Improved Robotic Mapping with Deadlock Solution Based on Ant Colony Algorithm

Jin Zhou[1,2,a], Michael Biney[1,b], Xu E[1,2,c,*], Chunxiao Liu[1,d] and Yanhong Li[1,e]

[1]*College of Information and Technology, Bohai University, Jinzhou, PR China*
[2]*College of Food science and Engineering, Bohai University, Jinzhou, PR China*
a. zhoujin@bhu.edu.cn, b. michael.biney@rocketmail.com, c. exu21@163.com
d. xiaoxiao198525@163.com, e. zcfl613@126.com
**corresponding author: Xu E*

Keywords: Deadlock, mapping, ACO, stigmergy, retraction.

Abstract: Deadlock in robotics can be a frustrating problem if ignored. It is described as a halt state where two threads wait on each other to release a resource. The recent rise in the use of ant colony algorithms in solving problems is what inspired this paper. Several heuristic methods for finding a solution to this problem have been designed hence the improvement of the ACO algorithm. This paper is devoted to solving deadlock problems in robotic mapping with the ant colony algorithm by improving on the traditional ACO to achieve efficient recovery techniques during robotic mapping. It also outlines the use of mathematical functions of the ACO to influence robots in their probabilistic decision making in a deadlock state. Results from this paper prove that the improved ACO with the retraction mechanism is more efficient and increases the differences of results and is useful to discover the ideal way.

1. Introduction

Recently, the deadlock problem has gained a more topical value in many fields of study. The deadlock problem has been a significant issue in mapping, along with a few others. It is impossible for moving ants not to have jams, likewise mobile robots. The reason is when speed ($v=d/t$) and density (p) hit tipping point jams are inevitable. Therefore this situation is directly dependent on the solution to this problem. In this work, the possibility of applying an improved ant colony algorithm to various deadlock problems is considered. A few endeavors have been made to fathom the deadlock, some of which are still finding the best solution to the problem. According to (frontiers 2019), [2] the problem can be best solved using an improved ant colony algorithm and the MAX-MIN ant systems in a constructed grid environment model. According to (Reactive Mission and Motion Planning with Deadlock Resolution), [1] the idea of using linear temporal logic to improve the mission and motion planning of robots that will be performing a task in a complicated environment with moving obstacles. A few others like Ahmed Saleh, Guido Maione, and John Fredrick Chiong lo all have made contributions, mostly tackling the problem at the base level by outlining a few detection techniques and preventions. This paper adds to the pool of knowledge of robotics and ACO in an unending cycle of updates and improvements. As we still journey on into a more sophisticated era of computing and technology, note that every idea and addition is not little but can be harnessed for the future of robotics. The ant colony optimization was proposed by an Italian scientist named Marco Dorigo as part of his Ph.D. in 1992. Furthermore, the first ant-inspired algorithm was called the ant system, but these days we use an improved version of the algorithm, which is now called the ant colony optimization algorithm. The main inspiration from the ant colony optimization comes from stigmergy. This is the interaction and coordination of organisms in nature by modifying the environment. In stigmergy, the trace left in the background by a body stimulates the performance of the next body by the same or different agent.

2. Fundamentals

2.1.Deadlock

Models of the environment are required for a vast extent of automated applications, from search and rescue to robotize vacuum cleaning. Robots that can acquire an accurate model of their environment on their own are regarded as fulfilling a significant precondition of genuinely autonomous agents.

The mapping issue is, by and large, respected as one of the fore most critical issues within the interest of building genuinely independent, versatile robots. When a robot or an ant is found in a complex environment or an unstructured mapping environment, there is the likelihood of a deadlock situation. A deadlock is two threads waiting on each other to release a resource. When more of the ants are trapped in a deadlock type of situation, the rate of reaching the optimal goal is severely reduced, resulting in a decrease in the diversity of solutions. When related to mobile robots, it takes much time to kick out of the situation.

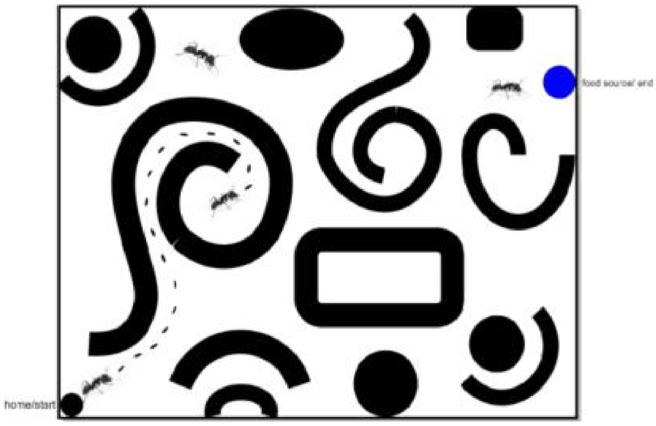

Figure 1: A scene of a deadlock situation.

Figure 1, the above figure, explains the deadlock situation. When ant x leaves home or a robot is set to start from the starting point x to go search for food or retrieve the blue ball, it seeks to find the shortest possible route. In the case of an ant, when caught in a deadlock like in Figure 1, it applies a natural mechanism called the retraction mechanism by moving a step back and then further back until it is out of the halt state. This situation can be bit frustrating for ants, especially in a complex environment with movable and non-movable obstacles. If ant x does not escape the deadlock on time other ants will catch up using the pheromone trails left behind by the initial ant, and then collision and overcrowding will take place. It increases the mortality rate of the ants. Since the ants and robots use stigmergy, their actions are a bit probabilistic; therefore, they can be altered. In the case of the mobile robot, where its task is to retrieve the blue ball, it communicates with its environment with sensors until it retrieves the ball. However, what if it also finds itself in a deadlock, without the natural response of the ants it will get stuck until it manually moved or the obstacle moved if in a simulated environment. Nevertheless, what this paper seeks to do is to apply the same natural response used by the ants to kick out from a halt in mapping in robotics.

2.2.The ACO Algorithm

ACO algorithm

```
Begin;
   Initialize the pheromone trails and parameters;
      Generate population of m solutions (ants);
       For each ant k∈m:  calculate fitness (k);
      For each ant determine its best position;
     Determine the best global ant;
     Update the pheromone trail;
   Check if termination = true;
End;
```

The Ant Colony Algorithm

Natural ants can find the shortest path from a food source to their nest, without using visual cues by exploiting pheromone data. Whereas strolling, ants store pheromone on the ground and take after in the probability of the pheromone previously deposited by other ants. Ants arrive at a choice point in which they ought to choose whether to turn left or right. Since they have no clue approximately, which is the most excellent choice, they select arbitrarily. It can be anticipated that, on average, half of the ants choose to turn left and the other half to turn right. This happens both to ants moving from cleared out to the right and those moving from right to left. The decision making the process of the ants and in this case, will be applied to a mobile robot and the use of pheromone simulation influenced by time to make a decision and to increase the probability of choosing the right path and backing out from a deadlock situation. This is the mathematical model used to represent the pheromone model on a graph.

$$\Delta\tau_{i,j}^{k} = \left\{ \frac{1}{Lk_n} \right. \quad (1)$$

So basically, delta tau $\Delta\tau$ shows the amount of pheromone that an ant deposits. Bear in mind an ant walks on a graph and so it moves from one point to the next, and that is the reason why the subscripts *i* and *j* are needed. *i* and *j* show the edge connecting the node *i* to the node *j,* and *k* is the *kth* ant. This is equal to 0 if the ant does not move to any of the edges. For example, if the *kth* ant refuses to move, then the pheromone level becomes 0. Otherwise, it is 1divided by Lk, which is the length of the path by the *kth* ant. Note that the reason for dividing by 1 is that the shorter the path, the higher the pheromone should be deposited by the ant. Seeing that we have multiple ends, how do we now calculate the amount of pheromone on each edge?

$$\tau_{i,j}^{k} = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \text{ (without evaporation)} \quad (2)$$

The $\sum$ *of k=1* to m where *m* = the total number of ants, delta tau *i* and *j,* and the superscript *k.* This is without evaporation because the pheromone is added over time, but if the evaporation is to simulated, then add the equation below.

$$\tau_{i,j}^{k} = (1-\rho)\,\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \text{(withevaporation)} \quad (3)$$

1- ρ Multiplied by the current pheromone $(\tau_{i,j})$ level plus the new pheromone level that should be deposited by all ants. Rho ρ is a constant that helps you to define the evaporation rate. When ρ = 0 there is no evaporation because 1- 0 multiplied by $\tau_{i,j}$. When ρ = 1, then it would be 1*1 multiplied by Δ. That is where the evaporation is at the maximum level. Let us take a look at the idea between the pheromone level and the evaporation rate.

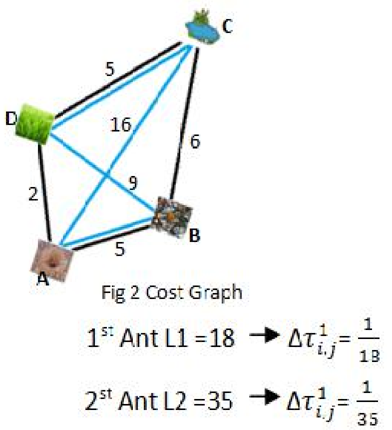

Figure 2: Cost Graph.

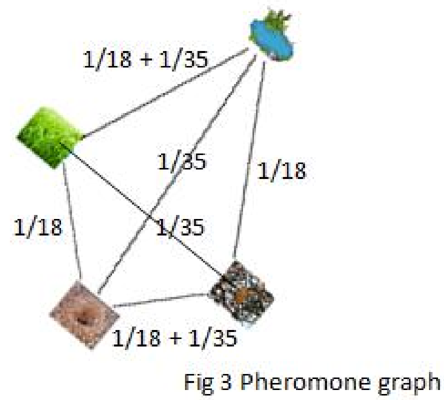

Figure 3: Pheromone graph.

Figure 2 indicates the paths taken by both the blue and black ants. The black lines indicate the steps of the 1st ant, and the first step is to calculate the length of those paths. Therefore L1: 5+2+5+6 $\Delta\tau^1_{i,j}$ =1/18 and L2: 5+9+16+5 $\Delta\tau^1_{i,j}$=1/35. Now how do you add that to your pheromone graph?

Areas that were traveled by the 2nd ant with the blue pheromone trails are where you add 1/35. On the contrary, the lines with black pheromone trails must add up 1/18, which has already been calculated. In the areas where you find both blue and black pheromone trails depicting that both ants traveled on them, you add both 1/18 and 1/35, using the formular below.

$$\tau_{i,j} = \sum_{k=1}^{m} \Delta\tau^k_{i,j} \tag{4}$$

Therefore if an ant is to move from the ant hive from Figure 3, it will likely move to the closest destination (the tree) because of the high pheromone concentration. Consider an initial pheromone level of 1 deposited on each of the edges. Using the formular:

$$\tau_{i,j} = (1-\rho)\,\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau^k_{i,j} \tag{5}$$

ρ=0.5, meaning every time 0.5%of the pheromone level evaporates. Therefore the paths taken by the black ant will be:

Distance from A-B; 0.5*1+1/8+1/35
Distance from B-C; 0.5*1+1/8
Distance from C-D; 0.5*1+1/8+1/35
Distance from D-A; 0.5*1+1/18
Distance from D-B; 0.5*+1/35
Distance from A-C; 0.5*+1/35

So to get the idea, evaporation removes a little bit of pheromone and adds a little bit depending on the graphs or edges traveled by the ant.

2.3.Calculating the Probabilities

This step, which is the most crucial thing, is to use the pheromones calculated in the first step to choose a path. As I have discussed before, ants use probabilities to decide on the shortest path and also to kick out of a deadlock situation. Below is an equation to simulate this movement.

$$P_{i,j} = \frac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum((\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta})} \text{Where: } \eta_{i,j} = \frac{1}{L_{i,j}} \quad (6)$$

This means that the probability of choosing the edge i and j is equal to $(\tau_{i,j})^{\alpha}(alpha)$ multiplied by $(\eta_{i,j})\beta$, divided by the sum ($\sum$) of the same equation in the denominator. Eta (η) indicates the *i,j* edge on the map. With the parameters alpha and beta, the impact of tau (τ) or eta (η) can be increased or decreased in the process of decision making. The denominator of this equation as the pheromone and quality of all edges that can be considered on edge *i.* this probability of all the edges is calculated for all the edges connected to the current node and is usually a number between the number of 0 and 1. If you want to make a decision based on the pheromone level, then you can remove eta (η) from the equation. Since we are interested in the shortest path, $\eta_{i,j}$ indicates that the length or cost of an edge how right it is in the process of calculating the probability of that edge.

Below is a numerical example to better understand the process of calculating the probability of choosing each edge on a graph.

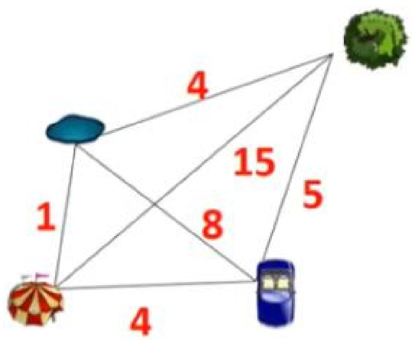

Figure 4: Cost graph.

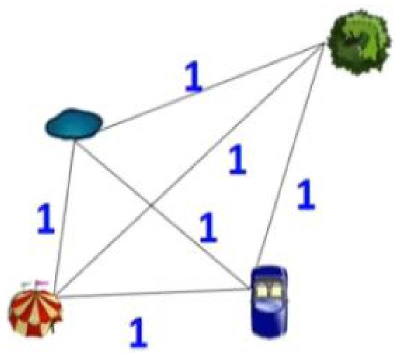

Figure 5: Pheromone graph.

For example, in case there is a dark ant that is going to make a decision on which paths to take; therefore, the probability of choosing the pond:

$$P=\frac{1\times\frac{1}{1}}{(1\times\frac{1}{1})+(1\times\frac{1}{15})+(1\times\frac{1}{4})}=0.7595$$

Probability of choosing the car:

$$P=\frac{1\times\frac{1}{4}}{(1\times\frac{1}{1})+(1\times\frac{1}{15})+(1\times\frac{1}{4})}=0.1899$$

Probability of choosing the tree:

$$P=\frac{1\times\frac{1}{15}}{(1\times\frac{1}{1})+(1\times\frac{1}{15})+(1\times\frac{1}{4})}=0.0506$$

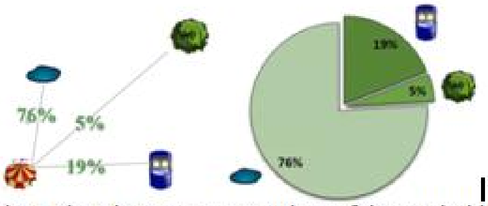

Fig 6 Pie Chart representation of the probabilistic calculation

Figure 6: Chart representation of the probabilistic calculation.

This means that the ant is likely to choose the pond over the tree and the car. Judging from the pie chart, it is obvious the ant did not go to the tree because it had a stronger connection weight than the car and pond. The ant is likely to go to the pond because of its short distance. Finally, 76% shows a good indication of the impact of the pheromone and the cost of calculating the probability of choosing one of the destinations. To alter the simulation process, the pheromone level can be changed from one to any number of did not a choice, and the result is also going to change, indicating that the most significant percentage to become the shortest path taken by the ant. This means that ants consider the quality and the amount of pheromone deposited on a path. The ultimate question now is how to use probabilities to mathematically choose a destination? Using a technique called the roulette wheel, this can be achieved. Using the probabilities from the previous simulation (0.76, 0.19 & 0.05), calculate the cumulative sum, as the first step.

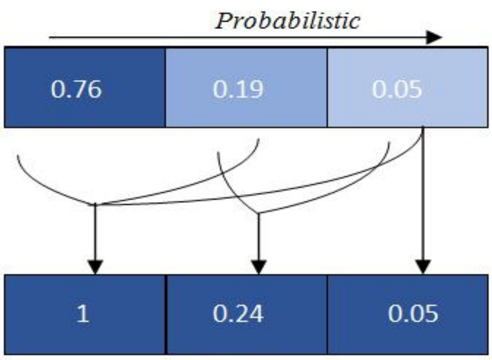

Fig 7 Cumulative

Figure 7: Cumulative.

To utilize the total chat to choose the goal, a random number (r) between [0, 1] is used. Therefore if the number is between $\{0.24< r \leq 1.00\}$, then the ant will choose the pond; otherwise, if it is between $\{0.05< r \leq 0.24\}$, then the ant chooses the car. On the other hand, if it is between $\{0.00< r \leq 0.05\}$, then it chooses the tree.

3. Improved Robotic Mapping with Deadlock Solution Based On Ant Colony Algorithm

The conventional ACO has taking after inadequacies: the robot may drop into a deadlock state in which the robot cannot move to the surrounding networks. Within the network outline, the conventional ACO may have more difficult times in a deadlock. Therefore, this paper makes the following changes. Handle deadlock. Concurring to the taboo table, it is the judge whether in a halt state. In case the ants are in a halt state, the withdrawal component will be adopted-complicated, and the ant or robot will withdraw a step back. If, after withdrawing, it still finds itself in a deadlock, it will still withdraw until it's clear of deadlock and then can now find the optimal path and then after update the pheromone trail.

Algorithm: Find deadlock and extract robot from the halt.

```
check if deadlock = true
    while (deadlock)
        move back steps plus 1
    determine a new path
```

Improved ACO

```
Begin;
  Initialize the pheromone trails and parameters;
    Generate population of m solutions (ants);
    For each, ant k∈m:  calculate fitness (k);
    For each ant determine its best position;
    Determine the best global ant;
    Check if deadlock = true
        while (deadlock)
            Move back steps plus 1
            Determine a new path
    Update the pheromone trail;
  Check if termination = true;
End;
```

4. Experimentation

Robots act in the same way as honeybees when choosing their nest, therefore when the robot environment becomes too is not, movement becomes sophisticated, especially in a deadlock. Robots locally interact with their environment and each other, and they display their opinion through LEDs. Undecided robots may discover an option and commit to it. How about when it is unable to come up with such discovery? The ACO algorithm has proven to be one of the best algorithms when it comes to its problem-solving capabilities. Wang and Yu (2011) [8], [14] received the new passing technique, which erased the ants caught in a deadlock state from the ant colony and did not upgrade the global pheromone. In any case, when more of the ants are caught within the deadlock state, the number of ants that can reach the objective is substantially diminished, which results in a decrease within the differing qualities of solutions and, conducive to the look of ideal way for ants.

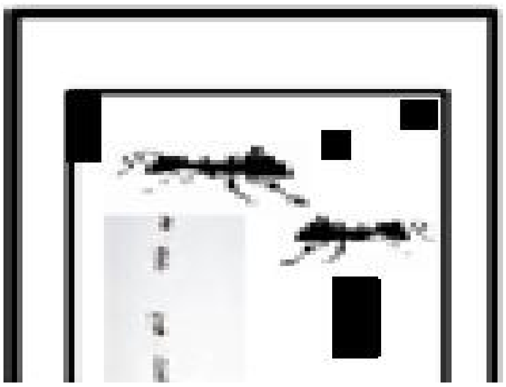

Figure 8: Shows an image of ants trapped in a deadlock.

If the ant moves from edge m to edge n, the next optional point is edge s; at this time, the insect is caught in a halt state, and it cannot move to its encompassing framework. This paper adopts the retraction mechanism. As shown in the figure above, the ant, which has strolled into the grid, is caught within the deadlock state, and the improved technique permits the ant to roll back one step and upgrades the grid table data. In case the robot is still caught into a deadlock state, the robot will proceed to rollback until grid*s*. At this minute, the ant or robot gets away from the deadlock zone. Since the deadlock edge may be the portion of the ideal global way, no pheromone update is carried out on the deadlock edge. The withdrawal mechanism cannot avoid ants or robots from entering a deadlock state, but it lets the halted ants or robot return back to the previous grid until there is an available grid around the ants.

According to (Stützle and Hoos, 2000), [16] this problem can also be solved using the MAX-MIN ant system. With this system, the pheromone trail is updated. The pheromone is into upgrade history after each emphasis trial in the conventional ant colony algorithm. Whereas within the MMAS, only the path pheromone of the ideal network is upgraded after the cycle is completed. Appropriately, the adjusted pheromone path upgrade rule is expressed by:

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \tau_{i,j}^{best} \quad (7)$$

$$\Delta\tau_{i,j}^{k}(t) = \frac{Q_1}{L^{best}} + \frac{Q_3}{C_{turns}^{best}} \quad (8)$$

$$\frac{Q_3}{C_{turns}^{best}} = \text{1Cals (l) + w2 turns (l)} \quad (9)$$

$$\text{w1} = \frac{V_{robot}}{W_{robot}} \quad (10)$$

$$\text{w2} = V_{robot} \times t_a \quad (12)$$

Where Q3 may be a constant more than 1.Lbest denotes to the briefest way right now found by the calculation, Cals (l) speaks to the whole of all the points of turning on the most excellent optimized way. Turns (l) are the whole of the turns on the most excellent optimized way. W1 and w2 speak to distinctive weight coefficient and are set by analyzing the robot's structure and kinematics (Wu et al., 2013; Li et al., 2017). [8], the w1 and w2 can change the overturning point and turning times into grid length, individually. Vrobot speaks to the steady speed of a portable robot. Wrobot speaks to the precise speed of a portable robot because it turns. Ta speaks to the time of increasing speed and deceleration as the portable robot turns once. The improved ACO is better by reason of considering how quickly the sequence converges. If put side by side to the traditional ACO, the improved ACO has a higher percentage of about 60% in the number of iterations and about 40% in reducing bends.

5. Results

This paper takes a bold step towards modifying the ACO by applying the retraction mechanism in complicated maps for the portable robot, especially in the retraction of robots in a deadlock situation; by utilizing the retraction mechanism and the improved algorithm during mapping planning of the robot. The issue of ant deadlock is unraveled and the calculative probability is improved. Results of the ACO with retraction mechanism and ACO without retraction mechanism run on different iteration maps below proves the efficiency of the improved ACO in different simulations. The results below indicate the defined movement of the robot with and without retraction mechanism and whether or not it is been run on the improved ACO.

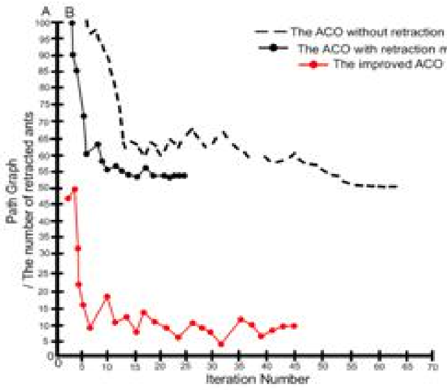

Figure 10: Performance comparison on a trough map.

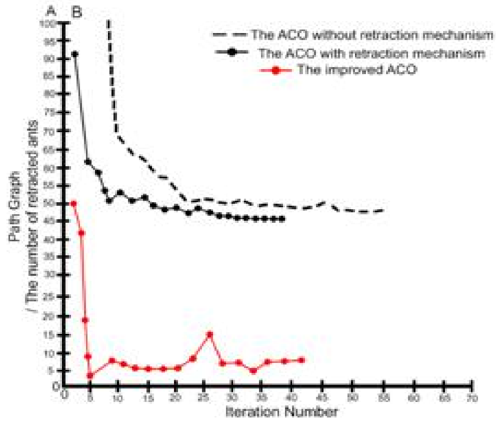

Figure 11: Performance comparison on a baffle map.

6. Conclusions

ACO with the withdrawal mechanism has higher proficiency and less emphasis than ACO without the withdrawal component. When ants or robot drops into a deadlock state, the withdrawal mechanism is utilized to replace the new passing procedure, which avoids a vast number of ant deaths in one cycle. Subsequently, each insect can get away by utilizing the retraction component, which increases the differences of results and is useful to discover the ideal way.

Acknowledgments

This work was supported by National Key R&D Program of China under Grant No.2019YFD0901605, and by National Natural Science Foundation of China under Grant No.61602056, and by Natural Science Foundation of Liaoning Province of China under Grant No. 20170540005, and by Social Science Foundation of Liaoning Province of China under Grant No. L19BGL016, and by Doctoral Research Initiation Foundation of Liaoning Province of China under Grant No. 201601353, and by Scientific Research Foundation of Department of Education of Liaoning Province of China under Grant No. LZ2016005&LQ2017002 and the robotics club of Bohai University.

References

[1] Reactive Mission and Motion Planning with deadlock Resolution. (n.d.). 1-23.
[2] S.A, F. M. (2007 - 2019). Articles. Retrieved November Tuesday, 2019.
[3] A. A. Kazharov and Kh. A. Kazharov, "Genetic Methods of Solving Vehicle Routing Problem," inThe Youth and Modern Information Technologies2008 (Tomsk, 2008) [in Russian].
[4] S. D. Shtovba, "Ant Algorithms," Exponent Pro, Matematika v Prilozheniyakh, No. 4, 70–75 (2003).
[5] F. Bonavear and M. Dorigo, Swarm Intelligence: From Natural to Artificial Systems (Oxford University Press, Oxford, 1999).
[6] D. Corne, M. Dorigo, and F. Glover, New Ideas in Optimization (McGravHill, 1999). M. Dorigo, Website of Marko Dorigo on Ant Optimization, http://iridia.ulb.as.be/dorigo/ASO/ASO.html.
[7] Analysis of Algorithms: An Active Learning Approach (Jones & Bartlett Publishers, 2001; Tekhnosfera, Moscow, 2004).
[8] Wu, J., Li, T., and Xu, B. (2013). Force optimization of planar 2-DOF parallel manipulators with actuation Redundancy considering deformation. Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.227, 1371–1377.

[9] A. A. Kazharov, "A Modification of Ant Algorithms and their Application to the ravelling Salesman Problem," in Microelectronics and Informatics (MEPhI, Moscow, 2008) [in Russian].L. Gladko V.M. Kureichik, and V. V. Kureichik, Genetic Algorithms (OOO "Rostizdat," at Don, 2004) [inRussian].
[10] Li, W. H., Yang, C. G., Jiang, Y. M., and Su, C. Y (2017). Motion planning for Omnidirectional wheeled mobile robot by potential field method. J.Adv. Transport. 3, 1–11.
[11] Stützle, T., and Hoos, H. H. (2000). MAX-MIN ant system. Fut. Gen. Comp. Syst. 16, 889–914.
[12] He, W., Dong, Y., and Sun, C. (2016b). Adaptive neural impedance control of a Robotic manipulator with input saturation. IEEE Trans. Syst. Man Cybernet. Syst. 46, 334–344.