Al-Khwarizmi Engineering Journal,Vol. 12, No. 4, P.P. 12- 25 (2016)

Al-Khwarizmi Engineering Journal

Path Planning of an autonomous Mobile Robot using Swarm Based Optimization Techniques

Ibraheem Kasim Ibraheem* **Fatin Hassan Ajeil****

** ,** Department of Electrical Engineering / College of Engineering / University of Baghdad*

*Email: Ibraheemki@coeng.uobaghdad.edu.iq

**Email: Fatin.Hassan0@gmail.com

(Received 19 April 2016; accepted 23 August 2016)

http://dx.doi.org/10.22153/kej.2016.08.002

Abstract

This paper presents a meta-heuristic swarm based optimization technique for solving robot path planning. The natural activities of actual ants inspire which named Ant Colony Optimization. (ACO) has been proposed in this work to find the shortest and safest path for a mobile robot in different static environments with different complexities. A nonzero size for the mobile robot has been considered in the project by taking a tolerance around the obstacle to account for the actual size of the mobile robot. A new concept was added to standard Ant Colony Optimization (ACO) for further modifications. Simulations results, which carried out using MATLAB 2015(a) environment, prove that the suggested algorithm outperforms the standard version of ACO algorithm for the same problem with the same environmental conditions by providing the shortest path for multiple testing environments.

Keywords: *robotics, Path planning, ant colony optimization, static environment, and collision-avoidance.*

1. Introduction

Route forecasting is a vital part of steering of the mobile robot; it is recognized as the art of obtaining an enhanced collision-free path from the initial point toward destination station. In computational complexity theory, route forecasting is categorized as an NP (nondeterministic polynomial time) comprehensive problem. That is, the computational time that is required to solve such problem rises dramatically (typically at an exponential rate) while the size (or dimension) of the problem raises. The researchers of route forecasting begin in late 60's, and several algorithms have been proposed, comprising the roadmap method, cell decomposition, Potential fields, and mathematical programming, etc. It has been found that these methods are either ineffective, due to the significant computational cost; or imprecise, due to the trapping in local minima. To overcome these drawbacks, many heuristic methods have been implemented, such as the application of artificial neural networks. One of the main benefits of heuristic algorithms is that it can produce an acceptable result very fast, which is especially suitable to solve NP-complete problems. The objective is to find the shortest and collision-free route (if the path exists) between an initial point and an end point in a network [1].

In [2], authors proposed two kinds of route forecasting in the robotic field. These techniques are identified as global route forecasting and local path prediction. In global route prediction, robots have complete information about the situation This global path can aid the robot to navigate to the real location because the feasible optimal route has been found within the location before robot start to navigate) whereas in local route forecasting robots do not have information about the situation or the information is not comprehensive (incomplete information). The main difficulties for constructing route forecasting algorithms for mobile robots are efficiency and security. Efficiency means it should take best time and safety means avoiding collision with obstacles. The robot route forecasting methods could be classified into different kinds based on number of robot (single or multi robot) or type of environment (static or dynamic). According on the environment where the robot is situated, the route forecasting methods can be categorized into two kinds as shown in Figure 1.

Environment	Static	Dynamic
Known	Case1	Case3
Un Known	Case2	Case4

Fig. 1. Classification of Path planning based on environment.

Looking into Figure 1, for each of these two kinds could be additionally divided into two sub-classification depending on how much the robot recognizes the complete information of the surrounding locations:

1-Robot route forecasting in a known environment in which the automation previously knows the position of the obstacles before it starts to navigate.

2- Robot route forecasting in a partially known or inexact environment in which the robot estimates the environment using sensors to get the local information of the location, shape and dimension of obstacles and then uses the information to ensure local path planning.

In this paper, we present case 1 when the environment is static and known obstacles.

2. Related Works

Many types of research studies path planning in recent years. [3] describes the use of a genetic algorithm to find the optimal path in a grid environment; the mobile robot has to find the best route which decreases the number of stages to be taken between the initial point and the end point by allowing four-neighbor movements of the robot. Authors in [4] performed two types of path planning: global path planning using two versions of Artificial Bee Colony (ABC) and local path planning by hybrid Bacterial Foraging Optimization (BFO) and Artificial Potential Field (APF) algorithms. While the implementation of the Particle Swarm Optimization (PSO) to find the optimum path, obstacle avoidance is done by translating robot to the adjacent safe point around the obstacle's border which is pre-defined and calculated; this work has been presented in [5]. The researchers in [6] Introduced path planning in a dynamic environment by combining heuristic algorithm and simulated annealing algorithms. The researchers in [7] Implemented intelligent water drops (IWDs) algorithm to solve robot route forecasting problem, the suggested algorithm has two stages; the first stage, finds the best global path. The second stage does a local search at relatively close distances of the global route and decreases its length and response time. In [8] G. Yogita *et al.* describe the use of two metaheuristic algorithm based on solitary intelligence, these are Cuckoo Search algorithm (CS) and Bat search (BS) algorithms. They are used to find global planning for robot in the same static environment contain twenty Obstacles, and assume that the robot move with theta and distance from source station toward destination, they assume that if the robot encounters obstacle, it will back one step size and choose new position that satisfies minimum distance with destination, the simulation result showed that the Bat search outperforms the Cuckoo search in term of number of iteration and complexity of the environment. In [9] H. Hsu-Chih *et al*. Implemented Artificial Immune System (AIS) which is based on natural immune system ability to detect cells foreign to the body. It has been used to construct initial feasible shortest path for mobile robot from start point to the end after that they smoothed the resulting path by applying most effective curve interpolation called B-spline interpolation. The environment implemented for the work was a grid and contain only static obstacle, the simulation results show the effectiveness of the proposed algorithm in I-shape environment.

Several algorithms have been suggested for this purpose, Ant Colony Optimization (ACO) is one of the most used algorithms. This paper is organized as follow:

Section 1 Introduction, Section 2 Related work, section 3 Ant Colony Optimization, section 4 Modeling of the Environment, Section 5 Problem Definition, Section ACO and MACO Based Robot Path Planning, section 7-result, and discussion, section 8 conclusion, 9Acknowledgement, and 10-References.

3. Ant Colony Optimization (ACO) Algorithm

ACO is a met heuristic method that belongs to the arbitrary searching algorithm. This algorithm was firstly introduced by M. Dorigo [10] who made the complete use of the likenesses between the routes of ant colony searching for food and the well-known travel salesman problem (TSP). To solve the TSP by an artificially simulating the procedure of ant searching for the food, discovery the shortest route from ant nest to food places through the interchange of information and shared collaboration [11]. All ants of the same anthill move along the same path by following one another. This is because every ant releases a substance called “Pheromone” while moving. The other ants sense the intensity of pheromone and follow the path having a higher concentration of pheromone. This is their way to find an optimized path. Initially, the ants wander randomly to find their way to the destination. Every ant releases pheromone along the route. On their back tour ant senses pheromone intensity and choose the path having a higher concentration of pheromone. The pheromone evaporates with time and hence the concentration of pheromone would be higher along the shortest path as the time taken to cover the shortest path would be minimum as compared to other paths. Hence, almost every ant would be attracted by the higher intensity of pheromone along the shortest path and select the optimized path. Pheromones: Biochemical material left by an ant when traveling; separately ant probabilistically favors to follow route rich in pheromone relatively than a lesser one [12].

If the ants encounter obstacle, they travel into available paths with equal probability (re-initialize pheromones) as shown in the figure below:

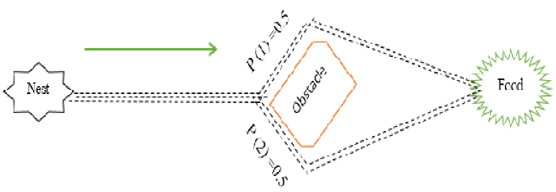


Fig. 2. Ants Obstacle Avoidance.

3.1. Mathematical Model of the ACO algorithm [13]

1-For kth ant at node i and want to move to next node j using probability formula.

$$Pij(k) = \frac{[\ \tau ij]^{\alpha} * [\ \eta ij]^{\beta}}{\sum_{k=1}^{m} [\ \tau ij]^{\alpha} * [\ \eta ij]^{\beta}} \qquad \dots (1)$$

Where $\alpha \,\&\, \beta$ are degree of importance of pheromones and heuristic function respectively.

2-after the ants complete their tour, the pheromones trial values are updated according to the following formula:

$$\tau ij(t+n) = (1-\rho)s\tau ij(t) + \Delta\tau ij \qquad \dots (2)$$

where ρ is the pheromones decay parameter ρ in range (0, 1)

$\Delta\tau ij$: is the amount of pheromones added by ant.

$$\Delta\tau ij = \textstyle\sum_{k=1}^{m} \Delta\tau ij.^{\wedge}k \qquad \dots (3)$$

$$\Delta\tau ij.^{k} = Q/fitness \qquad \dots (4)$$

where *Q* is pheromones update constant, *Fitness* is the performance index needs to be minimized.

3.2. Modified Ant Colony Optimization Algorithm (MACO)

The optimal path is the main component of the robotics domain, is the path needed to satisfy specific criteria such as distance, this problem is solved by using optimization algorithm such as Ant Colony Optimization (ACO), the new concept is added to standard ACO called **Aging.** The Approach to alleviate stagnation is pheromone control. Pheromone control adopts several approaches to reduce the influence of experience and encourage the exploration of new paths that are non-optimal.

Aging: A past experience can also be reduced by controlling the amount of pheromone deposited for each ant according to its age. This approach is known as aging. In aging, an ant deposit lesser and lesser amount of pheromone as it moves from one obstacle to another obstacle. Aging is based on the rationale that old ants are less successful in locating the optimal paths since they take a longer time to reach their destination. Both aging and evaporation encourage discoveries of new paths that are previously non-optimal [14].

Since the amount of pheromones deposit by every ant is giver by Eq. (4), that’s mean the $\Delta\tau ij$ is varying with corresponding to ants age, the suggested equation to change the amount of pheromones is given by:

$$Q_m = Q_{max} - \frac{Q_{max}-Q_{min}}{2} \times rand - \frac{Q_{max}-Q_{min}}{M} \times m \qquad \dots (5)$$

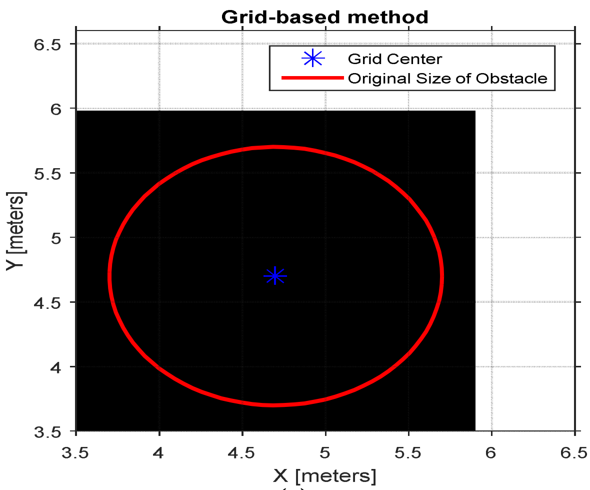
Where Q_{max} is the upper limit of pheromones and Q_{min} is the lower limit of pheromones, M is the total number of ants, and m is the index of ant in the colony.

4. Mobile Robot Environment Modelling

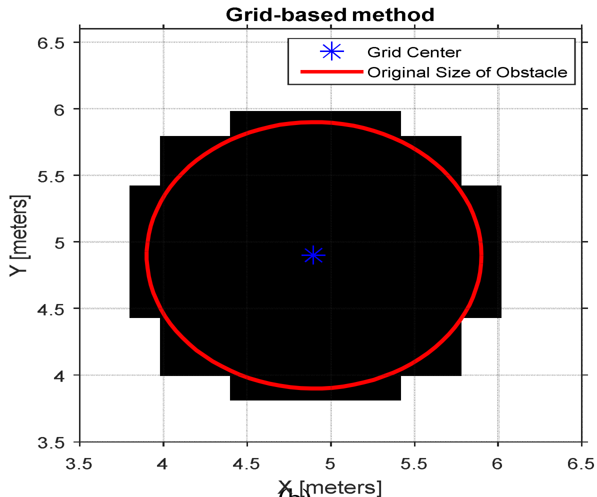
The first phase of route forecasting is to create an environmental model for the 2-D workspace of the mobile robot. There are several approaches of environment forming: Grid-Based method and Free Space-Based method. Each of these methods has advantages and disadvantages.

4.1. Grid-Based Method

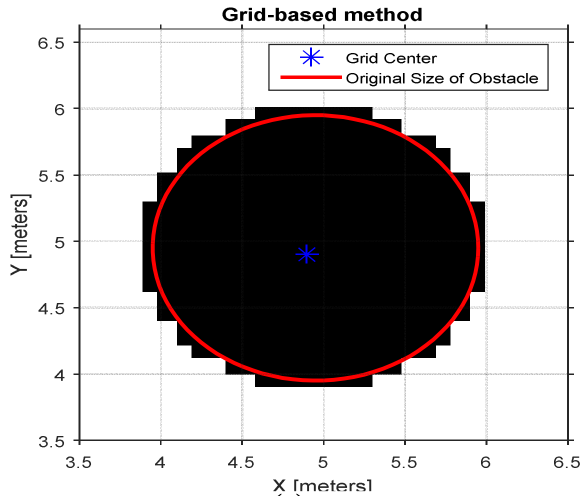
This environment is represented by a grid of (usually square) cells; each cell is either traversable or obstructed Object (on a traversable cell) can move to any adjacent traversable cell. The features of the Grid-Based method it is conceptually simple representation, local changes have only local effects, well suited for dynamic environments. Limitation of Grid-Based method are: imprecise representation of arbitrary obstacle increased resolution in one area increases complexity everywhere (potentially large memory size) as shown in Figure 3 below:



(a)



(b)



(c)

Fig. 3. (a) Grid-based method with resolution (1), (b) Grid-based method with resolution (5) (c) Grid-based method with resolution (10).

4.2. Free Space-Based Method

The environment is an initially empty simple shape, represent obstacles as virtual circle, advantages: arbitrary polygon obstacles, arbitrary motion angles, and unlimited resolution so memory is efficient. Limitations of the free space-based method are: complex code, Point localization takes more than constant time, and a waste of space (see Figure4).

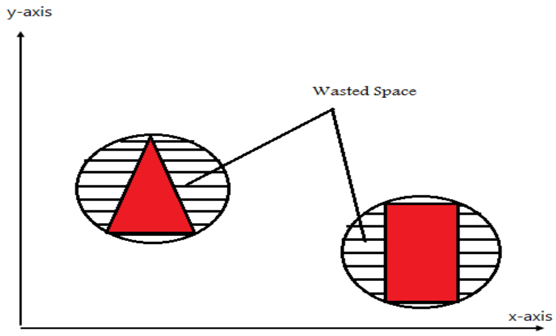


Fig. 4. Free space-based method.

In this paper, we present path planning in Binary Grid Method, each cell in the occupancy grid has a value representing the occupancy status of that cell. An occupied location is represented as true (1) and a free location is represented as false (0).

5. Problem Definition

Assume a 2-D square map covered with a uniform arrangement of net points. The size of the environment can be changed randomly; here, the environment comprises of a 10 ×10, 20 × 20, 30×30, and 40×40 grid cell. The left highest place of the environment is the beginning point for a route while the right lowest place of the environment is the end point for a route. The shapes and the Size of an obstacle are variable, the positions of the obstacles are arbitrarily closed and can be located at any network location in the map excepting at locations adjacent to the initial position region or adjacent to the position destination region (let two grid location left). Also, several obstacles are possible; Figure 4 below shows an example of such a procedure. Since the mobile robot is not a point, the dimension of the robot is added to the dimension of an obstacle to ensuring the safety of robot while piloting in the environment as displayed in Figure 5.

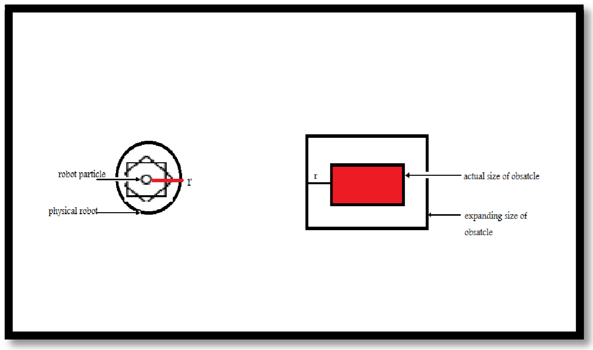


Fig. 5. Expanding the size of obstacle corresponding to robot size

In the figure above the size of obstacles is enlarge by the radius of the robot (r).
Beginning from the net position (1, 1), an ant iteratively travels from its current location to one of its surrounding locations. When ant at node i (row, col), ant A can select the next location (j) by selecting one of its 8 surrounding locations: [(i+1, j-1), (i+1, j), (i+1, j+1) ...] …C, where C typically is the set of all surrounding nodes of the current node. The ant A takes its next node arbitrarily, founded on the probability given by equation (1)

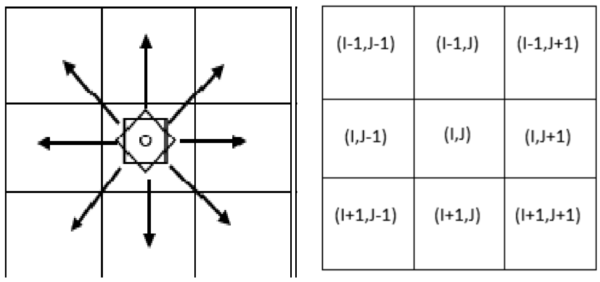


Fig. 6. An ant searching the surrounding cells.

The ant colony optimization algorithm is a joining product of positive response and some heuristic function. In ant colony optimization algorithm, ants select the route mainly depend on two stimulating message, the pheromone concentration and the visibility of next node. The concentration of pheromone does not have the large effect on the process in the early iterative process of algorithm. So, ant discovers enhanced solutions primarily according to the objective function. When used in route forecasting with grid model, the objective function is the shortest distance determined as follows:

$$D(i,j) = \sqrt{(Xi - X_g)^{2} + (Yi - Y_g)^{2}} \quad \ldots (6)$$

Where g represents the goal node and i represent the next node.

6. ACO and MACO Based Robot Path Planning

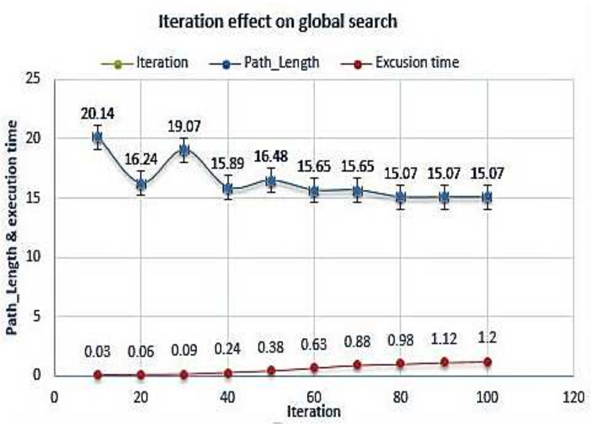
The Ant Colony Optimization technique can be applied to the robots to find an optimized path while navigating in an environment. Artificial Ant: are the mobile robot that is stimulated from the natural ants. The moving of the artificial ants is directed by a probabilistic function that depends on heuristic and trail functions. They transfer in the search space having all possible solutions and choose best solutions among them. Artificial ant favors routes having larger pheromone concentration. The place of ants and quality of the solution is recorded so that best solutions can be obtained.

Flow code for MACO algorithm [12]

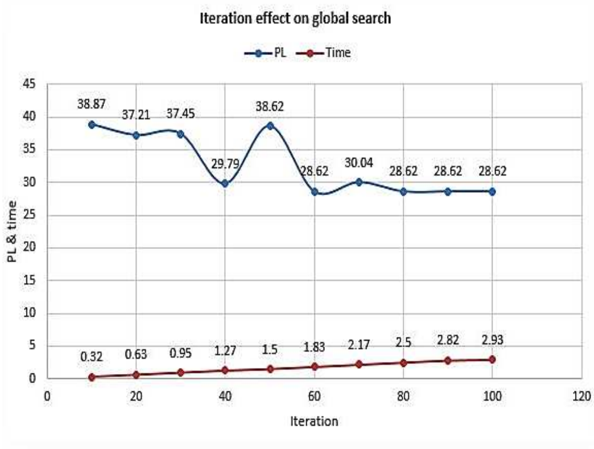
1- Do for iteration=1, 2, …N
Initialization stage
Pre-initialize the taboo table of each ant and the pheromone intensity on all sides.
Each ant needs to choose the next destination based on the restriction of the taboo table.
2-Do for ant=1, 2, …. K
3- Do for step=1,2, …. M
4-Compute the probability of the kth ant's next node using equation (1).
5- If the next node occupied by obstacle? Then
Neglect it
Else
Move to a next node by the computed probability Store the history of past node locations in an array
6- If the current location is equal to the end point? Then
- Obtain the path passed
-Update the pheromone evaporated on the entire map generated using equation (2)
Else
Go to step 4
End.

7. Results and Discussions
7.1. Effect of Design Parameters

In this section we present the influence of design parameter: number of Iteration, number of ants, and evaporation factor (ρ) on global search. The results are applying to 10×10, and 20×20 map size as shown below:



(a)

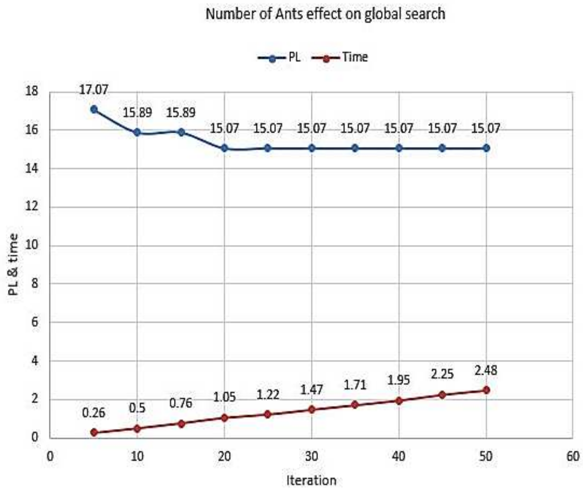


(b)

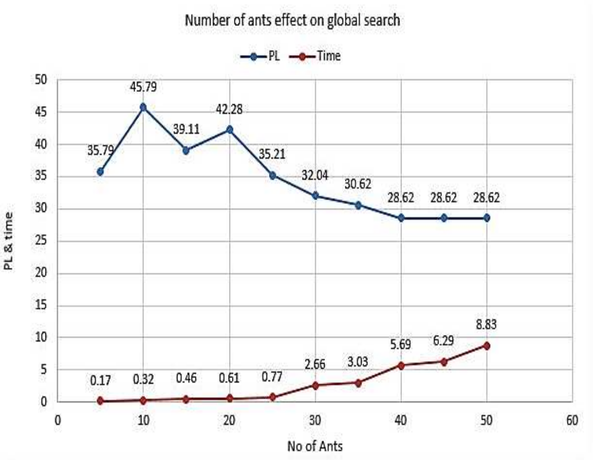
Fig. 7. Effect of no of iterations on path length andexecution time for 10 ×10 map, and (b) 20×20 map.

From the previous figure, it's evident that the algorithm finds an optimal path with increase total number of iteration, as shown in iteration (80) the path length are (15.07), (28.6274). Also, the whole time require is increased

By changing the total number of ant in the colony from (5) to (50), it's obvious the number of ants should be increased with increased search space as shown in the figure below:

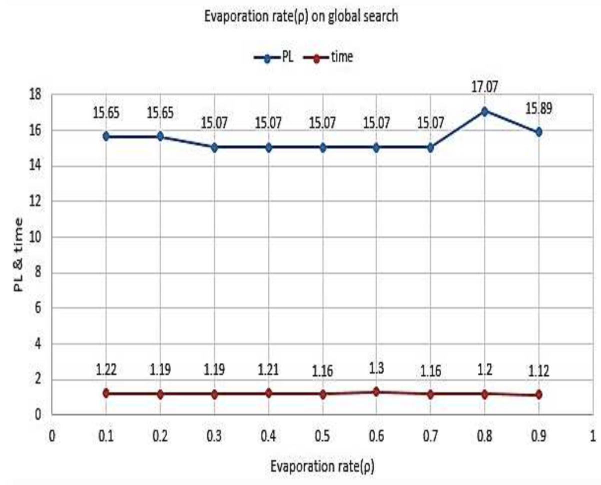


(a)

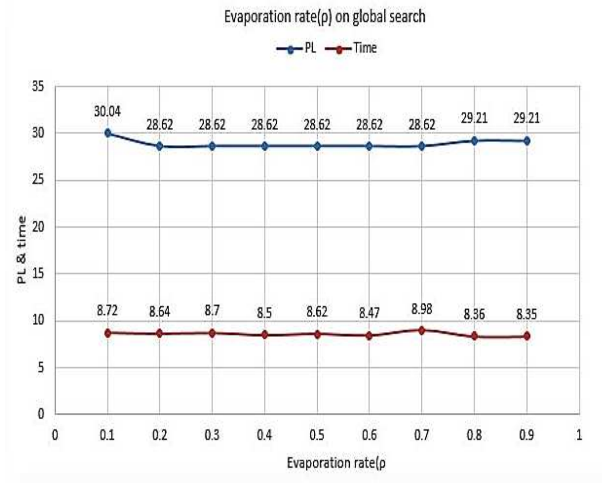


(b)

Fig. 8. Effect of no of ants on path length and execution time for (a) 10×10 map, and (b) 20×20 map.



(a)



(b)

Fig. 9. Effect of evaporation factor (ρ) on path length and execution time for (a) 10×10 map, and (b) 20×20 map.

From the previous figure, it's obvious that the best number for 10×10 map is (20), and for 20×20 map is (40). Also, the total time require is increased. The best value of evaporation factor (ρ) is between (0.3 to 0.7) for all dimension of search space, and has no effect on computation time as shown in the figure below:

7.2. ACO and MACO path planning Results

The results of implementing standard ACO Algorithm is introduced in this section. To make objective situations, four experiments were showed. All with different complexity (size of search space, Position, Shape, and some an obstacle) experiments were done with MATLAB R2015 (a). The parameter setting is shown in Table 1 below. For all the case studies the pheromone evaporation rate is set to ρ=0.3.

Table 1,
Parameter specification.

Parameter	Value
Iteration(K)	80-100
Number of Ant(M)	20-50
α(pheromone coefficient)	1
β(heuristic coefficient)	$1.605\text{-}17 \times (\text{map size})^2 + 0.22 \times (\text{map size}) + 6.181e^{-15}$
ρ (evaporation rate)	0.3

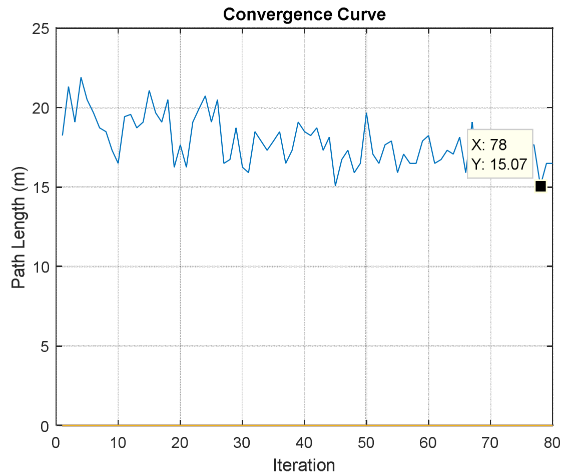


Fig. 11. The Convergence Curve for Experiment 1

7.2.1. Standard ACO

Experiment (1)

In this experiment the size of search space is (10×10) grid cell, and five obstacles are located in random manner for obtain the optimal path, its show that the shortest path is (15.07). The Figures 10 and 11 shows the results of SACO algorithm.

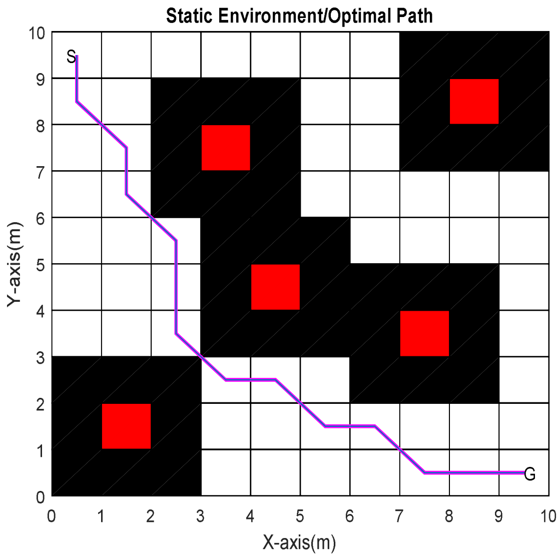


Fig. 10. Best Path found by Standard ACO for Experiment 1.

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (15.07) in iteration (78) with the total computation time equal to (1.217323) sec.

Experiment (2)

In this experiment the size of search space is (20×20) grid cell, and ten obstacles are located in a random manner for obtaining the optimal path, its show that the shortest path is (28.63). The Figures 12 and 13 shows the results of SACO algorithm for experiment 2.

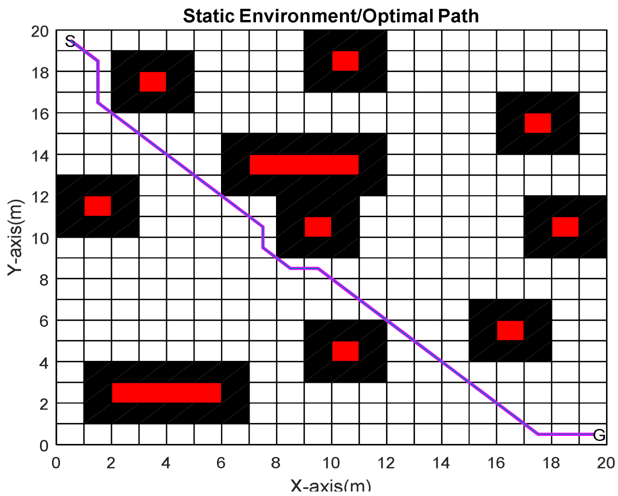


Fig. 12. Best Path found by Standard ACO for Experiment 2.

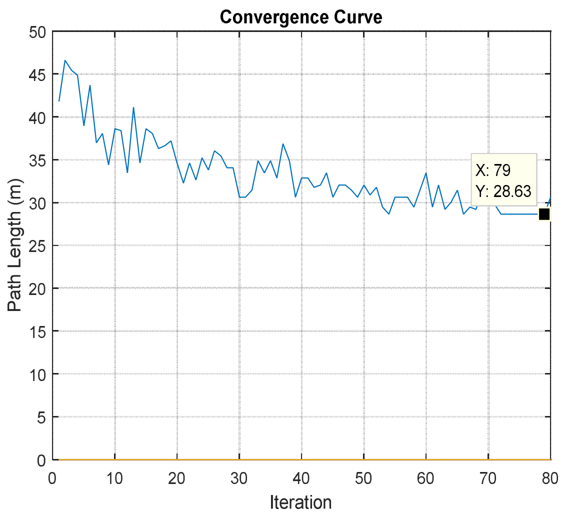


Fig. 13. The Convergence Curve for Experiment 2.

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (28.63) in iteration (79) with the total computation time equal to (9.072979).

Experiment (3)

In this experiment the size of search space is (30×30) grid cell, and obstacles are located in a random manner for obtaining the optimal path, its show that the shortest path is (45.9411). The Figures 14 and 15 shows the results of SACO algorithm for experiment 3.

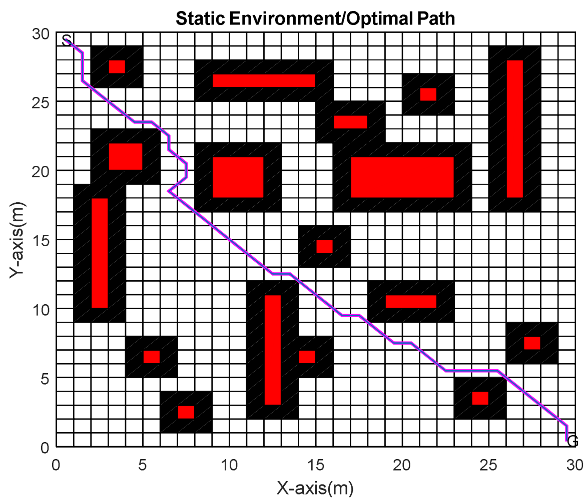


Fig. 14. Best Path found by Standard ACO for Experiment 3

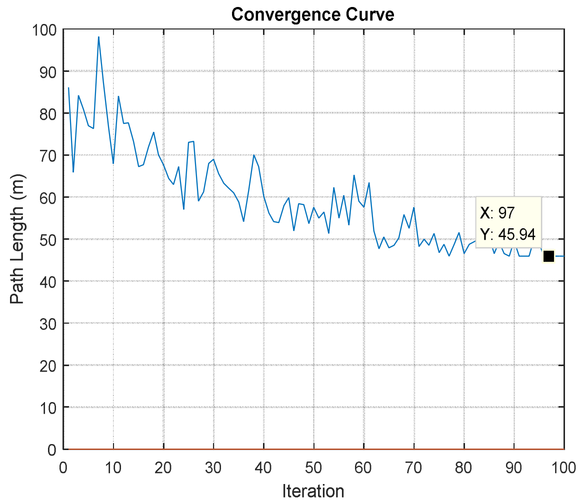


Fig. 15. The Convergence Curve for Experiment 3.

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (45.9411) in iteration (97) with the total computation time equal to (52.853641).

Experiment (4)

In this experiment the size of search space is (40×40) grid cell, and obstacles are located in a random manner for obtaining the optimal path, its show that the shortest path is (62.6690). The Figures 16 and 17 shows the results of SACO algorithm for experiment 4.

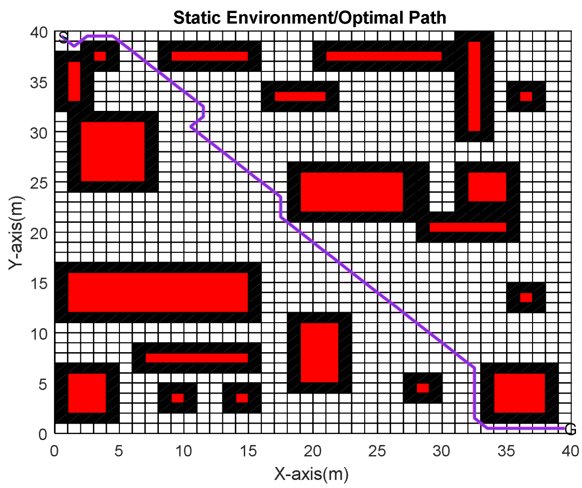


Fig. 16. Best Path found by Standard ACO for Experiment 4

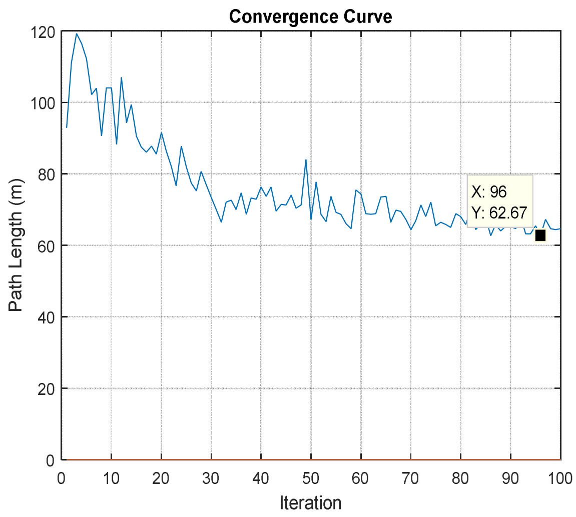


Fig. 17. the Convergence Curve for Experiment 4

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (62.6690) in iteration (96) with the total computation time equal to (134.770856).

7.2.2. Modified ACO (MACO)

Experiment (1)

In this experiment the size of search space is (10×10) grid cell, and five obstacles are located in random manner for obtain the optimal path, its show that the shortest path is (15.07). The Figures 18 and 19 shows the results of MACO algorithm.

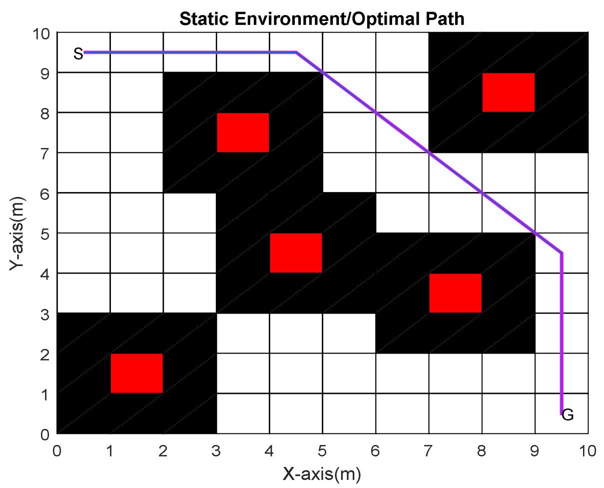


Fig. 18. Best Path found by Modified ACO for Experiment 1.

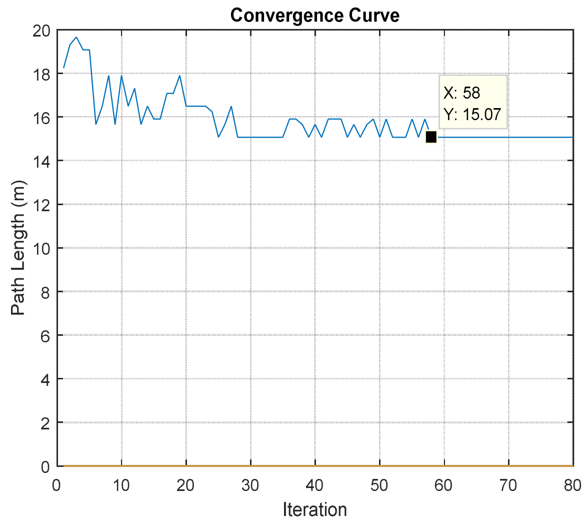


Fig. 19. The Convergence Curve for Experiment 1

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (15.07) in iteration (58) with the total computation time equal to (1.154530).

Experiment (2)

In this experiment the size of search space is (20×20) grid cell, and ten obstacles are located in a random manner for obtaining the optimal path, its show that the shortest path is (28.63). The Figures 20 and 21 shows the results of MACO algorithm for experiment 2.

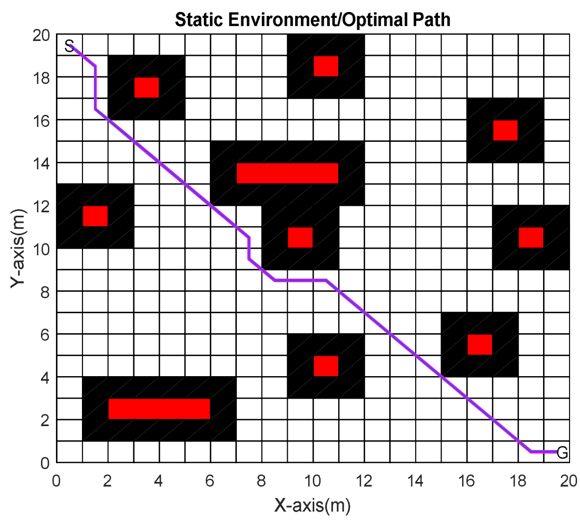


Fig. 20. Best Path found by Modified ACO for Experiment 2.

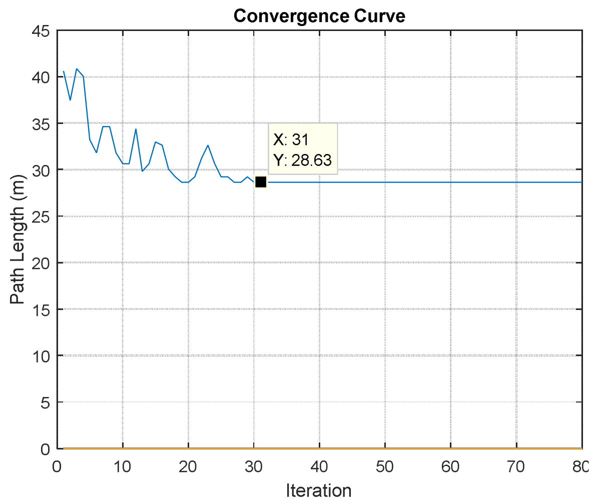


Fig. 21. The Convergence Curve for Experiment 2

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (28.63) in iteration (31) with the total computation time equal to (7.838582).

Experiment (3)

In this experiment the size of search space is (30×30) grid cell, and obstacles are located in random manner for obtain the optimal path, its show that the shortest path is (45. 1127). The Figures 22 and 23 shows the results of MACO algorithm for experiment 3.

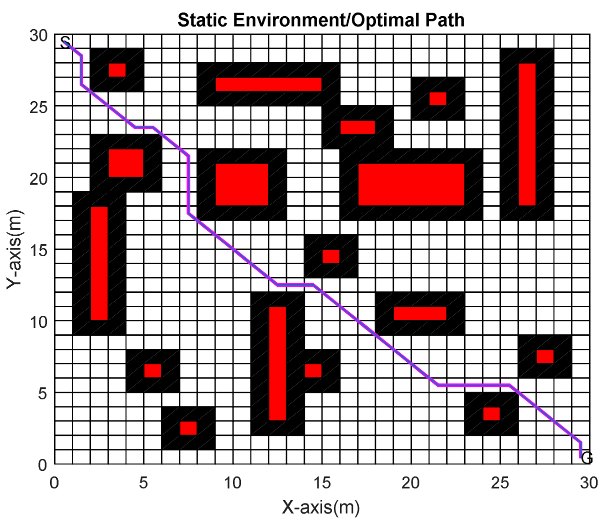


Fig. 22. Best Path found by Modified ACO for Experiment 3.

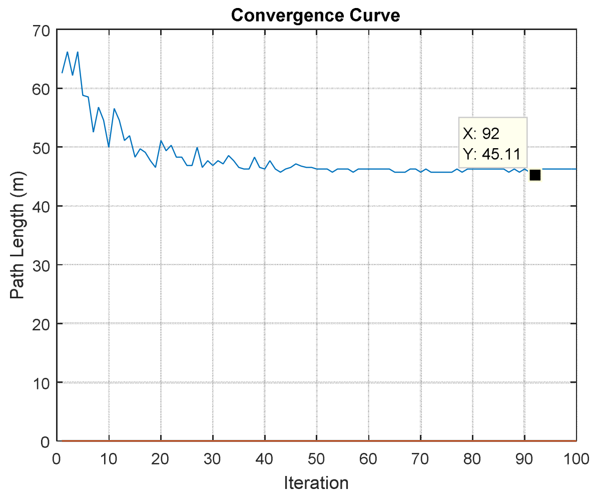


Fig. 23. The Convergence Curve for Experiment 3

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (45.1127) in iteration (92) with the total computation time equal to (52.017160).

Experiment (4)

In this experiment the size of search space is (40×40) grid cell, and obstacles are located in a random manner for obtaining the optimal path, its show that the shortest path is (61.8406). The Figures 24 and 25 shows the results of MACO algorithm for experiment 4.

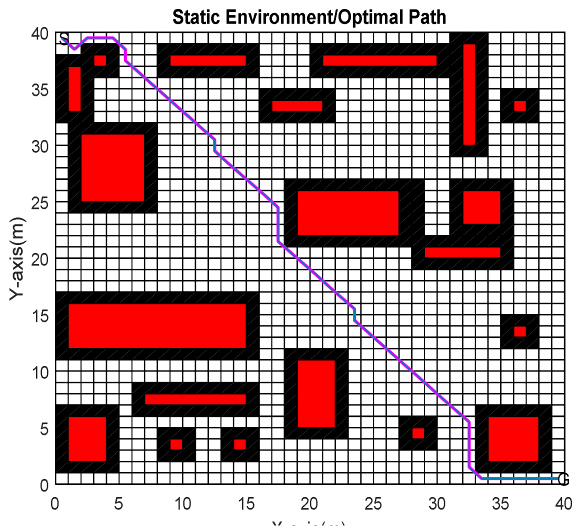


Fig. 24. Best Path found by Modified ACO for Experiment 4.

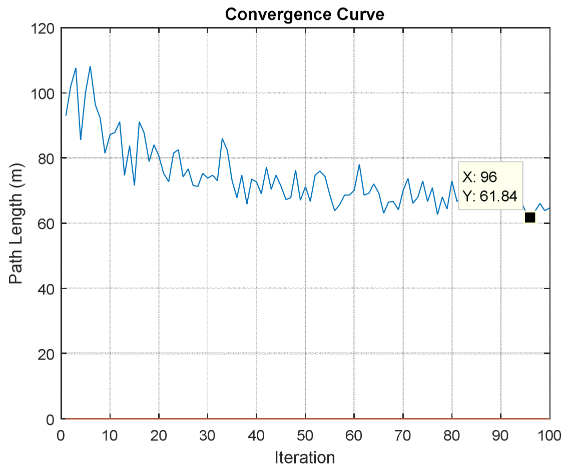


Fig. 25. The Convergence Curve for Experiment 4.

From the previous two figure, the mobile robot start avoids static obstacles with the shortest path is (61.8406) in iteration (96) with the total computation time equal to (132.678247).

Table 2,
Comparison between Standard ACO and MACO in term of path length and execution time.

Map Size	Standard ACO /Path Length	Modified ACO /Path Length	Standard ACO /Computation time(sec)	Modified ACO /Computation time(sec)
10×10	15.07	15.07	1.217323	1.154530
20×20	28.6274	28.6274	9.072979	7.838582
30×30	45.9411	45.1127	52.853641	52.017160
40×40	62.6690	61.8406	134.770856	132.678247

By looking to convergence curve for each map, we noticed the proposed algorithm is reached to global optimum faster than the standard version of the algorithm, and from Table II, the proposed algorithm requires less time, and path length either equal or less than the standard version.

The time requires to solve problem increased dramatically with increasing search space, (from 1.217323 to 134.770856) sec for standard ACO and (from 1.154530 to 132.678247) sec for modified ACO; this is because non-deterministic polynomial nature of path planning as mention earlier.

8. Conclusions

This paper presented route forecasting of a mobile robot using a new proposed metaheuristic algorithm called the ant's age colony optimization. It reaches the goal using the capability of the optimization of ant system algorithm in the assumed stationary environment. The Number of case studies has been directed by varying the number, size and position of the obstacle. The result is found to be best and satisfying for the individual problem with the parameters (α = 1, ρ= 0.3 and number of ant's m =50). With rising in the complexity of the problem, i.e. with an increase in some obstacles existing in the map; this algorithm will need a larger amount of execution time.

9. Acknowledgement

I would like to acknowledge all the staff of the Electrical Engineering Department at Baghdad University for their help.

10. References

[1] B. Michael et al." Ant Colony Optimization Algorithm for Robot Path Planning", International Conference on Computer Design and Applications (ICCDA) 2010.

[2] A. A. David M.W. Powers, "Review of classical and heuristic-based navigation and path planning Approaches", 2013.

[3] AL-Taharwa. Ismail, "A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment", Journal of Computer Science (J. Computer Sci.), Vol. 4, no 4, pp. 341-344, 2013.

[4] A.H. Nizar and A.M. Farah, "Improvement of Path Planning for Autonomous Mobile Robots Using Population-Based Optimization Algorithms". Master Thesis, Electrical Engineering Dept., University of Baghdad, 2014.

[5] M. A. Memon, "Autonomous Robot Path Planning Using Particle Swarm Optimization in Static and Obstacle Environment", Sindh Univ. Res. Jour. (Sci. Ser.), Vol. 47, No. 4, pp.653-658, 2015.

[6] M. S. Ganeshmurthy and G. R. Suresh, "Path Planning Algorithm for Autonomous Mobile Robot in Dynamic Environment", International Conference on Signal

Processing, Communication and Networking (ICSCN), 3rd 2015.

[7] S. Sohaila et al." An Intelligent Water Drops algorithm for solving robot path planning problem", IEEE International Symposium on Computational Intelligence and Informatics, 2013.

[8] G. Yogita et al.," A Comparison between Bat Algorithm and Cuckoo Search for Path Planning", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 3, Issue 5, May 2015.

[9] H. Hsu-Chih et al. "A Metaheuristic Artificial Immune System Algorithm for Mobile Robot Navigation",2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

[10] [10]-D. Marco and S. Thomas, "Ant Colony Optimization', Massachusetts Institute of Technology, 2004.

[11] Ying Pei, "Basic Ant Colony Optimization", International Conference on Computer Science and Electronics Engineering, 2012.

[12] Xin-She Yang, "Engineering Optimization an Introduction with Metaheuristic Applications", Cambridge, United Kingdom, 2010.

[13] Singiresu S. Rao, "Engineering Optimization Theory and Practice", fourth edition, 2009.

[14] G. Yogita and G. Kusum, "Artificial Intelligent in Robot Path Planning", International Journal Soft Computing and Engineering(IJSCE), ISSN:2231-2307, Vol.2, Issue.2, May 2012.

[15] K. Sapna, T.Chhavi, Y.Deepak,and T. Mayank," Robot Path Planning in Static Environment using Ant Colony Optimization", International Journal of Emerging Trends in Electrical and Electronics(IJETEE), Vol. 5, Issue. 3, July-2013.

التنبؤ بأفضل مسار للروبوت المتنقل باستخدام خوارزميات الاسراب الأمثليه

أبراهيم قاسم أبراهيم*   فاتن حسن عجيل**

، قسم الهندسة الكهربائية / كلية الهندسة/ جامعة بغداد*

*البريد الالكتروني: Ibraheem151@coeng.uobaghdad.edu.iq

**البريد الالكتروني: Fatin.Hassan0@gmail.com

الخلاصة

الهدف من البحث هو أيجاد افضل مسار للروبوت المتنقل وخال من الاصطدام بالعوائق الثابته بعد اعطاء نقطة البداية والنهاية بأستخدام خوارزميات الأسراب الامثليه "خليه النمل " خوارزمية النمل الامثليه استلهمت من السلوك الطبيعي للنمل في البحث عن الغذاء، تم اضافه مفهوم جديد (عمر النملة وتأثيرة على نسبة الفرمون الحشري)، الخوارزمية تضمنت ايجاد اقصر واسلم مسار في بيئات مختلفه بتعقيدات مختلفة، اضيف حجم الروبوت الى الحجم الاصلي للعوائق الثابتة ثم معاملة الروبوت بوصفه نقطة، تم تنفيد النتائج باستخدام برنامج الماتلاب، واثبتت صحة الخوارزمية المقترحة بالمقارنة مع الخوارزمية الاصلية لنفس للهدف والظروف نفسها لأيجاد اقصر مسار لعدة تنفيذات.