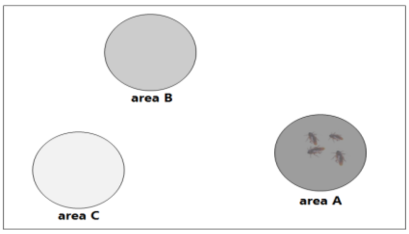
In this paper, a novel cockroach-inspired algorithms, Cooperative Learning CCO (CLCCO), is proposed for solving the RPP problems. Our aim is to provide a path planning method with fewer control parameters and higher efficiency. In CLCCO, the gird method is used for environment modeling. The strategy of cooperative learning is proposed and applied to the motion of cockroach individual. The follow and diversion behaviors of cockroach are simulated. The pheromone value is dynamically updated and computed according to the length of feasible path. By the pheromone value, the individual cockroaches learn from each other and decide whether to follow or diversion. Especially, the CLCCO requires only two controlling parameters to be set.

The remainder of this paper is organize as follow. In section 2, the background of the CLCCO algorithm is introduced, including the relevant biological theory and the basis of the algorithm. Section 3 gives some definitions on CLCCO. The details of CLCCO are introduced in section 4. The simulation experiments and compression path technique are given in section 5. Section 6 concludes this paper.

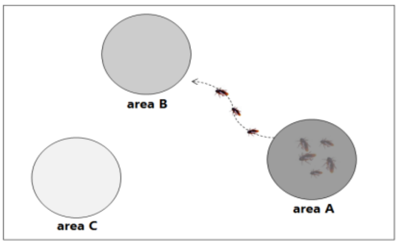
2. Background

2.1. Cooperative Behaviour of Cockroach Colony

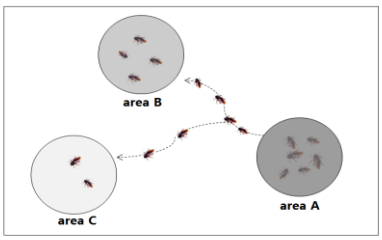
Most cockroaches have poor vision but a good sense of smell. Cockroach colony can communicate by the pheromones to organize and make decisions. Cockroach society is democratic and has no absolute leadership [15]. In recent years, a number of biologists have been studying on the cockroach's distinctive social lifestyle [1, 7, 15]. A series of experiments is designed, which showed that cockroaches have evolved smart behavior of cooperation to survive.



a) Cockroaches gather in area A.



b) Some cockroaches choose separation.



c) Some cockroaches choose diversion.

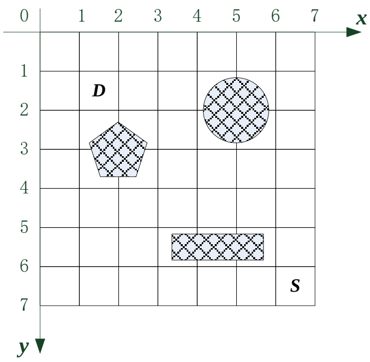
Figure 1. Experimental process on cockroach's social lifestyle.

Figure 1-a) to (c) show one of the experimental process [7]. In Figure 1-a), three dark areas are set in a closed container. Notice that area A is darker than area B and area B is darker than area C. When some cockroaches were put into the container, they all gathered in area A. The reason is that cockroaches like darker areas. As more cockroaches are placed in area A, some of them will move from area A to B (see Figure 1-b)). With one cockroach moving, other cockroaches may choose follow (the probability is about 60%) [7]. This process is called as separation. When cockroaches crawl from A to B, some pheromone will be left on the route. When the level of pheromone is too high, the cockroaches may choose to diversion. The diversion means that some of them may move to area C (see Fi Figure 1-c)). This experiment show that pheromones play an important role in cockroach behavior. By pheromones, cockroaches communicate and learn from each other. When the pheromone level is too high, the cockroach will also choose to separation or diversion in order to increase the survival probability.

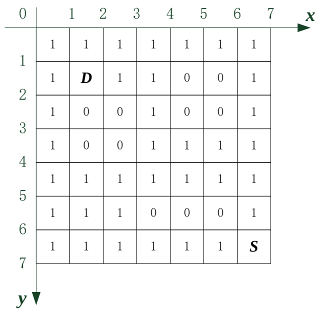
2.2. Grid Map

The workspace is modeled as an $X \times Y$ grid and stored by the set Map. Set Map is compose of the cell c_θ, where Map=$\{c_\theta \mid \theta=1, 2, \ldots, X \times Y\}$.

In rectangular coordinates, the coordinate (x, y) of the point on the lower-right corner of cell c_θ is regarded as the cell coordinate, where $x=1 \ldots X$ and $y=1 \ldots Y$. The process of generated grid map is presented in Figure 2. In Figure 2, S is the start cell and D is the destination cell. "0" represents the obstacle cells and "1" means the free cells.



a) Workspace modeling.



b) Grid map.

Figure 2. The process of generated grid map.

2.3. λ-Geometry and Multi-Objective Search

λ-geometry and multi-object search are efficient path planning techniques, which come from our previous research (See literature [2]).

λ-geometry is the directions in which an cockroach individual might move. In our algorithm, λ is defined as 16 (See Figure 3), that is, 16- geometry. With this strategy, planned routes are smoother and store less information.

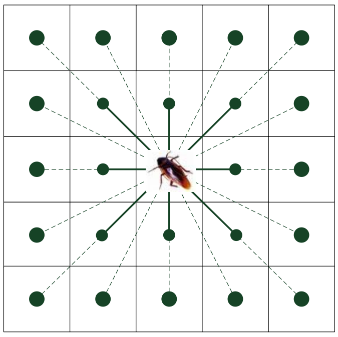


Figure 3. 16- geometry.

Multi-objective search technology can greatly improve the efficiency of the algorithm to find the path. 32-geometry is used to generate some search targets near the destination cell. Notice that these search targets can reach destination cell in a straight line. When moving, a cockroach may encounter some search targets. It means the cockroach finds some paths. Figure 4 shows that the cockroach finds three search targets at one point. That is, the cockroach find three paths. During a single crawl from the start cell to the destination cell, multi-target search allows the cockroach to find multiple paths.

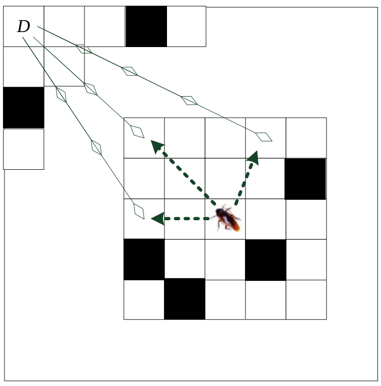


Figure 4. Multi-object searching.

3. Definitions for CLCSO

- *Definition* 1: the moving path of cockroach individual is recorded by set $P_i=\{ p_{i,}^1, p_i^2, \ldots, p_i^t \}$. p_i^t means the cell which the ***i***-th cockroach reaches at time *t*. Notice that there is no the same cells in P_i. Note that duplicate cells cannot appear in P_i in order to avoid circular paths.
- *Definition* 2: each cockroach has the set E called search field. E_i^t denotes the search field of the *i*-th cockroach at *t* time. With 16- geometry, search field E is compose of the 24 adjacent cells of p_i^t. Figure 5 illustrates the structure of E_i^t. Thus, cockroaches can move 1,2, $\sqrt{2}$, $2\sqrt{2}$ or $\sqrt{5}$ distances in a single step.

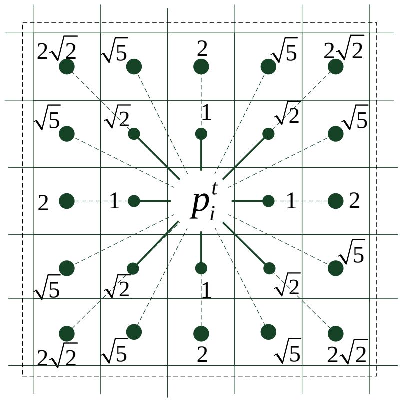


Figure 5. Search field.

- *Definition* 3: set F, called reachable fields, is compose of the reachable cells of search field E_i^t. F_i^t means the reachable field of the ***i***-th cockroach at *t* time. F_i^t is described as follow:

$$F_i^t = \left\{ f_j \mid j = 1 \ldots J, J \le 22, f_j \notin P_i \text{ and } f_j \in E_i^t \right\} \quad (1)$$

In order to avoid the same path being walked repeatedly, the reachable field does not include the cells that the individual cockroach has walked, therefore, J<22.

- *Definition* 4: pheromone value is symbolized as *ph*. ph_θ is pheromone value of the θ-th cell in grid map. At the initialization stage of algorithm, all cells have the same pheromone value, which is computed as follow:

$$ph_\theta = 1/(X \times Y), \quad (\theta = 1, 2, \cdots, X \times Y) \quad (2)$$

After initialization, the cooperative learning search will be executed. In this stage, pheromone value is computed according to the shortest path. Supposing the ω is the length value of the shortest path passing the cell c_θ, we can compute the ph_θ as follow:

$$ph_\theta = 1/\omega \quad (\theta = 1, 2, \cdots, X \times Y) \quad (3)$$

4. Cooperative Learning Cockroach Colony Optimization

In the remainder of this article, for a clear description, the cells are denoted with different symbols at different computing stage. As an example, cell in F_i^t and P_i are symbolized as f_j and p_i^t, respectively, where $p_i^t \in$ Map, $f_j \in$ Map, $F_i^t \subset$ Map and $P_i \subset$ Map.

4.1. Initializing Search

The process of initial search is a simulation on the experimental stage of Figure 1-a). In this process, the colony of cockroaches searches for a nest. This searching is somewhat random.

The population size of CLCCO is symbolled as I. At t time, p_i^t means the i-th cockroach individual has passed t cells. The path is recorded by set P_i. Thus, the p_i^{t+1} is computed as follow:

$$p_i^{t+1} = \begin{cases} Min\{f_j \mid f_j \in F_i^t\} & r_0 \le 0.5 \\ Rand\{f_j \mid f_j \in F_i^t\} & r_0 > 0.5 \end{cases} \tag{4}$$
$$\text{where:} \quad (j = 1, 2 \ldots J),\ (i = 1, 2 \ldots I)$$

Where, F_i^t is reachable field of the i-th cockroach at t time. r_0 is a random number and r_0~U(0,1). For the i-th cockroach, r_0 is regenerated at every step. $Min\{c_\theta | c_\theta \in F_i^t\}$ is the greedy search. By $Min\{c_\theta | c_\theta \in F_i^t\}$, the i-th cockroach at p_i^t chooses one cell from F_i^t as p_i^{t+1}, which has the smallest heuristic information. The heuristic information refers to that the Euclidean distance from the reachable cells in reachable field F_i^t to the destination cell D. $Rand\{c_\theta | c_\theta \in F_i^t\}$ denotes the i-th cockroach randomly chooses a cell from F_i^t as p_i^{t+1}. When D is included in F_i^t, cockroach individuals end up crawling once. Moreover, when fining Q paths, the CLCCO finishes the initialization stage.

After the initial search, the pheromones for the grid map will be updated. Supposing that the ω is the length value of the shortest path passing the cell c_θ, ph_θ is updated as follow:

$$ph_\theta = \begin{cases} 1/\omega & 1/\omega > ph_\theta \\ ph_\theta & 1/\omega \le ph_\theta \end{cases} \tag{5}$$
$$\text{where:} \quad (\theta = 1, 2, \ldots, X \times Y)$$

4.2. Cooperative Learning Search

Cooperative learning is essentially machine learning, which is a simulation on the experimental stage showed in Figure 1-b) and (c). This process includes the diversion and follow behaviors.

The optimal path of the whole colony is recorded by P_{best}. $Path_i$ is the optimal path that found by the i-th colony. P_{best} is described as follow:

$$P_{best} = Opt\{Path_i \mid i = 1, \ldots, I\} \tag{6}$$

Cooperative learning search is described as follow:

$$p_i^{t+1} = Coop\{f_j \mid f_j \in F_i^t\} \quad (j = 1, 2 \ldots J),\ (i = 1, 2 \ldots I) \tag{7}$$

$Coop\{f_j | f_j \in F_i^t\}$ denotes the cooperative learning search. The computing process of $Coop\{f_j | f_j \in F_i^t\}$ is showed in Figure 6.

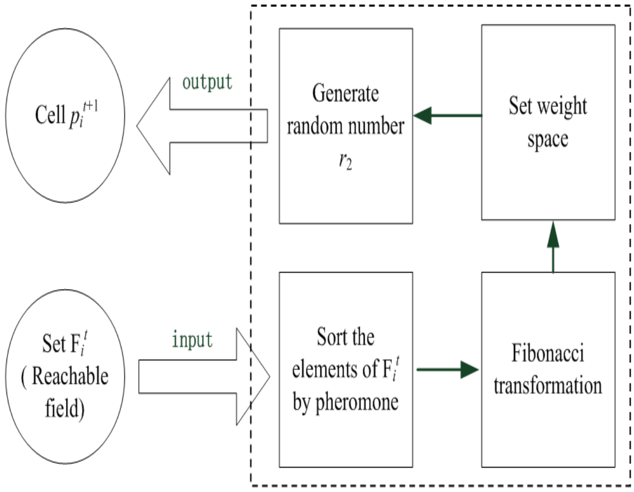


Figure 6. Cooperative learning search.

Firstly, the cells in the set F_i^t are sorted from small to large according to the pheromone value. And then perform the Fibonacci transformation. In detail, the weight value of the first cell is set as δ_1=10, the second one is set as δ_2=20. The n-th ($n \ge 3$) one is computed by Fibonacci sequence, that is, $\delta_n = \delta_{n-1} + \delta_{n-2}$. Thus, the weight space of the first cell is $s_1 = [1, \delta_1]$, the n-th one is $s_n = [\delta_{n-1}+1, \delta_n]$ and the last one is $s_J = [\delta_{J-1}+1, \delta_J]$. The Fibonacci transformation makes cells with larger pheromones have larger weight spaces. In Figure 6, r_2 ($r_2 \in N^*$) is a random positive integer, and r_2~U (1, δ_J). If $r_2 \in s_n$ ($\delta_{n-1}+1 \le r_2 \le \delta_n$), then the n-th cell is as p_i^{t+1}.

For example, suppose there are six cells in the set F_i^t, Figure 7 shows the calculation result of weight value and weight space. From Figure 7, we can find that δ_1, δ_2 and δ_3 have the same weight space. On the other hand, there is more and more weight space from δ_4 to δ_6. This situation means that the first three cells have the same probability of being selected as p_i^{t+1}. Starting with the fourth cell, the higher the pheromone value, the more likely the cell is to be selected as a p_i^{t+1}.

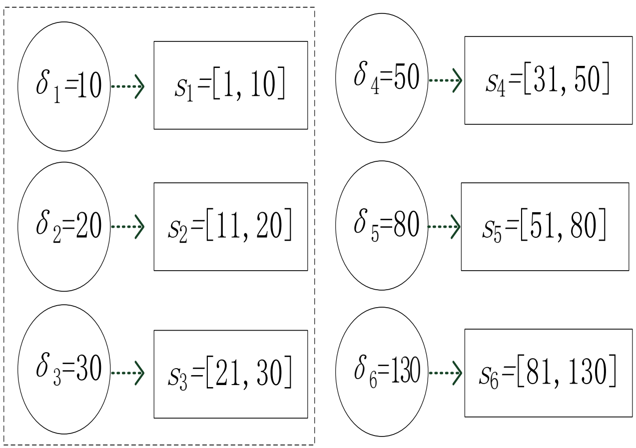
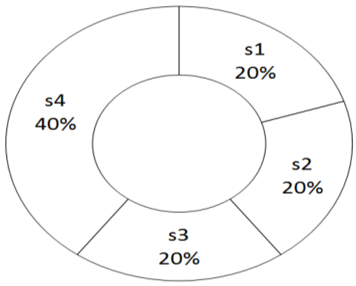
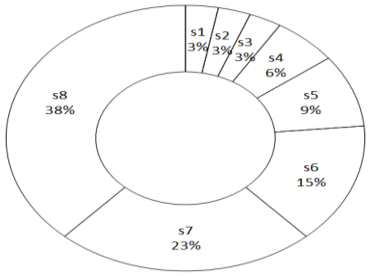


Figure 7. Weight value and weight space.

Figure 8 shows the probability that four and ten cells in F_i^t are selected as p_i^{t+1} based on the weight space.



a) 4 cells.



b) 10 cells.

Figure 8. Probability of weight spaces.

Based on the above conclusions, the relationship between cooperative learning and bionic cockroach behavior is as follows:

1. When the environment is simple, that is, there are fewer feasible cells in F_i^t, cooperative learning tends to choose randomly. The reason is that the weight space of each cell is similar. This process is the bionics of diversion behavior.
2. If F_i^t has more feasible cells. That means that cockroach individual faces a complex environment. Cooperative learning tends to choose the cell with the shortest path. This process completes the simulation of follow behavior.

Furthermore, for the *i*-th cockroach, if the path length of P_i is bigger than that in $Path_i$, then the *i*-th cockroach will execute a new search for the start cell *S*.

Instead, if the *i*-th cockroach finds a new path and its length is smaller than that in $Path_i$, then $Path_i$ is updated as follow:

$$\mathrm{Path}_i = Opt\{\mathrm{P}_i, \mathrm{Path}_i\} \tag{8}$$

After updating $Path_i$, the CLCCO algorithm will update P_{best}. The process is as follow:

$$\mathrm{P}_{best} = Opt\{\mathrm{P}_{best}, \mathrm{Path}_i\} \tag{9}$$

After updating P_{best}, the CLCCO algorithm will update the pheromone of the cells on $Path_i$. The computing process of updating the pheromone is as following:

$$ph_\theta = \begin{cases} 1/\tau & 1/\tau > ph_\theta \\ ph_\theta & 1/\tau \le ph_\theta \end{cases} \tag{10}$$

where:

$$(\theta = 1, 2, \ldots, X \times Y)$$

Where, the τ denotes the length of $Path_i$. During the executing of CLCCO, the pheromone of Map is updated dynamically and in real time.

4.3. Arithmetic Flow of CLCCO

For a more detailed presentation of CLCCO, this section presents the overall flow of the algorithm.

Algorithm1: Overall flow of CLCCO

Step1: Build a grid map for a workspace, remark the start cell S and destination cell D, set the value of pheromone for each cell (Eq. (2))
Step2: Initialize the number of cockroaches as I, the number of iterations of CLCCO as M. Set m=0 and i=0.
Step3: Execute the initializing search (Eq. (4)).
Step4: When finding the Q collusion-free paths, CLCCO finishes the initializing search.
Step5: According the Q collusion-free paths, CLCCO updates the pheromone of Map (Eq. (5)).
Step6: $i \leftarrow i+1$.
Step7: If i>I, then i =1.
Step8: The i-th cockroach executes the cooperative learning search (Eq.(7)).
Step9: If the length of $Path_i$ *is better than that of* P_i*, then the i-th cockroach return to the start cell,* P_i *is emptied and CLCCO jumps and executes Step6.*
Step10: If the i-th cockroach doesn't find the destination cell D, then CLCCO jumps and executes Step6.
Step11: If the length of $Path_i$ *is better than that of* P_i*, the i-th cockroach return to the start cell,* P_i *is emptied and CLCCO jumps and executes Step6.*
Step12: Update $Path_i$ *with* P_i *(Eq.(8)) and update the pheromone of the cells on* P_i*(Eq.(10)).*
Step13: If the length of P_{best} *is better than that of* $Path_i$*, the i-th cockroach return to the start cell,* P_i *is emptied and CLCCO jumps and executes Step6.*
Step14: Update P_{best}*with* $Path_i$ *(Eq.(8)).*
Step15: $m \leftarrow m+1$.
Step16: If $m \le M$*, then CLCCO jumps and executes Step6.*
Step17: Output P_{best}.

Step 1 to 4 belong to the initial search phase with time complexity $O(n^3)$. Step 5 to 17 belong to the cooperative learning search phase with time complexity $O(n^3)$. Therefore, the time complexity of CLCCO is $O(n^3)$.

5. Simulation Studies

To demonstrate the feasibility and effectiveness of CLCCO, a variety of experiments are carried out by computer simulation. Section 5.1 is the simulated experiments of CLCCO for the RPP problem. Section 5.2 gives the comparison among CLCCO, CCO and ACO. The computer configuration for experiments is that on Windows (32-bit versions) operations, Core(TM) i5-4300U CPU, 4GB memory. The programming language is Java.

5.1. Simulation Experiment on CLCCO

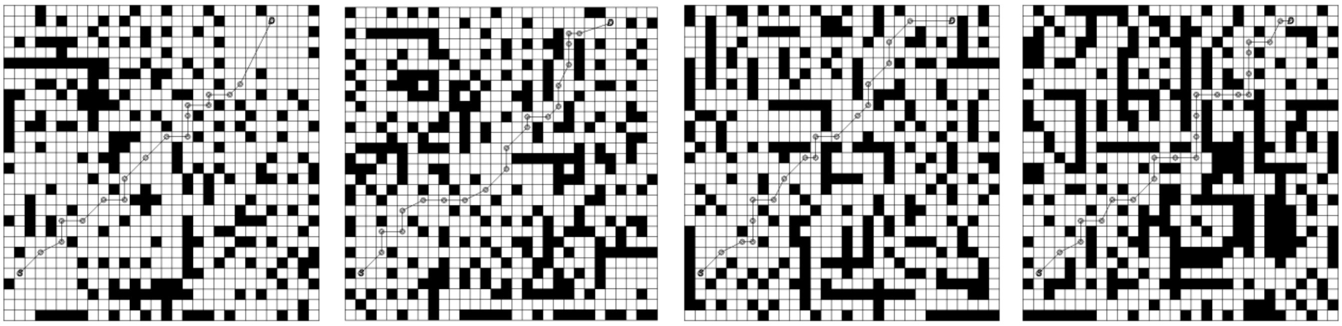
In this section, three groups of simulation experiments are conducted and twelve maps are chosen as workspace. To test the effectiveness of the algorithm under different conditions, all maps were divided into three groups according to the location of the start cell and the destination cell. The details are as follows:

1. Map(A) to (D) are set to start at $S(2,26)$ and target at $D(26,2)$.
2. Map(E) to (H) are set to start at $S(2,2)$ and target at $D(26,26)$.
3. Map(I) to (L) are set to start at $S(15,28)$ and target at $D(15,2)$.

The parameters of CLCCO are set as follow:

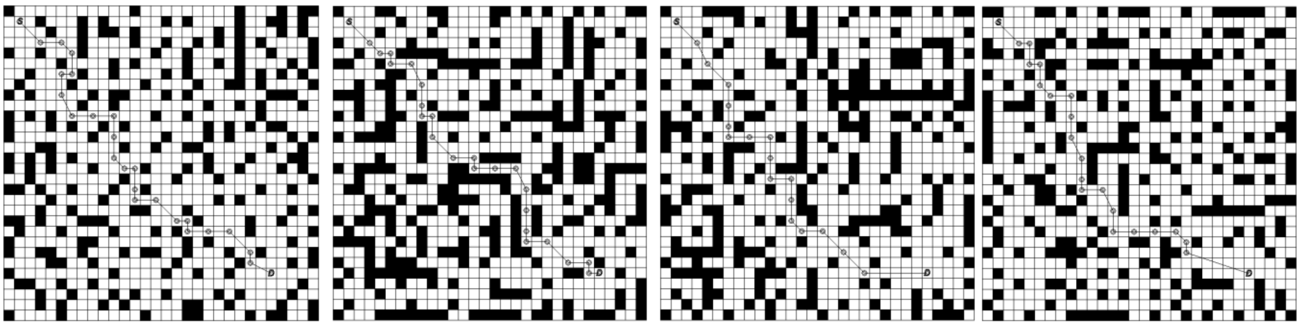
1. The number of cockroaches as I=20.
2. The number of iterations of CLCCO as M=50.

Figure 9, 10, and 11 show that the CLCCO algorithm has successfully planned a collision-free path for all the complicated workspaces. The symbols "◎" mark the planned route. By using the 16-geometry and multi-objective search, the planned path is hence shown to be smooth. Notice that cells marked only as "◎" need to be stored as path nodes. Therefore, the path planned by CLCCO needs to store less information, which is conducive to sending the path to the real robot.



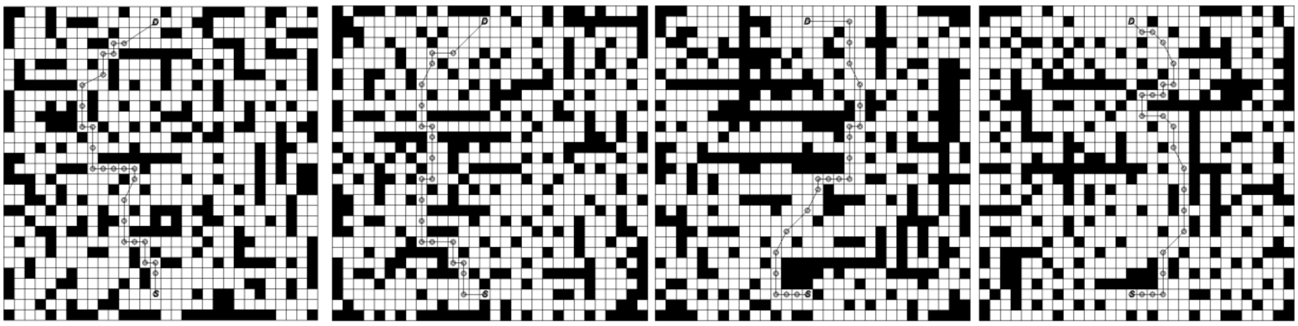
Map (A) Map (B) Map (C) Map (D)

Figure 9. Simulation experiment 1 (start cell: S (2,26) and destination cell: D(26,2)).



Map (E) Map (F) Map (G) Map (H)

Figure 10. Simulation experiment 2 (start cell: S (2,2); destination cell: D(26,26)).



Map (I) Map (J) Map (K) Map (L)

Figure 11. Simulation experiment 3(start cell: S (15,28); destination cell: D(15,2)).

5.2. Comparison between CLCCO and other Algorithms

In this section, the tests focus on the comparisons among CLCCO, CCO [2] and ACO [22]. CCO is the cockroach-inspired algorithm and ACO is a classic bionic algorithm. To make a fair comparison, the ACO and CCO algorithms were set to a group size of 20 and an iteration number of 50, respectively. Other parameter settings are derived from literature [2] and [22]. The twelve maps in section 5.1(Map (A) to (L))are as workspaces. Each algorithm is continuously executed 20 times. Table 1 to 3 show the detailed experimental data. In Table 1, Table 2 and Table 3, the symbol *"best"* and *"mean"* refer to the optimal and average path length respectively. The symbol *"std"* represents the standard deviation. Since the CCO algorithm is a deterministic search algorithm, and the path length of each experiment is the same. Therefore, for CCO, it is not necessary to calculate the values of standard deviation.

From the three data tables, we can find that CCO is superior to ACO in optimal and average path length. The main reasons are that CCO uses16- geometry and multi-objective strategies. These strategies are described in section 2.3. However, the determination of search strategy makes CCO lose its flexibility. It is why CLCCO is better than CCO in average and optimal path length. The value of standard deviation represents the stable performance of the CLCCO algorithm.

Figure 12 illustrates the convergence characteristics of map A, B, C, and D in terms of the mean performance of the total runs. The termination criterion is that the number of iterations reaches 50. Every five iterations, the current mean path lengths found by the algorithms are recorded. The average of all twenty runs of the algorithms are shown.

Table 1. Comparison experiment 1 (start cell: S(2,26) and destination cell: D(26,2)).

MAP	CLCCO			CCO			ACO		
	best	***mean***	***std***	***best***	***mean***	***std***	***best***	***mean***	***std***
(A)	39.67	40.25	0.61	41.34	41.34	—	42.46	43.53	1.09
(B)	40.77	40.94	0.19	42.46	42.46	—	44.94	45.33	0.44
(C)	40.03	40.38	0.50	42.43	42.43	—	43.46	43.69	0.30
(D)	43.37	43.75	0.39	46.17	46.17	—	48.79	49.67	0.97

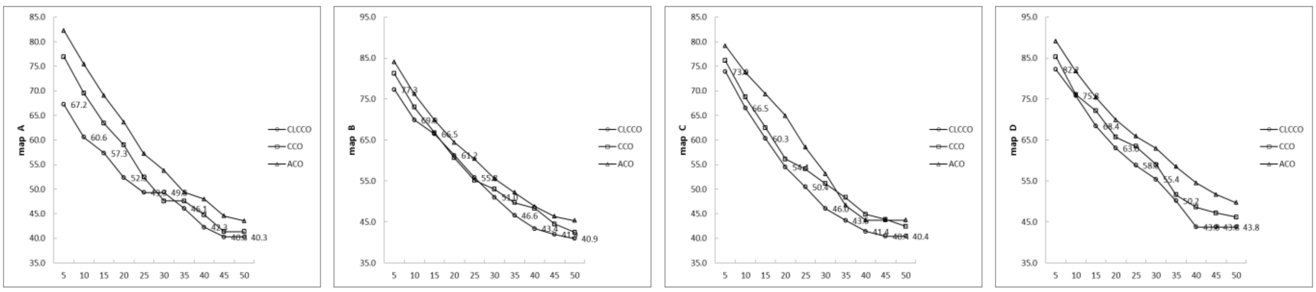
Table 2. Comparison experiment 3 (start cell: S(2,2) and destination cell: D(26,26)).

MAP	CLCCO			CCO			ACO		
	best	***mean***	***std***	***best***	***mean***	***std***	***best***	***mean***	***std***
(E)	43.79	43.92	0.34	46.63	46.63	—	48.87	49.10	0.20
(F)	42.37	42.66	0.25	44.46	44.46	—	45.54	46.46	0.77
(G)	41.96	42.01	0.09	44.34	44.34	—	45.85	46.28	0.52
(H)	42.45	42.57	0.21	46.75	46.75	—	49.84	49.99	0.15

Table 3. Comparison experiment 2 (start cell: S(15, 28) and destination cell: D(15, 2)).

MAP	CLCCO			CCO			ACO		
	best	***mean***	***std***	***best***	***mean***	***std***	***best***	***mean***	***std***
(I)	39.08	39.23	0.19	43.18	43.18	—	46.26	46.49	0.44
(J)	37.48	37.70	0.28	42.81	42.81	—	43.64	44.66	0.87
(K)	38.54	38.74	0.25	40.77	40.77	—	42.65	43.27	0.63
(L)	37.54	37.87	0.39	42.27	42.27	—	43.25	43.58	0.39

Figure 12 illustrates that the convergence rate of CLCCO is generally better than that of CCO and ACO. The reason is that technology of cooperative learning search plays an important role. Because 16- geometry and multi-objective strategies are used, the convergence speed of CCO is better than that of ACO. Therefore, cooperative learning search, 16- geometry and multi-objective are highly efficient robot road force planning strategies.



a) Convergence Graph for map A.

b) Convergence Graph for map B.

c) Convergence Graph for map C.

d) Convergence Graph for map D.

Figure 12. Median convergence characteristics of CLCCO, CCO and ACO on map A, B, C, and D.

5.3. Compression Path Method

Notice that the performance of CLCCO to plan the path has been validated in sections 5.1 and 5.2. Whether an optimal path can be executed correctly by real robot depends largely on the mechanical control of real robot. The research objective of this paper is to provide the planned path information to the robot, but not the mechanical control.

In general, the memory resources of robot are limited, but it requires high speed of path information transmission. In order to save storage resources and improve the speed of information transmission, we design an information compression method.

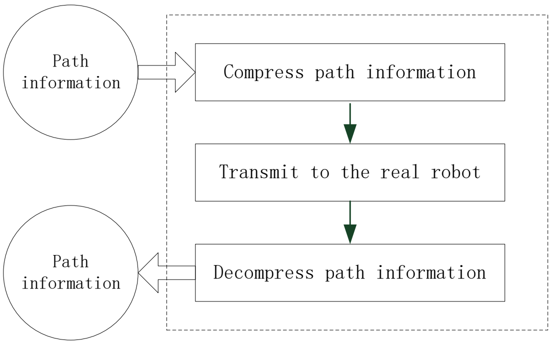


Figure 13. Path compression transmission

The path compression transfer process is shown in Figure 13. Path compression is achieved by Equation (11).

$$\varphi = (c_\theta^{(y)} - 1) \cdot X + c_\theta^{(y)} \quad (11)$$

Here, c_θ denotes any cell on the optimal path. $c_\theta^{(y)}$ is the y-coordinate of cell c_θ in grid map. X is the number of cells in a row. By Equation (11), the two-dimensional coordinate is compressed as an integer φ. Path decompression is implemented by Equation (12).

$$\begin{cases} c_\theta^{(x)} = \left((\varphi - 1) \bmod X\right) + 1 \\ c_\theta^{(y)} = \lfloor (\varphi - 1) / X \rfloor + 1 \end{cases} \quad (12)$$

Here, $c_\theta^{(x)}$ is the x-coordinate of cell c_θ in grid map. The symbol "mod" represents the modulus operation and the symbol "$\lfloor \ \rfloor$" means round down. Equation (12) is going to convert the compressed path information φ to the two-dimensional coordinates of the original path.

The information compression method is to compress the binary coordinate (x, y) into a single value integer. This can save up to 50% of storage space. This compression path method has been tested with a real robot and a grid workspace (See Figure 14).

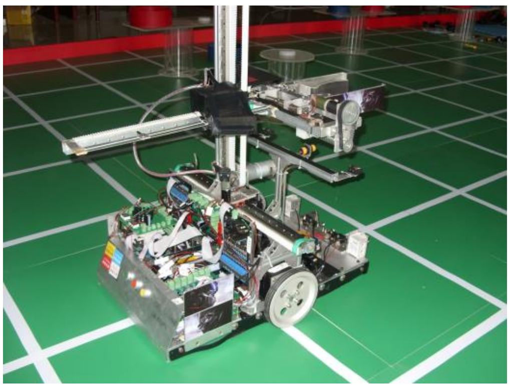


Figure 14. Mobile robot in real workspace.

The workspace was randomly blocked and the route of the real robot was computed by CLCCO algorithm. The real environments are simpler than the maze grid maps in the previous simulation experiment. In most cases, real robots can find shortest paths by avoiding obstacles. It proves that CLCCO algorithm can be applied to real robots.

5.4. Discussions on CLCCO

According to the testing results, we can performance the analyses on CLCCO as follow:

1. The process of initializing search changes the distribution of pheromone that is the crucial information of cooperative learning search.
2. The pheromone value is inversely proportional to the path length. By cooperative learning, the cockroach individual tends to choose the position with bigger pheromone value as the next cell.
3. In essence, the cockroach individual can move and search along the relatively good path by cooperative learning, which can make the cockroach individual find a better path.
4. The pheromone is always dynamically updated during the execution of CLCCO, which means that all cockroaches learn from each other.
5. Because of the application of cooperative learning, CLCCO need only set two parameters, that increases the level of controllability of CLCCO.
6. Path compression technology makes CLCCO algorithm control real robot more real-time.

Overall, CLCCO can be applied to the RPP problem with good performance.

6. Summary and Conclusions

By introducing the strategy of cooperative learning, a novel CLCCO algorithm for RPP problem is presents in this paper. The idea of CLCCO is mainly based on the strategy of cooperation and machine learning. Some new methods, such as Fibonacci transformation and multi-objective search, etc., are proposed. Especially, this paper presents a new Compression Path Method, which can save 50% of the storage space. The CLCCO algorithm only needs two parameters. By the comparisons with state-of-the-art cockroach-inspired algorithm, CLCCO demonstrates the high-performance. Among the possible perspectives, the CLCCO algorithm will be used for the workspace with weight regions or be extended in volume to solve the RPP problem for 3-D workspace.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.51975239), the Ministry of Education Research of Social Sciences (No.17YJC790002), the National Planning Office of Philosophy and Social Science (Grant Nos. 19BJY255), the Natural Science Foundation of Jiangsu Province

(No. BK20191214), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No.20KJA120001), the Qing Lan Project, the Huaian City Science and Technology Plan Project (No.HAB202070, No.HAP201909), the Innovation Foundation of Jiangsu Vocational College of Electronics and Information (No.JSEIYY2020004, No.JSEIYQ2020002, HABL202129).

References

[1] Ame J., Halloy J., Rivault C., Detrain C., and Deneubourg J., "Collegial Decision Making Based On Social Amplification Leads to Optimal Group Formation," *Proceedings of the National Academy of Sciences*, vol. 103, no. 15, pp. 5835-5840, 2006. doi: 10.1073/pnas.0507877103.

[2] Cheng L., Han L., and Zheng X., "Adaptive Cockroach Colony Optimization for Rod-Like Robot Navigation," *Journal of Bionic Engineering*, vol. 12, no. 2, pp. 324-337, 2015.

[3] Cheng L., Chang L., Song Y., Wang H., Xu Y., and Bian Y., "A Bionic Optimization Technique with Cockroach Biological Behavior," *Chinese Journal of Electronics*, vol. 30, no.4, pp. 644-651, 2021. https://doi.org/10.1049/cje.2021.05.006

[4] Cheng L., Song Y., and Bian Y., "Cockroach Swarm Optimization Using a Neighborhood-Based Strategy," *The International Arab Journal of Information Technology*, vol. 16, no. 4, pp. 784-790, 2019.

[5] FeiXiang X., XinHui L., Wei C., Chen Z., Bing-Wei C.,"Fractional Order PID Control for Steer-by-wire System of Emergency Rescue Vehicle Based on Genetic Algorithm," *Journal of Central South University*, vol. 26, no. 9, pp. 2340-2352, 2019.

[6] Friedrich T., Kötzing T., and Krejca S., and Sutton A., "Robustness of Ant Colony Optimization to Noise," *Evolutionary Computation*, vol. 24, no. 2, pp. 237-254, 2016. DOI: 10.1162/EVCO_a_00178

[7] Halloy J., Sempo G., Caprari G., Rivault C., Asadpour M., Tâche F., and et al., "Social Integration of Robots Into Groups of Cockroaches to Control Self-Organizined Choices," *Science*, vol. 318, pp. 1155-1158, 2007. doi: 10.1126/science.1144259.

[8] Hu W. and Yen G., "Adaptive Multiobjective Particle Swarm Optimization Based on Parallel Cell Coordinate System," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp.1-18, 2015. DOI: 10.1109/TEVC.2013.2296151

[9] Le C., "New Bionic Algorithm: Cockroach Swarm Optimization," *Computer Applications in Engineering Education*, vol. 44, no. 34, pp. 44-46, 2008.

[10] Liang Y., Jiang P., and Xu J., and Wu M., "Initial Alignment of Compass Based on Genetic Algorithm-Particle Swarm Optimization," *Defence Technology*, vol. 16, no. 1, pp. 257-262, 2020. https://doi.org/10.1016/j.dt.2019.08.001

[11] Lin Q., Liu S., and Zhu Q., Tang C., Song R., and Chen J., "Particle Swarm Optimization with a Balanceable Fitness Estimation for Many-objective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32-46, 2018.

[12] Obagbuwa C., Adewumi O., and Adebiyi A., "Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Function Problems," *Mathematical Problems in Engineering*, vol. 2014, no. 1, pp. 1-12, 2014.

[13] Obagbuwa I. and Abidoye P., "Adaptive Cockroach Swarm Algorithm," *in Proceedings of the International Conference on Numerical Analysis and Applied Mathemati*cs, Rhodes, pp. 1-7, 2017.

[14] Tsai C., "Roach Infestation Optimization with Friendship Centers," *Engineering Applications of Artificial Intelligence*, vol. 39, no. 7, pp. 109-119, 2015.

[15] Watanabe H., Mizunami M., Rustichini A., "Pavolv's Cockroach: Classical Conditioning of Salivation in an Insect," *PloS One*, vol. 2, no. 6, pp. 521-529, 2007. https://doi.org/10.1371/journal.pone.0000529

[16] Xia X., and Zhou Y., "Performance Analysis of ACO on the Quadratic Assignment Problem," *Chinese Journal of Electronics*, vol. 27, no. 1, pp. 26-34, 2018. https://doi.org/10.1049/cje.2017.06.004

[17] YongHuan M., Bo Q., and ShiYa W., "Regression Prediction of Photometric Redshift Based on Particle Warm Optimization Neural Network Algorithm," *Spectroscopy and Spectral Analysis*, vol. 39, no. 9, pp. 2693-2697, 2019.

[18] Yuan H., YuQing Z., and GuangHua Z., "Android Driver Vulnerability Discovery Based on Black-Box Genetic Algorithm," *Chinese Journal of Computers*, vol. 40, no. 5, pp. 1031-1042, 2017.

[19] Yuan Q., Huang W., and Li R., "Dynamic Fusion of Artificial Fish-Swarm Algorithm and Cockroach Swarm Optimization with Differential Evolution Mutation and its Application in Grid Task Scheduling," *Computer Applications and Software*, vol. 29, no. 5, pp. 175-177, 2012.

[20] ZhaoHui C. and HaiYan T., "Cockroach Swarm Optimization," *in Proceedings of the IEEE International Conference on Computer Engineering and Technology*, Chengdu, China, pp. 653-655, 2010. DOI:10.1109/ICCET.2010.5485993

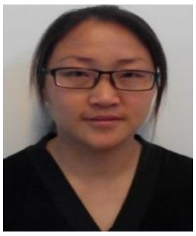
[21] ZhenYue L., YongGe W., and XiaoHui H., "A Genetic Algorithm for Stress Tensor Inversion

and its Application to The Northeast Margin of the Tibetan Plateau," *Chinese Journal of Geophysics*, vol. 63, no. 2, pp. 562-572, 2020.

[22] Zhu Q., "Ant Algorithm for Path Planning of Mobile Robot in Complex Environment," *Acta Automatica Sinica*, vol. 32, no. 4, pp. 586-593, 2006.

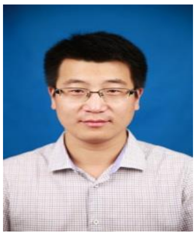
Le Cheng received his Ph.D. degree in computer and information from Hohai University, China in 2015. He is currently a professor at the College of Computer Science and Communication, Jiangsu Vocational College of Electronics and Information, China. His research interests include evolutionary algorithms, motion planning, and machine learnin.

Lyu Chang received his Ph.D. degree from Jilin University, Jilin, China. He is currently a Professor with the Department of Computer Science and Engineering, Huaian Vocational College of Information Technology, Huaian, China. His current research interests include evolutionary algorithms and machine learning. He has hosted and participated in a number of project sponsored by the Natural Science Foundation of China.



Yanhong Song is currently an Assistant Professor at the College of Computer Science and Communication, Jiangsu Vocational College of Electronics and Information, China. Her research interests include motion planning and machine learning.

Haibo Wang is currently a professor at Jiangsu Industrial Cloud Edge Collaborative Technology Engineering Research Center, China. His research interests include evolutionary algorithms and machine learning.



Yuetang Bian received his Ph.D. degree from Southeast University, Nanjing, China. He is currently an Associate Professor with the Department of Management Science, School of Business, Nanjing Normal University, Nanjing, China. His current research interests include network theory and methods, Optimization Theory and Social behavior in Networks. He has hosted and participated in a number of project sponsored by the Natural Science Foundation of China.