



Development of Path Planning Algorithm Using Probabilistic Roadmap Based on Ant Colony Optimization

Mohammed I. Abdulakareem^{a*}, Firas A. Raheem^b

^aControl and Systems Department, University of Technology, Baghdad, Iraq.
61284@student.uotechnology.edu.iq

^bControl and Systems Department, University of Technology, Baghdad, Iraq. 60124@uotechnology.edu.iq

*Corresponding author.

Submitted: 26/06/2019

Accepted: 31/07/2019

Published: 25/03/2020

KEY WORDS

Path Planning, Probabilistic Roadmap (PRM), Ant colony optimization, B-spline curve.

ABSTRACT

In this paper, a unique combination among probabilistic roadmap, ant colony optimization, and third order B-spline curve has been proposed to solve path-planning problem in complex and very complex environments. This proposed method can be divided into three stages. First stage is to construct a random map depending on the environment complexity using probabilistic roadmap algorithm. This could be done by sampling N nodes randomly in complex and very complex static environments, then connecting these nodes together according to some criteria or conditions. The constructed roadmap contains huge number of possible random paths that may connect the start and the goal points together. Second stage includes finding path within the pre-constructed roadmap. Ant colony optimization is selected to find or to search the best path between start and goal points. Finally, the third stage uses B-spline curve to smooth and reduce total length of the found path in the previous stage where path's length has been reduced by 1% in first environment and by 15% in second environment. The results of the proposed approach ensure feasible path between start and goal points in complex and very complex environment. In addition, the path is guaranteed to be shortest, smooth, continues and safe.

How to cite this article: M. I. Abdulakareem and F. A. Raheem, "Development of path planning algorithm using probabilistic roadmap based on ant colony optimization," *Engineering and Technology Journal*, Vol. 38, Part A, No. 03, pp. 343-351, 2020.

DOI: <https://doi.org/10.30684/etj.v38i3A.389>

This is an open access article under the CC BY 4.0 license <http://creativecommons.org/licenses/by/4.0>.

1. Introduction

Path planning is a process of obtaining reasonable, collision free route between the start and goal point. The need of path planning becomes important for fully or partially automated process. There are two types of environments according to how much information is known about the environment. The environment considered as known when the location of obstacle(s) is defined previously while considered as unknown when there is no information about it [1]. Path planning can be divided into two categories.

First category, according to time: On-line and Off-line. In On-line path planning, path computed during motion according to data coming from sensors but in Off-line planning the path computed according environment model. Second category, according to environment: dynamic environment in which, environment contains moving and static obstacles while in static environment contains only static obstacles [2]. Another category can be added depending on type: global planning in which the path may be found from start to goal point before robot starts to move. Therefore, the information must be fully known about the environment. Second type is local planning in which, path cannot be fully obtained before robot starts to move because knowledge about environment is partially known or unknown. It is important to know path planning categories and environments types because good path planning leads to good robot navigation. Global path planning starts from 1980 until these days and many algorithms had been proposed by researchers. Research leads to improve global planning year by year. At the beginning, path planning was focusing on finding path to goal only. Then it became bigger issue by not only reach the goal but also to consider optimization criteria [3]. There are many algorithms used to solve path-planning problem such as heuristic methods, meta-heuristic, and randomized method. A*, D*, and Dijkstra can be considered as heuristic methods. A* algorithm is considered as greedy, graph, and heuristic search algorithm that able to find sub optimal but, not optimal path. [4]. D* algorithm can be considered as dynamic A* which is able reforming the path according to new information coming from sensors [5]. Meta-heuristic methods include particle swarm optimization (PSO), and ant colony optimization (ACO) [6]. In addition, there are intelligent methods, like artificial potential field (APF), genetic algorithm (GA), APF is used in different types of robots where the field of forces is applied on robot. These forces are attractive force to goal and repulsive force from obstacles [7]. Randomized methods can be divided into two categories: rapidly exploring random tree (RRT) which is more suitable for dynamic environment and probabilistic roadmap (PRM) for static environment.

PRM algorithm has been applied successfully to very complex static environments. PRM computation can be divided into two phases: the preprocessing (construction) phase and the query phase. In construction phase, Roadmap can be generated by generating N number of free nodes randomly and connecting these nodes by simple planner called local planner. Edge is a straight path computed by local planner connecting two nodes together so, roadmap constructed from nodes and edges. After construction phase, the query phase starts by connecting given start and goal points to constructed roadmap and then obtain sequence of edges that grantee feasible and collision free path [8].

The presented work gives a new approach, which consists of combining a random grid of roadmap, ant colony optimization, and third order B-spline curve. At the end, the approach obtains interpolated, smoothed, and shortened path in complex and very complex environment.

The general structure of this paper will be as follows. Section 2 will give brief explanation about related work while section 3 will clarify proposed method, the classic PRM, and its algorithm. Section 4 will contain the theory of ant colony optimization (ACO). Section 5 for b-spline curve theory. Section 6 will contain results and finally conclusion of presented work will be written in section 7.

2. Related Work

In order to overcome path-planning problems, many algorithms have been studied extensively. One of these algorithms is probabilistic roadmap (PRM) algorithm. Starting with Kavraki who used PRM to find path in high dimensional space. In other words, PRM used with a robot has many degrees of freedom in know static environment. Kavraki use uniform sampling to distribute nodes [8]. Valerie Boor propose another strategy for sampling node called Gaussian sampler. The new strategy proposed to grantee better coverage of difficult part of space (narrow passage) [9]. Wilmarth propose another strategy for sampling nodes in narrow passages called medial axis. The basic concept of medial axis sampling is to sample a node randomly. The sampled node retracted to the medial axis between two obstacles, regardless if the sampled node is collision free or not [10]. Geraets list many advanced sampling strategies such as: Gaussian, nearest contact, obstacle based, obstacle based*, medial axis and bridge test. In bridge sampling strategy two nodes sampled randomly and the distance between them are predefined. If the mid-point between two sampled nodes is, collision free while two sampled nodes are not free, then mid-point is added otherwise no node will be added. Obstacle based strategy sample a node randomly if it is collision free the node will be added, otherwise, in random direction the sampled node moves with pre-defined steps till it becomes collision free while in obstacle based* the sampled node will be discarded if its collision free [11]. In 2005, David Hsu propose a hybrid sampling strategy. This strategy uses multiple sampling methods to take advantage of each one and reduce disadvantages of each one [12]. Shwail use A* algorithm

with probabilistic roadmap to find near optimal path between start and goal points after that, improved genetic algorithm used to enhance found path and make it more optimal [2]. Generally, probabilistic roadmap (PRM) used with static environment but, Zhang propose a method to plan a path for mobile robot in unstructured and dynamic environment. The algorithm based on two phases: first phase, construct collision free roadmap and store it as graph; second phase is Q-learning, reinforcement learning method, is collaborated with PRM to obtain good path to catch goal. In this way, the robot uses past experience to improve its performance in avoiding dynamic and static obstacles. This method called PRM-Q method [13].

2. Proposed Method

In this paper, a new combination between probabilistic roadmap algorithm, ant colony optimization, and B-spline curve has been proposed as shown in Figure 1. Mainly, the proposed methods divide path-planning solution into three stages. First stage is construction of probabilistic roadmap, which will be clarified in current section. Second stage is searching path by ant colony represented and explained in section 4, while the last stage is smoothing and enhancing path using B-spline curve. The third stage is discussed in detail in section 5.

This section will clarify the first two blocks of the proposed methods while next sections will consider the remaining blocks. Starting with first block, each path-planning problem starts with a task and environment as shown in Figure 2. Normally, task is start and goal point and sometimes is more, like make the path to pass through specific points. Second thing in path planning problem is the environment. The environment could be static or dynamic, and known or unknown. Whenever the path-planning problem defined very well, the suitable path-planning algorithm can be chosen.

Second block in the proposed method is probabilistic roadmap construction. Roadmap (R) can be constructed in preprocessing phase where roadmap contains edges and nodes $R = (N, E)$. N is a set of nodes chosen randomly to be collision free with obstacle while E is edge, which can be defined as very simple, straight, and feasible path. These edges or paths called local paths founded by not powerful planner, but very fast planner called local planner. At the beginning, the roadmap $R = (N, E)$ is empty. Frequently, random nodes are sampled and added to N if it is collision free. Now, for each new node q some nodes q' selected from previously sampled nodes to connect them with new node by local planner. This process leads to find new edge $E = (q, q')$. The neighbor's nodes N_q specified according to Eq. (1) [8]. if the distance between current node and any node in R is less than or equal to pre-defined threshold, the node considered as neighbor otherwise; node can be neglected as shown in Figure 3 and the process repeated for all nodes. Usually, threshold value is (1-10) units

$$N_q = \{q' \in N \mid D(q, q') \leq D_{max}\} \quad (1)$$

Where D_{max} is predefined threshold, q is new generated node, q' is neighbor node, and $D(q, q')$ is Euclidean distance Eq. (2) [8].

$$D(q, q') = \|q - q'\| \quad (2)$$

Another way can be used to specify node neighbors is to choose nearest K number of nodes to the current node.

General algorithm of roadmap construction: -

$N \leftarrow \emptyset$

$E \leftarrow \emptyset$

Loop

q ← is randomly generated node

$N_q \leftarrow$ number of nodes neighboring node q

For all $q' \in N_q$, in order of increasing $D(q, q')$ do

$E \leftarrow E \cup \{(q, q')\}$, Update R

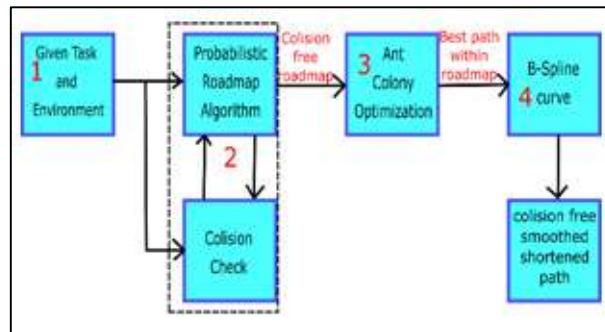


Figure 1: Proposed method

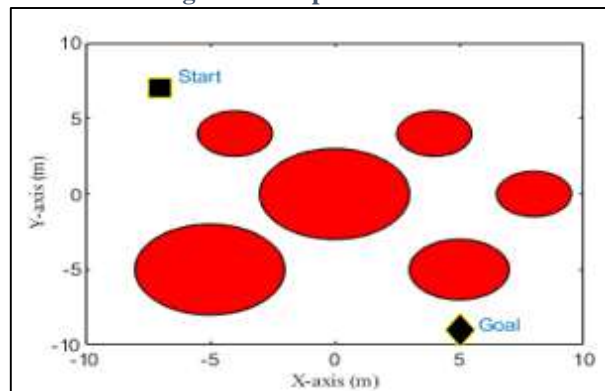


Figure 2: Path planning problem

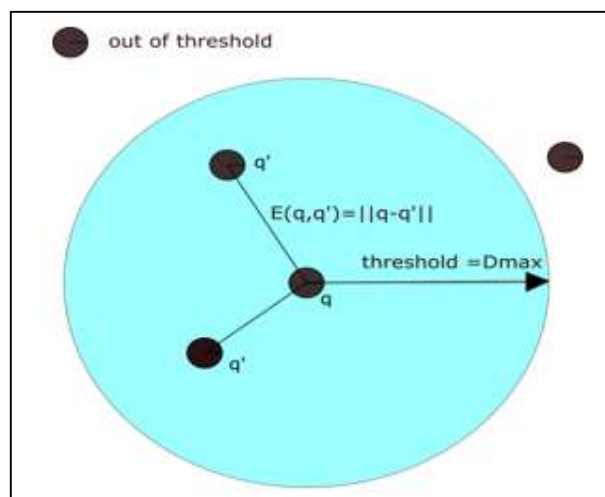


Figure 3: Selecting neighbor

The last step in roadmap construction is related to local planner to check if the edge is collision free or not. There are several ways to check collision some of them stated as flows:

- Incremental: In this method, the edge divided into X steps starting from start point to goal point and with each step the collision checked with obstacles [11].
- Binary: This method is different from last mentioned method in the way of if dividing edge. The division of edge start from middle and if the mid-point collision free, the two portions of the edge is divided again in half and the collision checked again too till predefined number of division X is completed. If all collision checks are free, then the edge is collision free [11].

The output of the second block is collision free roadmap. The roadmap, for example as shown in Figure 4, now is ready for next phase, which is query phase. The objective of query phase is to try connecting given start and goal point with constructed roadmap. Next step is to obtain feasible path between start and goal point. Now, the question is how to obtain or search shorter path within roadmap [8]. Ant colony meta-heuristic method is used to serve previous goal.

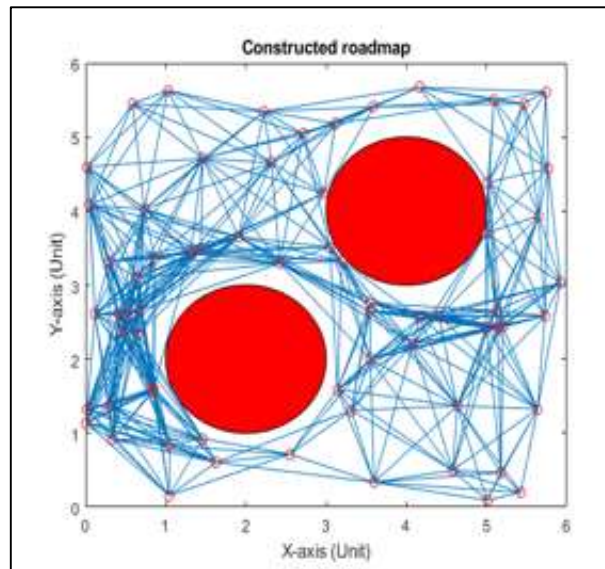


Figure 4: Construction of PRM

3. Ant Colony Optimization (ACO)

This section represents third block in proposed approach. The objective of this part is to select feasible path from constructed roadmap. In the world of ants, it is well known that ants are capable to obtain shortest path to food using swarm intelligence. Ant Colony Optimization (ACO) has been developed to simulate behavior of ants to find heuristic solution for optimization problem. ACO proposed in 1992 by Dorigo. When the ants travel to find food, they deposit chemical matter called pheromones. Pheromones attract other ants to food source. When more ants travel to specific food source, the quantity of pheromones on the path increased. The shortest path to food source contains more pheromones, because more ants travel through this path. The pheromone is chemical matter that evaporate over time, thus it reduces the chance of taking path with low pheromone level, where pheromone level in shortest path will increase because, deposition process of pheromone overcome evaporations process [14]. Consider a network, which is in our case the constructed roadmap, the probability that ant K travels from node i to next node j can be calculated by Eq. (3) [14].

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij}^k)^\alpha (\eta_{ij}^k)^\beta}{\sum_{i \in N_i^k} (\tau_{ij}^k)^\alpha (\eta_{ij}^k)^\beta} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad (3)$$

τ_{ij}^k refers to pheromone levels, while η_{ij}^k represents Euclidean distance between node i (current node) and node j (next node). α and β is weight of pheromone level and distance respectively between two nodes. When β equal to zero then the probability of choosing node j depends on nothing but pheromone level. In other words, when α equal to zero then the probability depends only on heuristic distance (the closest node to current node). K is ant number while N_i^k is set of neighboring nodes.

As mentioned before, the pheromone evaporate overtime as expressed in Eq. (4) [14]. Evaporation rate denoted by ρ ($0 \leq \rho < 1$) where

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad (4)$$

It is clearly that when ρ equal to zero, no evaporation happens, in other hand, when ρ equal to one, pheromone is fully evaporated. After evaporation process, the pheromone level updated by additional levels deposited by ants according to Eq. (5) [14].

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (5)$$

Where m is population, and $\Delta \tau_{ij}^k$ can be calculated by Eq. (6) [14].

$$\Delta \tau_{ij}^k = \frac{1}{c^k} \quad (6)$$

c^k reward of ant k for choosing path. Figure 5 show the flow chart of ACO

4. B-Spline Curve

The output of the most path-planning algorithm is straight lines segment. Actually, path with straight line segments is undesirable because of discontinuity issue, mechanical wear, localization error, and

slipping. In addition, the need of robot to stop at each turning edge, then turn, and lastly move forward again can be considered as another disadvantage. Dubin's, Reed, and Shepp's paths generate smooth paths by combining lines and circles but the path was discontinuous [15].

B-Spline, Bezier curve, and NURBUS (Non-Uniform Rational B-Spline) curve are the most famous spline curve used in computer aided design. Each type of those curve use control points to control their shape and they were studied to use them in path planning. Bezier curve has unwanted properties like; number of control point will determine curve order. Whereas B-spline curve, order independent on number of control points, which give advantage over Bezier curve.

NURBUS curve is derived from B-Spline and the only different is each control point in NURBUS curve has its own weight. These weights give high flexibility to modify the generated curve by changing the weight of each control point [16]. A B-spline curve has n control points P_i ($i= 0, 1, \dots, n$) and k order may be defined by Eq.(7) [17].

$$p(t) = \sum_{i=0}^n p_i N_{i,k}(t) \quad t_{k-1} \leq t \leq t_{n+1} \quad (7)$$

Where $N_{i,k}(t)$ are basis function of B-spline with k order defined over the knot vector $T= \{t_0, t_1, \dots, t_k, \dots, t_n, \dots, t_{n+k}\}$. Basis function can be obtained by the following recursive de Boor-Cox Eq. (8) [16].

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t \leq t_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (8a)$$

$$N_{i,k}(t) = \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(t) \quad (8b)$$

$$t_i = \begin{cases} 0, & i < k \\ 1 + i - k, & k \leq i \leq n \\ 2 + n - k, & i > n \end{cases} \quad (8c)$$

This is the last block of proposed approach and its output is the results of the approach. The results are shown in next section.

5. Simulation and Results

In this paper, two environments have been used for simulation purpose. First environment contains less obstacles than second one. More details will be discussed in next two sub-sections.

I. First environment

First environment may be easy relatively to second one. It has six circular obstacles with different sizes. Environment ranges, $x= [-10 \ 10]$, $y= [-10 \ 10]$. PRM algorithm used to construct roadmap Then ACO used to find the shortest path within roadmap as shown in Figure 6. Finally, path smoothed and shortened by B-spline curve as shown in Figure 7. After ACO implementation length of path was 20.927 m but this path represented by segments of straight lines. After smoothing process path length was 20.73 m, which is shorter than path found by ACO. It is good to mention that third order B-spline curve has been used.

II. Second environment

In second environment, same steps are repeated again. Environment range $x= [-8 \ 8]$, $y= [-8 \ 8]$. Simulation figures are shown in Figures 8-9. The length of found path by ACO is 62.3397 m which is reduced to 52.9779 m using B-spline curve.

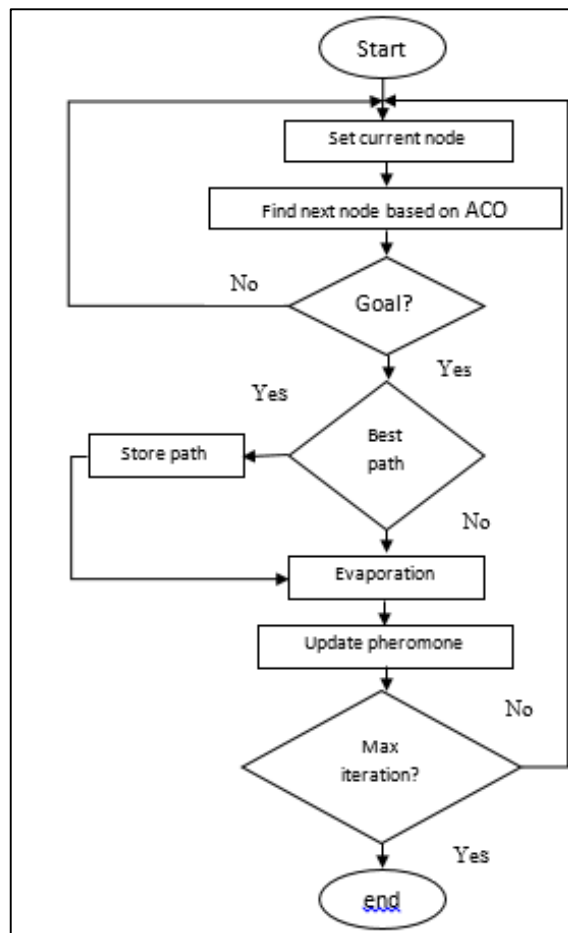


Figure 5: ACO flow chart

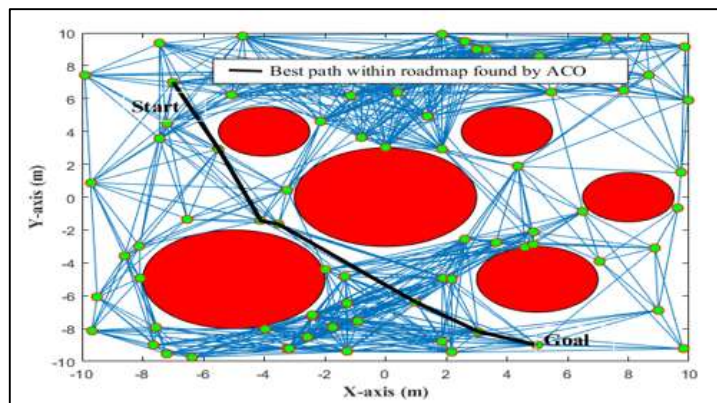


Figure 6: Given Environment, constructed roadmap, and shortest path in roadmap for Environment 1

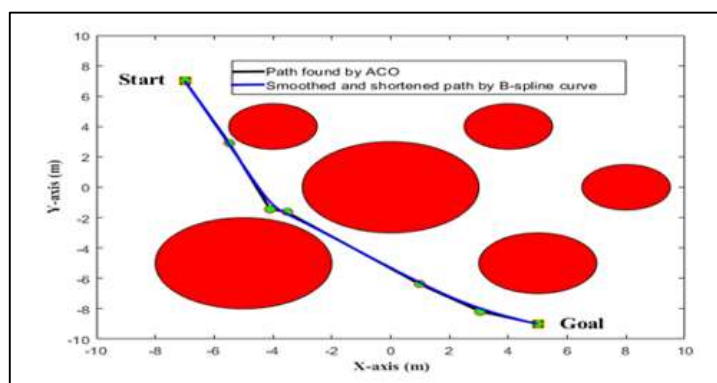


Figure 7: Smoothed and shortened path by B-spline for Environment 1

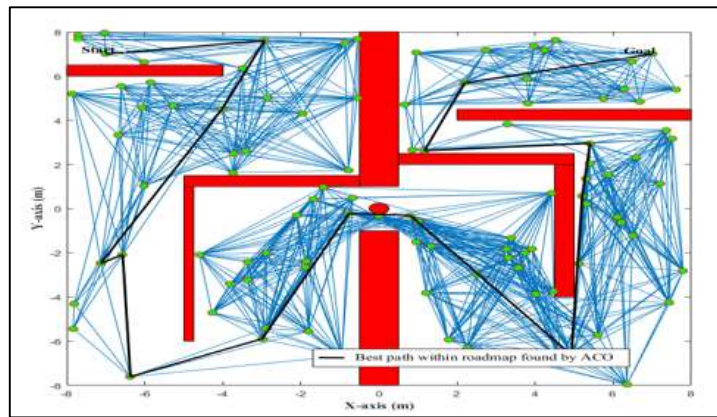


Figure 8: Given Environment, constructed roadmap, and shortest path in roadmap for Environment 2

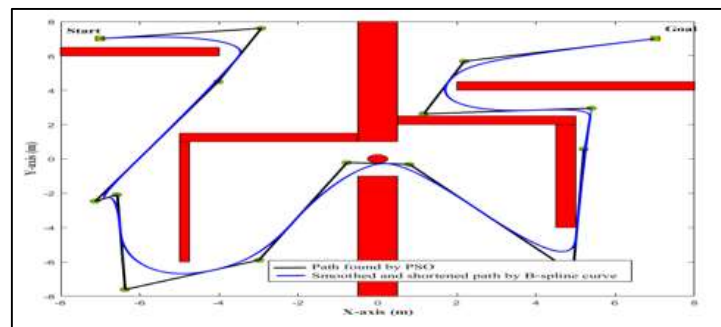


Figure 9: Smoothed and shortened path by B-spline for Environment 2

Table 1 shows some information about PRM algorithm parameter and ACO parameter, also the length of found path. It is noticed that number of nodes in second environment is more than first one. This because second environment is more complex than first one. Even if it is smaller than first one. Also, for same reason, number of population and iteration in second environment is bigger than second one. Second thing could be noticed is affection of B-spline in shortening the path is depend on how much bad or good path found by ACO.

For example, in first environment the found path by ACO is relatively good so, B-spline curve reduce path length only about 0.2 m but, in second environment the path was bad and B-spline curve had more effect on path length by about 10 m.

Hardware used in simulation Intel® core i3 CPU @ 2.4GHz and 4.00 GB RAM. MATLAB 2018 is used as simulation software.

Table 1: Summarized Information

Environment	PRM		ACO			ACO path (m)	B-spline path (m)
	node	threshold	population	iteration	alpha beta Rho		
1	80	8	300	600	1 1 0.01	0.01	20.927
2	100	8	400	700	0.8 0.5 0.01	0.01	62.3397

5. Conclusions

This paper proposes to use the combination of probabilistic roadmap (PRM) and ACO technique to solve path-planning problem for complex, very complex, static, and known environment. The proposed method solves the path-planning problem by dividing it into three stages. In first stage, PRM used to model environment by constructing collision free roadmap where this roadmap contains more than one possible path to reach goal here the first stage finished. Second stage is finding path stage by using ACO to obtain the shortest path within roadmap. Found path by ACO is represented by straight lines. Third stage is shortening and smoothing the path found by ACO. From results, it can be noticed that B-spline curve reduce path's length in first environment about 1% while in the second environment, it reduced by 15 % therefore; it can be said that the effect of third stage depends on the quality of the path found in second stage. If the ant colony optimization stage gives high quality path the affection of B-spline is small, otherwise the use of B-spline is very effective robot

moves. In addition, it is noticed that the quality of the path depends on probability of the node distribution. But, at the same time probabilistic roadmap algorithm can ensure to find several paths to the goal regardless the complexity of the environment. The complexity of the roadmap depends on complexity of the environment. In other words when the environment is complex, more nodes should be distributed which leads to increase number of ways that may be selected to reach the goal. So, the complexity of roadmap is increased at this case. At the end, the approach has been simulated using two different in complexity environment, and the path has been approved to be shortest, smooth, continues, and collision free.

References

- [1] F. A. Raheem, and U. I. Hameed, "Interactive heuristic D* path planning solution based on PSO for two-link robotic arm in dynamic environment," *World Journal of Engineering and Technology*, Scientific Research Publishing, Vol. 7, No. 1, pp 80-81, February 2019.
- [2] S. H. Shwail, and A. Karim, "Probabilistic roadmap, A*, and GA for Proposed decoupled multi-robot path planning," *Iraq journal for applied physics*, Vol. 10, No. 2, pp.3, June 2014.
- [3] N. B. Sariff, and N. B. Amin, "Ant colony system for robot path planning in global static environment," 9th WSEAS international conference on System science and simulation in engineering, Japan, pp. 192, 2010.
- [4] F. A. Raheem, and A. A. Hussain, "Applying A* path planning algorithm based on modified C-space analysis," *Al-Khwarizmi Engineering Journal*, Vol. 13, No. 4, pp. 125, 2017.
- [5] F. A. Raheem, and U. I. Hameed, "Path planning algorithm using D* heuristic method based on PSO in dynamic environment," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, Vol. 49, No. 1, pp. 258, 2018.
- [6] Y. Gigras, and K. Gupta, "Metaheuristic algorithm for robotic path planning," *International Journal of Computer Applications*, Vol. 85, No. 3, pp. 1, January 2014.
- [7] F. A. Raheem, and M. M. Bader, "Development of Modified path planning algorithm using artificial potential field (APF) based on PSO for factors optimization," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, Vol. 37, No. 1, pp. 317, 2017.
- [8] L. E. Kavraki, P. Svestka, J. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transaction on Robotics and Automation*, Vol. 12, No. 4, pp. 566-570, August 1996.
- [9] V. Boor, M. H. Overman, and A. F. Stappen, "The gaussian sampling strategy for probabilistic roadmap planners," *IEEE International Conference on Robotics and Automation*, Michigan, pp. 1018, 1999.
- [10] S. A. Wilmath, N. M. Amato, and P. F. Stiller, "MAPRM: a probabilistic roadmap planner with of the free space," *IEEE International Conference on Robotics and Automation*, Michigan, pp. 1024, 1999.
- [11] R. Geraerts, and M. H. Overmars, "Sampling and node adding in probabilistic roadmap planners," *Robotics and Autonomous Systems*, ELSEVIER, Vo. 54, No. 2, pp. 169, December 2005.
- [12] D. Hsu, G.S. Ante, and Z. Sun, "Hybrid PRM sampling with a cost-sensitive adaptive strategy," *IEEE International Conference on Robotics and Automation*, Barcelona, pp. 3874, 2005.
- [13] Y. Zhang, N. Fattahi, and W. Li, "Probabilistic roadmap with self-learning for path planning of a mobile robot in a dynamic and unstructured environment," *IEEE International Conference on Mechatronics and Automation*, Takamatsu, pp. 1074, 2013.
- [14] M. Brand, M. Masuda, and N. Wehner, X. Yu," Ant colony optimization algorithm for robot path planning", *International Conference on Computer Design and Applications ICCDA*, China, pp.436-437, 2010.
- [15] M. Elbanhawi, M. Simic, and R. Jazar, "Continuous-curvature bounded trajectory planning using parametric splines," *Frontiers in artificial intelligence and applications*, Vol. 262, pp.514, 2014.
- [16] M. Elbanhawi, M. Simic, and R. Jazar, "Continuous path smoothing for car-like robots using B-spline curves," *Journal of Intelligent and Robotic Systems*, Vol. 80, pp. 25-26, 2015.
- [17] C. Tai, S. Hu, and Q. Husang, "Approximate merging of B-spline curves via knot adjustment and constrained optimization," *Computer-Aided Design*, Vol. 35, pp. 894, 2003.