Journal of Engineering Science and Technology
Vol. 15, No. 5 (2020) 3034 - 3054


# DEVELOPMENT OF A* ALGORITHM FOR ROBOT PATH PLANNING BASED ON MODIFIED PROBABILISTIC ROADMAP AND ARTIFICIAL POTENTIAL FIELD

FIRAS A. RAHEEM, MOHAMMED I. ABDULKAREEM*

Control and Systems Department, University of Technology-Iraq,
Al-sina'ah Street, 19006, Al-Rusafah, Baghdad, Iraq.
*Corresponding Author: 61284@student.uotechnology.edu.iq

## Abstract

Path planning is one of the most interesting topics in robotics field for researchers. It responsible to find the best path between the start and the goal point for a given environment and task. In this paper, a new approach has been proposed to solve the path planning problem by combining the methods of probabilistic roadmap (PRM) and artificial potential field (APF), where the attractive potential filed is used to enhance the construction of roadmap by improving the nodes' location. These new locations of the nodes ensure better path planning possibilities in a given complex static known environment. A* heuristic method is used to find the shortest path within the constructed roadmap. This path represented by segments of straight lines therefore, a non-uniform rational B-spline (NURBS) curve is used to smoothen the path and reduce the path length. Particle swarm optimization (PSO) is used to obtain the optimized weights that needed for each control point that participate to form the spline curve. The optimized weights ensure shortest and collision free path. The results that come out from the proposed approach can guarantee the path feasibility and reasonability between the start and the goal points in complex, static, and known environment. Moreover, the final path ensured to be continues, smooth, safe and absolutely optimal in term of path length.

Keywords: A* algorithm, Artificial potential field (APF), NURBS curve, Probabilistic roadmap (PRM), Particle swarm optimization (PSO).

## 1. Introduction

Path planning is a process of obtaining reasonable, collision free route between the start and the goal points. Path planning becomes an important issue for fully or partially automated process. There are two types of environments according to how much information is known about 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 may be divided into two categories. First category, according to time: On-line and Off-line path planning. In On-line path planning, path computed during motion according to data coming from sensors while in Off-line planning, the path computed according to environment model. Second category in path planning is according to environment: dynamic and static environment. Usually, the dynamic environment includes moving and non-moving obstacles therefore, sometimes called a hybrid environment. In other hand, the static environment contains only non-moving 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 has to be fully known about the environment. Second type is local planning in which, path cannot be fully obtained before robot starts to move because the knowledge about environment is partially known or fully unknown. It is important to know path planning categories and environments types because it leads to good path planning, which leads to good robot navigation. Global path planning starts from 1980 till these days and many algorithms proposed by researchers. Researches improve global planning year by year. At the beginning, path planning was focusing on finding the path to the goal only. Then it became bigger issue by not only reach the goal but, also to consider optimization criteria. [3].

There are a lot of algorithms used to solve path planning problem such as heuristic methods, meta-heuristic, and randomized method. A*, D*, and Dikstra can be considered as heuristic methods. A* algorithm is 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 from sensors [5]. Meta-heuristic methods include particle swarm optimization (PSO), and ant colony optimization (ACO) [6]. Also, there are intelligent methods, like artificial potential field (APF), genetic algorithm (GA). APF is used with different types of robots where the forces result from potential field is applied on robot. These forces are attractive force to the goal and repulsive force from the obstacles [7]. Randomized methods can be divided into two categories rapidly exploring random tree (RRT) which is more compatible for dynamic environment while 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 pre-processing (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. A straight path computed by local planner used to connect two nodes together. This straight line called edge so, roadmap constructed from nodes and edges. After construction phase, the query phase started by given pair start and goal points to the constructed roadmap and then obtain sequence of edges that grantee feasible and collision free path [8].

The contributions of this novel work are proposing a new approach utilizes the combination between the probabilistic roadmap and the artificial potential field method to enhance the roadmap construction. Moreover, a heuristic A* method has been used with the enhanced roadmap as a search path method. Lastly, a Non-uniform rational B-spline (NURBS) curve based on particle swarm optimization (PSO) has been used for enhancing the path in terms of length and smoothness, where PSO is used to find the optimized weights of the control points of NURBS curve. These weights ensure shortest and collision free path.

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. Section 4 will contain the theory of classical probabilistic roadmap algorithm and its modification based on APF while A* heuristic method explained in Section 5. Section 6 for Non-uniform rational B-spline (NURBS) curve theory based on particle swarm optimization (PSO). Section 7 will contain results and finally conclusion of presented work will be written in Section 8.

## 2. Related Work

To overcome path planning problems, many algorithms were studied extensively. One of these algorithms is probabilistic roadmap (PRM). Starting with Kavraki et al. [8] who used PRM to find a path in high dimensional space. In other words, PRM used with a robot has many degrees of freedom in know static environment. L.E. Kavraki use uniform sampling to distribute nodes. Geraets and Overmars [9] lists many advanced sampling strategies such as: Gaussian, nearest contact obstacle based, obstacle based-star, medial axis and bridge test. In the 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 the mid-point will be 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-star the sampled node will discarded if its collision free. In 2005, Hsu et al. [10] proposes a hybrid sampling strategy. The proposed strategy uses multiple strategies to take the advantage of each strategy and reduce disadvantages of each one. Shwail and Karim [2] used A* algorithm with probabilistic roadmap to obtain near optimal path between the start and the goal points after that, improved genetic algorithm used to shorten found path and make it more optimal. Generally, probabilistic roadmap (PRM) used with static environment but, Zhang et al. [11] proposed 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 reach the 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.

## 3. Problem Formulation and Proposed Method

This section gives general description about the problem that has to be solved, also the proposed method for this purpose.

### 3.1. Problem formulation

The main problem is to solve a given task include start and goal points for specific complex environment, then try to sense and compute all possibilities and situations according to the environment degree of complexity for finding path solution that should be feasible, collision free, and optimal path.

### 3.2. Proposed method

This section will clarify several points about proposed method. Firstly, solving path planning problem divided into three stages: constructing roadmap using probabilistic roadmap algorithm, finding best path in the constructed roadmap by A* method and enhancing the obtained path by A* in terms of smoothness and length. This approach gives a unique combination to obtain feasible path in complex and very complex static known environment. In addition to the combination, new method is used to sample nodes. All sampling methods which used in constructing roadmap did not consider the location of the start and the goal points. The new method uses artificial potential field to serve the objective of sampling nodes with considering the location of start and goal point. Artificial potential field has two components. First one is attractive field which its direction toward the goal point and second component is repelling field which its direction away from the obstacles. The new method uses the attractive potential field to enhance the location of the normally distributed nodes and focusing them towards goal point. After constructing roadmap, A* algorithm used to obtain the best path with in roadmap. This path is represented by straight lines which undesired in path planning because of discontinuity issue. Non-uniform rational B-spline (NURBS) curve used to enhance the path in term of smoothness and the length of the found path. NURBS curve gives high flexibility to change the shape of its spline by changing the weights of each control points that participate in forming the spline. Particle swarm optimization is used to optimize these weights to make the spline shortest and collision free. Each step in the proposed approach will explained in detail in the next sections. For better understanding, Fig. 1 shows the block diagram of the proposed approach.

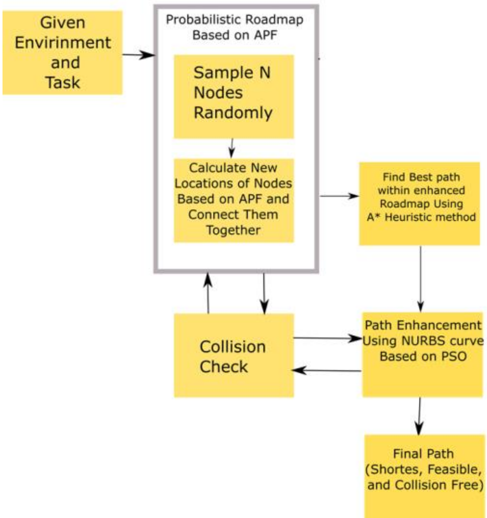


**Fig. 1. Block diagram of proposed method.**

## 4. Probabilistic roadmap (PRM) based on artificial potential field

This section describes PRM algorithm in details. Roadmap (*R*) is constructed in pre-processing phase where the roadmap contains edges and nodes *R*= (*N*, *E*). *N* is a set of nodes distributed randomly to be collision free with obstacles while *E* is edge which can be defined as very simple, straight, and feasible path connects two nodes together. These edges or paths called local paths founded by not powerful planner but very fast planer called local planner. At the beginning, the roadmap *R*= (*N*, *E*) is empty. Frequently, a random free node is produced and added to *N*. For each new node has location *q*, some nodes with location 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 neighbours nodes set *Nq* is specified according to Eq. (1), where Dmax is predefined threshold by user according to [8], *q* is new generated node, q' is neighbour node, and *D* (*q, q'*) is Euclidean distance Eq. (2). If the distance between current node and any node in *R* is less than or equal to pre-defined threshold, the node considered as neighbour otherwise; the node neglected as shown in Fig. 2. Another condition can be added to connect two nodes. The condition is the edge between two selected nodes has to be collision free. The process repeated again for all nodes. Another way can be used to specify node neighbours is to choose nearest K number of nodes to the current node [8].

$$Nq = \{q' \in N | D(q,q') \leq Dmax\} \quad (1)$$

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

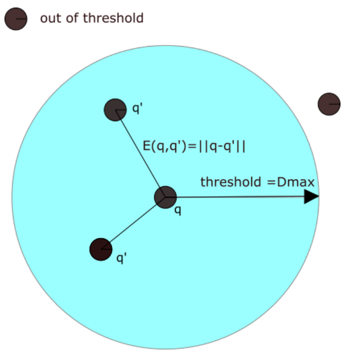


**Fig. 2. Selecting neighbor.**

As previously explained, probabilistic roadmap uses normal distribution to sample nodes that needed to model the environment. Also, there are several sampling techniques developed by researchers such as: Gaussian, nearest contact obstacle based, obstacle based*, medial axis and bridge test [9]. No one of these sampling techniques consider the location of the start and the goal points. The new method of sampling nodes depends on artificial potential field. The idea behind using APF in sampling nodes to make nodes’ location interacts with the start and the goal points.

Firstly, the basics and theory of APF has to be clarified to see how APF may be used in nodes distribution. The principle idea of the potential field ($U$) is that robot considered as a point that has coordinate $q$ in the environment.

The potential field is applied on this point. The filed drag point to the goal and at the same moment it repeals the point (robot) from obstacles boundaries. So, there are two components of fields. The first one is attractive field ($U_{att}$) and the second one is repulsive field ($U_{rep}$) as shown in Eq. (3) [12, 13].

$$U(q) = U_{att}(q) + U_{rep} \tag{3}$$

The artificial force (F) as shown in Eq. (4) maybe defined as negative gradient of the artificial potential field that guide the point to goal. Artificial force has two components attractive force ($F_{att}$) and repulsive force ($F_{rep}$) [12, 13].

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{att} \tag{4}$$

$$F(q) = F_{att}(q) + \ F_{rep}(q) \tag{5}$$

Before explaining how APF can be used in nodes sampling. Each component of artificial field should be defined. Depending if the point is within the goal area or not, the attractive field is defined by the quadratic or conic Eq. (6), where, $\beta$ is a positive coefficient of attractive field, and $\rho_g$ is Euclidean distance between the current ($q$) and the goal ($q_g$) position. The goal area can be recognized by a circle with a center at goal point and radius ($D$). Quadratic equation gives large magnitude of attractive force so; it is not preferable to use it when the point is far away from goal. Large magnitude of attractive force leads to fast driving to the goal which may cause collision with obstacles so, it is more suitable to use conic equation. Normally, the radius of goal area is relatively small therefor; quadratic equation is used inside goal area. In the other hand, the gradient of attractive field and it is expressed by Eq. (7). At $D$=$\rho_g$, the gradient of quadratic and conic potential are equal. Finally, the attractive force $F_{att}(q)$ can be shown in Eq. (12) [12].

$$U_{att}(q) = \begin{cases} \frac{1}{2}\beta\rho_g(q)^2 & \rho_g(q) \le D \\ D\beta\rho_g(q) - \frac{1}{2}\beta D^2 & \rho_g(q) > D \end{cases} \tag{6}$$

$$\nabla\, U_{att}(q) = \nabla\frac{1}{2}\beta\rho_g(q)^2 \tag{7}$$

$$\nabla\, U_{att}(q) = \nabla\frac{1}{2}\beta \left|\left|q - q_g\right|\right|^2 \tag{8}$$

$$\nabla\, U_{att}(q) = \frac{1}{2}\beta\nabla\sum\left(q^i - {q_g}^i\right)^2 \tag{9}$$

$$\nabla\, U_{att}(q) = \beta\left(q^1 - {q_g}^1, \dots, q^n - {q_g}^n\ \right) \tag{10}$$

$$\nabla\, U_{att}(q) = \beta\left(q - q_g\right) \tag{11}$$

$$F_{att}(q) = \begin{cases} -\beta\left(q - q_g\right) & \rho_g(q) \le D \\ -D\beta\dfrac{\left(q - q_g\right)}{\left|\left|q - q_g\right|\right|} & \rho_g(q) > D \end{cases} \tag{12}$$

The second component of the potential field is repulsive potential field $U_{rep}$ which is responsible to push the point away from obstacles and it is defined by Eq. (13). $\eta$ is positive coefficient of repulsive field, and $\rho_o$ is predefined distance which is determine if the repulsive field has influence on point or not. $\rho_q$ is Euclidean distance between current position ($q$) and the position of closest obstacle to point ($q_c$). The repulsive force is derived to be expressed by Eq. (14) [12, 13].

$$U_{rep}(q)U_{rep}(q) = \begin{cases} \frac{1}{2} * \eta * \left(\frac{1}{\rho_q} - \frac{1}{\rho_o}\right)^2 & \rho_q \geq \rho_o \\ 0 & \rho_q < \rho_o \end{cases} \tag{13}$$

$$F_{rep}(q) = \begin{cases} \eta \left(\frac{1}{\rho_q} - \frac{1}{\rho_o}\right) \frac{1}{\rho(q)^2} \nabla \rho(q) & \rho_q \geq \rho_o \\ 0 & \rho_q < \rho_o \end{cases} \tag{14}$$

$$\nabla \rho(q) = \frac{(q - q_c)}{||q - q_c||} \tag{15}$$

The proposed method for sampling nodes in the environment is to generate N nodes randomly then for a given start and goal points the attractive force calculated for each location ($q$) of the randomly sampled nodes. These nodes can be derived to new position ($q_{new}$) according to the direction and magnitude of attractive force ($q_{new}$). Normally, the direction of these nodes is towards the goal point and the magnitude depends on how much these nodes are far from the goal. The new position of node after influence of attractive force can be calculated by Eq. (16).

$$q_{new} = q_{old} + \mu F_{att} \tag{16}$$

$\boldsymbol{\mu}$ is positive number, between (0-1), determine affection of attractive force on the sampled nodes. Fig. 3 shows the old and new location of nodes. The blue star represents old position ($\boldsymbol{q_{old}}$), red circle is new position ($\boldsymbol{q_{new}}$), black arrow shows the direction and magnitude of the attractive force, black points represent start and goal, and finally yellow circle is obstacle. It is good to mention that the repulsive force is neglected because probabilistic roadmap algorithm keeps only collision free nodes.

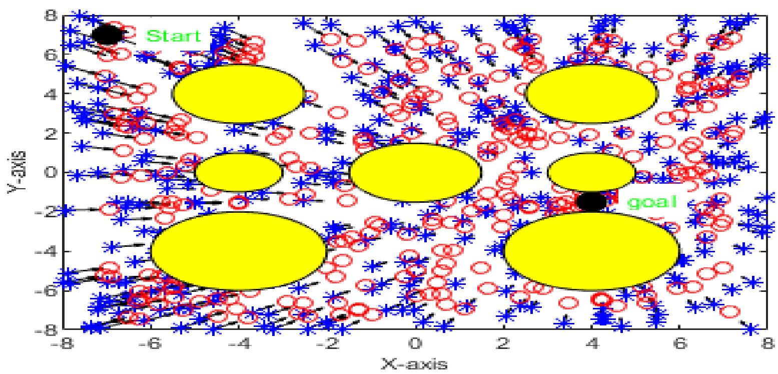


**Fig. 3. Old and new location of node.**

The new position of nodes is more focused in the direction of the goal points, which is lead to construct better roadmap. After finding the new location for all nodes, these nodes connected together by Eq. (1) as previously explained. The last step related to local planner is to check if the edge is collision free or not by dividing edge into $X$ steps, starting from the start point to the goal point and with each step the collision checked with obstacles. The new point at each step can be calculated by Eq. (17)

$$x_i = \left(1 - u_i \frac{d}{X}\right) x_{start} + (u_i \frac{d}{X}) x_{end} \tag{17}$$

$$y_i = \left(1 - u_i \frac{d}{X}\right) y_{start} + (u_i \frac{d}{X}) y_{end} \tag{18}$$

where $d$is total length of edge, $u_i$ is division number, $i$= {1, 2, 3… $X$-1}. Figure 4 shows equation terms for better understanding. The following algorithm shows the steps of new sampling method in constructing roadmap:

1. Define environment, start point, and goal point.
2. Generate N nodes randomly in environment space.
3. At each node, calculate APF force by Eq. (12).
4. Apply attractive forces on nodes locations and calculated new locations by Eq. (16).
5. Connect nodes together according to Eq. (1) to be collision free.

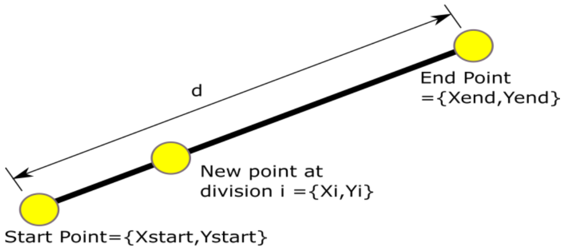


**Fig. 4. Collision check of road map edge.**

At the end, the roadmap, for example as shown in Fig. 5, is ready for the next phase which is query phase. The objective of query phase is to try connecting given start and goal point with constructed roadmap. In other words, the path to be obtained between start and goal point. Now, the question is how to obtain or search shorter path within constructed roadmap [8]. A* heuristic method is used to serve previous goal.

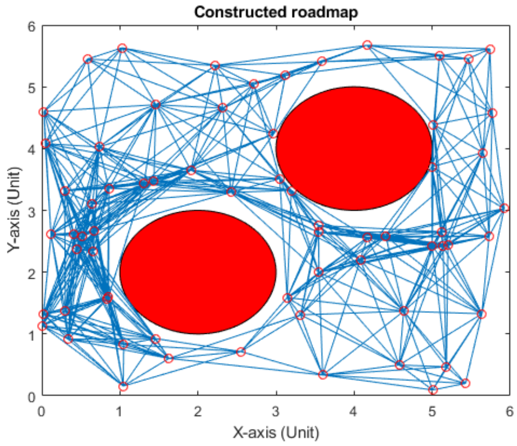


**Fig. 5. A simple example of constructed roadmap.**

## 5. A* Heuristic Method

A* algorithm is one of the most popular algorithm used to obtain the feasible path between the start and the goal point. It is very efficient and simple to implement. Normally, A* use eight neighbour nodes shaped as square [14]. A* algorithm is the best first search that combines the benefits of uniform-cost and greedy searches using fitness function Eq. (19) [15].

$$f(n) = g(n) + h(n) \tag{19}$$

$g(n)$ represents accumulative cost from the start node to the current node while h(n) is heuristic estimated function from the current node to the goal node. Usually, h (n) is calculated by Euclidean distance [15]. In our case, a new type of grid used with A* method. This grid is nothing but a random complex constructed roadmap, where the constructed roadmap used as A* grid and the path is a sequence of roadmap's edges as shown in Fig. 6. The solid green line shows g(D), while the dashed line represents h(D) which is the Euclidean distance from node D to goal node. A* heuristic algorithm has two lists. First list called open list which contains all nodes that can be chosen as next node. Second list called is close list, which contains all visited nodes and cannot be visited again. Algorithm choose the next node that has minimum cost function f(n) as shown in Fig. 7. Current node is A, and all possible next nodes are B, C, and D. Next node is C because it has lowest cost. After choosing node C as shown in Fig. 8, open list is B, D, E, and F. it can be noticed node B, and D are already in open list therefore, their cost must be updated. Node D may be chosen as example to update its cost. If the new cost of node D is lower than old one then the cost should be updated.

Otherwise node D maintain its cost. From the new open list, node F is selected and node C is added to close list. The process is repeated till reach goal node. Figure 9 shows A* algorithm flowchart.

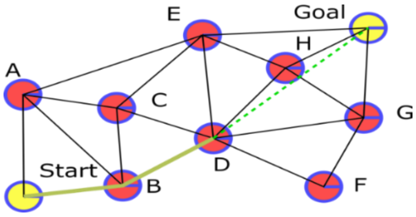


**Fig. 6. Example for A* grid.**

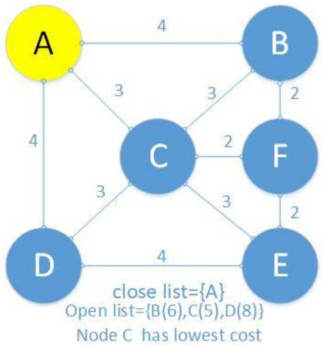


**Fig. 7. Selecting closest node C.**

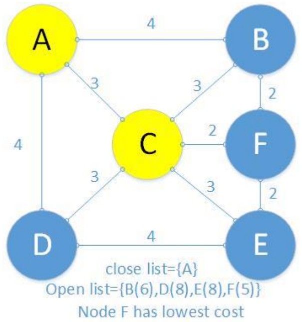


**Fig. 8. Selecting closest node F.**

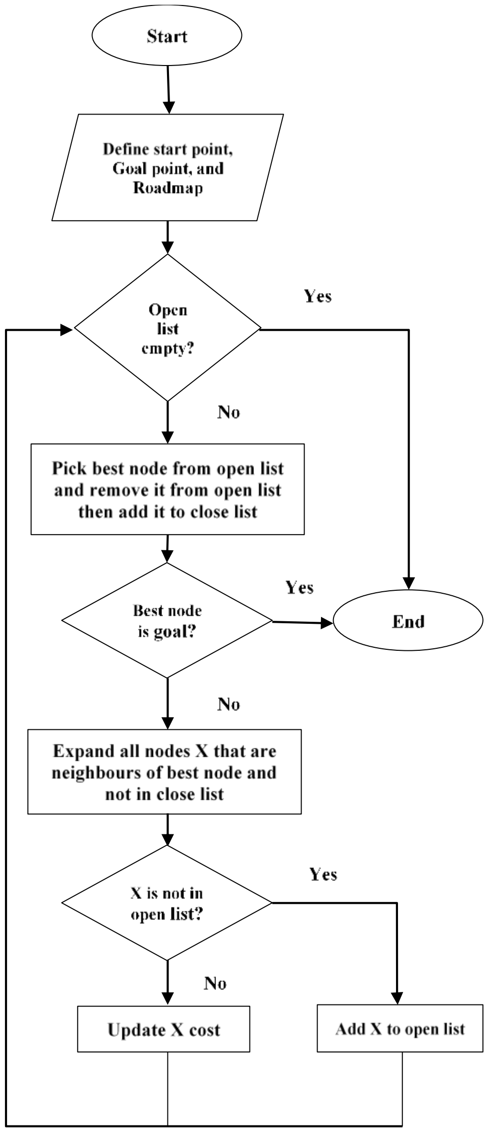


**Fig. 9. A* algorithm flowchart.**

## 6. NURBS Curve Based on PSO

Generally, most path planning algorithms produce straight lines segment. Actually, these paths with straight line segments are undesirable because of discontinuity issue, robot mechanical wear, localization error, and slipping. Besides that, the robot needs to stop at each turning edge, then turn, and lastly move forward again can be considered as another disadvantage. Moreover, Dubin's, Reed, and Shepp's paths generate smooth paths by combining lines and circles but, the path was discontinuous [16].

The most used curves in computer aided design are B-Spline, Bezier, and Non-Uniform Rational B-Spline (NURBUS) curve. Each type of those curve use control points to control their shapes as shown in Fig. 8 and they were studied to use them in path planning. Bezier curve has unwanted properties such as; number of control point will determine curve order which means, high number of control points increase order complexity. Whereas B-spline curves order independent on number of control points which give advantage over Bezier curve [17]. A B-spline curve has $n$ control points $p_i(i = 0,1 \ldots n)$ and k order may be defined by Eq. (20) [17, 18].

$$p(t) = \sum_{i=0}^{n} p_i N_{i,k}(t) \qquad t_{k-1} \le t \le t_{n+1} \tag{20}$$

$N_{i,k}(t)$ are basis function of B-spline with k order defined over the knot vector $T= \{t_0, t1\ldots tk, \ldots t_n, \ldots t_{n+k}\}$. Basis function can be obtained by the following recursive de Boor-Cox Eq. (21) [18].

$$N_{i,1}(t) = \begin{cases} 1, & t_i \le t \le t_{i+1} \\ 0, & otherwise \end{cases} \tag{21}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t) \tag{22}$$

$$t_i = \begin{cases} 0, & i < k \\ 1 + i - k, & k \le i \le n \\ 2 + n - k, & i > n \end{cases} \tag{23}$$

The disadvantage of B-spline is that the curvature of spline is fixed and cannot be changed only if the position of control point is changed. NURBS curve is derived from B-Spline. The only difference is each control point in NURBUS curve has its own weight determine how much the control point can participate in forming spline curve. These weights give high flexibility to modify the shape of spline curve by changing the weight of each control point. NURBS curve can be defined by Eq. (15), where $w_i$ represents the weight of i$^{th}$ control point [17].

$$p(t) = \frac{\sum_{i=0}^{n} w_i p_i N_{i,k}(t)}{\sum_{i=0}^{n} w_i p_i} \qquad t_{k-1} \le t \le t_{n+1} \tag{24}$$

The flexibility of NURBS curve can be used in the path planning, when the generated spline is unfeasible or in other words, the path is not collision free. Increasing the weights of the control points leads to attract the curve towards the control points. Changing the weights of some control points can avoid the collision as shown in Fig. 10. It also noticed that there is a wide range of weights that meet collision free condition but, this range cannot ensure shortest path for these control points. For previous reason, an optimization technique has to be used to optimize all the weights of control points. The final optimized weights ensure that the path is shortest and collision free. Particle swarm optimization (PSO) is used to optimize NURBS weights for its unique searching mechanism, computational efficiency, simple concept, and easy implement

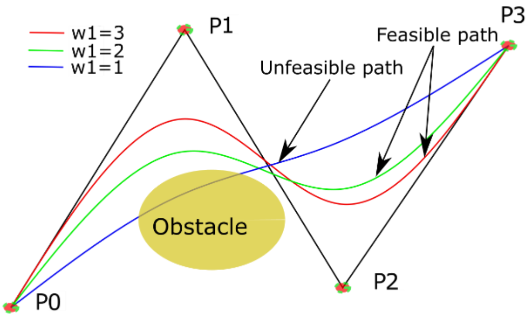


**Fig. 10. Flexibility of NURBS curve.**

Inspired by the social behaviour of bird flocking search for food, PSO has been proposed as optimization technique [19]. It is known as a stochastic, computational intelligence oriented population based on global optimization technique. PSO can be used to solve multi-dimensional problems or functions. PSO is based on population which is nothing but particles. Each problem has fitness function f(x) used to evaluate the solution found by these particles. Each particle has its own position ($x_i$) and velocity ($v_{i)}$ where (i) is particle number. Position of particles represents different feasible solution for the optimization problem. Also, each particle has personal best position ($x_{pbest}$) which represent best solution found by particle i. There is another term in PSO called global best position denoted by ($x_{gbest}$) which represent best solution found by all particles. Particles moves around best solution till reach it. The new position of particles can be calculated by Eq. (25). t is current iteration number and t+1 is next iteration. Previous equation is not applicable until the velocity of next iteration calculated by Eq. (26) [14].

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{25}$$

$$v_i(t+1) = wv_i(t) + c_1r_1\left(x_{pbest} - x_i(t)\right) + c_2r_2\left(x_{gbest} - x_i(t)\right) \tag{26}$$

$c_1$ and $c_2$ is positive acceleration constant of cognitive and social component respectively. $r_1$ and $r_2$ are random numbers number between (0-1). $w$ represents inertia weight. Large value of w is good for global exploration while for local exploration it is preferred to use small value. Normally, PSO algorithm converge to optimal solution iteration by iteration therefore, the search should become more local by reducing inertia weight for each iteration by using Eq. (27) [14].

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{t_{max}}\right)t \tag{27}$$

where $w_{max}$, and $w_{min}$ are maximum and minimum possible value of inertia weight respectively. $t_{max}$ is maximum iteration number. One last thing has to be discussed about Partical swarm optimization is generating initial particles within search space by Eq. (19) [14]

$$x_i = X_{min} + r_1(X_{max} - X_{min}) \tag{28}$$

$$v_i = V_{min} + r_2(V_{max} - V_{min}) \tag{29}$$

$[X_{min}, X_{max}]$ and $[V_{min}, V_{max}]$ are search space limits. In our case, Position of the particles represents the weights of control points of NURBS curve, and the fitness function is the length of the path. Only if the path is collision free the solution is accepted. PSO algorithm can be summarized in following points [19]:

- Define control points, number of population, number of iteration, $c_1$ , $c_2$, $r_1$, $r_2$, $w_{max}$, $X_{min}$, $X_{max}$, $V_{min}$, $V_{max}$, and $w_{min}$
- Generate initial position and velocity for each particle with in problem search space by Eq. (27).
- For each particle, evaluate fitness function Eq. (30), where $n$ is the total number of points that participate in forming path and $x$ is the point coordinate that belongs to the path.
- If the solution of particle cause path collision, then its fitness value assigned as infinity otherwise the solution evaluated by fitness function.

$$fitness\ function = \sum_{i=1}^{n-1} ||x_{i+1} - x_i|| \tag{30}$$

- Compare each particle's fitness value with its old fitness value. If the value of current particles is less than old one then set current value as best value and current position as $x_{pbest}$.
- Identify the particle that has the best fitness value. The value of its fitness function is identified as global best value and its position as $x_{gbest}$.
- For all particles, update velocities and position by Eq. (24) and (25).
- Steps 3-5 repeated till stopping condition is met.

## 7. Results and Discussion

In this section, the proposed approach tested by using it with a complex known environment. The process of constructing roadmap based on artificial potential field can be approved by constructing two roadmaps for the same environment and task. Environment range is X= [-10, 10], and Y= [-10, 10], start point= [-6, 9], and goal point= [4, -9] as shown in Fig. 11.

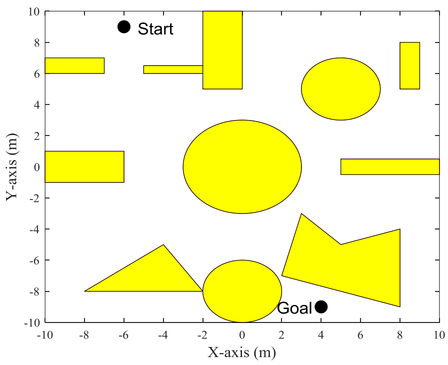


**Fig. 11. Given environment and task.**

Firstly, a random roadmap, as shown in Fig. 12, is constructed by using random distribution. The best path within constructed roadmap is found by A* heuristic method. The length of the path is found to be 32.34498 m. The found path is smoothed and shortened by NURBS curve based on PSO. It can be noticed that eleven control points participated to form NURBS curve. Therefore, there are eleven control points needed to be optimized. After optimization process, it found that the optimized length of the path is 30.1055 m. The path before using NURBS curve and after using it is shown in Fig.13. The convergence of PSO towards best weights is shown in Fig. 14.

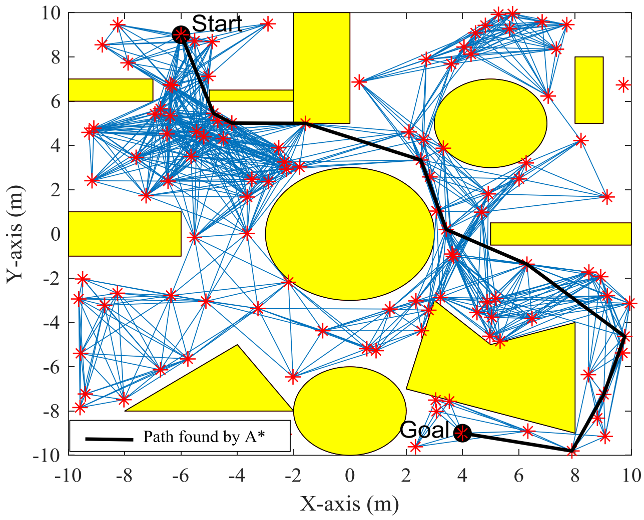


**Fig. 12. Classical constructed roadmap.**

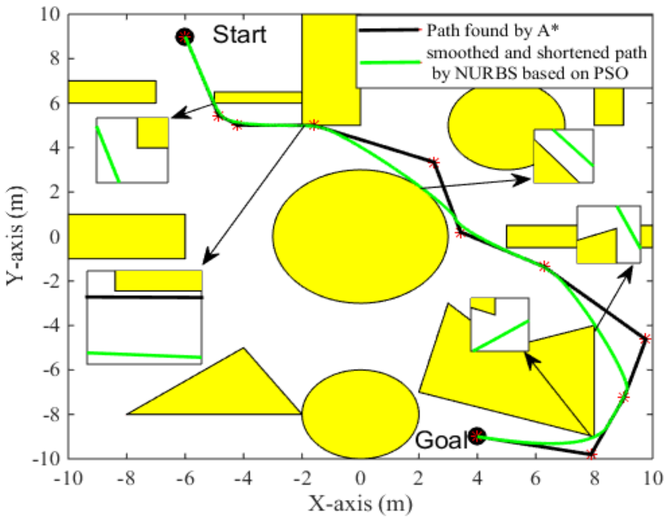


**Fig. 13. Found path before and after using NURBS curve.**

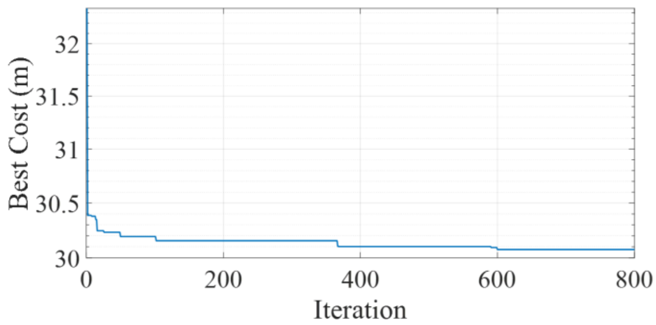


**Fig. 14. PSO convergence towards optimized weights of NURBS curve.**

Secondly, the same randomly distributed nodes are modified by attractive force of artificial potential field, where the locations of nodes are enhanced as shown in Fig. 15. As a result of nodes distribution enhancement, less randomly roadmap based on APF constructed as shown in Fig. 16, which is different from the old one. As in the first roadmap, A* algorithm is used again to obtain the best path within the enhanced roadmap and it is found that the path length is 22.48007 m. The path length is reduced by NURBS curve based on PSO to be 21.1655 m as shown in Fig. 17, where eight control points participate to form a spline curve. The convergence of PSO towards best weights is shown in Fig. 18. Table 1 shows the difference in path length between classical roadmap and enhanced roadmap based on APF, where the path's length of the enhanced roadmap shorter than the path's length of classical roadmap by 10 m. Also, the reader can notice that using NURBS curve based on PSO reduce the length of the path with better smoothing enhancement. This approves the efficiency of the new distribution method and the efficiency of the whole approach. Table 2 shows the parameters of PSO, while Tables 3 and 4 show the optimized weights for each control point of NURBS curve. Control points control the shape of the spline where, PSO choose the optimized weight for each point that ensure shortest path.

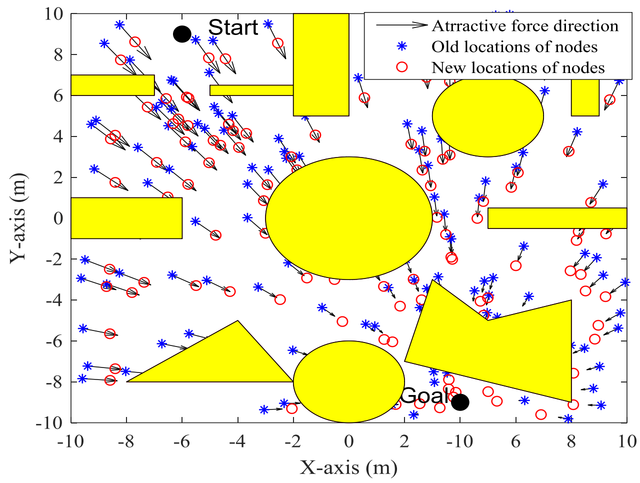


**Fig. 15. New locations of nodes.**

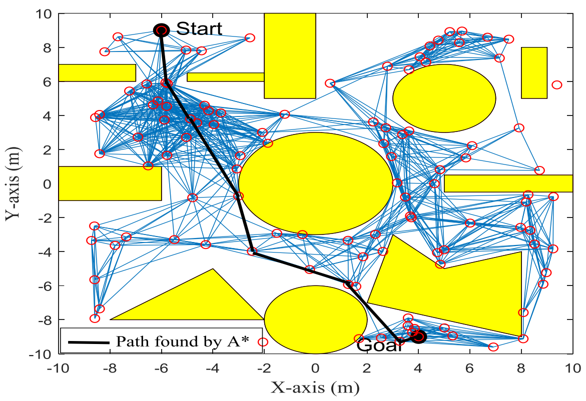


**Fig. 16. Constructed roadmap based on attractive potential filed.**

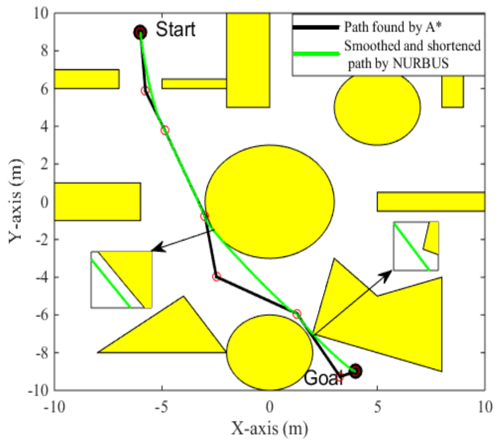


**Fig. 17. Found path before and after using NURBS curve.**

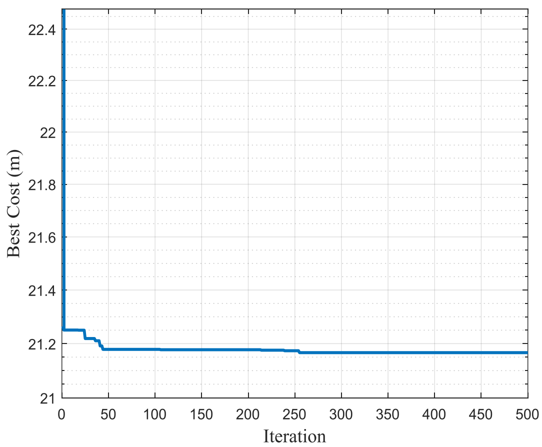


**Fig. 18. PSO convergence towards optimized weights of NURBS curve.**

**Table 1. Comparative results between classical PRM and PRM based on APF.**

| Approach | Nodes | Threshold(m) | A* path length (m) | NURBS path length (m) |
|---|---|---|---|---|
| **Classical PRM** | 117 | 5 | 32.34498 | 30.1055 |
| **PRM based on attractive potential field** | 110 | 5 | 22.48007 | 21.1655 |

**Table 2. PSO parameters.**

| Approach | $w_{max}$ | $w_{min}$ | Population | Iteration |
|---|---|---|---|---|
| **Classical PRM** | 0.9 | 0.6 | 50 | 800 |
| **PRM based on APF** | 0.9 | 0.6 | 50 | 500 |

**Table 3. Results of PSO for classical PRM.**

| Control point number | Optimized weight |
|---|---|
| 1 | 0.468 |
| 2 | 3.6662 |
| 3 | 1.1756 |
| 4 | 2.35 |
| 5 | 0.2562 |
| 6 | 1.349 |
| 7 | 2.772 |
| 8 | 0.349 |
| 9 | 2.3797 |
| 10 | 2.6147 |
| 11 | 3.1534 |

**Table 4. Results of PSO for PRM based on APF.**

| Control point number | Optimized weight |
|---|---|
| 1 | 3.1231 |
| 2 | 1.206 |
| 3 | 2.9642 |
| 4 | 1.8473 |
| 5 | 0.137 |
| 6 | 3.3072 |
| 7 | 0.5088 |
| 8 | 2.8681 |

## 8. Conclusions

This paper proposes a novel approach to solve path planning problem in known complex environment. This approach can be divided into three stages: roadmap construction stage, path search stage, and path enhancement stage. First stage use a new combination among probabilistic roadmap (PRM) and artificial potential field (APF) to construct an enhanced roadmap which gives better path possibilities than classical random roadmap. Second stage is using A* heuristic method to obtain the shortest path (basic path) within classical or enhanced roadmap constructed in the previous stage. The basic path represented by segments of straight lines connecting selected nodes of roadmap. Therefore, the path needs further enhancement, which is done in the third stage by interpolating, smoothing and shortening the path using NURBS curve based on particle swarm optimization. PSO technique has been used to find the optimized weights of the control points of NURBS curve for ensuring shortest, feasible, and collision free path. It is clear that the proposed method can be used in complex and very complex known environment and the path is globally planned before robot moves. From the simulation and results section reader can notice following things:

- The new method of distributing nodes based on artificial potential field enhance nodes to move towards better locations, which mean better path can be found. In other words, APF increase the probability of the node to be distributed in better way, where the path length has been reduced by 30.5% if it is compared with the classical random roadmap.
- Forming NURBS curve based on PSO ensure feasible path for a given control points, where the path length has been effectively reduced and smoothed. The length of the basic path has been reduced by 6.92%, when the classical random roadmap used. Moreover, when the enhanced roadmap used, A* path reduced by 5.84%.
- Because the extracted path from enhanced roadmap is already better than the path extracted from the random roadmap, PSO reaches faster to optimal weights in case of using enhanced roadmap.

At the end, the approach has been simulated using complex environment, and the path has been approved to be shortest, smooth, continues, and collision free. The future directions for developing this work can be done by applying it practically on mobile robot or robot arm manipulator, where the proposed method has to be modified to accommodate robot's constraints. These constrain come from joint limits, robot body, geometry uncertainty, etc. Also, the proposed approach can be modified to use it with dynamic environment by combining D* heuristic method with PRM.

**Nomenclatures**

| | |
|---|---|
| $C_1$ | Positive acceleration constant of cognitive component |
| $C_2$ | Positive acceleration constant of social component |
| $D_{max}$ | Predefined threshold, m. |
| $d$ | Diameter of goal area, m. |
| $E$ | Edge |
| $F$ | Force, N |
| $F_{att}$ | Attractive force, N |
| $F_{rep}$ | Repulsive force, N |
| $f(n)$ | Fitness function, m. |
| $g(n)$ | Accumulative cost, m. |
| $h(n)$ | Heuristic cost, m. |
| $N$ | Set of nodes |
| $N_q$ | Set of neighbours nodes |

| | |
|---|---|
| $N_{i,k}$ | Basis function |
| $P_i$ | Control point i |
| $q$ | Coordinate of node |
| $q'$ | Coordinate of neighbour nodes |
| $q_c$ | Coordinate of closest node |
| $q_{new}$ | New coordinate of node |
| $q_g$ | Coordinate of goal node |
| $R$ | Roadmap |
| $r_1$ | Random number of cognitive component |
| $r_2$ | Random number of social component |
| $T$ | Knot vector |
| $t$ | Iteration number |
| $t_{max}$ | Maximum iteration |
| $U$ | Potential field |
| $U_{att}$ | Attractive potential field |
| $U_{rep}$ | Repulsive potential field |
| $V_{max}$ | Maximum velocity, m/s. |
| $V_{min}$ | Minimum velocity, m/s. |
| $v_i$ | Velocity of iteration i, m/s. |
| $w_i$ | Inertial weight of iteration i |
| $w_{max}$ | Maximum weight of inertia |
| $w_{min}$ | Minimum weight of inertia |
| $X_{max}$ | Upper limit of variable |
| $X_{min}$ | Lower limit of variable |
| $x_i$ | Position of particle i |
| $x_{pbest}$ | Personal best |
| $x_{gbest}$ | Global best |

***Greek Symbols***

| | |
|---|---|
| $\beta$ | Positive coefficient of attractive field |
| $\eta$ | Positive coefficient of repulsive field |
| $\rho_q$ | Distance from current node to goal node |
| $\rho_o$ | Predefined distance determine influence of repulsive force |
| $\rho_q$ | Distance from current node to closest obstacle |
| $\mu$ | Positive fractional number |

**Abbreviations**

| | |
|---|---|
| ACO | Ant colony optimization |
| APF | Artificial potential field |
| GA | Genetic algorithm |
| NURBS | Non-uniform rational b-spline |
| PRM | Probabilistic roadmap |
| PSO | Particles swarm optimization |
| RRT | Rapidly-exploring random tree |

## References

1. Raheem, F.A.; and Hameed, U.I. (2018). Interactive heuristic D* path planning solution based on PSO for two-link robotic arm in dynamic environment. *World Journal of Engineering and Technology*, 7(1), 80-81.
2. Shwail, S.H.; and Karim, A. (2014). Probabilistic roadmap, A*, and GA for proposed decoupled multi-robot path planning. *Iraqi Journal of Applied Physics*, 10(2), 3.
3. Sariff, N.B.; and Buniyamin, N. (2010). Ant colony system for robot path planning in global static environment. *In 9th WSEAS International Conference on System Science and Simulation in Engineering* (*ICOSSSE'10*). 192.
4. Raheem, F.A.; and Hussain, A.A., (2017). Applying A* path planning Algorithm based on modified C-space analysis. *Al-Khwarizmi Engineering Journal*, 13(4), 124-136.
5. Raheem, F.A.; and Hameed, U.I. (2018). Path planning algorithm using D* heuristic method based on PSO in dynamic environment. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, 49(1), 258.
6. Gigras, Y.; and Gupta, K. (2014). Metaheuristic algorithm for robotic path planning. *International Journal of Computer Applications*, 85(3), 1.
7. Raheem, F.A.; and Badr, M.M. (2017). 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)*, 37(1), 317.
8. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; and Overmars, M.H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4), 566-570.
9. Geraerts, R.; and Overmars, M.H. (2006). Sampling and node adding in probabilistic roadmap planners. *Robotics and Autonomous Systems*, 54(2), 169.
10. Hsu, D.; Sánchez-Ante, G.; and Sun, Z. (2005). Hybrid PRM sampling with a cost-sensitive adaptive strategy. *In Proceedings IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 3874.
11. Zhang, Y.; Fattahi, N.; and Li, W. (2013, August). 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, Japan, 1074.
12. M.M. Bader, *Development of a robot path planning Algorithm based on a modified artificial potential field*, M.Sc. Thesis, Department of Control and System Engineering, University of Technology, Baghdad, Iraq, 2017.
13. Li, G.; Yamashita, A.; Asama, H.; and Tamura, Y. (2012, August). An efficient improved artificial potential field based regression search method for robot path planning. *IEEE International Conference on Mechatronics and Automation*, Chengdu, China, 1227-1228.
14. A.A. Hussain, *New Approaches for C-Space map construction for Heuristic path solutions based on PSO technique*, M.Sc. thesis, Department of Control and System Engineering, University of Technology, Baghdad, Iraq, 2017.

15. Ammar, A.; Bennaceur, H.; Châari, I.; Koubâa, A.; and Alajlan, M. (2016). Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments. *Soft Computing*, 20(10), 4149-4171.
16. Elbanhawi, M.; Simic, M.; and Jazar, R.N. (2014). Continuous-curvature bounded trajectory planning using parametric splines. *Frontiers in Artificial Intelligence and Applications*.
17. Elbanhawi, M.; Simic, M.; and Jazar, R.N. (2015). Continuous path smoothing for car-like robots using B-spline curves. *Journal of Intelligent and Robotic Systems*, 80(1), 25-26.
18. Munira, A.; Jaafar, N. N.; Fazilah, A. A.; and Nooraizedfiza, Z. (2014). Review on non-uniform rational B-Spline (NURBS): Concept and optimization. *Advanced Materials Research*, 903, 338-343.
19. Binitha, S.; and Sathya, S.S. (2012). A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering*, 2(2), 140-141.