Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 290–297

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

A Robust Path Planning For Mobile Robot Using Smart Particle Swarm Optimization

Harshal S. Dewang[a], Prases K. Mohanty[a]* Shubhasri Kundu[b]

[a]*National Institute of Technology, Arunachal Pradesh, 791112,India.*
[b]*KIIT University, Bhubaneswer, 751024, India.*

Abstract

In this paper, a new approach is presented for getting a solution of the mobile robot path planning problem based on Adaptive Particle Swarm Optimization (APSO). The proposed APSO algorithm is smarter than conventional PSO and widely used for solving the real time problems. In this work an objective function is framed considering the distance between robot to goal and obstacle respectively. The objective function is optimized with of APSO for solving the path planning process of robot. The different simulated experiments are performed to test the ability of the proposed algorithm. The performance of the robot path planning using APSO is compared to the performance of the conventional PSO in terms path length and time in static environments. It is focused that using new approach the robot can successfully avoid obstacle and reach the target with shorter time than conventional PSO.

*Keywords:*Particle Swarm Optimization (PSO); Mobile Robot; Path Planning; Obstacle Avoidance.

1. Introduction

Now-a-days mobile robots are vastly used in many areas such as in military purposes, space research, emergency situations like fire hazard, medical use etc. The robot completed above type of tedious tasks efficiently and effectively without any human interruption. To cope up with such situation “Path Planning” term has been introduced. In path planning, robot needs to navigate on a particular route whether the environment is familiar or not to the robot. During navigation of mobile robot various types of obstacles or hurdles comes across the robot and it needs to overcome those hurdles safely without collision and find the suitable path from source to goal point.

* Corresponding author. Tel.: +91-948-523-6260
*E-mail address:*pkmohanty30@gmail.com,

10.1016/j.procs.2018.07.036

Nomenclature

v	velocity of particle
w	inertia weight which correlate between the exploration and exploitation characteristics in PSO
c_1, c_2	constant balancing factors for the particle best parameters and social parameters or acceleration constant
rand ()	random number between 0 to 1
f	fitness function

Generally robot path planning problems are categorized as off-line or global path planning and on-line or local path planning [1]. In off-line planning, environment is familiar to the robot as well as the immobile obstacles are present. In this type of path planning before the robot starts its movement, the algorithm has to ready the whole path with co-ordinates using various types of methods. On the contrary, local path planning runs over completely in unfamiliar environment i.e. online path generation occurs as the obstacle occurrence between the source and destination point. In these both methods, path decided by the information given by the sensors about the environment.There are mainly four steps towards the navigation of mobile robot [2]. 1) Perception: Using sensors, robot generates information required about the environment. 2) Localisation: Robot reminds its own position and orientation in the environment by each iteration. 3) Cognition and Path planning: To reach at the goal point, path is determined by the robot. 4) Motion control: By controlling the motion, robot tracks its path.

Many researchers solved the mobile robot path planning problems using classical and heuristic algorithms. In the classical path planning technique includes 1.Road map method 2.Potential field method 3.Cell decomposition method 4.Subgoal Method 5.Sampling based Method and many more [3-4]. In Roadmap technique, path is generated by both curved and straight lines by basically using two approaches 1) Visibility Graph 2) Voronoi Diagram [5]. In visibility graph technique, for short length path, algorithm generates path close to the obstacle as much as possible so that the possibility of accident is increase in this case. Whereas in Voronoi diagram, path generation is away from the obstacle so that robot will stay safe without any damage but path is not claim as shortest. This method gives maximum clearance to the path but become more complex for implementation. In potential field method, total concept is based on the repulsive and attractive forces acted on the goal and obstacle respectively. Robot is allure towards the goal point due to attractive force and push backward from the obstacle due to repulsive force. It becomes more complicated when the number of points in the environment increases and it is easily fall into local minima [6]. In Cell decomposition method, total environment is divided into small segments, some of them are free, and some are occupied with obstacles. The boundaries are accomplished between the cells [7]. This method is good for optimal path finding in classical approach, but the path is decided by the grid resolution of the environment. In the subgoal method, list of accessible configuration from the initial configuration is abandon and as the goal configuration is accessible, the path is formed [8]. The Sampling based methods are efficient for high dimensional problems but solution is probabilistically obtained. In this method the rate of the convergence is totally depends upon the use of local planner [9]. In classical methods the main disadvantage is that it suffers the local trap condition and inefficient and complex to implement. Therefore in recent years researchers are motivated towards the heuristic approaches and it takes less computational time and easier than classical methods.

In heuristic approach, the fuzzy logic technique is mainly depends upon the “IF-THEN” rules [10]. The total number of fuzzy rules for designing a logic system is decided on the number of input parameters. By help of these rules framing the target approach, obstacle avoidance and the path planning for robot have to be decided. Selection of appropriate membership functions as well as rule sets is the most important and difficult task in this technique. The Artificial Neural Network (ANN) is very popular among the heuristic approach method because its learning capability and the parallel processing. When we introduce ANN to path planning problem mainly three parameters are to be validated 1) collection of sensory data 2) obstacle avoidance 3) path planning [11-13]. However in path planning problems, ANN is time-consuming method. The path planning problems for mobile robot using Neuro-Fuzzy technique discussed by many researchers. In fuzzy logic controller, ANN is used as a preprocessing part. It provides better result as compare to standalone techniques [14]. Nature-inspired algorithms like Ant colony Optimization (ACO) [15], Genetic Algorithm [16], Particle swarm optimization (PSO) and hybrid techniques [17-18] are analyzed by many researchers for path planning in mobile robot.

This paper concentrates on path planning of mobile robot in a static environment, where a new approach is proposed based on Particle Swarm Optimization (PSO). The PSO algorithm was introduced by [19] and successfully implemented in many engineering areas. By using features of PSO algorithm, find the optimum solution for mobile robot in a static environment.

Saska et al. [20] proposed path formation using PSO with string of cubic splines and compares result with potential field and visibility Graph. The generated path is smooth and easy to execute as they applied parameter like weight of safeness so that it will help to go for bad decision instead of having no decision as robot can move the obstacle. PSO and probabilistic roadmap method (PRM) used by Masehian et al. [21] for path smoothening in path planning. They connected the PSO particles as pseudo nodes and random nodes by PRM and calculate angle between shortest path and connected nodes to obtain smoothen path and validate result using Dijkstra's algorithm. Zhang et al. [22] vary the particle size and test PSO on minimum and maximum hypervolume environments and used the resample method to obtain optimum path also they combine sampling results and uniform mutation to update particle positions. Mathematical model for localization of robot and accelerated particles on optimal path using PSO is applied by Mohamed et al. [23]. Time required to reach the goal and optimal path following both depends upon population size, as size increases, time required to reach goal increase and we get more optimal path and if size decreases, time decreases to reach the goal but not that much optimal path is obtain. Yong et al. [24] worked on 4-dimensions (3 spatial and 1 time based) obstacle. By using parameterization method, calculus of variation problem is converting to time varying nonlinear program. The solution for this problem is solved with PSO using penalty function. Biogeography based optimization (BBO), voronoi boundary network (VBN) and PSO all are combined by Hongwei et al. [25] in static environment. They applied updating particle characteristics of PSO to increase variety of population in BBO. Then use updated algorithm to find the optimal path by VBN. Daniel et al. [26] tries to solve unification and optimization problem in local path planning using PSO and used coverage control points for trajectory constraints using Splines, Bezier curves and Radial basis functions. Conventional PSO on various non–convex shapes of obstacles with help of random sampling performed by Alam and Rafique [27] to get optimal results. They also tested the trap situation so that the local minima not trap the robot.

Many researchers are working on PSO algorithm but very few literatures are found on adaptive nature of PSO. The total implementation of PSO is only about 6% in mobile robot navigation compare to all the classical and heuristic algorithms [28]. Hence, we motivated to deploy APSO algorithm for resolve the path-planning problem of mobile robot. An objective function has designed to track the shortest path, avoid collision with obstacle and reach at the goal. The simulation results are presented using conventional PSO and APSO in different environments. It noticed that APSO performed better results than conventional PSO.

2. Particle Swarm Optimization (PSO)

The PSO is a population based algorithm, in which swarm of particles are get into the objective function in each iteration and compare nature with personally and socially with fitness function and obtain high convergence rate[19]. The velocity of each particle is update in each iteration with respect to the internal and the social behavior of the particles. Due to this updating characteristic of PSO, it becomes unique in all other evolutionary algorithms. The velocity and position of each particle is update using equation (1) and (2). [19]

$$\left[v(i,j)\right]_a^{b+1} = \begin{bmatrix} w \times v(i,j) + c_1 \times rand(\ ) \times \left(partb(i,j) - x(i,j)\right) \ldots \\ + c_2 \times rand(\ )\left(globb(i,j) - x(i,j)\right) \end{bmatrix}_a^b \tag{1}$$

$$\left[x(i,j)\right]_a^{b+1} = \left[x(i,j)\right]_a^b + \left[v(i,j)\right]_a^{b+1} \tag{2}$$

Where; a= particle no. in the total population; b=iteration no. ; (i, j)=co-ordinates no. in x and y axis respectively; c_1 term=Cognitive parameter; c_2 term=Social parameter; $partb(i,j)$=particle best value for that particular particle that up to the performed iterations; *globb(i,j)*=Global best value among all the particles that up to the performed iterations.

2.1. PSO Algorithm

Initialize the PSO parameters
For maximum no. of the iteration for optimum solution
  For every particle
    Calculate the fitness value from objective function
  end
  For every particle
    Calculate the Velocity and apply velocity constraints to groups of particles
  end
  Update the fitness value of every particle if previous one is better then keep it same otherwise change it with the new one
  Update the **particle best** to each particle
  Update the **global best** particle
End

2.2. Adaptive Particle Swarm Optimization (APSO)

Adaptive means which will capable to accept the change for that any one parameter in velocity equation (1) like inertia weight or acceleration parameters etc. have to change continuously in every iteration. As the large value of inertia weight gives effect on the c_2 term and small value of inertia weight parameter is responsible for the c_1 term. The search capabilities of PSO algorithm automatically gets increase by keep varying the inertia weight. The inertia weight can be varying in two ways.

1) Constant and varying Inertia weight

According to Shi and Elberhart [29], "*w*" must be more than 0.9 and less than 1.2 to get optimum result but with these values, for some problems low exploration occurs and the result may stuck in local minima. In [30], a varying value of "*w"* (normally between 0.5 and 1) utilized to obtain the better results in the dynamic environment in equation (3).

$$w = 0.5 + \frac{rand(\ )}{2} \tag{3}$$

2) Iteration based varying inertia weight

In this type the inertia weight is varying according to the number of iteration in linear or non-linear manner.

$$w = w_{\max} - \frac{(w_{\max} - w_{\min}) \times z}{\max ite} \tag{4}$$

Where; $w_{\max}$=maximum value assign to the inertia weight according to the algorithm; $w_{\min}$=minimum value assign to the inertia weight according to the algorithm; z=current iteration no.; max*ite*=maximum no. of iteration in the algorithm.

According to the equation no. (4), initially the value of the inertia weight is high and as the no. of iteration increases, the value of inertia weight is decreases. At we run the algorithm with high value of inertia weight exploration occurs and as we come closer to the result with low value of inertia weight exploitation occurs. This trend contributes to vary the inertia weight parameter continuously so that local optimal may be avoided during the path planning. It's convenient to implement as well as effective method to get optimal solution. Therefore, we implemented APSO with iteration based varying inertia weight for getting an optimal route for mobile robot in unknown environment.

3. Architecture of mobile robot path planning using APSO

In mobile robot navigation, the robot has to frame its own path using its intelligence behavior with the condition that it should not have any collision with the obstacles and reach to the goal with minimum time and energy. To fulfill above stated criteria the robot either has to accelerate itself or track the shortest path to reach at the goal. If the robot follows the shortest path, then distance between robot and goal should be minimum as shown in Fig. 1. The minimum distance can be determined by following formula (5).

$$f_1(\mathrm{i}) = distRG(\mathrm{i}) = \left[\left|(Rx - Gx)^2 + (Ry - Gy)^2\right|\right]^{1/2} \tag{5}$$

Where; i=no. of iteration; *distRG(i)* = Distance between Robot and Goal in the i^{th} iteration; *Rx*=x-co-ordinate of the robot current position; *Ry*=y-co-ordinate of the robot current position; *Gx*=x-co-ordinate of the goal point; *Gy*=y-co-ordinate of the goal point; $f_1(i)$= distance function of robot and goal.

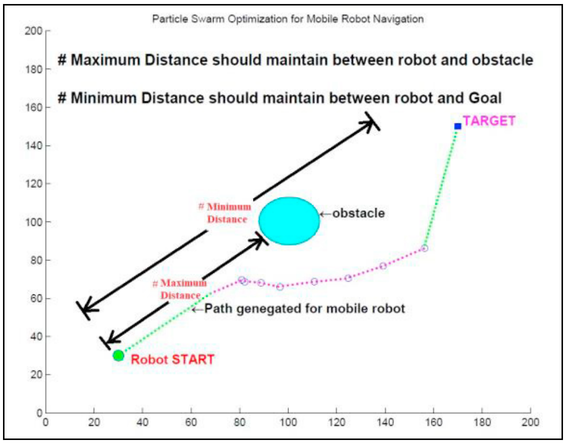

Fig. 1. Architecture of robot navigation.

During tracking the shortest path, robot also has to take care of the obstacle position. If the robot collides with the obstacle then the robot's parts may get damage so to avoid such situation robot should have to maintain the maximum distance from the obstacle as shown in Fig. 1. This maximum distance can be determined using distance formula (6).

$$f_2(i) = distRO(i) = \left[\left|(Rx - Ox(\mathrm{i}))^2 + (Ry - Oy(\mathrm{i}))^2\right|\right]^{1/2} \tag{6}$$

Where; i=no. of iteration; *distRO(i)*=Distance between Robot and obstacle in the i^{th} iteration; *Rx*=x-co-ordinate of the robot current position; *Ry*=y-co-ordinate of the robot current position; *Ox(i)*=x-co-ordinate of the obstacle point; *Oy(i)*=y-co-ordinate of the obstacle point; $f_2(i)$= distance function of robot and obstacle.

Each particle in the swarm must have to calculate the distance between the robot to goal and obstacle to avoid the collision and reach safely to goal. The objective function or fitness function for each particle can be calculated as follows (7),

$$f(i) = f_1(i) + f_2(i) \tag{7}$$

From objective function (7), the particle's local best and the Global best position are decided. Based on particle's global best position, the new position for robot is determined. The iteration continues until the robot reached at goal position.

3. Simulation results and discussion

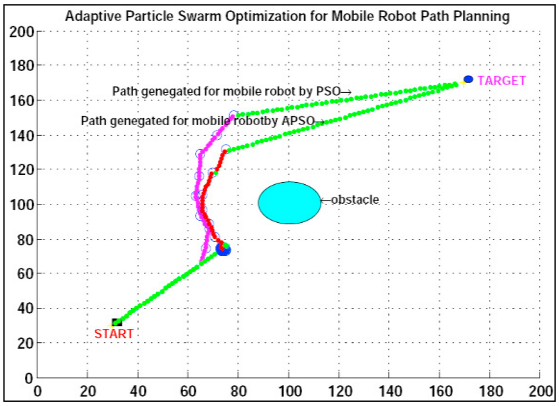

Fig. 2. Single obstacle avoidance using PSO and APSO.

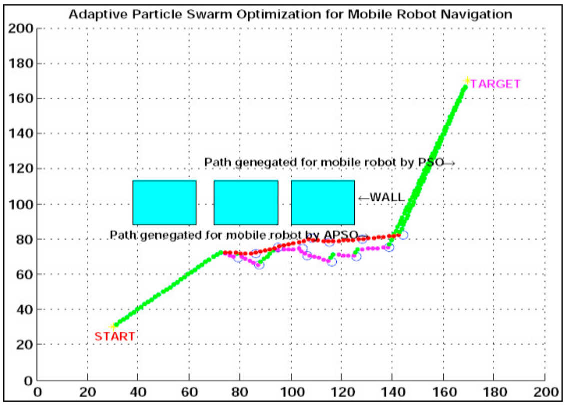

Fig. 3. Wall following behavior using PSO and APSO.

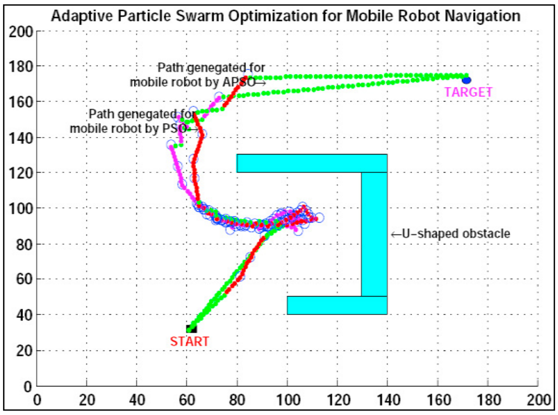

Fig. 4. Escaping from trap condition using PSO and APSO

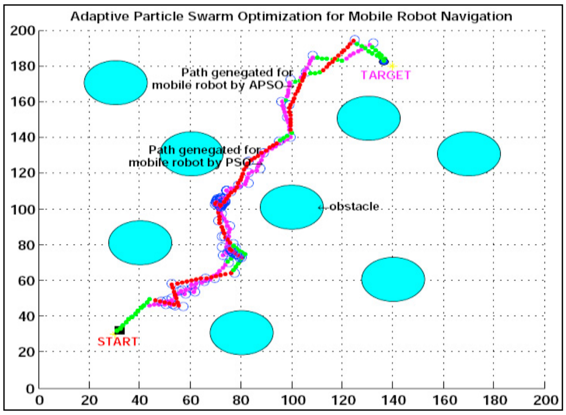

Fig. 5. Obstacle avoidance in maze environment.

The simulation results are compiled through MATLAB in a PC of windows 8.1 OS, Intel(R) Core(TM) i3-4160 CPU with 3.60GHz and 4GB RAM. First, we create an environment of 200x200 units, the unit is assumed as centimetres. The starting point and goal point for robot are represented by black square and blue circle respectively and the robot is considering as a point. The obstacles are scattered randomly in the environment. In the Fig. 2, initially the robot is far away from the obstacle so robot moves directly towards the goal without using any intelligence technique as shown by the green colour path. When the robot enters to the threshold region of the obstacle, the APSO algorithm is initiated to search a suitable position for robot using equation no.7. Using the proposed algorithm, the robot completely avoids the obstacle and follows the target path. The various parameters for APSO are given in Table No.1. The path generated using conventional PSO is marked with pink colour path where as in APSO path shown in red colour respectively. In Fig. 3 the robot has successfully defended the wall following behavior using APSO. The robot escaped from trapped condition as shown in Fig.4. In Fig.5 the mobile robot successfully navigate in a maze environment. The efficacy of the anticipated APSO algorithm has been verified with conventional PSO algorithm in terms of duration requires and path length for robot to reach at the goal. The comparison results are presented in Table-2.

Table 1. Parameters used in simulation for APSO.

Parameters	Value
No. of particles	80
C_1,C_2	2
w_{min}	0.8
w_{max}	1.1

Table 2. Path length and time covered by robot using PSO and APSO.

Figure No.	Path length covered by robot (in pixels)		Duration requires for robot to reach at the target from start point (in seconds)	
	PSO	APSO	PSO	APSO
Fig. 2	441.72	434.14	3.466	3.360
Fig. 3	447.42	428.14	3.895	3.638
Fig. 4	600.81	580.11	6.605	6.199
Fig. 5	466.93	456.32	5.129	3.975

From Table-2, it can be concluded that the APSO is performing better results than the conventional PSO. The path length covered by robot in conventional PSO in all type of environment is more than APSO, so more energy loss occurs. Hence using APSO the robot can achieve goal more efficiently. We also record the time required by the robot to achieve the goal for both algorithms, it is noticed that the adaptive one provides better result than conventional. So APSO requires minimum time to achieve the target with following collision free path.

4. Conclusions

The following conclusions are presented on robot path-planning using adaptive PSO algorithm:

- With the help of distance between robot to obstacle and goal, a new objective function has been constructed. The objective function is optimized using adaptive PSO algorithm to figure out the solution for path planning in mobile robot.
- In proposed APSO algorithm the exploration and exploitation rate are more efficient than conventional PSO and the local minima situations are precisely handled.
- The simulation experiment using APSO has been performed in different environmental situations and it has been observed that the robot has capability to successfully avoid the obstacle and follows the target path.
- Comparison in between conventional PSO and APSO has been carried out with the help of parameters like duration required and path length travelled by the robot to reach goal with different types of environmental simulations. It has been noticed that using APSO the robot can effectively perform navigation process with shortest path and time than conventional PSO.

In future, the suggested path planning algorithm can be tested in more complex environments and multiple robots may be deployed in place of single mobile robot.

References

[1] Prases K. Mohanty and Dayal R. Parhi (2013) “Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques: a Review” *Journal of Advance Mechanical Engineering* (**1**): 24-39.

[2] R. Siegwart, I. Nourbakhsh and D. Scaramuzza (2011) “Introduction to autonomous Robots” *second edition The MIT press. Cambridge, Masachusetts, London, England*.

[3] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran and Robin De Keyser (2016) “Heuristic approaches in robot path planning: A survey” *Elsevier- Robotics and Autonomous Systems* (**86**):13–28.

[4] Ellips Masehian and Davoud Sedighizadeh (2007) “Classic and Heuristic Approaches in Robot Motion Planning– A Chronological Review” *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering Vol:**1**, No:5*: 228-233.

[5] Lydia E. Kavralu, Petr Svestka, Jean-Claude Latombe and Mark H. Overmars (1996) “Probabilistic roadmaps for path planning in high-dimensional configuration spaces” *IEEE Transactions on Robotics and Automation* (**12**): 566-580.

[6] S.S. Ge and Y.J. Cui (2002) “Dynamic Motion Planning for Mobile Robots Using Potential Field Method” *springer- Autonomous Robots* (**13**): 207-222.

[7] A. Howard, K. Iagnemma and A. Kelly (2016) “Field and Service Robotics” *Springer-Verlag London*.

[8] Christian Hofner and Giinther Schmidt (1995) “Path planning and guidance techniques for an autonomous mobile cleaning robot” *Elsevier-Robotics and Autonomous Systems* (**14**): 199-212.

[9] Konstantinos I. Tsianos, Ioan A. Sucan and Lydia E. Kavraki (2007) “Sampling-based robot motion planning: Towards realistic Applications” *Elsevier- computer science review* (**1**): 2-11.

[10] Dayal R. Parhi (2005) “Navigation of mobile robots using a fuzzy logic controller” *Journal of intelligent and robotic systems*, 42(**3**):253-273.

[11] An-Mi Zou, Zeng-Gaung Hou, Si-Yao Fu and Min Tan (2006) “Neural Networks for Mobile Robot Navigation: A Survey.” *Springer-Verlag Berlin Heidelberg*: 1218–1226.

[12] Dayal R. Parhi (2008) “Neuro-Fuzzy Navigation Technique for Control of Mobile Robots” *INTECH- Mobile robot motion planning*: 409-432.

[13]Wei He, Y. Chen and Z. Yin (2016) “Adaptive Neural Network Control of an Uncertain Robot With Full-State Constraints” *IEEE-Cybernetics* (**46**):620-630.

[14] Prases. K. Mohanty and Dayal R. Parhi (2014) “Navigation of autonomous mobile robot using adaptive network based fuzzy inference system” *Springer- Journal of Mechanical Science and Technology* 28 (**2**): 2861-2868.

[15] TAN Guan-Zheng, HE Huan and Sloman Aaron (2007) “Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots” *Acta automatica Sinica* (**33**): 279-285.

[16] Yanrong Hu and Simon X. Yang (2004) “A Knowledge Based Genetic A1gorithim for Path Planning of a Mobile Robot” *IEEE-international Conference on Robotics and Automation*: 4350-4355.

[17] Prases K. Mohanty and Dayal R. Parhi (2015) “A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach.” *Springer-Verlag Berlin Heidelberg* (**7**):255–273.

[18] Hao Mei, Yantao Tian and Linan Zu (2006) “A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment” *International Journal of Information Technology* (**12**): 78-89.

[19] James Kennedy and Russell Eberhart (1995) “Particle Swarm Optimization” *IEEE*: 1942-1948.

[20] Martin Saska, Martin Macas, Libor preucil and Lenka Lhotska (2006) “Robot Path Planning using Particle Swarm Optimization of Ferguson Splines” *IEEE- Emerging Technologies and Factory Automation*: 839-833.

[21] Ellips Masehian and Davoud Sedighizadeh (2010) “A Multi-Objective PSO-based Algorithm for Robot Path Planning” *IEEE- International Conference on Industrial Technology*: 465-470.

[22] Yong Zhang, Dun-wei Gong and Jian-hua Zhang (2012) “Robot path planning in uncertain environment using multi-objective particle swarm optimization” *Elsevier- Neuro-computing* (**103**):172–185.

[23] Abdullah Zawawi Mohamed, Sang Heon Lee, Hung Yao Hsu and Namrata Nath (2012) “A faster path planner using accelerated particle swarm optimization” *Springer- Artificial Life and Robotics* (**17**):233-240.

[24] Yong Ma, M. Zamirian,Yadong Yang, Yanmin Xu and Jing Zhang (2013) “Path Planning for Mobile Objects in Four-Dimension Based on Particle Swarm Optimization Method with Penalty Function” *Hindawi Publishing Corporation- Mathematical Problems in Engineering* .

[25] Hongwei Mo and Lifang Xu (2014) “Research of biogeography particle swarm optimization for robot path planning” *Elsevier-Neuro-computing* (**148**): 91–99.

[26] Nancy Arana-Daniel, Alberto A. Gallegos, Carlos Lopez-Franco and Alma Y. Alanis (2014) “Smooth Global and Local Path Planning for Mobile Robot Using Particle Swarm Optimization, Radial Basis Functions, Splines and Bezier curves” *IEEE- Congress on Evolutionary Computation*: 175-182.

[27] Muhammad Shahab Alam and Muhammad Usman Rafique (2015) “Mobile Robot Path Planning in Environments Cluttered with Non-convex Obstacles Using Particle Swarm Optimization” *IEEE- International Conference on Control, Automation and Robotics*: 32-36.

[28] S. H. Tang, W. Khaksar, N. B. Ismail and M. K. A. Ariffin (2012) “A Review on Robot Motion Planning Approaches” *Pertanika Journal Sci. & Technology* (**20**):15-29.

[29] Yuhui Shi and Russell Eberhart (1992) “A Modified Particle Swarm Optimizer” *IEEE- Computation Proceedings*: 69-73.

[30] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh and Reza Safabakhsh (2011) “A novel particle swarm optimization algorithm with adaptive inertia weight” *Elsevier- Applied Soft Computing* (**11**): 3658–3670.