

# Metaheuristic Algorithm for Robotic Path Planning

Yogita Gigras  
ITM University  
Gurgaon  
India

Kusum Gupta  
Banasthali University  
Banasthali  
India

## ABSTRACT

This paper presents a metaheuristic optimization algorithm for mobile robot path planning problem. A comparative study between particle swarm and ant colony optimization algorithm is conducted. The experimental study shows that the ant colony optimization algorithm outperforms over particle swarm optimization in terms of computational time.

## General Terms

Path planning, Ant colony optimization algorithm (ACO) and Particle swarm optimization (PSO).

## 1. INTRODUCTION

Heuristic algorithms are intended to find out the good solution to an optimization problem by 'trial-and-error' in a reasonable amount of time. But here heuristic means to find or search by trial-and-error method. There is no guarantee to find the best or optimal solution, although it can be a better or improved solution than an educated guess. Any reasonable good solution can be suboptimal or near optimal and could be good enough for such problems. Broadly, local search methods are heuristic methods as their parameter search is focused on the local variations and the optimal or best solution can be well outside the local region [3][7]. A high quality feasible solution in the local region of interest is usually accepted as a good solution in many optimization problems in practice as time is the major constraint in these problems. Metaheuristic algorithms are higher-level heuristic algorithms where 'meta' means higher level or beyond therefore metaheuristic means to find the solution using higher level techniques, although certain trial-and-error processes are still used. Metaheuristic are treated as higher level techniques and tactics for exploration and exploitation of the huge space for parameter search. In recent years, the 'metaheuristics' refers to all modern higher level nature inspired algorithms including particle swarm optimization and ant colony optimization algorithm [4].

## 2. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) algorithm are [2] based on the behavior of a colony or swarm of insects such as ants, termites, bees and wasps, a flock of birds or a school fish. The PSO reflects the behavior of these social organisms. Each particle in a swarm behaves in a distributed way using its own intelligence and the collective or group intelligence of the swarm. If one particle finds a good path to food, the rest of the swarm will also be able to follow the good path instantly if their location is far away in the swarm.

In multivariable optimization, the swarm is assumed to be of fixed size with each particle located initially at random locations in the multidimensional design space. Each particle has two characteristics: - one is position and second is velocity. Each particle moves in the design space and remembers the best position in term of target position or objective function value it has discovered. The particles communicate information to each other and adjust their

individual position and velocities based on the information received on the target position.

## Pseudocode for PSO

1. Set the number of particle as N.
2. Choose the initial population randomly and represented by  $X_1(0), X_2(0), X_3(0) \dots \dots X_n(0)$ .
3. Evaluate the objective function values at  $X_j(0)$  where  $j = 1, 2, 3 \dots n$  as

$$F_1 = F[X_1(0)]$$

$$F_2 = F[X_2(0)]$$

$$F_3 = F[X_3(0)]$$

.

.

$$F_n = F[X_n(0)]$$

4. Set the initial velocities of each particle is to zero  $V_1(0) = V_2(0) = \dots = V_n(0)$ .
5. Set the iteration number as  $i = 1$ .
6. In the  $i^{th}$  iteration, following two parameters are calculated which is used by particle j:-
  - (a) The best particle of  $X_j(i)$ ,  $P_{best,j}$  i.e coordinates of  $j^{th}$  particle in the current iteration is equal to the highest value of the objective function  $F[X_j(i)]$ , encountered by particle j in all the previous iteration.
  - (b) The best value of  $X_j(i)$ ,  $G_{best,j}$  i.e coordinates of all particles up to that iteration is equal to the highest value of the objective function  $F[X_j(i)]$ , encountered in all the previous iteration by any of the N particle.
7. Find the velocity of the particle j in the iteration:-
$$V_j(i) = V_j(i-1) + r_1[p_{bestj} - X_j(i-1)] + r_2[G_{bestj} - X_j(i-1)], j = 1, 2, 3 \dots n$$
Where  $r_1$  and  $r_2$  are uniformly distributed random numbers in the range 0 and 1.
8. Find the position or coordinate of the  $j^{th}$  particle in the  $i^{th}$  iteration as
$$x_j(i) = x_j(i-1) + V_j(i), j = 1, 2 \dots n$$
9. Evaluate the objective function values corresponding to the particle as  $F[X_1(0)], F[X_2(0)], F[X_3(0)] \dots F[X_n(0)]$ .

10. If all the particles converge to the same set of values, then the method is converged otherwise goto step 6 by updating  $i = i + 1$ .

### 3. ANT COLONY OPTIMIZATION

ACO is based on the cooperative behavior of real ant colonies, which are able to find the optimal path from nest to food source [1]. The ant colony optimization process can be viewed by multilayered graph in which ant has to move from node to node by depositing the pheromone chemical, this chemical attract the other ants also and this chemical is a volatile chemical[5][6]. The probability by which any ant k, when located at node i, uses the pheromone trail  $\tau_{ij}$  to compute the probability of choosing j as the next node:-

$$P_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{j \in N_i(k)} \tau_{ij}^\alpha} & \text{if } j \in N_i(k) \\ 0 & \text{otherwise} \end{cases}$$

Where  $\alpha$  denotes the degree of importance of the pheromones and  $N_i^{(k)}$  denotes the set of neighboring nodes of ant k when located at node i contains all the nodes except the predecessor node. An ant travels from node to node and reaches the destination node. Afterward pheromone updating takes place by the given formula.

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

Where  $\rho \in (0,1]$  is 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 aim of pheromone updating is to increase the pheromone value associated with optimal path. The pheromone deposited on arc  $(i,j)$  by the best ant k is  $\Delta\tau_{ij}^{(k)}$ .

Where

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

Here Q is a constant and  $L_k$  is the length of the path traversed by the best ant k.

#### Algorithm

1. Assume the number of ants in a colony is N.
2. A set of permissible discrete values is  $X_1, X_2, X_3 \dots \dots X_n$ .
3. Each ant can choose any of the path or discrete value i.e.  $X_1, X_2, X_3 \dots \dots X_n$ .
4. Set the iteration number  $l=1$ .
5. Compute the probability  $P_{ij}$  of selecting edge  $X_{ij}$  as  

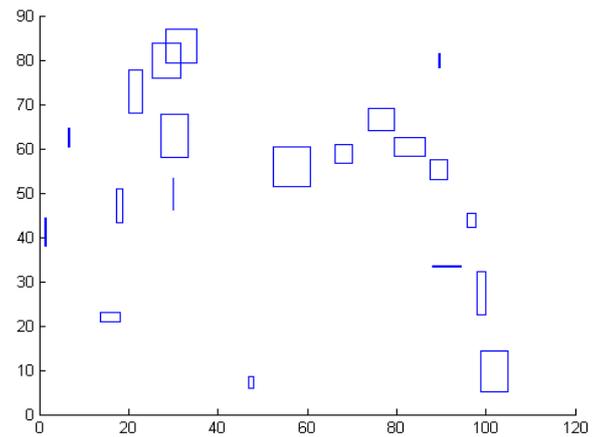
$$\frac{\tau_{ij}^{(l)}}{\sum_{m=1}^n \tau_{im}^{(l)}} \quad i = 1,2,3 \dots n \text{ and } j = 1,2,3 \dots n.$$
6. Generate N random numbers  $r_1, r_2, r_3 \dots \dots r_n$  in the range (0,1) one for each ant.
7. Calculate the cumulative probability range associated with the different path using roulette-wheel selection.
8. Determine the path taken by ant k as one of the cumulative probability which include  $r_k$  random number in it.

9. Calculate the objective function values for all the ants  $F_k = F(X^{(k)}) \quad k = 1,2,3 \dots n$
10. Find out the best and worst paths among the N paths by different ants.
11. This process is assumed to be converged if all N takes same best path.
12. If convergence is not achieved then all the ants return home and start again in search of food.
13. Update the pheromone on arc when all the ants complete their tour  

$$\tau_{ij}^{(k)} = \tau_{ij}^{(old)} + \sum_{k=1}^N \Delta\tau_{ij}^{(k)}$$
14. Set  $\tau_{ij}^{(old)} = (1 - \rho)\tau_{ij}^{(l-1)}$
15. Repeat this process until  $l = l_{max}$  (maximum no. of iteration).

### 4. ROBOTIC PATH PLANNING

Consider the moving space of robot of 100\*100 with twenty randomly generated obstacles therefore the position of the obstacle is not known the robot.



**Fig 1: Generation of randomly generated obstacles**

Consider the robot as asterisk structure with initial position  $(X_s, Y_s)$  and target position  $(X_t, Y_t)$  moving in a straight line with two variables  $d$  and  $\theta$ .

$$\text{Where } d = \sqrt{(x_t - x_s)^2 + (y_t - y_s)^2}$$

And

$$\theta = \left( \frac{\text{abs}(y_t - y_s)}{\text{abs}(x_t - x_s)} \right)$$

With the use of these two variables we can calculate  $(X_{snew}, Y_{snew})$

Where

$$X_{snew} = X_s + \text{stepsize} \times \cos \theta$$

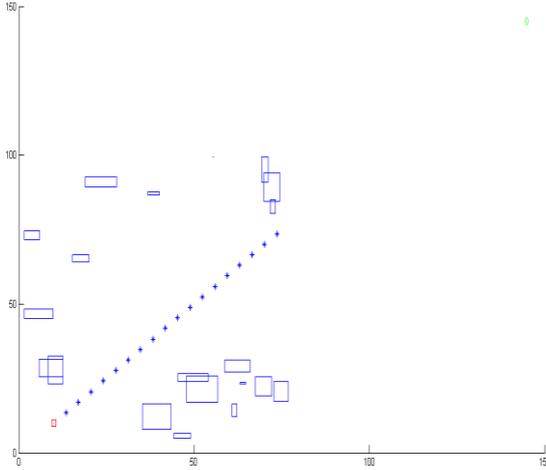
And

$$Y_{snew} = Y_s + \text{stepsize} \times \sin \theta$$

Assume  $\text{stepsize} = 1$

Again calculate the  $\theta = \left( \frac{abs(y_t - y_{snew})}{abs(x_t - x_{snew})} \right)$

When encounter with the obstacle then it take the three steps back here we can apply the ACO and PSO to bypass the obstacles and afterwards compares the performance of the algorithms.



**Fig 2: Robot encounter with obstacle**

Firstly apply the PSO, in which assume number of particle equals to twenty-five and generate initial random population with position  $(X_1, X_2, X_3 \dots X_{25})$ . In two dimensional space the position of the particles are represented by ST1X in X-direction and ST1Y in Y-directions as:-

$$ST1X = \cos(X(i))$$

$$ST1Y = \sin(X(i))$$

After this calculate the distance function for each particle:-

$$fun\_val(i) = \sqrt{(x_t - ST1X(i))^2 + (y_t - ST1Y(i))^2}$$

Evaluate the minimum function value of the particle i.e. minimum distance for the target with its index by  $fun\_val(i)$ .

$$[oV oI] = \min (fun\_val)$$

Where oV represents the value and oI represents the index. Afterwards update the velocity of all particles in  $VX(i)$  for X-direction and  $VY(i)$  for Y-direction.

$$VX(i) = VX(i) + r_{no(1)} \times (pbestx(i) - ST1X(i)) + r_{no(2)} \times (gbestx - ST1X(i))$$

$$VY(i) = VY(i) + r_{no(1)} \times (pbesty(i) - ST1Y(i)) + r_{no(2)} \times (gbesty - ST1Y(i))$$

Where  $P_{best,i}$  i.e. coordinates of  $i^{th}$  particle in the current iteration is equal to the lowest value of the distance function  $fun\_val(i)$ , encountered by particle  $i$  in all the previous iteration. We have to minimize the distance function so that robot can reach to the destination while following the minimum distance.

And

$G_{best,i}$  i.e. coordinates of all particles up to that iteration is equal to the lowest value of the distance function  $fun\_val(i)$

encountered in all the previous iteration by any of the  $N(N = 25)$  particle.

Update the position of the particle in X-direction and Y-direction as:-

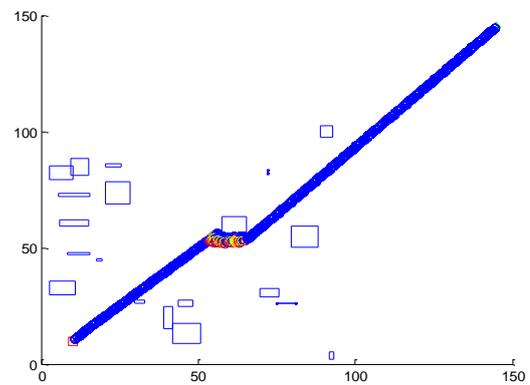
$$ST1X(i) = ST1X(i) + VX(i)$$

$$ST1Y(i) = ST1Y(i) + VY(i)$$

While moving if any of the particle lie insides the obstacle then assume the distance function value as  $fun\_val(i) = 10^5$  so that the particle never reaches to the destination end otherwise distance function value

$$fun\_val(i) = \sqrt{(x_t - ST1X(i))^2 + (y_t - ST1Y(i))^2}$$

Follow these steps so that all the particles converge to the same set of values.



**Fig 3: Robot reaches destination by using PSO**

Then apply ACO

Assume number of ants equal to twenty five and take eight discrete angles for path taken by an ants i.e.  $ant\_path = [0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4]$ . In two dimensional space the position of ants are represented in X-direction and Y-direction as:-

$$X = \cos(ant\_path(i))$$

$$Y = \sin(ant\_path(i))$$

Initially amount of pheromone in all eight directions  $\tau_{ij} = 1$  and probability of choosing the next path by an ant is given as:-

$$\frac{\tau_{ij}^{(l)}}{\sum_{m=1}^8 \tau_{im}^{(l)}} \quad i = 1,2,3 \dots n \text{ and } j = 1,2,3 \dots n.$$

The specific path chosen by any ant is determined by random number generated in the range(0,1). For this, Calculate the cumulative probability associated with discrete path. The path taken by an ant  $k$  is the one for which cumulative probability range includes corresponding random number generated for ant  $k$ .

If an ant goes inside the obstacle than the value of distance function  $10^5$  so that ant never reaches to the destination end.

Otherwise

$$fun\_val(i) = \sqrt{(x_t - ST1X(i))^2 + (y_t - ST1Y(i))^2}$$

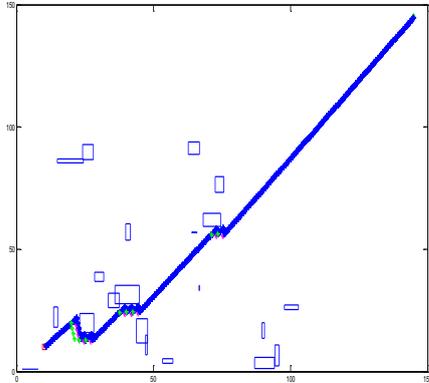
Update the pheromone chemical for the best path:-

$$\tau = \tau + 2 \times \min(fun\_val)/\max(fun\_val)$$

And for remaining paths

$$\tau = \tau \times (1 - \rho), \rho = 0.5$$

Follow these paths until all ants choose the same path.



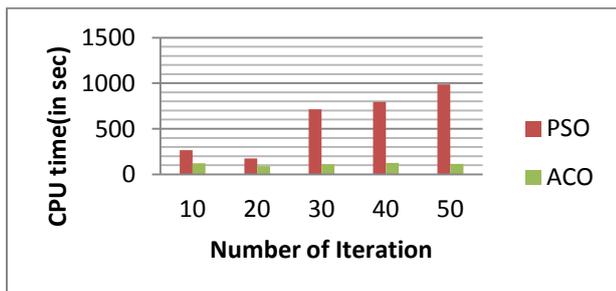
**Fig 4: Robot reaches destination by using ACO**

## 5. RESULT

Implementing ACO and PSO in matlab so that CPU time (in sec) is calculated for different iterations as shown in table 1

**Table 1. Number of iteration vs. CPU time in sec)**

No.of iteration	PSO(sec)	ACO(sec)
10	267	123
20	174	90
30	715	112
40	795	125
50	987	114



**Fig 5: Performance of PSO and ACO algorithm on path planning problem**

## 6. CONCLUSION

This paper presented a metaheuristic optimization based algorithm to the problem of mobile robot path planning. The ACO algorithm has potential of finding near optimal solution in shorter time than PSO algorithm. The future aspects of research is to hybrid these two algorithm to give more better algorithm in terms of computational time as well as handling the more complex dynamic environment with moving obstacles.

## 7. REFERENCES

- [1] Beatriz A. Garro, Humberto Sossa and Roberto A. Vázquez. 2006, "Path Planning Optimization Using Bio-Inspired Algorithms", International Conference on Artificial Intelligence, 319 – 330.
- [2] Abdullah Zawawi Mohamed, Sang Heon Lee , Mahfuz Aziz, Hung Yao Hsu,Wahid Md Ferdous. 2010"A Proposal on Development of Intelligent PSO Based Path Planning and Image Based Obstacle Avoidance for Real Multi Agents Robotics System Application", International Conference on Electronic Computer Technology (ICECT), 128 – 132.
- [3] X.-S. Yang. 2009,"Harmony Search as a Metaheuristic Algorithm", in: *Music-Inspired Harmony Search Algorithm: Theory and Applications* (EditorZ. W. Geem), Studies in Computational Intelligence, Springer Berlin, Vol. **191**, 1-14.
- [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] 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.
- [6] Alireza rezaee, Mohammad jafar pour jalali. 2009 "Multi-Resolution Ant Colony A New Approach to Use Swarm Intelligence in Continuous Problems", International Conference on Information and Multimedia Technology, pp: 529 – 532.
- [7] 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.