ANT COLONY BASED PATH PLANNING ALGORITHM FOR AUTONOMOUS ROBOTIC VEHICLES

Yogita Gigras[1] and Kusum Gupta[2]

[1]Department of Computer Science and Engineering, ITM University Gurgaon, India.
gigras.yogita@gmail.com
[2]Department of Computer Science, Banasthali University, Banasthali, India.
gupta_kusum@yahoo.com

ABSTRACT

The requirement of an autonomous robotic vehicles demand highly efficient algorithm as well as software. Today's advanced computer hardware technology does not provide these types of extensive processing capabilities, so there is still a major space and time limitation for the technologies that are available for autonomous robotic applications. Now days, small to miniature mobile robots are required for investigation, surveillance and hazardous material detection for military and industrial applications. But these small sized robots have limited power capacity as well as memory and processing resources.

A number of algorithms exist for producing optimal path for dynamically cost. This paper presents a new ant colony based approach which is helpful in solving path planning problem for autonomous robotic application. The experiment of simulation verified its validity of algorithm in terms of time.

KEYWORDS

Ant colony optimization(ACO).

1. INTRODUCTION

Today mobile robots operate not only in the industrial, commercial or military environment but also interact with humans about the decision-making process. In dynamic and unknown environment, robots must have a capability of processing large amount of information, make planning and navigation decisions very quickly. Conventional methods which address robotics control issues rely upon strong mathematical modeling and analysis. The various conventional approaches are suitable for control of industrial robots and automatic guided vehicles which operate in controlled environments and perform simple repetitive specific tasks that require only end-effectors positioning or simple motion along fixed paths [6]. However, operations in unstructured environments require robots to perform more complex tasks for which analytical models for control can often not be determined . Even in cases where models are available, uncertainty and imprecision are generally not accounted sufficiently. This gives the space for exploring unconventional soft computing techniques for the control of robotic systems. The aim of soft- computing is to tolerate imprecision, uncertainty, and approximation to achieve robust and low cost solution in small timeframe. Where mobile robots[8] have additional processing needs i.e. used to collect, plan and execute their route as well as perform specific tasks to achieve a specific mission. An autonomous mobile robot[7] requires efficient path planning approach. Path planning is one of the important problems in the field of robotics as it tries to find out the optimal path from source position to target position. Besides optimization, it helps the robot to move without any collision in the entire path as it follows from initial position to final

DOI : 10.5121/ijaia.2012.3604

position [4-5]. This means that an algorithm has to avoid the obstacles and reach the destination point in minimum time. This comes under the domain of robot navigation. The path planning can be done in a static as well as dynamic environment. In a static environment, the robot map is constant and does not change with respect to obstacle as it is not moving. In a dynamic environment the maps keeps on changing due to the moving obstacles. For this path planning, a new learning algorithm i.e. ant colony optimization (ACO) is chosen in this paper [2-3].

ACO is a metaheuristic approach for solving the combinatorial optimization problem. In simulation, it has been shown that ACO finds out the optimal path from source point to destination point. It was inspired by real ant colonies as they are capable of finding the shortest path from nest to food source by depositing the pheromone information, where pheromone is a volatile chemical substance deposited by the ants. While walking an ant deposits pheromone on the path, and probabilistically, chooses that path which has higher amount of pheromone concentration.

2. ACO

The ACO algorithm is applied for bypassing the obstacles in the dynamic environment. ACO is used to solve the optimization problem that can be expressed in terms of feasible paths on a graph. ACO tries to find that path whose cost is minimum. The information that the ants collect during the search process is stored in the pheromone trail τ with edges. The ant cooperates in finding the solution by exchanging information by the help of the pheromone chemical [9]. Edges are also associated with heuristic value η which represents information about the problem instance or its value is inverse of the distance. This algorithm is implemented in two steps. In first step, the edge is selected by a probability formula. Assume that ant k is located at node i, uses the pheromone τ_{ij} deposited on the edge (i,j) to compute the probability of choosing next node is given below by the equation 1:-

$$\mathrm{Pij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}}{\sum_{j \varepsilon N_i(k)} \tau_{ij}^{\alpha}} & \text{if } j \varepsilon N_i(k) \\ 0 & \text{otherwise} \end{cases}$$

Equation 1

Where α denotes the degree of importance of pheromone trail and $N_i^{(k)}$ indicates the set of neighbour of ant k when located at node i expect the predecessor node (the last node visited by ant k). This will prevent the ant k for returning to the same node. An ant travels from node to node until it reaches to the target node and come back to the starting node.

In the second step, once all the ants complete their tour, then updating the pheromone trail takes place by the equation 2.

$$\tau_{ij} = (1-\rho)\,\rho + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$$

Equation 2

Where $\rho \; \varepsilon \, (0,1]$ is the evaporation rate and $\Delta\tau_{ij}^{(k)}$ is the amount of pheromone deposited on the edge (i,j) selected by the best ant k. The motive of pheromone updating is to increase the pheromone value associated with best path. The pheromone deposited on arc (i, j) by the best ant k is $\Delta\tau_{ij}^{(k)}$.

Where

$$\Delta\tau_{ij}(k) = \frac{Q}{L_k}$$

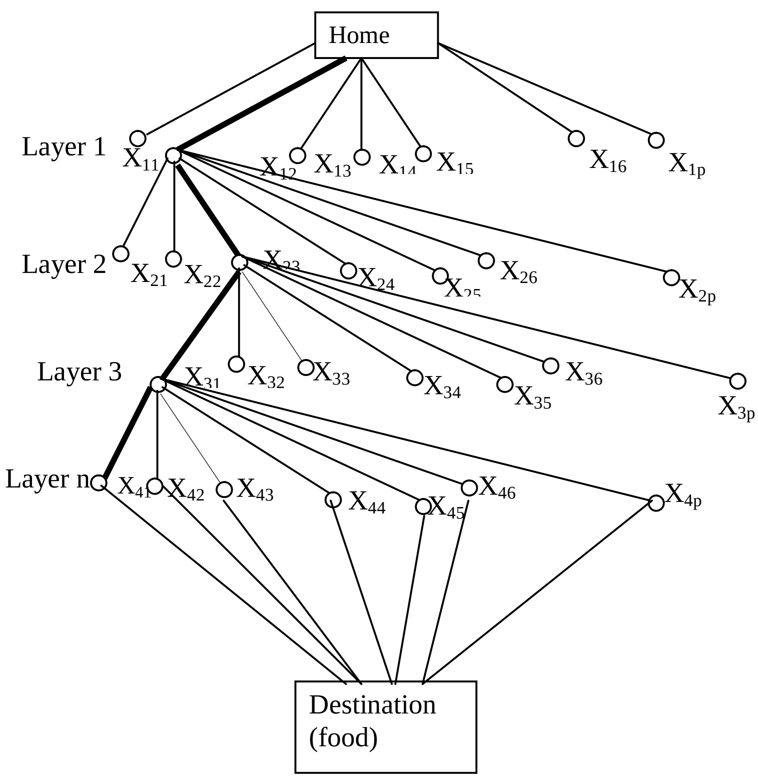
Where Q is a constant and L_k is the length of the path traversed by the best ant k. This equation is also implemented as:-

$$\Delta\tau_{ij}^{(k)} = \begin{cases} \dfrac{\varsigma f_{best}}{f_{worst}} & if (i,j) \varepsilon tour \\ 0 & otherwise \end{cases}$$

Where f_{worst} is the worst value and f_{best} is the best value of the objective function taken by the N ants and ς is parameter for controlling the scale of global updating of pheromone. The larger value ς, signifies more pheromone deposited on the global best path and better exploitation ability [1].

3. ACO Algorithm

The step-by-step procedure of the ACO algorithm for finding the optimal path on multilayer graph (Figure 1) is described here



“Figure 1: Optimal path on multilayer graph”

1) Assume the number of ants in a colony is N.
2) No. of layer in a graph is n.
3) Each layer contain p nodes i.e. (x_{i1}, x_{i2},……, x_{ip}).
4) Initially all the edges contain equal amount of pheromone $\tau_{ij}^{(1)}{}_{=1}$ where superscript denotes the iteration number.
5) Set the iteration number l=1.
6) Compute the probability p_{ij} of selecting edge x_{ij} as

$$P_{ij} = \frac{\tau_{ij}(l)}{\sum_{m=1} \tau_{im}(l)}; i = 1, 2 \ldots .., n$$

$$\text{and} \quad j = 1,2 \ldots .n.$$

Assume α=1

7) Generate N random numbers r_1, r2, r3.....r_n in the range (0,1), one for each ant.
8) Calculate the cumulative probability range associated with the different path using roulette- wheel selection.
9) Determine the path taken by ant k as one of the cumulative probability which Include r_k random number in it.
10) Repeat the step 9, for all the layers i=1, 2, 3……..n.
11) Calculate the objective function values for all the ants $F_k=F(X^{(k)})$; k=1,2,………n Find out the best and worst paths among the N paths by different ants.
12) This process is assumed to be converged if all N takes same best path.
13) If convergence is not achieved then all the ants return home and start again in search of food.
14) Set l=l+1.
15) Update the pheromone on arc when all the ants complete their tour

$$\tau_{ij}^{(k)} = \tau_{ij}^{(old)} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$$

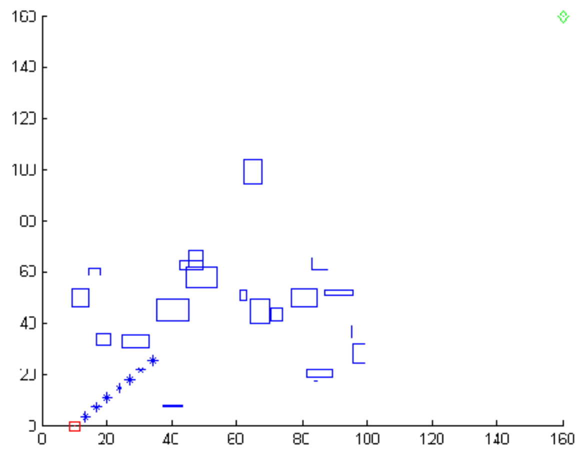
16) Set $\tau_{ij}^{(old)} = (1-\rho)\,\tau_{ij}^{(l-1)}$
17) Repeat this process until $l=l_{max}$ (maximum no. of iteration).

4. ROBOTICS PATH PLANNING

We introduce the simulation of a robotics path planning to justify our proposed method. The path planning is used to obtain the shortest path from a given starting point to a target point without any collision of robot with obstacles. The environment taken is changeable and complex to show efficiency of the algorithm .

4.1 Results and discussion

In our experiment, we assume a robot of asterisk structure, where the source is of square shape and the target is of diamond shape in moving space of 160*160 cm^2. The source co-ordinate (x_s,y_s) and target coordinate (x_t,y_t) are fixed and the robot moves forward from source-end to destination-end in a straight line without colliding with twenty randomly generated rectangular obstacles in the range [1,100]. But the robot does not know the position of obstacles as we consider the dynamic environment.



"Figure 2: Robot encounter with obstacle"

The robot is moving in a straight line with the step size of 5 units by using two variable d and θ .Where d= $\sqrt{(x_t - x_s)^2 + (y_{t-}y_s)^2}$ and $\theta = \tan^{-1}(abs(y_t - y_s)/abs(x_t - x_s))$.

The next point selected by the robot as a source point (x_{snew}, y_{snew}) where $x_{s_{new}} = x_s + stepsize * \cos(\theta_{new})$ and $y_{s_{new}} = y_s + stepsize * \sin(\theta_{new})$ and new $\theta_{new} = \tan^{-1}(abs(y_t - y_{s_{new}})/abs(x_t - x_{s_{new}}))$ until it encounter with the obstacle (figure 2). At that time robot moves back with 3 units i.e. $x_{s_{new}} = x_s - 3 * stepsize * \cos(\theta_{new})$ and $y_{s_{new}} = y_s - 3 * stepsize * \sin(\theta_{new})$ and again calculate the value of θ_{new} with the above mentioned θ_{new} formula. At this place we have to apply the ACO algorithm to bypass the obstacles.

Assume the number of ants is equal to 50 and take eight discrete angles for path chosen by the ants that is ants_path = [0,π/4, π/2, 3 π/4, π,5 π/4,3 π/2,7 π/4] after taking the three steps in backward direction by the robot. There are eight new paths (x_{ants_new}, y_{ants_new}) that ants choose can be calculated by equation 3:-

$$x_{anew} = \cos(ants\_path(i))$$
$$y_{anew} = \sin(ants\_path(i))$$

Equation 3

Initially equal amount of pheromone τ_{ij} in all eight paths τ_{ij} =1. The probability of selecting the path by an ant is calculated by given equation 4.

$$P_{ij} = \frac{\tau_{ij}}{\sum_{m=1} \tau_{im}}$$

Equation 4

The specific path chosen by any ant can be determined by a random number generated in the range (0,1). For this, we have to calculate the cumulative probability ranges associated with discrete paths. The path taken by k^{th} ant as the one for which cumulative probability range includes the value of corresponding random number generated for k^{th} ant.

If an ant goes inside the obstacle then the value of objective function i.e. fun_val $=10^5$, it implies that the ant never reach to the destination point otherwise the value of objective function is computed by equation 5.

$$\text{fun\_val}=\sqrt{(x_t - x_{ant\_new})^2 + (y_t - y_{ant\_new})^2}$$

Equation 5

Evaluate the best path among the eight path chosen by different ants is F_{best}=min (fun_val) and correspondingly update the value of pheromone by equation 6.

$$\tau = \tau + 2 * \min(\text{fun\_val}) / \max(\text{fun\_val})$$

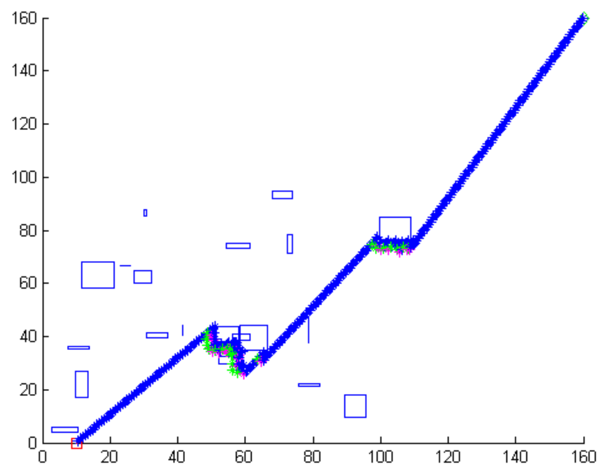
Equation 6

For other paths the value of pheromone is updated by equation 7.

$$\tau = \tau * (1-\rho), \rho = 0.5.$$

Equation 7

It gives the best path for the first step. Apply the same procedure five times to bypass all the obstacles and reach to the destination by following the optimal path. The simulated result in Matlab is shown in figure3:-



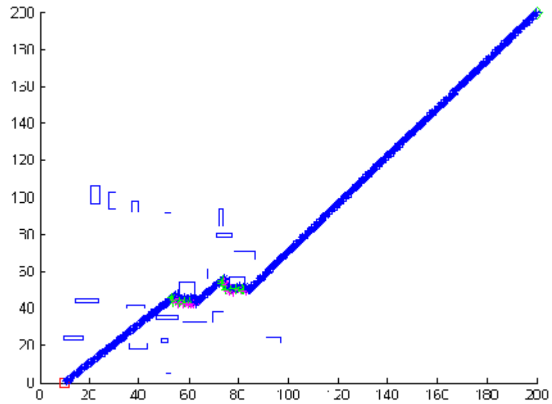
"Figure 3:Robot reaches at destination-end by following optimal path"

We check this algorithm by considering many constraints which is explained in the following cases:-

Case 1

Increase the distance between source and destination

Increase the distance of target from (160,160) to (200,200) apply the same algorithm and find simulated result.

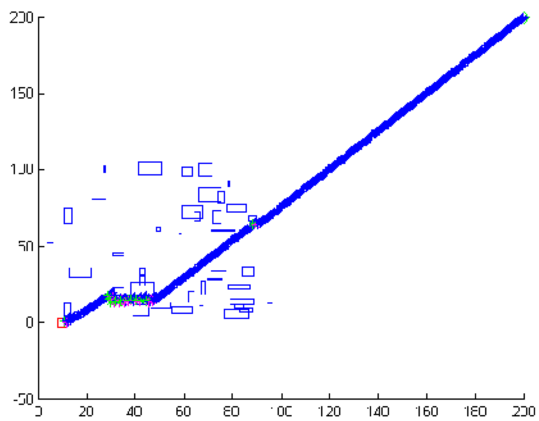


“Figure 4: Increase the distance between source and destination”

Case 2

Increase the number of obstacles

Increase the number of obstacles from 20 to 40 again implements this algorithm to verify its results

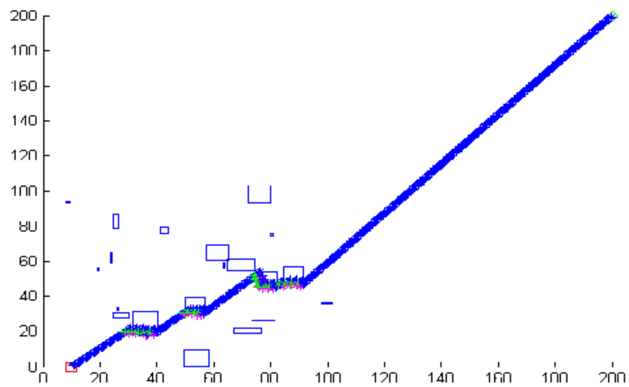


“Figure 5: Increase the number of obstacles”

CASE 3

Increase the number of ants

Increase the value of ants from 50 to 150; again implement the algorithm to validate it.



“Figure 6: Increase the number of ants”

This algorithm requires the elapsed time of 27.911018 sec.

5. CONCLUSION

The ACO algorithm is a kind of intelligent bio-inspired optimization technique which is applied widely for combinatorial optimization problems. The ACO algorithm not only has the feature of finding the global optimization but also has the feature of high performance communication collaboration between individuals and effective binding with information. In this paper, we introduced the ACO algorithm into the path planning of robot for bypassing the obstacles in dynamic environment. The simulation results validate the effectiveness of proposed ACO algorithm as it requires the elapsed time of 27.9 sec approximately.

6. REFERENCES

[1] Chia-Feng Juang, Senior Member, IEEE, and Chun-Ming Lu (2009). Ant Colony Optimization Incorporated With Fuzzy Q-Learning for Reinforcement Fuzzy Control. IEEE Transactions on Systems, Man, and Cybernetics —Part A: Systems and Humans,Vol.39, No. 3,pp 597 – 608.

[2] Gireesh Kumar .T, Vinodh. P. Vijayan (2007). A Multi-agent Optimal Path Planning Approach to Robotics Environment. International Conference on Computational Intelligence and Multimedia Applications, Vol.1, pp 400 – 404.

[3] Rahul Kala, Anupam Shukla, Ritu Tiwari (2009). Fusion of Evolutionary Algorithms and Multi-Neuron Heuristic Search for Robotic Path Planning. World Congress on Nature & Biologically Inspired Computing, pp 684-689.

[4] M. Sabry Hassouna, Alaa E. Abdel-Hakim, and A.A. Farag (2005). Robust Robotic Path Planning Using Level Sets. IEEE International Conference on Image Processing, Vol. 3,pp. III-473-6.

[5] Yili Fu, Han Li and Yulin Ma (2006). Path Planning of Cooperative Robotics and Robot Team. IEEE International Conference on Robotics and Biomimetics, pp 1250 – 1255.

[6] Araki M., PID Control (2010). Published in ELOSS of Control Systems, Robotics, And Automation – Vol. II, pp. 58-79.

[7] K.S. Senthilkumar K.K. Bharadwaj (2009). Hybrid Genetic-Fuzzy Approach to Autonomous Mobile Robot. IEEE International Conference on Technologies for Practical Robot Applications, pp. 29-34.

[8] P. Vadakkepat, O. Miin, X. Peng, and T. Lee (2004).Fuzzy behavior-based control of mobile robots. IEEE Trans. Fuzzy System. Vol. 12, No. 4, pp. 559–565.

[9] Fabrice Lauri, Abderrafiâa Koukam (2008). A Two-Step Evolutionary and ACO Approach for Solving the Multi-Agent Patrolling Problem. IEEE Congress on Evolutionary Computation, pp: 861 – 868.