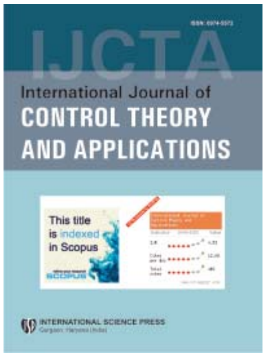

International Journal of Control Theory and Applications

ISSN : 0974–5572

Volume 9 • Number 40 • 2016

Optimal Trajectory Planning Based on Improved Ant Colony Optimization with Ferguson's Spline Technique for Mobile Robot

Priyanka Sudhakara[a] Velappa Ganapathy[a] Karthika Sundaran[a] and Manickavasagam Balasubramanian[a]

[a]*School of Computing, SRM University, Kancheepuram, TamilNadu, India*

E-mail: priyanka.k@ktr.srmuniv.ac.in, ganapathy.v@ktr.srmuniv.ac.in, karthika.su@ktr.srmuniv.ac.in, bmanickavasagam90@gmail.com

Abstract: This paper presents a modified design and development of an improved Ant Colony Optimization algorithm with cubic based spline method for an optimal trajectory planning of a wheeled mobile robot. An Enhanced Ant Colony Optimization (ACO) algorithm for a mobile robot navigation, to find the most optimal path, is enhanced by simplifying the equations, enlarging the area of the simulation framework, extending the task capabilities of the robot and testing the algorithm through simulation. Cubic based-spline called cubic Ferguson's Spline technique is also used to create a shortest path with minimal time, is used to create a smoother trajectory from a set of jagged points. The proposed algorithm has to calculate optimal trajectory for the mobile robot to perform the following tasks: target-searching, boundary following and obstacle avoidance. The total length of the path traversed determines the efficiency of path traced. This proposed method enhances the utility of the ACO algorithm by designing and creating a feasible ACO graphical user interface. Further we carried out the research on ACO algorithm by performing systematic testing and simulations.

Keywords: *Ant Colony Optimization, Cubic Spline, Path Planning, Mobile Robots, Obstacle Avoidance Wall Following.*

1. INTRODUCTION

In the late 1960s, the research for autonomous mobile robot was established. Shakey robot, an autonomous mobile robot developed by Stanford University in 1966 was a great success. Independent planning, controlling, reasoning and various other tasks are said to be autonomy of the mobile robots, in complex situations with the application of artificial intelligence. The development of sensor technologies and computer applications researches on autonomous mobile robots reached a high-rise at the end of 1970s. In the mid 1980's, a platform for mobile robots was started to develop by a large number of well-known world-wide industries. The prime applications of autonomous mobile robot were used in research experiments in research laboratories and institutes and were also used to promote the multi-directional learning of robotics. Since the 1990s, the development of high adaptability of mobile robot control technology, high level information processing and sensor technologies, emerging of programming technology under the real-time scenarios, and the elevated research levels of mobile robots attracted several researches to do more research in robotics fields. Latterly, autonomous mobile robots are widely used in ocean development, military purpose, space exploration, factory automation, atomic energy, mining, construction, agriculture, etc.

An Intelligent autonomous mobile robot is integrated with multiple operations and functions with dynamic perceptions of environments, should be dynamic in behavior controlling, planning, executing and decision-making. Trajectory planning in autonomous mobile robots is one of the core technologies in the study of robotics research, which programs and studies the functions of algorithms. The enhancement of robot's trajectory planning to meet the necessity of the practical applications is a lead to the issues in mobile robotics. The trajectory tracking programming is choosing the collision-free trajectory from the start position to the goal position avoiding the obstacle in the environment, according to a certain evaluation standard. The primary issues in mobile robotics include choosing the optimal path or approximate-optimal one from the source point to the destination point avoiding the collision through the free trajectory and an algorithm to build a model that is reasonable by utilizing the scenario's information provided by the mobile robot. In [1], the model which is developed is able to cope-up with trajectory tracking errors and uncertainty factors in the scenario, minimizing the influence of utilizing external objects to the mobile robot, resulting in a better decision [1].

In this paper, an improved Ant Colony Optimization Combined with a cubic based spline method called cubic Ferguson's spline technique for trajectory planning of a mobile robot is proposed. First the trajectory for the robot has been mapped using improved ACO and then Ferguson's spline method is applied to smoothing the curves of the path that has been tracked. Thus it will be shown that the time taken will be minimized for the robot to reach the target.

The rest of this paper is organized as follows. Section II presents the description of related work. The proposed improved ant colony optimization algorithm with cubic Ferguson's spline technique and its operation are explained in Section III. Simulation results and discussions are given in Section IV and finally concluded the paper with future directions in Section V.

2. RELATED WORKS

The classical method for trajectory planning is performed using simulation test based on graph. The common method is based on the global trajectory planning. Latterly, local and global approaches include voronoi graph, A* algorithm, grid method, visibility graph, topology, artificial potential field method, etc.

The authors [2] have considered the task of robot navigation planning in frameworks in which non-rigid obstacles exist. They have combined simulation of physical objects and probabilistic roadmaps and evaluated an optimal path which has become a trade-off between the distance to be travelled and the cost of deformation. Autonomous robots path planning techniques [3]_are analyzed for optimal motion. A local path planning algorithm based on Bug algorithm is proposed. Using sparse information, detected by the sensors which are closest to the obstacles is used by the Bug algorithm for the robot to travel to the destination. The obstacle borders are identified when the robot circumvents the obstacles so as to satisfy some conditions specified in the algorithm so that the robot safely avoids the obstacles and travel towards its destination. A Point Bug algorithm for robot navigation is also introduced by them in which the use of outer perimeter of an obstacle is minimized by looking at only certain important points on the obstacles' outer perimeter and hence the robot avoids the obstacle and turns towards the goal and finally generates a whole path from source to destination.

Multi-agent trajectory planning with obstacles present in an environment was tackled [4]. A unique optimal multi-agent trajectory planning is described using a model of the environment which is represented graphically. The environment which is represented as a graphical model is changed dynamically avoiding obstacle collisions and also correcting the path accordingly. Changes in the robot paths are introduced in the new algorithm which makes the robot to move away so that collisions are avoided. An optimal path between a source and target points in a specified grid environment [5] is found using genetic algorithm so as to control the mobile robot to reach the target safely. An algorithm which uses search spaces that are complicated and having low complexities and allowing the robot move to four-neighboring states has been developed.

A trajectory planning algorithm [6] either global or local path planning using sensors is able to assimilate the complete updated information about obstacles. The Voronoi Diagram is used to obtain safe areas available in the environment. Then the safe areas are collected by Fast Marching Method to get the traversing path. To provide a safe and reliable robot movement, map-based and sensor-based planning operations are clubbed. A path planning algorithm for traversing a mobile robot is developed with a webcam [7]. This technique is used to find minimal path length for the robot to travel and reach the goal, at the same time avoiding collision with any of the obstacles present along the way. The obstacles' locations are traced by the images captured by the webcam. Then the shortest distance to the goal is estimated using Voronoi Diagram. The authors [8] have proposed an online path planning algorithm, based on the so-called network simplex method. They have assumed that the sensing range of the robot is short compared to the individual path lengths that it plans and hence the scenario is modelled as a graph comprised of nodes and arcs. The network simplex method is adapted to execute the re-planning issue. The utility of the method is demonstrated by combining it with a navigation control strategy.

The D* Lite Algorithm [9] has been extended and the new algorithm has been named as Enhanced D* Lite Algorithm. It prevents mobile robot from traversing across obstacle's sharp corners in between two obstacles and also avoids irregular obstacles. Virtual walls are created to avoid traversing to the same place again and again. It is capable of re-planning quickly to traverse and also remembering the path. The authors [10] have explored the robot's environment using Artificial Neural Networks and Fuzzy Logic techniques and aid the robot to reach its goal following a shortest path. Path Remembering Robot Algorithm is proposed and this will assist the robot to avoid acute obstacles. Virtual Wall building method is also proposed to prevent the robot traversing the same acute obstacles again and again. The Classical Q-Learning [11] algorithm (CQL) has been extended and the new algorithm named as Improved Q-Learning (IQL) algorithm. The CQL utilizes a Q-table to store the Q(S, a) where 'S' are states and 'a' are actions. Thus, it needs an array of size ($n \times m$). In the IQL, it is advantages to store Q-values only at a specific state S. 'n' number of Q-values are there for 'n' states. Apart from storing Q-values to represent the current status of a specific state, 'n' Boolean Lock Variables are required. The Q-value at a particular state is not to be updated further if the Lock variable is at state 1.

A Parallel Elite Genetic Algorithm (PEGA) [12] is applied for generating a global trajectory planning for a mobile robot which is to navigate in a well-organized environment. The PEGA consists of two types and having a migration operator. This operator helps preserving good population diversity, keeping parallelism and preventing premature convergence as compared to that of conventional Genetic Algorithms. Cubic B-spline is used to smoothen the initially created feasible trajectory using PEGA so that a near optimal collision free trajectory can be constructed. One Field-Programmable Gate Array (FPGA) chip using the concept of system on a programmable chip technology and pipelined hardware implementation scheme both used to improve the computational speed, is utilized for trajectory planning and smoothing.

To improve the search ability, an adaptive-velocity-mutated operation [13] technique, integrated with the trajectory planning system, has been proposed. PSO utilizes a framework incorporating crossover and mutation operations and the new framework is named as Evolutionary-Group-Based Particle-Swarm-Optimization (EGPSO) for trajectory planning. For selecting parents in crossover operations, particle updates and replacements, different groups are formed using EGPSO. Fuzzy Controller designed by EGPSO is utilized to navigate a mobile robot in an unknown environment. For merging target seeking and boundary following navigations, a behavior supervisor method is presented. Dead cycles existing in the system are clearly identified and accounted. Fermet points, Improved Dijkstra Algorithm and Delaunay Triangulation methods are utilized to construct a reduced visibility graph with a minimal path length [14].

A composite global method of optimal selection of trajectory and efficient dynamic avoidance of obstacles is presented [15]. For global path selection and dynamic obstacle avoidance the authors have used Distance Transform and GJK (Gilebert-Johnson-Keerthi) algorithm. Whereas the authors [16] have used Tabu list technique for dynamic obstacle avoidance and Honey Bee Mating Optimization (HBMO) algorithm for global optimal path selection. Their simulation results obtained are found to be much better than Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) algorithms. A robot trajectory planning using a wavelet network control scheme is proposed [17]. The optimal Wavelet Neural Network parameters (WNN) and the Proportional Integral Derivative (PID) controller parameters are determined by the PSO. The PSO method and PID controller are utilized for controlling the non-holonomic mobile robot that involves path tracing using two optimized WNN controllers one to control the speed and another to control the azimuth angle. Matlab Simulink is used to model the robot and the PSO algorithm is written in MATLAB codes. Grid method [18] has been used for scenario modelling and to boost up searching speed, ACO algorithm with bipartite parallel searching approach is utilized to boost up searching speed. Trajectory planning of a mobile robot in a static known environment is explained [19] and the planning procedure is split up into simpler tasks. The first task is to create a trajectory from starting point to target point which is free of collisions and PSO is used to create the trajectory. Second task involves using a radial basis function for trajectory generation on the collision free path obtained. The third and final task is to use a PID controller to properly plan the robot path. Without using an optimal path to move from a random starting state to a final target state for trajectory planning, Q-learning can be used to reduce the convergence time [20]. Q-value for the best possible action at a state is stored by the proposed algorithm and hence significant storage is effective. Their proposed algorithm is then compared with classical Q-learning and is studied using Khepera-II robot. Trajectory planning of a mobile robot is attempted by combining cloud theory and rough set [21]. Their existing evolutionary algorithm's shortcoming is precocity and in order to overcome this issue they have used an algorithm which has considerably increased the speed and accuracy of robot movement.

Genetic algorithm has been used for robot path planning in a 3-D static environment to find a collision free optimal path [22]. The whole 3-D space is split into layers of 2-D space and a shortest path is found to reach the goal avoiding obstacles. A cost effective and indigenously developed educational framework to explain concepts involved in robotics and mechatronics areas has been presented [23]. In this framework, the robot is able to change its shape from a wheeled mobile robot to a humanoid thus facilitating conduct of large number of experiments. General Purpose Operation System (GPOS) performance with Real Time Operating System (RTOS) using a mobile robot as a real time application is analyzed [24]. The mobile robot module was implemented on a single board computer having ARM11 as its core. The method used to calculate response is real-feel method which uses dedicated timer and interrupt to calculate the response. A review is carried out bringing out all types of architectures [25] now. Dynamic modelling for motion planning system and geometric modelling for path planning system are developed using a hybrid technology to accommodate the design complexity. Agent based path planning system and motion planning system are used for the path planner control which plays a critical role in controlling the movement of a non-holonomic robot. A model based on machine learning method has been used to achieve early detection of App-DDOS [26] by multitude request flood. The following metrics "Request chain length, request chain context, ratio of packet types, ratio of packet count, route context, router chain context and ratio of request intervals" are defined in the model ARTP.

3. PROPOSED METHOD

3.1. Ant Colony Optimization

Two similar but functionally different algorithms based on the concepts of ACO were written for the purpose of this paper, one for the purpose of performing target-searching in an obstacle filled environment, the second for the purpose of performing boundary-following in a closed area. The pseudo-codes for both algorithms are presented below. Although mentioned separately, it is to be noted that the obstacle avoidance capabilities of the robot have been merged with the target-searching and boundary-following functions, and thus performed as a single task.

Target-Searching Algorithm :

Initialisation:

Load simulation map

Initialise variable

Iterations:

For current iteration

While there are ants moving

For current ant

Calculate probability to move to each neighbour

Store current position into move history

Control ant distance from wall

Prevent backtracking

Prevent 4 square looping

Assign new position based on probability

Move the ant

Check if termination criteria met

End – for current ant

End – while there are ants moving

Pheromone evaporation

Depositing pheromones

End – current iteration

Wall-following pseudocode:

Initialisation :

Load simulation map

Initialise variables

Iterations :

For current iteration

While there are ants moving

For current ant

Assigning probability to move to each neighbour

Store current position into move history

Control minimum ant distance from wall

Control maximum ant distance from wall

Prevent backtracking

Prevent 4 square looping

Assign new position

Move the ant

End – for current ant

End – while there are ants moving

Depositing pheromones

End – current iteration

Blacking out non-explored paths

Considering that this algorithm is to be used on environment sizes much larger than that of previous researchers, many of the general ACO algorithm equations used to calculate various parameter values have been simplified and modified to help reduce calculations and simulation time. A few statements and equations to be noted are as follows:

1. For the Target-searching algorithm, the probability for an ant to move to any one of its immediate neighbors in a single ant step is calculated according to the formula:

$$\text{Probability to move to X} = \frac{\text{Pheromone in X}}{\sum \text{Pheromone in all neighbours}} \tag{1}$$

2. For the boundary-following algorithm, the probability for an ant to move to any one of its immediate neighbors in a single ant step is calculated according to the formula:

$$\text{Probability to move to X} = \frac{\sum \text{Pheromone in all neighbours}}{\text{Pheromone in X}} \tag{2}$$

This equation is such that it promotes the exploration of new walls that have not been discovered by ants.

3. The amount of pheromones deposited on a successful path is determined using:

$$\text{Pheromone deposited} = \left(\frac{1}{N}\right)^{\frac{P}{2}} \tag{3}$$

Where, N = Number of steps taken along the path and

P = Pheromone deposition constant.

Negative pheromones are deposited according to the formula:

$$\text{Pheromone} = \frac{-0.001}{\text{mapsize}} \tag{4}$$

Pheromone evaporation conducted at the end of each iteration:

$$R = E \times P \tag{5}$$

Where, R = Remaining pheromones,

E = Pheromone evaporation constant and

P = Currently available pheromones.

3.2. Enhancement of Ant Colony Optimization Algorithm

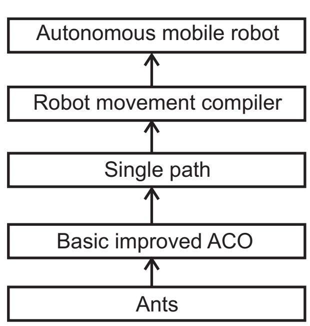

Figure 1: Flow Diagram of ACO

The full extent of this paper involves not only the design and writing of an Amended Ant Colony Optimization algorithm, but also another two related programs that work on top of the ACO algorithm, which ultimately enables real time implementation of ACO in an autonomous mobile robot. The following Figure 1 flow diagram shows the hierarchy of various algorithms created for this paper.

3.2.1. Single Trajectory Selection Algorithm

This algorithm takes as input the pheromone map from either the target-searching or boundary-following function and from it chooses a single best path for the robot to travel on. While seemingly redundant in purpose considering that the ACO algorithm would be able to accomplish the same task, this second algorithm actually helps to reduce the simulation time.

The ACO algorithm is a relatively slow algorithm that requires longer computational time as it employs hundreds of "Ants" over hundreds of iterations. Such long computational times could prove detrimental to the success of its implementation in real time. On the other hand, the Single Trajectory Selection algorithm is a very simple and much faster algorithm, although much of the capabilities available in the ACO algorithm are lost in the tradeoff for speed. With these two options at hand, instead of running the slower ACO algorithm as long as it takes to converge to a single path, our proposed algorithm allows the ACO algorithm to run until a relatively narrow spread of pheromones is found. At this point, the Single Trajectory Selection algorithm would take over the task and perform a much faster and more effective convergence process to a single path.

This Single Trajectory Selection algorithm works in similar ways to the ACO algorithm in that it chooses the next step to take to depending on the amount of pheromones in the four main directions. This is done as shown in the following page.

$$\sum \text{Pheromone in X direction} \;=\; \sum_{1}^{N} \frac{\text{Pheromone M units away}}{\text{M}^2} \tag{6}$$

Where,

N is the distance from the robot's current position to the nearest obstacle in X direction and

$$\text{M} \;=\; 1, 2, 3 \ldots \text{N}$$

The next step to take would be to move in the direction with the most summed pheromones among the 4 main directions. The above formula is such that it places more importance on its immediate surroundings compared to locations farther away from the robot. This is done so that the robot does not get distracted by less optimal paths that run in parallel with the path it is travelling on, and also by patches of high pheromone locations along unsuccessful paths.

This algorithm differs with the ACO algorithm in that it uses only a single agent, the "virtual robot", in a single set of iterations, to travel along the half-converged pheromone path output by the ACO algorithm. It is because of this simplicity that the algorithm gains its speed.

3.3. Cubic Ferguson Splines

Splines are a good choice for robot's trajectory movements for its ease of implementation and high degree of smoothness. Considering the start position P0 and the target position P1 of the spline and their corresponding tangent vector P\`0 and P\`1, we can seek Ferguson spline using the following equation in [27]:

$$\text{X}(a) \;=\; \text{P}_0 f_1(a) \;+ \text{P}_1 f_2(a) + \text{P}_1{}' f_4(a) \tag{7}$$

Where corresponding f_1, f_2, f_3, f_4 are Ferguson-multinomials and a ε [0, 1] is parameter which are described by

$$f_1(a) = 2a^3 - 3a + 1 \quad (8)$$
$$f_2(a) = 2a^3 - 3a^2 \quad (9)$$
$$f_3(a) = a\,(a - 1)^2 \quad (10)$$
$$f_4(a) = a^2\,(a - 1) \quad (11)$$

Equations (7) through (11) simply the source position P0 and target position P1 can be obtained by X(0) and X(l). Values of the positions P\`0 and P\`1 are obtained by derivative substitution.

$$f_1'(a) = 6a^2 - 6a \quad (12)$$
$$f_2'(a) = -6a^2 + 6a \quad (13)$$
$$f_3'(a) = 3a^2 - 4a + 1 \quad (14)$$
$$f_4'(a) = 3a^2 - 2a \quad (15)$$

By using the equations (1) and (12 – 15), we can conclude $P_0' = X'(0)$ and $P_1'\ X'(1)$.

3.3.1. Food Coding and Fitness function

To simplify the issues of trajectory planning, mathematical notations of Ferguson spline are showed in $2d$ space as follows:

$$\begin{aligned} r(a) &= (x(a), y(a)) \\ &= b_0 + b_1 a + b_2 a^2 + b_3 a^3 \end{aligned} \quad (16)$$

Where

$$\begin{cases} b_0 = 2P_0 - 2P_1 + P_0' + P_1' \\ b_1 = -3P_0 + 3P_1 - P_0' + P_1' \\ b_2 = P_0' \\ b_3 = P_0 \end{cases} \quad (17)$$

Each spline is defined only by points P0 and P1 and vectors P_0' and P_1'. In equation (16), two neighboring splines in the string share the corresponding vector and one of the terminal points. The whole trajectory that is defined by total number of variables in 2D space is $4(n + 1)$, where n is the number of splines in the string.

A prime part of an bio-inspired algorithm is to choose a better fitness function for evaluation. Global minimum of this function could correspond to smooth trajectory that is safe and short. The function in [27] consists of two parts that penalize the long distance of the trajectory and the trajectories that causes collision with obstacles. The fitness function is as follows:

$$f = f_1 + af_2 \quad (18)$$

Where a is a weight factor that adjusts the proportion of the length. f_1 is defined by following equation:

$$f_1 = \frac{L}{L_{min}} \quad (19)$$

Where

L is the length of trajectory and

L_{min} is Euclidian distance between source and target point. f_2 is defined by following equation:

$$f_2 = \begin{cases} L, & d_{min} > d_{safe} \\ e^{(D\_(safe+1)/d\_min+1)+1} & d_{min} \le d_{safe} \end{cases} \quad (20)$$

Where d_{safe} is a constant that determine the influence of the obstacles.

4. EXPERIMENTAL RESULTS

The performance of the proposed model was evaluated and tested it in seven various environmental scenarios. In the performance evaluation, we developed various situation scenarios in which source and destination positions were chosen randomly. The scenario was defined as a 10x10 environment includes the randomly positioned obstacles. For finding an optimal path between two positions in any scenarios, ACO algorithm starts with 25 food sources and continued to optimize the Ferguson's spline parameters in continuous 100 iterations cycles. Ferguson's splines was used to develop a trajectory between designated positions (number of splines (n) = 1) and d_{safe}= 0.6 were chosen to consider a safe distance between the obstacles and trajectories.

Figure 2: ACO Mapped Path (Result I)

Figure 3: ACO Mapped Path with Ferguson's Spline (Result I)

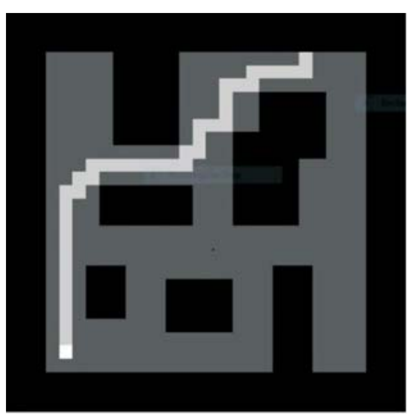

Figure 4: ACO Mapped Path (Result II)

Figure 2 shows the Test I results of the path traced by the bee using improved ACO algorithm. Smoothened curved path using Ferguson spline is shown in figure 3. As the rough turning points of the path are smoothed to curves, it will be easy for the robots to turn and move forward. This will reduce the time taken for the robots to reach the target as the robot doesn't want to stop while turning directions.

Figures 4 & 6 shows the Test II & III results of the path traced by the bee using improved ACO algorithm whereas figures 5 & 7 shows the smoothened curved path using Ferguson spline. Table 1 shows the Time Taken by the mobile robot using improved ACO algorithm and after applying Ferguson spline in the scenarios Test 1, 2 and 3.

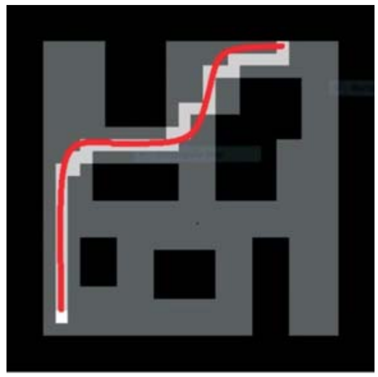

Figure 5: ACO Mapped Path with Ferguson's Spline (Result II)

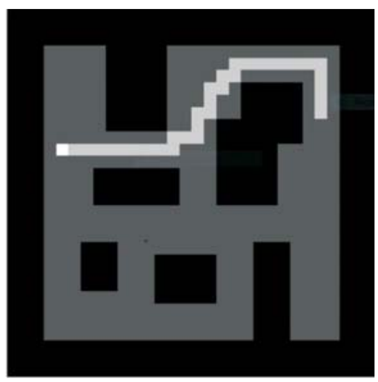

Figure 6: ACO Mapped Path (Result III)

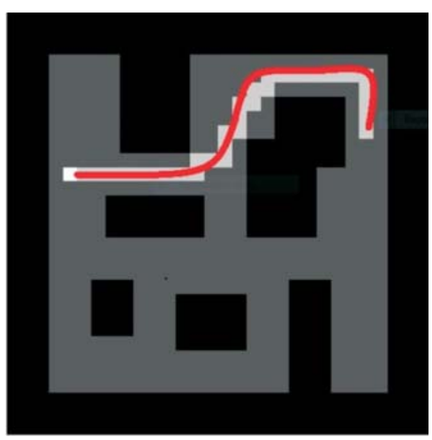

Figure 7: ACO Mapped Path with Ferguson's Spline (Result III)

Table 1
Time Taken using Improved ACO algorithm and after applying Ferguson spline in the scenarios Test 1, 2 and 3

Time Taken	*Test 1 (in sec)*	*Test 2 (in sec)*	*Test 3 (in sec)*
Using Improved ACO algorithm	49.0023	41.0003	33.0019
After applying Ferguson Spline	37.0031	28.0055	17.0041

5. CONCLUSION

In this paper, a result for the issue of trajectory planning is proposed using Ant Colony Optimization and Ferguson Splines. The trajectory between two desired positions was described by ACO algorithm and Ferguson function strings, was used to find the desirable spline parameter values, which finally leads to take an optimal trajectory regardless the obstacles between the two positions. The results in simulations showed a better outcome for ACO algorithm which implicitly indicates that the results are more optimal than other PSO algorithms. Although ACO algorithm took longer to run, ACO algorithm model finds the optimal path for robot motion in less iteration and in less time. For future enhancements, ACO can be tuned to get better outcomes and applications of other spline functions like cubic B-splines can be applied to gain better results with less complexity and more smoothness. Replacing ACO with other evolutionary models as well as changing the fitness function to the trajectory planning issues to obtain better results can be also considered as other future works.

REFERENCES

[1] Tan Min, WANG Shuo.(2013) Research Progress on Robotics. ACTA AUTOMATICA SINICA, 39(7):pp.963-972.

[2] Buniyamin N, Ngah WAJW, Sariff NZ, Mohamad A. Simple local path planning algorithm for autonomous mobile robots. International Journal of Systems Applications, Engineering and Development. 2011; 5(2):151-9.

[3] Kolushev FA, Bogdanov AA. Multi-agent optimal path planning for mobile robots in environment with obstacles. LNCS: PSI'99; Berlin Heidelberg: Springer-Verlag; 2000; 17(55):503-10.

[4] Taharwa IA, Sheta A, Weshah MA. A mobile robot path planning using genetic algorithm in static environment. Journal of Computer Science. 2008; 4(1):341-4.

[5] Garrido S, Moreno L, Blanco D, Jurewicz P. Path planning for mobile robot navigation using voronoi diagram and fast marching. IJRA. 2011:42-64.

[6] Shojaeipour S, Haris SM, Gholami E, Shojaeipour A. Webcam-based mobile robot path planning using voronoi diagrams and image processing. WSEAS International Conference on Applications of Electrical Engineering; Malaysia. 2010. p. 151-6.

[7] Ersson T, Hu X. Path planning and navigation of mobile robots in unknown environments. IEEE/RSJ International Conference onIntelligent Robots and Systems; Maui, HI. 2001. p. 785-864.

[8] Ganapathy V, Yun SC, Chien TW. Enhanced D star lite algo-rithm for autonomous mobile robot. International Journal of Applied Science and Technology. 2011:58-73.

[9] Ganapathy V, Yun SCJNG. Fuzzy and neural controllers for acute obstacle avoidance in mobile robot navigation. IEEE/ASME International Conference on Advanced Intelligent Mechatronics Suntec Convention and Exhibition Center; Singapore. 2009. p. 14-7.

[10] Konar A, Chakraborty IG, Singh SJ, Jain LC, Nagar AK. A deterministic improved q-learning for path planning of a mobile robot. IEEE Transactions on Systems, Man and Cybernetics: Systems. 2013; 43(5):1141–53.

[11] Tsai CC, Huang HC, Chan CK. Parallel elite genetic algo-rithm and its application to global path planning for autonomous robot navigation. IEEE Transactions on Industrial Electronics. 2011; 58(10):4813–21.

[12] Juang CF, Chang YC. Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments. IEEE Transactions on Fuzzy Systems. 2011; 19(2):379–92.

[13] Jan GE, Sun CC, Tsai WC, Lin TH. An O(n log n) short-est path algorithm based on delaunay triangulation. IEEE/ ASME Transactions on Mechatronics. 2014; 19(2):660- 6.

[14] Tamilselvi D, Shalinie SM, Hariharasudan M. Hybrid approach for global path selection and dynamic obstacle avoidance for mobile robot navigation. Advances in Robot Navigation. Mexico: INTECH Open Access Publisher; 2011.

[15] Shoushtary MA, Nasab HH, Fakhrzad. Team robot motion planning in dynamics environments using a new hybrid algorithm (honey bee mating optimization-tabu list). Chinese Journal of Engineering. 2014: 1-8.

[16] Abdallan TY, Hamzah MI. Trajectory tracking control for mobile robot using wavelet network. International Journal of Computer Applications. 2013; 74(3):32-7.

[17] Zhao J, Fu X. Improved ant colony optimization algorithm and its application on path planning of mobile robot. Journal of Computers. 2012; 7(2):2055-62.

[18] Cosic A, Susic M, Katic D. Advanced algorithms for mobile robot motion planning and tracking in structured static environments using particle swarm optimization. Serbian Journal of Electrical Engineering. 2012; 9(1):9-28.

[19] Das PK, Mandhata SC, Behera HS, Patro SN. An improved Q-learning algorithm for path-planning of a mobile robot. International Journal of Computer Applications. 2012; 51(9):40-6.

[20] Ma Z, Ning X. The path planning algorithm and simula-tion for mobile robot. Journal of Theoretical and Applied Information Technology. 2013; 50(3):601-5.

[21] Suresh KS, Vaithiyanathan V, Venugopal S. Layered approach for three dimensional collision free robot path planning using genetic algorithm. Indian Journal of Science and Technology. 2015 Dec; 8(35).

[22] Arif A H, Waqas M, Rahman UU, Anwar S, Malik A, Iqbal J. A hybrid humanoid-wheeled mobile robotic educational platform – design and prototyping. Indian Journal of Science and Technology. 2014 Jan; 7(12).

[23] Murikipudi A, Prakash V, Vigneswaran T. Performance analysis of real time operating system with general pur-pose operating system for mobile robotic system. Indian Journal of Science and Technology. 2015 Aug; 8(18).

[24] Subramanian MB, Sudhagar K, Rajarajeswari G. Design of navigation control architecture for an autonomous mobile robot agent. Indian Journal of Science and Technology. 2016 Mar; 9(10).

[25] Prasad KM, Reddy ARM, Rao KV. Anomaly based Real Time Prevention of under rated App-DDOS attacks on web: An experiential metrics based machine learning approach. Indian Journal of Science and Technology. 2016 Jul; 9(27)..

[26] Zhu Qing-bao.(2005) Ants Predictive Algorithm For Path Planning Of Robot In A Complex Dynamic Environment. Chinese Journal of Computers, 28(11):pp.1898-1906.

[27] J. Ye, R. Qu, “Fairing of parametric cubic splines, “Mathematical and Computer Modelling, vo1.30, no. 5/6, pp.121-131, 1999.