Scientific Research Journal *Vol. 6 No. 1, 65-76, 2009*

Solving Robot Path Planning Problem Using Ant Colony Optimisation (ACO) Approach

Nordin Abu Bakar[1] and Rosnawati Abdul Kudus

Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA (UiTM), Malaysia
[1]Email: nordin@fskm.uitm.edu.my

ABSTRACT

Learning is a complex cognitive process; thus, the algorithms that can simulate learning are also complex. The complexity is due to the fact that little is known about the learning process that can be simulated in a machine. In this study two methods have been chosen to navigate a simulated robot to a target point; namely, Ants Colony Optimisation (ACO) and the Fuzzy Approach. The focus of this paper is primarily the ACO method and the Fuzzy Approach is used as a comparative benchmark. Three scenarios were designed: the Big Hall, the Wall Following and the Volcano Challenge. These experimental scenarios represent the respective navigation frameworks found in the literature used to test learning algorithms. The results indicate that the ACO's performance is inferior to the Fuzzy approach; justification for this has been discussed in relation to previous research in this area. Some future work to investigate this phenomenon further and improve the performance of the ACO algorithm is also presented.

Keywords: *ant colony optimisation, fuzzy approach, machine learning, robot navigation*

Introduction

The challenge in path-planning problems (PPP) is defined as follows: "given a robot and a description of an environment, plan a path between

ISSN 1675-7009

two specific locations. The path must be collision-free (feasible) and satisfy certain optimization criteria". "A description of an environment" tells us that the search space is finite and the world space must be defined before a path is planned. The trend in solving path planning problem is motivated by the current gap between available technology and new application demands [1]. Current industrial robots have low flexibility and pre-programmed sequences of operations that are not able to operate in unexpected situations. The emerging architectures include hierarchical architectures that partition the robot's functionalities into high-level and low-level layers, behaviour-based architectures that achieve complex behaviour by combining several simple behaviour producing units, and hybrid architectures that combine a layered organization with a behaviour-based decomposition of the execution layer [2]. Some researchers solved the free collision path planning problem by solving two sub-problems; firstly, a path is found from the robot's initial position to the goal and secondly the robot approximates this path as it avoids obstacles. This method is restrictive in that the robot is required to stay close to any given path. It would fail if the path moves through a passageway and/or is blocked by an unforeseen obstacle. Local solutions can lead the robot into local minima traps [3].

Robot path planning can be local as well as global. Global path planning usually makes use of a state space or map that has complete knowledge of the environment, whereas local path planning computes an actuation command in reaction to the information acquired by external sensors viewing the immediate environment [4]. A complete knowledge of the environment however, increases the computational complexity, causing the system to be inefficient in real-time applications [5]. The approach taken in this study computes a feasible path before the task is executed. The ACO technique has been adopted for robot path planning and to reach the target; this technique is later compared with a fuzzy logic approach.

The ACO Model

Ants are sometimes annoying especially when they invade our kitchen, going all over the place to search for food. Before you dig out the ant killer, consider how a group of ants can help teach us to solve problems. Try a simple experiment by breaking the trail made by those ants from behind the wall to your sugar jar. Make a line with your finger so that the trail is broken. You will see in an instance that those ants at the back of the trail will disperse and lose direction. But before you know it the broken trail will be amended (maybe slightly off track) and they continue the work as usual. This is the very ability that has stunned scientists – the ability to find the shortest path. There are three ideas from natural ant behaviour that are simulated in the ACO model:

1. The preference for paths with a high pheromone level
2. The higher growth rate of pheromones on shorter paths.
3. The trail mediated communication among ants.

The ACO model has been tested on several transportation problems such as the traveling salesman problem (TSP) and network routing problems. The solutions for both problems have made significant impacts. The ACO model algorithm for solving the TSP is as follows [6]:

Loop
Randomly position total_ants on total_cities
For I: = 1 to total_cities
For k: = 1 to total_ants
Choose the next city to move to by applying a ***probabilistic state transition*** *rule*
End-for
End-for
Update pheromone trails
Endloop

The probabilistic state transition rule [6] is as follows

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(\tau)^{\alpha} \ . \ [\varphi ij]^{\beta}}{\sum_{i \in j_i^k} [\tau_{ij}(\tau)]^{\alpha} \ .[\varphi_{ij}]^{\beta}} \tag{1}$$

where j_i^k is the set of cities still to be visited
Update pheromone trails [6] will use the following rule

$$\tau_{ij}(t+1) \leftarrow (1-\rho) \ . \ \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

where $\Delta\tau_{ij}(t) = \sum_{k=1}^{n} \frac{1}{L^k}$

ACO in the Robot Path Planning Problem

The use of the ACO method in the TSP gives a general framework to solve other problems of a similar nature. The same framework can be used to solve the robot's path planning problem. The aim of the problem is find the shortest tour while avoiding the obstacles [7, 8].

The robot landscape is defined as a two dimensional grid composed of 100 by 100 squares in both the x and y directions. This is the simulated workspace used throughout this experiment to implement the ACO method as an engine to drive the robot across the plain from a defined starting position to a final target destination. In between there will be obstacles to test the vulnerability of the ACO algorithm in manoeuvring the robot to avoid collisions [9].

(x-1,y+1)		*(x+1,y+1)*
	(x,y)	
(x-1,y-1)		*(x+1,y+1)*

Figure 1: An Ant Current Position (bold) with 4 Possible Next Positions (italic)

The ACO is considered a global path planning strategy which requires a complete knowledge of the environment. It searches paths within the valid region and establishes a connection between a start state and a goal state. A global planner stops the search when a valid path is found or no path is detected [5].

The Experiments

Robot path planning problems are one of the more interesting problems, which relates to much artificial intelligence research. This study focuses on how robots can be manipulated to learn the surrounding landscape. There are three primary testing ground landscapes; the Big Hall, the Wall Following and the Volcano Challenge.

The big Hall scenario is designed to be the simplest task that a robot could be assigned to do. A goal-seeking behaviour is adopted to find the target. This task yielded maximum performance for both learning algorithms; namely, the fuzzy approach and the ACO.

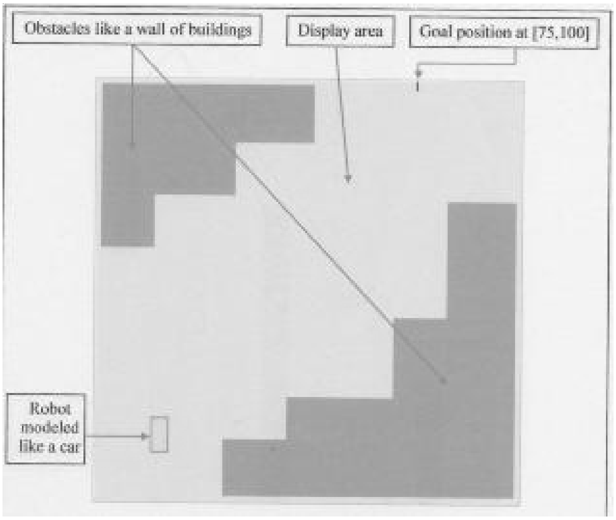

Figure 2 : The Big Hall Set up

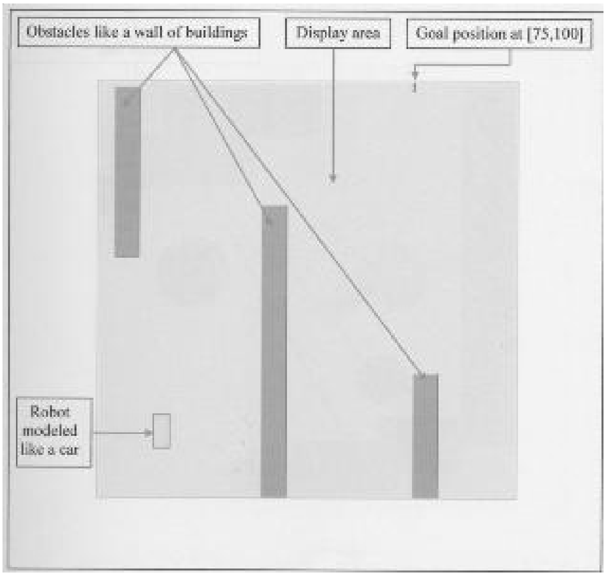

Figure 3: The Wall Following Set Up

The Wall Following scenario is designed to challenge the machine's learning capability. The simulated robot will need to recognize the obstacles and try to avoid them whilst searching for the target. The robot was stationed at different starting positions prior to the search and find activities. The significance of these different starting points is reflected in the results presented in the latter part of this report. Basically the increased distance the robot has to travel to the target directly impacts on the performance of one algorithm, but is insignificant to the other.

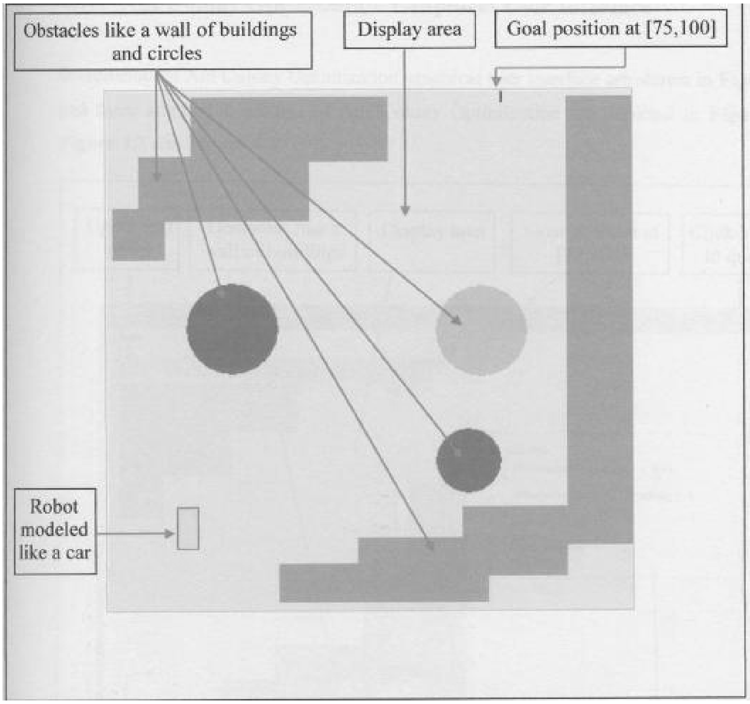

Figure 4: The Volcano Challenge Set Up

The volcano challenge increases the number of obstacles thus increasing the difficulty level for the machine learning algorithms, which must drive the robot to the target.

The Results

The performance for the two learning algorithms; the Ant Colony Optimization (ACO) method and the Fuzzy Logic Approach (Fuzzy), have been evaluated with respect to the distance travelled to reach the destination and the time required to find the target.

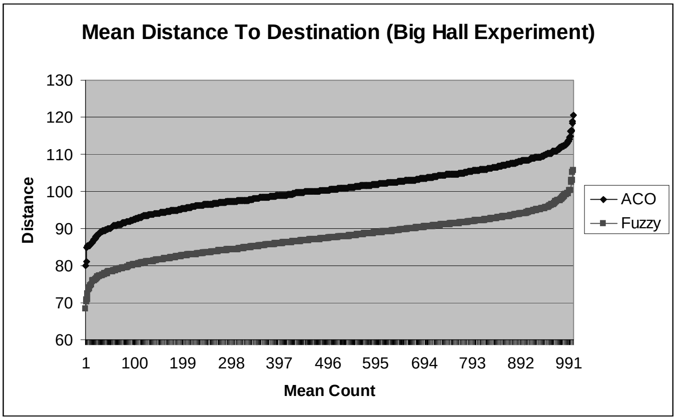

Figure 5: The Big Hall Experiment

The results of the Big Hall experiment are presented in Figure 5 and clearly indicate that the Fuzzy approach requires shorter distance to reach the destination. The average values presented are from 20 different starting positions.

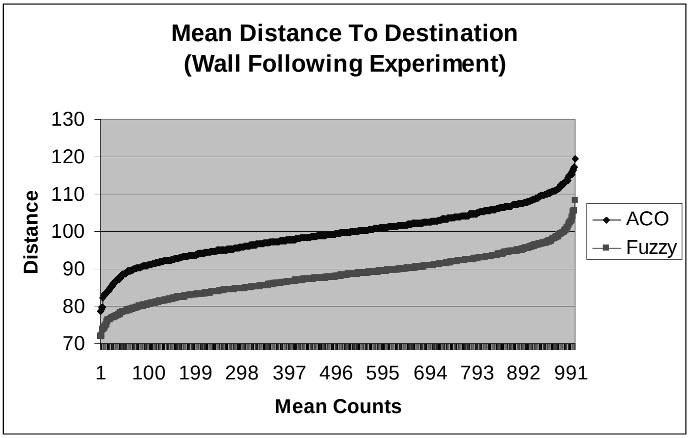

Figure 6: The Wall Following Experiment

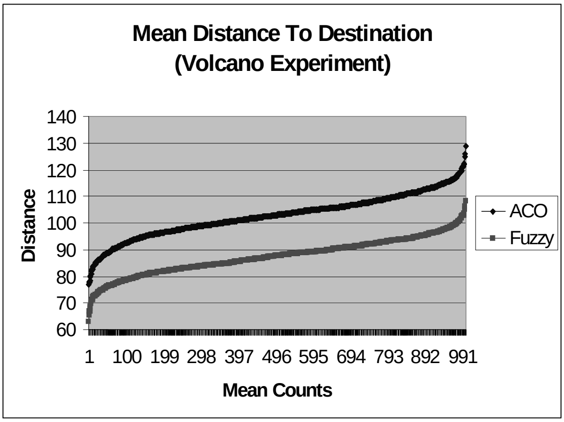

Figure 7: The Volcano Challenge

The distance results for the wall following experiment clearly indicate that the fuzzy approach performed better than the ACO method.

The volcano challenge, as with the other two test scenarios, shows a markedly better performance by the fuzzy method.

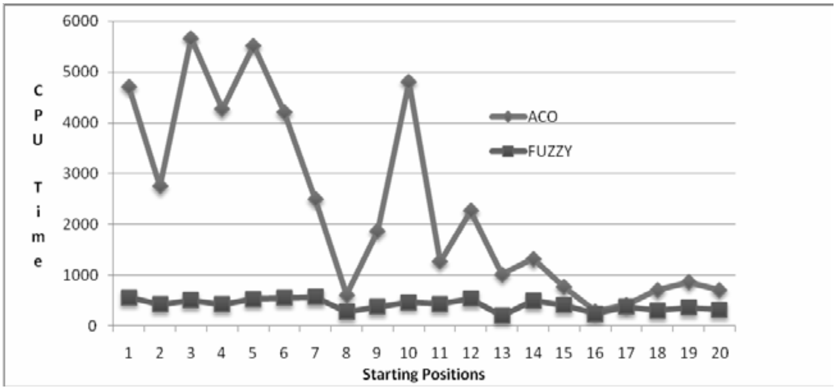

Figure 8: Time Performance for the Big Hall Experiment

Discussion

The results for the distance traversed by the robot from an initial position to the goal position, Figures 5, 6 and 7, indicate that both methods are

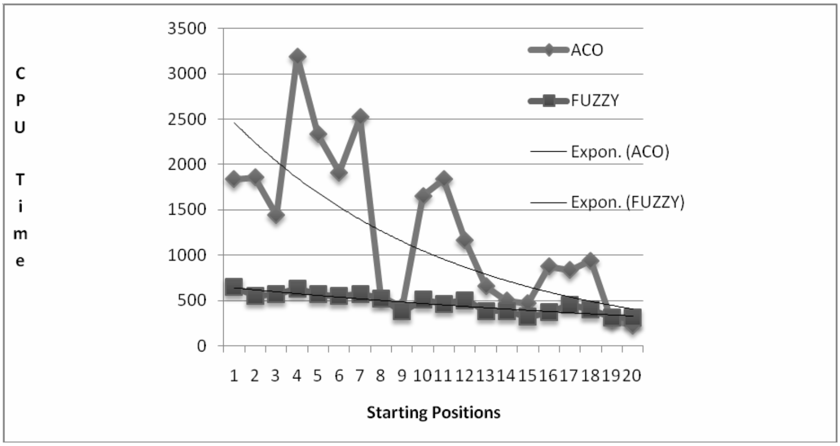

Figure 9: Time Performance for the Wall Following Experiment

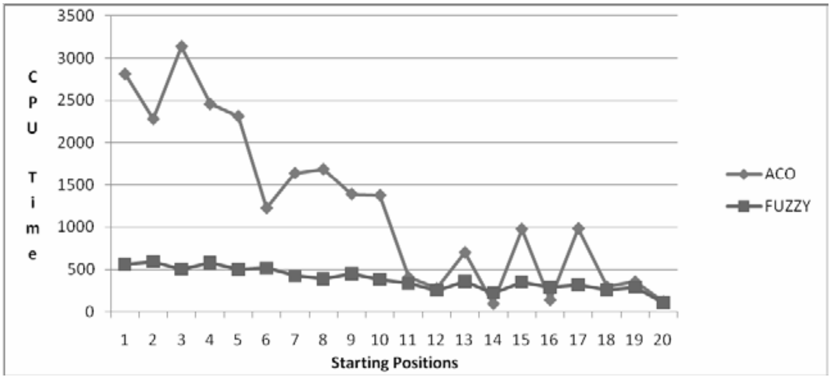

Figure 10: Time Performance for the Volcano Challenge

satisfactory in terms of goal seeking, wall following and obstacles avoidance. Both methods are capable of find the shortest path from an initial position to the goal position, while avoiding obstacles in a reasonable time. The ACO method consistently requires the robot to travel a greater distance in all three test scenarios.

For the CPU times required to complete the three scenarios, Figures 8, 9 and 10, clearly indicates that in most cases, the Fuzzy approach requires less CPU computational time than the ACO method. Specifically, the overall performance of Ant Colony Optimization in the three scenarios; the Big Hall, the Wall Following and the Volcano Challenge, indicates that the Ant Colony Optimization method consumed double the CPU time the Fuzzy approach required. A reasonable explanation for this is that the Ant Colony Optimization algorithm works by reinforcing good solutions and therefore more CPU time is required in order to generate good solutions.

From the overall performances of the two implemented methods, Figures 5-10, it can be concluded that the Fuzzy approach outperforms ACO in essentially all cases. The Fuzzy approach performances are better in terms of distance traversed and CPU time consumption due to the fact that the fuzzy based navigation strategy employs a sensor to guide local planner navigation and thus minimizes collision with stationary obstacles. In addition, the Fuzzy approach adopted navigation method, which is based on a confined sensor region surrounding the actual state, does not consider the entire state space. Hence the amount of computation time required for the Fuzzy Controller System is markedly reduced by using only the nearest obstacles to determine the robot direction.

The fuzzy based navigation can be described as follows: at each iteration, navigation makes a guess as to which is the best actuating command to be sent to the robot so that the robot's state is altered until it comes closer to the goal state. Any state change experienced by the robot is interpreted by the fuzzy based navigation as a new environmental situation to which the navigator reacts with a new actuating command. The fuzzy based navigation provides a collision free path in many cases, even though the navigator is a simple reactive mechanism with only a simple rule base. Conversely, the ACO method, which employs an Elitist Ant System and four neighbourhood stochastic search technique for path finding does not perform better than the Fuzzy approach in many of the scenario runs. This may be attributed to the fact that the Ant Colony Optimization method adopts a global navigational strategy, which requires complete knowledge of the entire environment. The ACO method searches for a path inside the region of valid configurations, thereby connecting a start state with a goal state and this exploration of the robot's entire state space is very time consuming. Based upon the results presented the Ant Colony Optimization method consumes essentially double the CPU time for the Fuzzy Navigator System.

Conclusion

This paper has discussed in detail the experimental results for the implementation of the Fuzzy approach and Ant Colony Optimization for robot path planning problems, namely the Big Hall, the Wall Following and the Volcano Challenge. The performances of the proposed methods have been evaluated with respect to goal seeking, wall following and obstacle avoidance behaviour, which was measured in terms of distance traversed from the initial position to the goal position and the CPU time consumed in order to complete the task.. Each of the scenarios considered was designed with a different environment and stationary obstacles configurations. It has been shown that both algorithms performed satisfactorily and are capable of directing the robot through the simulated environments quite well.

The main drawback of the suggested Fuzzy approach is that it is possible for the robot to get trapped within a particular obstacle configuration; this is known as dead lock. However, even if the robot is in dead lock, it behaves well and will not collide with obstacles. If no valid path exists, the system will behave as if it were in dead lock. Different strategies have been suggested to overcome such dead lock situations, these include Multi Level Path Planning Strategy, via point and random search.

The performance of both methods has been compared, and their potential and limitations identified. From the experiments performed the Fuzzy Navigator System outperforms the Ant Colony Optimization method in essentially all cases. The fuzzy approach is better in terms of distance traversed and CPU time consumed. In conclusion although the ACO method is out performed in the three scenarios considered in this paper it is possible that more complex scenarios, such as a maze, which require an overview of the overall environment may result in the ACO method outperforming the Fuzzy method.

References

[1] S. G. Tzafestas, 1999. *Advances in Intelligent Autonomous Systems*, Kluwer, Boston.

[2] A. V. Topalov and S. G. Tzafestas, 2000. Layered Multi-Agent Reactive Behaviour Learning in a Robotic Soccer, Proc. *SYROCO*.

2000: 6th IFAC Symp. on Robot Control, Vienna Univ. of Technology, Vienna, Austria.

[3] R. A. Brooks, 1986. A robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, RA-2(1), pp. 14-23.

[4] K. Althoefer, 1996. *Neuro-Fuzzy Path Planning for Robotics Manipulatos*, PhD Thesis, King's College London, U.K.

[5] M. Majdi, S. A. Hassan and S. Soleimanpour, 2008. Multi AGV Path Planning in Unknown Environment Using Fuzzy Inference Systems, *IEEE*.

[6] M. Dorigo, *et al*., 1991:1996. IRIDIA, Universite Libre de Bruxelles, Belgium, *IEEE*.

[7] J. Gil de Lamadrid and M. Gini, 1990. Path Tracking through Uncharted Moving Obstacles, *IEEE Transactions on System, Man, and Cybernetics*, 20(6), pp. 1408-1422.

[8] J. R. Firby, 1987. An Investigation into Reactive Planning in Complex Domains, *AAAI Conference*.

[9] M. Kaufman, 1987. An Architecture for Intelligent Reactive Systems.