

Ant Colony Optimization Algorithm for Robot Path Planning

Michael Brand¹, Michael Masuda¹, Nicole Wehner², Xiao-Hua Yu¹

¹ Department of Electrical Engineering, California Polytechnic State University
San Luis Obispo, CA 93407, USA

² Department of Electrical Engineering, University of Notre Dame
Notre Dame, Indiana 46556, USA

Abstract—Path planning is an essential task for the navigation and motion control of autonomous robot manipulators. This NP-complete problem is difficult to solve, especially in a dynamic environment where the optimal path needs to be rerouted in real-time when a new obstacle appears.

The ACO (Ant Colony Optimization) algorithm is an optimization technique based on swarm intelligence. This paper investigates the application of ACO to robot path planning in a dynamic environment. Two different pheromone re-initialization schemes are compared and computer simulation results are presented.

Keywords- Ant Colony Optimization, Robot Path Planning

I. INTRODUCTION

Path planning is an important issue for the navigation and motion control of autonomous robot manipulators. In computational complexity theory, path planning is classified as an NP (nondeterministic polynomial time) complete problem [2]. That is, the computational time that is required to solve such problem increases dramatically (usually in an exponential rate) when the size (or dimension) of the problem increases.

Let robot A be a single rigid object moving in a two- or three-dimensional Euclidean space (called work space W); and let obstacles B be rigid objects distributed in W . Assume there is no kinematic constraints that limit the motion of A (i.e., A is a “free-flying” object), then a path planning problem can be formulated as the following ([1]):

Given an initial position and its orientation, a goal position and its orientation of A in W , generate a path that specify a sequence of positions and orientations of A avoiding contact with B , starting from the initial position and terminating at the goal position. Report failure if no such path exists.

The studies of path planning started in late 60's ([1]) and many different algorithms have been proposed, including the roadmap approach, cell decomposition,

potential fields, and mathematical programming, etc. ([2]). It has been found that the above methods are either inefficient, due to the high computational cost; or inaccurate, due to the trapping in local minima ([2]).

To overcome these limitations, many heuristic approaches have been developed, such as the application of artificial neural networks. One of the major advantages of heuristic algorithms is that it can produce an acceptable solution very quickly, which is especially suitable for solving NP-complete problems.

The ant colony optimization (ACO) algorithm is a meta-heuristic approach inspired by the behavior of the biological ants in real world. It is one of the most successful examples of swarm intelligent systems and has been applied to solve many different types of problems, such as optimizing nonlinear objective functions and network routing in telecommunication networks.

In this research, the application of the ant colony optimization algorithm for robot path planning is investigated. The goal is to find the shortest and collision-free route (if exists) between a starting point and a destination point in a grid network. To simulate a dynamic environment, obstacles with different shapes and sizes are added after the optimal path is founded in the original (obstacle free) network. Two different pheromone re-initialization schemes (i.e., the local initialization and the global initialization) are discussed and compared. Computer simulation results are presented to demonstrate the effectiveness of the ACO algorithm.

II. THE ANT COLONY OPTIMIZATION ALGORITHM

It is well known that the biological ants in real world are able to utilize swarm intelligence to find the shortest route to nutrients. Ant Colony Optimization (ACO) algorithms have been developed to mimic the behavior of real ants to provide heuristic solutions for optimization problems. It was first proposed by M. Dorigo in 1992 in his Ph. D. dissertation.

When searching for food, biological ants exhibit complex social behavior based on the hormones they deposited (called pheromones). Pheromones attract other ants and outline a path to the food source that other ants can follow. As more ants walk along the path, more pheromone is laid, and the chance that more ants will take the path increases. The shortest path to the food builds up the most pheromones because more ants can travel it in less amount of time. This phenomenon was first observed in the famous Double Bridge experiment [2]: when given the choice between a short path and a long path to a food source, the ants consistently found the shortest path after a period of time. To prevent establishing a suboptimal path (when the solution is trapped into a local minimum), the pheromone also evaporates over time [4], thus reducing the change for other ants to take the path. On the other hand, the pheromone levels on the shortest path remain high because in this case, the pheromone deposit speed is faster than its evaporation speed.

Consider a network where ants can travel between different nodes. Using pheromone deposits, the probability that an ant k located in node i will choose to go to another node j in the network is given by ([4]):

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij}^k)^\alpha (\eta_{ij}^k)^\beta}{\sum_{l \in N_i^k} (\tau_{il}^k)^\alpha (\eta_{il}^k)^\beta} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad (1)$$

where pheromone levels are denoted by τ_{ij}^k ; as with real ants, the more pheromone on a path, the more possibility that the ant will take the path. The summation in the denominator takes into account all the possible choices (or neighboring nodes) in the set N_i^k when the ant is at node i . Both α , β , and η_{ij}^k are usually application dependent; where η_{ij}^k represents the heuristic information, and the values of α and β weigh the importance of the pheromone and heuristic values. When $\beta = 0$, $(\eta_{ij}^k)^\beta = 1$, then the probability only depends on the pheromone levels; on the other hand, when $\alpha = 0$, the probability only depends on heuristic values – that is, the node that is the closest one to the current node has the highest probability of being selected.

The pheromone levels of the path (from node i to j), can evaporate with a percentage ρ (also called the evaporation rate):

$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij} \quad (2)$$

where $0 \leq \rho < 1$. After pheromone evaporation occurs, the new pheromone levels are updated with the additional pheromone laid by the ant(s) that just crossed the path:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3)$$

$$\Delta \tau_{ij}^k = \frac{1}{C^k} \quad (4)$$

where C^k is the associated cost or reward of ant k for choosing this path.

III. ROBOT DYNAMIC PATH PLANNING BASED ON ANT COLONY ALGORITHM

In this section, the proposed ant colony optimization algorithm is applied for robot path planning in a grid network. Since our goal is to find the shortest path between the starting and ending positions, the total path length is chosen to be the cost or reward associated with each possible solution. The simulation is coded in Python and implemented on a Macintosh Powerbook G4 computer with a 1.5GHz processor.

In the simulation, three different sizes of grid network are considered, i.e., 20X20, 30X30, and 40X40. It is assumed that one ant can only occupy one node and moves to one of its adjacent nodes at a time in four different directions, i.e., up, down, left, and right. All the nodes are evenly distributed in the network; and the distance between any two adjacent nodes is normalized to 1 (or 1 unit block). Thus, the path length is represented in terms of the number of (unit) blocks.

The simulation starts with a “clean” environment; i.e., there is no obstacle in the original network. The upper-left corner is chosen to be the starting point and the lower-right corner is chosen to be the destination. All the pheromones are initialized as 0.1. The ant colony algorithm is then applied to find the shortest path and pheromones are deposited. A computational flow chart is shown in Fig. 1. Fig. 2 illustrates the path found by ACO algorithm in a 20X20 grid network convergences to its optimal value, where y-axis represents the path length (in number of blocks/grids) and x-axis represents the simulation run-time (in seconds).

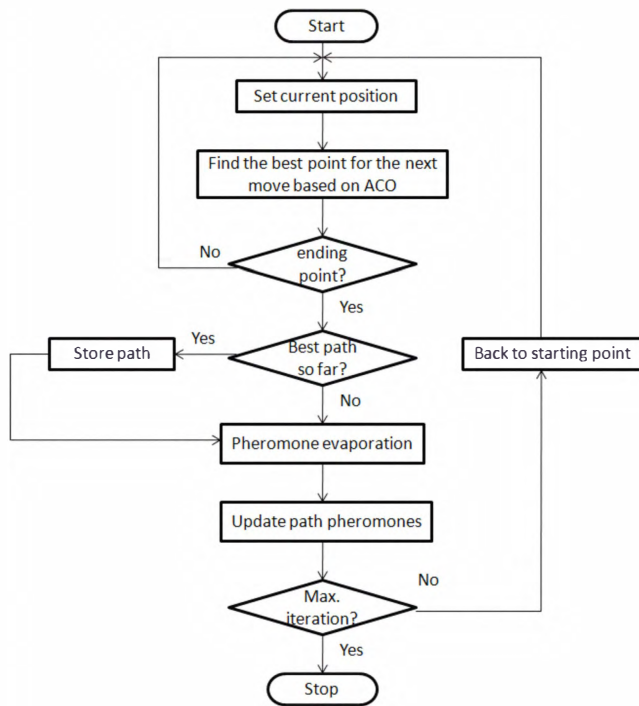


Figure 1. Computational flow chart of ACO

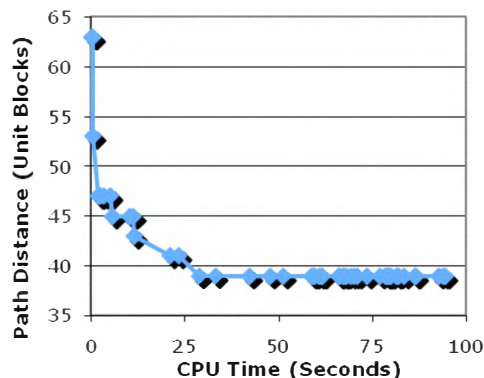


Figure 2. Optimal path found by ACO in a grid network

To simulate a dynamic environment, barriers (obstacles) are added after the algorithm converges (for a fair comparison, the sizes of obstacles are proportional to the sizes of networks). Fig. 3 illustrates such an example, where the two darker areas inside the 40X40 grid represent two arbitrary obstacles added at the positions that are randomly chosen. The ACO algorithm must be called again in order to find the shortest path in this new network with obstacles. The flow chart is shown in Fig. 4.

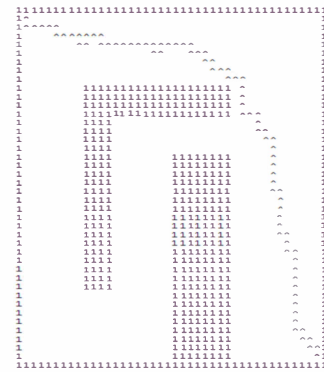


Fig. 3. The Optimal path found by ACO (with obstacles) in a 40X40 grid network

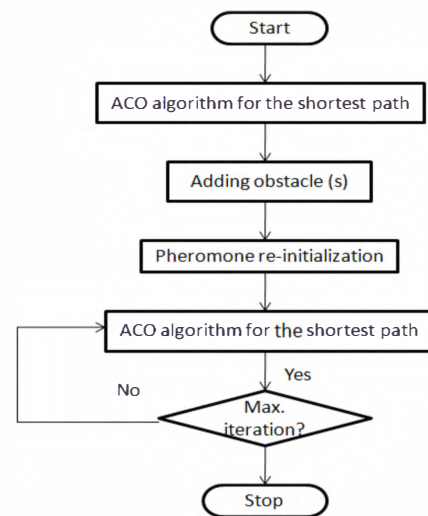


Figure 4. The flow chart of ACO in a dynamic environment

Pheromone initialization plays an important role in ACO algorithm. In this research, after the obstacles are added, the pheromones in the network are re-initialized. Two different re-initialization schemes, namely, the global initialization and the local initialization are tested and their performances are compared. In global initialization, all the pheromones in the entire network are uniformly reset back to the original pheromone level, which is 0.1. With local initialization, a “gradient” of pheromones is initialized around the object. The value of half the highest pheromone levels in the network is applied directly to the points that are next to the object. The pheromone levels are then decreased by a fraction (e.g., 50%) as the points move outward in a “circle” around the object. Fig. 3 shows the optimal path found in a 40X40 grid with obstacles using the local initialization method.

The performance of the two different initialization approaches are summarized in Table 1 (for local initialization) and Table 2 (for global initialization),

based on the additional number of iterations needed to find the optimal path after obstacles are added, as well as the length of the final optimal path. From the data, we can conclude that the local initialization outperforms the global initialization. For a 20X20 grid network, though they eventually found the optimal path with the same length, the numbers of additional iterations requested using the two methods are different. It only takes 122 additional iterations for the local initialization method while 151 iterations are needed for the global initialization method (an increase of 23.8%). The advantage of using local initialization is even more significant for larger networks. In a 40X40 grid network, the additional number of iterations needed by local initialization is 69 while the additional number of iterations needed by global initialization is more than doubled (i.e., 148). Even with more iterations, the path length found by local initialization is 7.8% shorter (128 blocks) than its counter algorithm (138 blocks).

Table 1. The additional number of iterations and the optimal path found by ACO with obstacles (local initialization)

Size	20X20	30X30	40X40
Number of iterations	122	84	69
Path length	39	64	128

Table 2. The additional number of iterations and the optimal path found by ACO with obstacles (global initialization)

Size	20X20	30X30	40X40
Number of iterations	151	277	148
Path length	39	66	138

Fig. 5 demonstrates another simulation with three obstacles in different shapes and sizes. In (a), ACO algorithm is implemented for only 1 iteration after these obstacles are added. Obviously, the path shown in (a) is not optimal; thus more iterations are needed. The much improved result is shown in (b), which is obtained after the ACO algorithm runs for 1000 iterations.



(a)



(b)

Fig. 5. The Optimal path found by ACO (with another obstacle) in a 40X40 grid network

- (a) The result of the first iteration after the obstacle is added
(b) The path resulted after 1000th iterations

IV. CONCLUSIONS

In this research, the ant colony optimization (ACO) algorithm is applied to find the shortest and collision-free route in a grid network for robot path planning. Obstacles with various shapes and sizes are considered to simulate a dynamic environment. Computer simulation results demonstrate that the ACO algorithm can successfully re-route the optimal path for the new network after obstacles are added. Future works may include the investigation on different ACO algorithms, such as the Elitist Ant System (EAS), the Rank-Based Ant System, and the MAX-MIN Ant System (MMAS). Also, simulation works can be performed with more complicated networks and obstacles.

REFERENCES

- [1] Latombe, J. (1990) Robot motion planning. Springer-Verlag.
- [2] Goss, S., Aron, S., Deneubourg, J.L., and Pasteels, J.M. (1989). Self-organized Shortcuts in the Argentine Ant. *Naturwissenschaften* Issue 76, pg. 579-581. Springer-Verlag.

- [3] Jackson, D. E. and Chaline, N. (2007). Modulation of pheromone trail strength with food quality in Pharaoh's ant, *Monomorium pharaonis*. *Animal Behaviour*. Vol. 74, Issue 3, pg.463-470.
- [4] Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimization*. Massachusetts Institute of Technology.
- [5] Vander Meer, R.K. and Alonso, L.E. (1998). Pheromone directed behavior in ants, pg. 159-192. *Pheromone Communication in Social Insects*. Westview Press, Boulder, CO.
- [6] Gengqian, L., Tiejun, L., Yuqing, P., Xiangdan, H. (2005). The Ant Algorithm for Solving Robot Path Planning Problems. *The 3rd International Conference on Information Technology and Applications*.
- [7] www.aco-metaheuristic.org. Accessed Feb. 20, 2009.