

# **Improvement of Robot Path Planning Using Particle Swarm Optimization in Dynamic Environments with Mobile Obstacles and Target**

**Maryam Yarmohamadi**

Computer Department  
Shahed University of Tehran, Tehran, Iran  
yarmohamadi@shahed.ac.ir

**H. Haj Seyyed Javadi**

Department of Mathematics and Computer Science  
Shahed University of Tehran, Tehran, Iran  
h.s.javadi@shahed.ac.ir

**Hossein Erfani**

Computer Engineering Department, Science and Research Branch  
Islamic Azad University, Tehran, Iran  
Hossein.erfani@gmail.com

## **Abstract**

Particle Swarm Optimization (PSO) has been demonstrated to be a useful technique in robot path planning in dynamic environment with mobile obstacles and goal. One or many robots are able to locate a specification target with high efficiency when

driven by an optimization PSO algorithm. The goal of the optimization is minimize the resultant path lengths. To avoid obstacles during movement trajectory, self organized trajectory planning is required. This study propose to use particle swarm optimization, which is motivate from the simulation of social behavior of fishes and birds, as a feasible approach for self organized control of robot to avoid obstacle throughout the trajectory.

**Keywords:** Robot Path Planning, Trajectory Planning, Particle Swarm Optimization, Dynamic Environments, Optimization, Moving Obstacles, Moving Target

## 1 INTRODUCTION

This paper addresses the main problem in robot path planning. Although many algorithms have been proven to be feasible and efficient for robot mutation planning and collision avoidance, classic techniques for path planning problem are general methods like Roadmap, Cell Decomposition, Potential Fields and Mathematical Programming. Most of path planning problems can be solved using these classic techniques. High time complexity in large problem space and trapping in local optimum are drawbacks for classic techniques. These drawbacks cause the classic techniques to be inefficient in a large problem space. In order to improve the efficiency of classic methods, probabilistic algorithms like PRM and RRT are proposed. For local optimum problem, many heuristic and meta-heuristic algorithms like Simulated Annealing [9], Genetic Algorithms [7] and Ant Colony Optimization [8] are used in path planning problem. Finally a smooth path planning for mobile robots is simulated in [6].

A PSO based algorithm for path planning mobile robots with mutation operator is presented in [4]. A perfect and precise path planner, which finds the path if one exists, otherwise reports no path existence, is NP complete [2]. PSO was proposed by Kennedy and Earhart in 1995, motivated by social behavior of organisms such as bird flocking [5]. The PSO algorithm can be easily modified in order to allow the particle/robots to successfully consider the risks of the dynamic environment [3] and obstacle avoidance while still continuing on an efficient trajectory leading toward total swarm convergence on the target.

This paper presents modification to the path planning robot using PSO presented to now include obstacle, we expect propose an efficient search algorithm as well as creating a small scale model of search system. So that it can plan a path from initial position to goal. Search spaces are such as building, lakes, rivers, and mountains that require the robot is to navigate around these obstacles and avoid collision. The rest of the paper organized as follows: In section 2, we give the detailed description of the particle swarm optimization algorithm. Also section 3 discusses PSO and robot Path planning problem and describes objective function and how to use PSO for parameter

optimization. In section 4 we discuss about the feasibility of the presented approach and we discuss how to solve the constraint path planning problem using PSO in section 5 and finally we discuss about simulation result in the last section.

## 2 PARTICLE SWARM OPTIMIZATION

In Particle Swarm Optimization (PSO), a problem space is covered with an initial population of solutions in while they are guided to search for the optimum over a number of generations. The concept of PSO is that each particle randomly searches as through the problem space by updating itself with its own memory and the social information gatherers from other particles. An attractive feature of PSO is that its implementation is simple and effective and if the path exist this algorithm can find it.

Suppose  $Z$  is the dimension of the searching space and  $N$  is the number of particle. The position and velocity of the  $i^{th}$  particle is represented as  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  and  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ , is the velocity of the  $i^{th}$  particle, each particle is maintains a memory of its previous best position denoted by  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  and the best position of the population ( $gbest$ ) is denoted  $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ . Each particle updates its position and velocity according to the following equations:

$$\begin{cases} V_i^{k+1} = wV_i^k + c_1r_1(P_i - X_i^k) + c_2r_2(P_g - X_i^k) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \quad (1)$$

Where  $w$  is the inertia in range  $[0.9, 0.4]$  that increasing linearly,  $c_1$  and  $c_2$  are two positive constant, usually we choose  $c_1 = c_2 = 2$ ;  $r_1$  and  $r_2$  are two random functions in the range  $[0, 1]$ .  $i = 1, 2, \dots, N$  that  $N$  is the size of the population;  $k = 1, 2, \dots, K$  that  $k$  is the maximum number of iterations.

PSO algorithm can be summarized in the following step (Algorithm 1):

Step 1: A population of robot is initializing in the search environment containing a target and an obstacle, with random position, velocities, personal best position  $pid$ , and global best position  $pg$ .

Step 2: The fitness value-Euclidean distance from the robot to the target, for each particle where  $z$  is the targets coordinate and  $x$  is the current coordinates of the individual robot.  $y$  is intermediate position to go from  $x$  to  $y$ .

$$Fitness = Euclidean(x, y) + Euclidean(y, z) \quad (3)$$

Step 3: The particle fitness is compared with it is previous best fitness  $pid$  for very iteration to determine the next possible coordinate position for robot in the search environment. The next possible velocity and position of each robot are

determined according to (1) and (2) where  $vid$  and  $xid$  represent the velocity and position of the robot at  $k+1$  iteration.

In PSO, the inertia weight  $w$  play important role, because balances between local path and global path related it. Calculate new inertia weight according to (4):

$$w = w_{start} - \frac{w_{start} - w_{end}}{K} k \quad (4)$$

Step 4: If the next possible position  $x_{id}$   $k+1$  re resides within the obstacle space, the obstacle avoidance part of the algorithm explained in this Section is employed, otherwise the robot moves to this new position and step 5 is implemented.

Step 5: The  $pid$  with the best fitness for the entire swarm is determined and the global best  $pid$  coordinate location  $pgd$ .  $pbest$ , is updated with this  $pid$ .

Until convergence is reached, repeat step 2 to 5.

### 3 PSO AND ROBOT PATH PLANNING PROBLEM

As the particle move through the search space, gaining one new position for iteration, a conditional statement checks to see if the position of the particle will fall within the boundaries of the obstacle. If this condition is true, the obstacle avoidance section of the algorithm is initiated.

Movement of a robot position is realized by the Particle Swarm Optimization algorithm. PSO convergence to the best solution by adjusting the trail of each individual particle toward its best location based on the best of itself and global best on the neighbor particles. The modification of a robot position is realized by position and velocity information. The position of a robot is represented by Cartesian coordinate, such as  $x$ ,  $y$  axis position and the velocity is modification by PSO.

The initial position for the robot is initialized with random coordinates. Robot is defined within the context of topological obstacles comprising itself in the population. Neighbors of robot can be carried out based on the distance of robot position. In this study, global path is considered for robot movement path planning. Global path is carried out based on the fitness function. Fitness value is evaluated for each particle over its obstacles based on the coverage of the target area. When a particle discovers a pattern that is better than any it has found is stores the coordinate as new best position. Robot rotates in order to reduce the possible collisions with obstacles.

Robot R is located at initialized position at the first,  $(x_r, y_r)$  in time  $(t)$ .  $(x'_r, y'_r)$  is the next position of robot in time  $(t+1)$  and  $(x_g, y_g)$  is location of goal. We find the between position of robot  $(x'_r, y'_r)$  with PSO algorithm. PSO is initialized with a population of random particles  $(X_1, X_2, \dots, X_D)$  which distribute uniformly around  $(x_r, y_r)$  at first. At the end of the algorithm execution, we choose the particle with the best position to be the next position of the robot. It can be seen in Fig.1.

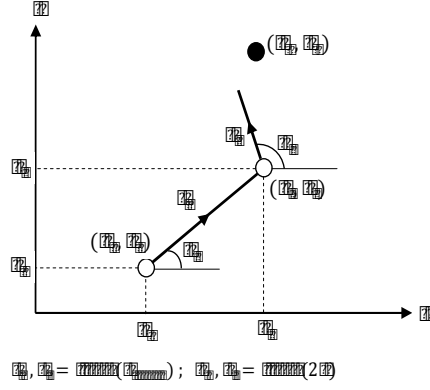


Figure 1. position of the  $i^{th}$  particle populated around  $(x_r, y_r)$

## OBJECTIVE FUNCTION

The optimization problem includes an objective function, concerning minimization of the Euclidean distance between the current positions of the robots with their respective goal position.

This function calculates the length of trajectory of the robot. This trajectory is the line joining  $\{(x_r, y_r)(x'_r, y'_r)\}$  and  $\{(x'_r, y'_r)(x_g, y_g)\}$  without any collision. This function calculates Euclidean distance that we minimize the total path length without touching the obstacle in Fig. 2.

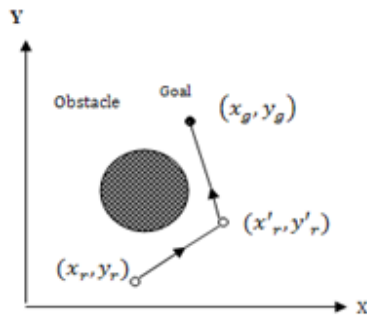


Figure 2. robot select of  $(x'_r, y'_r)$  to avoid collision with obstacle.

$$y_i = y_r + V_r \sin \theta_i \Delta t \quad (5)$$

$$x_i = x_r + V_r \cos \theta_i \Delta t \quad (6)$$

For the particle in Fig.2 that represents a solution for the next position of the robot. Substituting for  $x_i$  and  $y_i$  from expressions obtain:

$$F = V_r + \sqrt{(x_r + V_r \cos \theta_i - x_g)^2 + (y_r + V_r \sin \theta_i - y_g)^2} \quad (8)$$

#### 4 PROPOSED APPROACH

Unlike similar works penalty function [1] here we present a new penalty function based on the environment state. Robot planning direct path from initial position to goal until collision with obstacle accure. If robot seen obstacle, we have to developing the objective function by adding a penalty function. For this reason for penalize, we assume a penalty value for every obstacle. Robot always selects the shortest path for moving towards the goal. When the robot seen the obstacle, it decide the direction and rotate left or right. This rotation can be any degree of the motion radius  $0 \leq \alpha < 2\pi$ .

Influence of the adjacent obstacle is added to length of the arc for that portion. We assume direct path from robot to goal and a vertical line from center of obstacle to that line, let this crossover  $x'$ , we assume four triangles that every triangle has right angle in Fig. 3. Calculate length of two paths and select the minimum path to rotation left or right.

$$penalty(j) = \min\{arcj + \sum_{s=1}^S Tangent(s)\} \quad (9)$$

S is number adjacent obstacles with  $arcj$  and  $Tangent$  is a recursive function that calculate  $a+a'$  for left side and  $a''+a'''$  for right side and  $j \in \{1,2\}$ .

$$Tangent(s) = a + a' = \sqrt{b^2 + c^2} + \sqrt{b'^2 + c^2}$$

Calculate length of two paths and select the minimum path to rotation left or right.

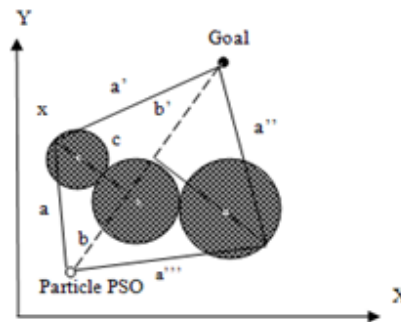


Figure 3. Four Triangles with right angle.

## 5 SOLVING THE CONSTRAINT PATH PLANING ROBOT USING PSO

In this section we propose a solution to path planning problem using PSO. The proposed scheme presumes current position of robot and their speeds, and determines the next position of robot by optimizing the given constrained objective function.

We propose an algorithm for trajectory of robot using PSO.

1. Add current position of robot to the trajectory.
2. If the distance between the current position of the robot and the goal is less than or equal to a predefined threshold, go to f.
3. Relocate the obstacles and the goal according to their corresponding relocation probability.
4. Initialize PSO swarm around current position of the robot. Evolve PSO swarm according to Algorithm 1.
5. Choose global best position the swarm to be the current position of the robot and go back to a.
6. Add the goal position to the trajectory and stop.

### Improvement

Robot has a limited footstep,  $V_{robot}$ , for which the robot cannot move further in each step.

There is also a footstep for goal,  $V_{goal}$ , that should be smaller than that of the robot's. Otherwise, although the goal is not ever moving and moves according to a probability  $P_{goal}$ , in the worst case it can escape from the robot all the time. Every obstacle in the environment has relocation probability  $P_{obs}$  that for static obstacles  $P_{obs} = 0$ . Regarding to this probability, all the obstacles change their position by  $V_{obs}$ , before the planning process starts in each step. Obstacles are assume to be circular. No two obstacles in the environment have overlaps, but they can be adjacent.

To take case of static and dynamic obstacles in the environment, we add a penalty function to the objective function.

We seen different between result of computer simulation and result of approached in [1] in Table1.

R1 is result of improvement penalty function and R2 is result in [1] that elapsed time and length of trajectory are in R1 less than R2 in 15 different environments. Parameters of environment are: GRP is Goal's relocation probability, VG is Goal's moving step, VR is Robot's moving step and VO is Obstacles' moving step.

ENVIRONMENT	R1:TIME/LENGHT	R2: TIME/LENGHT
GPR=1, VG=5, VR=5, VO=3		
1	54/130	57/138
2	22/98	33/117
3	23/101	30/108
GPR=0, VG=5, VR=30, VO=3		
4	27/150	56/158
5	30/138	31/135
6	37/158	44/162
GPR=0, VG=5, VR=10, VO=3		
7	39/166	40/173
8	22/183	25/190
9	37/159	40/165
GPR=1, VG=5, VR=10, VO=3		
10	23/108	31/136
11	25/117	33/143
12	22/98	30/135
GPR=0, VG=5, VR=5, VO=3		
13	44/162	59/200
14	37/158	40/183
15	61/201	67/215

Table 1: Comparison between R1, R2

## 6 EXPERIMENT AND COMPUTER SIMULATION

We test PSO algorithm for path planning robot in MATLAB under Windows XP on a computer with 3.20GHZ Pentium(R) and 1.00GB of memory. We assume the current position of robot is in  $[0,0]$  and the current position of goal is in  $[100,100]$ . In Fig. 5 the goal and obstacle are fixing in their position. The number, size and location of the obstacles toward each other have been varied in Fig. 4 and Fig. 5, whereas maximum velocity (step size) and relocation probability of the goal and obstacles has been varied in Fig. 5 and Fig. 6. In Fig. 7, Fig. 8 and Fig. 9 the green obstacles are mobile and goal is immobile that is also tabulated in Table 2.



Figure	No. of obstacle	No. of Static obstacle	No. of Dynamic obstacle	relocation probability of the goal	Velocity of goal	Velocity of robot
4	5	5	0	0	0	10
5	2	2	0	1	5	10
6	8	8	0	0.3	5	30
7	8	4	4	0	0	10

Table 2: Four environments

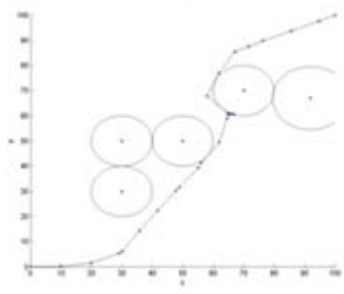


Figure 4

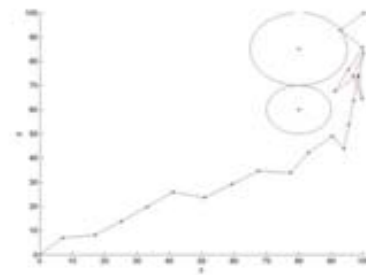


Figure 5

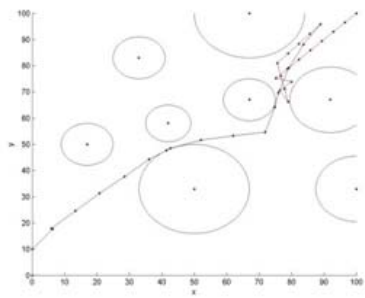


Figure 6

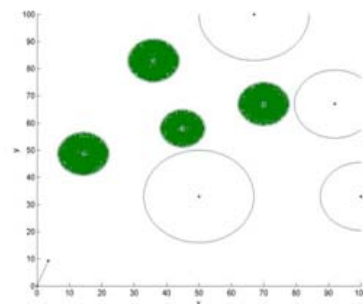


Figure 7. In start time

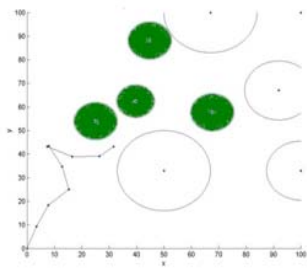


Figure 8. In middle path

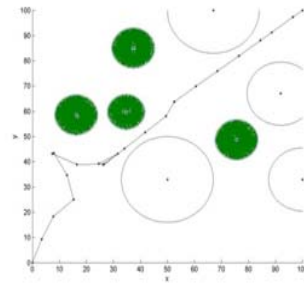


Figure 9. In end time

## ACKNOWLEDGMENT

This paper has presented a successful modification to a PSO algorithm reported show that applying particle swarm optimization to a path planning robot problem is still practical and efficient for search areas with variable obstacle. In the future we will work with multi-robot optimization models. This proposes can be applying in soccer robot, because environment is dynamic with mobile obstacle and mobile or immobile goal in soccer robot problem. Experimental results show that our approach performs better in escaping local optimum in comparison to the similar works.

## REFERENCES

- [1] A. Zargar Nasrollahy, H. Haj Seyyed Javadi, Using Particle Swarm Optimization for Robot Path Planning in Dynamic Environments with Moving Obstacles and Target *Proc. IEEE Int. ems*, pp.60-65, 2009.
- [2] J. Chakraborty, S. Saswati, Co-operative Multi Robot Path Planning Using Particle Swarm Optimization, 2005.
- [3] J. Kennedy, R. Eberhart, Particle Swarm Optimization, IEEE International Conference on Networks, Perth, WA, 1995, pp. 1942-1948, vol.4.
- [4] L.Shirong, M. Linbo and Y. Jinshou, Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems, *Proc. IEEE Int. Conf. on Mechatronics and Automation*,(2006) pp. 1733 – 1738.

- [5] M. Hua-Qing, Z. Jin- Hui, Z. Xi-Jing, Obstacle avoidance with multiobjective optimization by PSO in dynamic environment, *Proc. Int. Conf. Machine Learning and Cybernetics*, Vol. 5, (2005) pp. 2950-2956.
- [6] S. Doctor, G.KVenaygamoorthy and V.Gudise, Optimal PSO for Collective Robotic Search Application, IEEE Congress on Evolutionary Computation, June 19-23, 2004.Portland OR, USA, pp. 1390-1395.
- [7] X. Sijiang L. Qing, T. Xinhai and Z.Yingchun, Optimum Path Planning for Mobile Robots Based on a Hybrid Genetic Algorithm, *In Proc. HIS'06*. (2006) pp. 53-58.
- [8] Y. Guo, M. Yang, A novel knowledge included path planning strategy for the Mobile Robots, *International Journal of Computer Science and Network Security*, May 2008.
- [9] Z.Qidan, Y. Yongjie and X. Zhuoyi, Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing *Proc. ISDA'06*, (2006) pp. 622-627.

**Received: February, 2011**