Sindh Univ. Res. Jour. (Sci. Ser.) Vol. 47 (4) 653-658 (2015)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Autonomous Robot Path Planning Using Particle Swarm Optimization in Static and Obstacle Environment

M. A. MEMON[++], S. MEMON, Z. BHATTI, S. A. KHOWAJA, B.BALOCH*

Institute of Information and Communication Technology (IICT), University of Sindh, Jamshoro

Received March 2015 and Revised 27th September 2015

Abstract: This paper presents a simple and effective approach for mobile robot that detects and avoids densely populated and randomly distributed same size circular shaped static obstacles. The PSO (Particle Swarm Optimization) algorithm is implemented in the proposed technique to determine the optimum route of a robot from source to destination point until any obstacle is detected on its path. Once any obstacle is detected over the optimized path, the obstacle avoidance is done by moving robot towards the nearest safe point around the obstacle's boundary which is pre-defined and calculated. Furthermore, proposed technique calculates the next point around the obstacle's boundary which is nearest to the predefined target. For finding the effectiveness, PSO and GA (Genetic Algorithm) is applied and simulated on three different cases. Each case has variation in the location of starting point and goal. Finally, simulation results of all three cases are compared in-terms of Number of Iterations, Path Length and Execution Time. In result, the PSO performs better than GA in all three cases and provides a collision free smooth path in simulated environment.

Keywords: Mobile Robot, Static Obstacles, Particle Swarm Optimization, Genetic Algorithm, Path Length

1. INTRODUCTION

In the modern era, the robots are engaged in various services in which collision free path planning is always the key concern like delivery, cleaning, disaster rescue, and space exploration. Various algorithms have been researched and developed, however; still there is a huge need of improvement and enhancement in the performance of such techniques, which would make the locomotion of robot more autonomous in an environment with random obstacles. Furthermore, the robot system needs to be able to determine and opt for shortest feasible path, from source to destination, that ultimately save time, energy and cost of the system.

Path planning algorithms are divided into two categories, global path planning or off-line and local path planning or on-line. If complete information of environment is available in advance, for example location of static obstacles and other moving entities, then it is called a global path planning. Whereas, if the environment information is not known in advance, then the robot moves according to present scenarios and sensor readings, this is called as local path planning (Harshini and Ramji, 2015), (Leena and Saju, 2014).

In past, classical techniques were used for path planning like CD (Cell Decomposition), PF (Potential Fields), mathematic programming, and Roadmap. However, these techniques have various limitations such as they consume too much time to solve large space problem and stuck in local optimum position (Atyabi *et al.*, 2013), (Leena and Saju, 2014). Due to these limitations heuristic and meta-heuristic algorithms such as, ACO (Ant Colony Optimization), GA (Genetic Algorithm), ANN (Artificial Neural Network), MA (Memetic Algorithm), Tabu Search, SFL (Shuffled Frog Leaping), SA (Simulated Annealing) and PSO (Particle Swarm Optimization) were used for path planning (Atyabi *et al.*, 2013), (Leena and Saju, 2014). Mostly researchers used PSO algorithm, because it is simple, efficient and computationally less complex, and also it has high success rate and solution quality as compared to MA, ACO, GA, and SFL (Elbeltagi, *et al.*, 2005). In literature, various PSO based algorithms were suggested for obstacle avoidance and path planning like GSA (Gravitational Search Algorithm) (Purcaru, *et al.*, 2013), RRT (Rapidly-exploring Random Tree) (Zhou and Gon, 2013), fuzzy (Cai and Yang, 2013), Dijkstra (Wang and Yu, 2012) and PRM (Probabilistic Roadmap Method) (Masehian and Sedighizadeh, 2010), each one addressed distinct complex environments by assuming different shapes, size of obstacles and length of search space. The Genetic Algorithm was applied by (Ghorbani, Shiry and Nodehi, 2009), (Jun *et al.*, 2010), (Yun, *et al.*, 2010), and (Cui, *et al.*, 2013) to optimized the distance from target in obstacle environment, other algorithms like ACO (He *et al.*, 2013), (Mohanraj, *et al.*, 2014), (Ganganath and Cheng, 2013) and firefly (Li, *et al.*, 2014) were also used to address same problem.

[++]Corresponding author: memon.mudasir@usindh.edu.pk
*Department of Statistics, Sindh Agriculture, University, Tando Jam

In this paper, the problem of robot path planning in static and densely populated environment with obstacles is addressed and solved by PSO based obstacle detection and avoidance technique and results are compared with GA. The PSO algorithm is used as global path planner and finds optimized path from source to the destination, and the proposed technique is used to detect and avoid obstacles that results in collision free path.

2. GENETIC ALGORITHM

The GA (Genetic Algorithm) was developed by John Holland in 1975 (Joshi, 2014) and it was inspired by Darwin's theory of evolution. The rule was stated as "The survival of species depend on how strong he is". Later on, this concept of Darwins was utilized in computational algorithms to find optimized solutions for complex problems. In genetic algorithm, set of solution is called chromosomes or population. Genetic algorithm is composed of three basic stages, selection, crossover and mutation. In selection stage, parents or healthy genes identified through a process of fitness evaluation and passed to the next generation. In crossover stage, selected parents give birth to new offspring's by interchanging the part of information between them. In mutation stage, a new gene introduced randomly to enhance the global searching within the desired region. In Genetic Algorithm, optimized path for robot motion from origin to destination in static obstacle environment is pre-selected. The optimized Path is divided into obstacle free short length "N" segments; end point of each segment is treated as a target point (Ghorbani, Shiry, and Nodehi, 2009), (Jun *et al.*, 2010), and (Yun, *et al.*, 2010) as shown in **(Fig. 1).** For e.g. the path shown in figure is divided into 8 non equal segments represented by [start point to T_1], [T_1 to T_2], [T_2 to T_3], [T_3 to T_4], [T_4 to T_5], [T_5 to T_6], [T_6 to T_7], and [T_7 to Target], and in each segment, end points [T_1, T_2, T_3, T_4, T_5, T_6, T_7, Target] are the local targets for obtaining the collision free route.

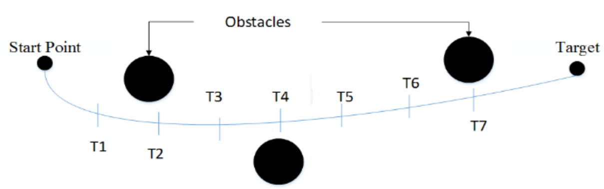


Fig. 1: GA Algorithm based Path Planning of Robot in Static and Obstacle Environment

The flow chart of GA based Path Planning of robot is shown in (**Fig. 2)**. First, the chromosomes, crossover rate, mutation rate, epsilon and target for the mobile robot are initialized. Then, fitness of each chromosome is evaluated using equation (1).

$$\text{Fitness Function} = |\text{Target} - \text{Current Location}| \qquad (1)$$

After calculating the fitness, parents have been found using Roulette Wheel Method (Joshi, 2014) then crossover operation is applied that generates new off springs for next generation. Mutation operation on child adds new gene that enhances the global searching capability of robot within target limits, and robot reaches its goal in target by target manner.

3. PARTICLE SWARM OPTIMIZATION

The PSO (Particle Swarm Optimization) algorithm was developed by Russell C. Eberhart and James Kennedy in 1995 (Talukder, 2011) and it was derived from the swarm behavior of birds while searching food or nest.

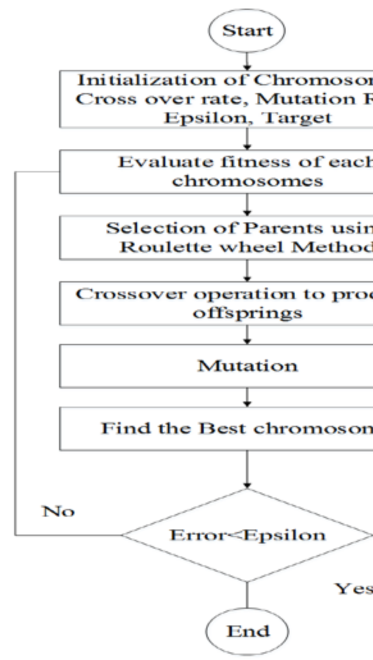


Fig. 2: GA based Path Planning Algorithm

The term "particle" means individual bird which has a solution for the problem. The PSO technique is comprised of two equations; Equation 2 mentioned below is used for updating the velocity of particle and equation 3 is used for updating the position of a particle.

$$v_{ij}^{t+1} = [v_{ij}^{t} \times \omega^{t+1}] + [c_1 \times r_{1j}^{t} \times [P_{best,i}^{t} - x_{ij}^{t}]] + [c_2 \times r_{2j}^{t} \times [G_{best} - x_{ij}^{t}]] \qquad (2)$$

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1} \qquad (3)$$

The term ω^{t+1} is the inertia represented by equation (4) and it linearly decreases from 0.9 to 0.4.

$$\omega^{t+1} = \omega_{max} - \left[\frac{\omega_{max} - \omega_{min}}{\text{Total Number of Iterations}} \times \text{Current Iteration}\right] \quad (4)$$

The individual best location or personal experience of a particle is represented by term "$P_{best,ij}^{t}$" and the particle which has best fitness (solution) in the swarm is represented by term "G_{best}". The term "x_{ij}^{t}" and "v_{ij}^{t}" are current position and velocity of a particle "i" in "j" direction, respectively. The influence of personal experience and groups experience is controlled by acceleration constants C_1 and C_2, whereas r_{1j}^{t} and r_{2j}^{t} are random numbers generated between 0 and 1 (Atyabi *et al.*, 2013).

4. <u>PROPOSED TECHNIQUE</u>

In this research, a simplest mechanism is proposed for obstacle detection and avoidance for environments that contains static circular shaped same size obstacles. The proposed approach is shown in **(Fig. 3).**

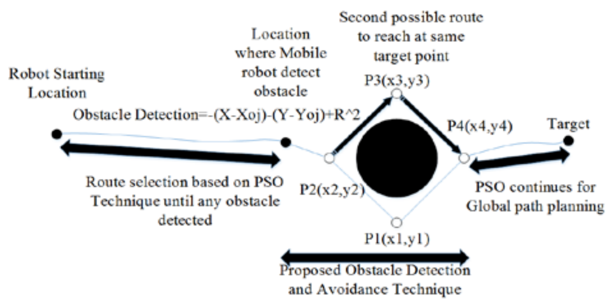


Fig. 3: PSO based Proposed Obstacle Detection and Avoidance Technique

Each obstacle has pre-defined four surrounding points for safe crossing of robot. At the initial stage, robot selects its optimized route from starting point to goal using PSO algorithm until any obstacle detected. In this experiment, grid based [100×100] environment is modeled as a work space for mobile robot as shown in **(Fig. 4).**

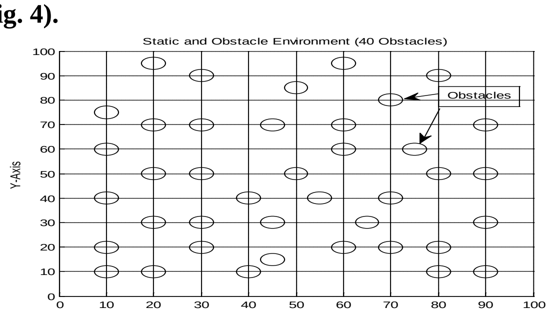


Fig. 4: Static and obstacle Environment

The 40 randomly distributed circular shaped objects with radius of 5 cm are supposed as obstacles and placed at random distances from each other. The flowchart of proposed algorithm is shown in **(Fig. 5).**

First step is the initialization of population with random positions, velocity, target, c_1 and c_2, and Epsilon. Then, the system evaluates the fitness of each particle using equation (1); here fitness is the distance of robot from target in particular direction. If any particle's current fitness is smaller than previous fitness in any axis, then this system saves the current position of the particle as P_{best}, otherwise recall previous best position of particle, where particle achieve less distance from the target. After calculation of personal best values of all particles, compare all particle's best positions (P_{best}) and select one particle that has best position among all and point out that location as Global best (G_{best}). Then, PSO algorithm is applied for finding the next velocity and location of each particle using equation (2) and (3), respectively.

Large values of velocity may results in wide change of particle's position which causes the divergence problems and particle may leave the search space. Therefore, to handle these situations, inertia weight and velocity clamping techniques are used to reduce the effect of previous velocity on the next velocity. The formula for Inertia weight "ω" is represented by equation (4).

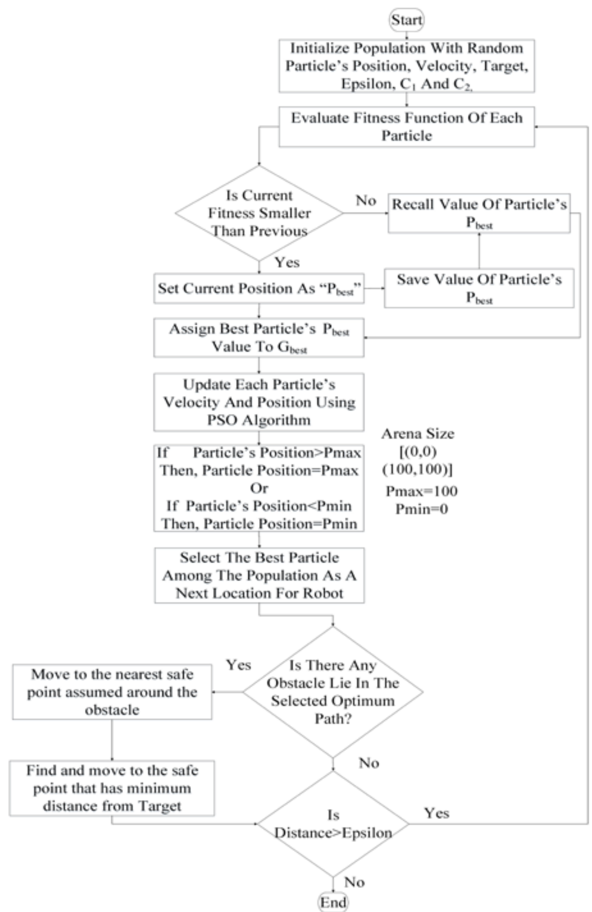


Fig. 5: Flow Chart of Proposed Technique

After application of PSO algorithm, if any particle's position value exceeds from the arena size [100×100] then this algorithm limits the particle position and velocity using equation (5) and (6), respectively, and if any particle's position value goes below the origin or comes out negative then this algorithm limits the particle position and velocity using equation (7) and (8), respectively.

If $x_{ij}^{t+1} > x_{max}$ then $x_{ij}^{t+1} = x_{max}$ $\therefore x_{max} = 100$ (5)

$v_{ij}^{t+1} = 0.5 \times v_{ij}^{t+1}$ (6)

If $x_{ij}^{t+1} < x_{min}$ then $x_{ij}^{t+1} = x_{min}$ $\therefore x_{min} = 0$ (7)

$v_{ij}^{t+1} = 0.5 \times v_{ij}^{t+1}$ (8)

After finding the next location, Equation (9) is used for obstacle detection that finds whether the next calculated position lies within the obstacle region or not.

$$\text{Obstacle Detection} = -(X - X_{o_j})^2 - (Y - Y_{o_j})^2 + R^2 \quad (9)$$

Where,

X and Y is the current location of Robot in-terms of X-axis and Y-axis.

X and Y_{o_j} is the location of particular obstacle.

R is the radius of obstacle.

The obstacle detection equation always generates negative output until next location of robot lies in obstacle region. After obstacle detection, first robot moves to the pre-defined point of a particular obstacle that have least Euclidean distance and it is calculated by equation (10), robot moves to $P_2(x_2,y_2)$ because of the shortest distance as shown in **(Fig. 3).**

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (10)$$

After that, proposed method calculates the distance of all four pre-defined surrounding safe points of particular obstacle from target and compare them. According to distance, $P_4(x_4, y_4)$ is the nearest point to the target. Then, from $P_2(x_2,y_2)$ to $P_4(x_4,y_4)$ there are two ways to reach at that particular point, one is in clock wise direction and second is in anti-clock wise direction, both has same distances from targeted safe point. After reaching at desired safe point, PSO algorithm is reapplied, until any further obstacle detects and stops when robot successfully reaches at the target.

5. SIMULATION RESULTS

In this paper, GA and PSO based proposed obstacle detection and avoidance algorithm is applied and tested to solve path planning problems of robot in static and obstacle environment. Matlab R2013b is used for coding with Intel® core™ i5-3210M CPU@2.50Ghz and windows 7 platform. The complexity of working environment is kept constant for both algorithms which facilitates in the comparison. The parameters used for both algorithms for simulation are listed in **(Table 1).**

Table 1: Values of Parameters used in GA and PSO

GA		PSO	
Chromosomes	10	Population	10
Crossover rate	0.9	C_1 and C_2	0.5
Mutation rate	0.1	ω_{max} and ω_{min}	0.9 and 0.4

In this experiment, three different cases are considered, and in each case starting position of robot and its goal is changed as mentioned in **(Table 2)**.

Table 2: Starting and Goal Points Considered for Different Cases

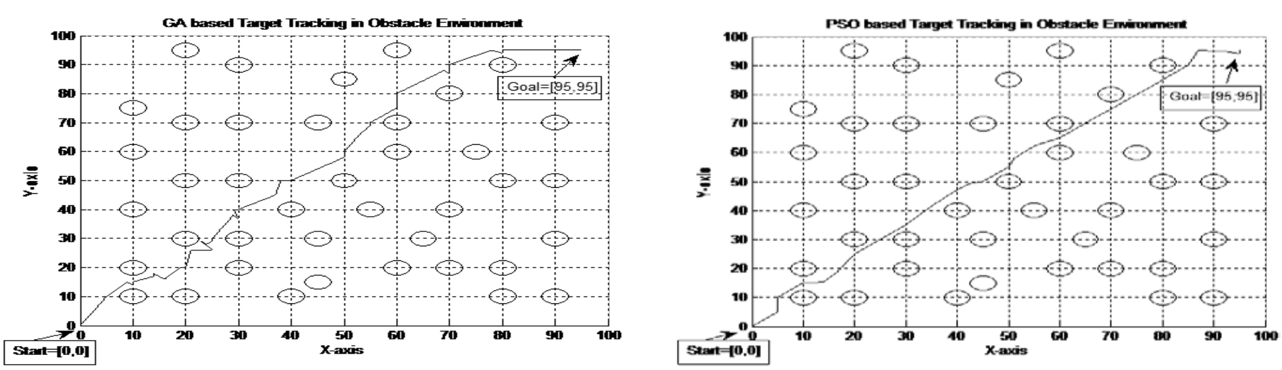
Case	Start point	Goal Point	Obstacles
1	[0,0]	[95,95]	40
2	[0,100]	[95,5]	
3	[100,100]	[5,5]	

The results of all cases are illustrated below.

Case-1:

Starting position and goal of robot in case-1 are [0, 0] and [95, 95], respectively as shown in **(Table 1)**. Simulation results for selected case using GA and proposed technique are shown in **(Fig. 6 (a)** and **Fig. 6 (b))**, respectively.

Form figures it is observed that, both algorithms creates collision free path, however in proposed technique robot smoothly reaches at goal, while in GA robot reaches at target in zigzag manner.

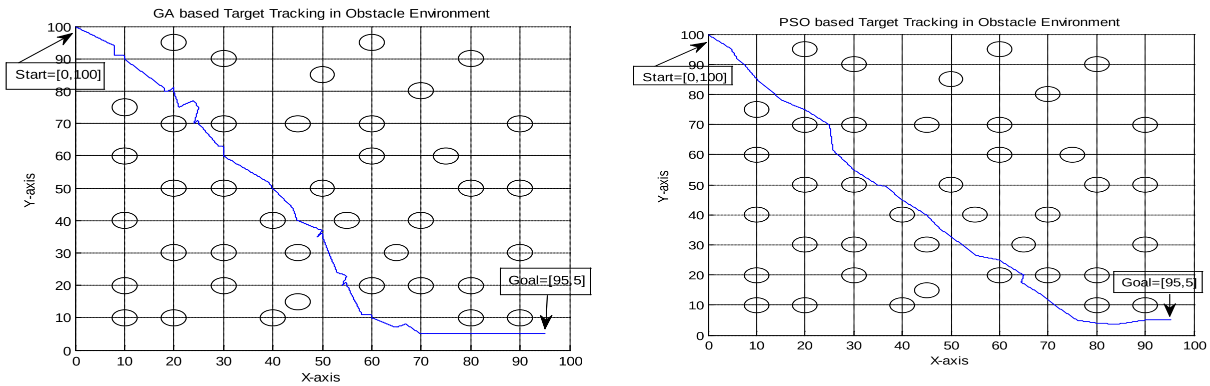


(a) (b)

Fig. 6: Simulation Results in Case-1

Case-2:

In case-2, robot starts from the location [100,100] and its destination is [95, 5]. The (**Fig. 7(a)** and **Fig. 7(b))** represent response of GA and proposed technique, respectively.

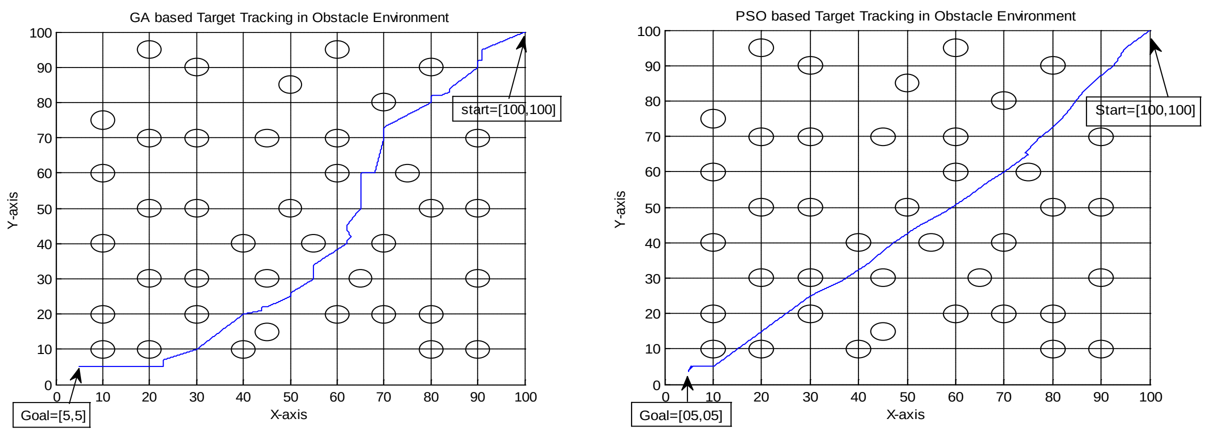


(a) **(b)**

Fig. 7: Simulation Results in Case-2

Case-3:

In case-3, starting position of robot and its goal are [100,100] and [5, 5], respectively, and simulation results for GA and proposed technique are shown in **(Fig. 8 (a)** and **Fig. 8 (b)).**



The results of Case-1, Case-2 and Case-3 are summarized in **(Table 3).** Performances of both algorithms are compared by three factors; 1) Iteration 2) Path Length and 3) Execution time. Iterations mean how many times a program repeated for finding clash free path from starting point to destination. Path length is the total distance covered by a robot from starting point to destination. Execution time is the total time consumed by program from start to termination.

Table 3: Comparison of PSO Based Proposed Technique and GA

Technique	**Case**	**Iterations**	**Path Length**	**Execution Time (Seconds)**
GA	1	216	164.3540	18.920721
	2	144	168.0758	15.503297
	3	70	156.2374	9.352444
Proposed Technique (PSO based obstacle Avoidance)	1	41	142.6817	2.706456
	2	39	145.5833	2.924204
	3	34	141.4792	2.822605

In all three cases, proposed technique generates collision free, smooth and optimum path in less number of iterations and executes program in short period of time as compared to GA.

6. CONCLUSION

In this paper, PSO based simple obstacle detection and avoidance technique is proposed for solving the path planning problems of mobile robot in static and randomly distributed, densely populated obstacle environment. The results of proposed method are compared with GA, both algorithms are tested on same working environment, and results proved that proposed technique is simple, effective, and performed better than GA under our designed environment. The result shows that, in proposed approach robot smoothly reached at its goal by selecting optimum path in less number of iterations with very short execution time.

The effectiveness of proposed technique is compared with GA; in future this can be compared with other heuristic algorithms. In addition, obstacles considered in this study are static and circular shaped, however; it needs further research to address same problem with irregular dynamic obstacles.

REFERENCES:

Atyabi A. and D. M. W. Powers, (2013). "Review of classical and heuristic-based navigation and path planning approaches," Int. J. Adv. Comput. Technol., vol. 5, no. 14, 1–14.

Cui, S. G., J.L Dong, (2013) "Detecting Robots Path Planning Based on Improved Genetic Algorithm", Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, 204-207. Shenyang, Liaoning, China.

Cai Y. and S. X. Yang, (2013) "A PSO-based approach with fuzzy obstacle avoidance for cooperative multi-robots in unknown environments," Proceeding of the IEEE International Conference on Information and Automation. 1386–1391, Yinchuan, China

Elbeltagi, E. T. Hegazy, and D. Grierson, (2005). "Comparison among five evolutionary-based optimization algorithms," Adv. Eng. Informatics, vol. 19, no. 1, 43–53.

Ghorbani, A., S. Shiry, and A. Nodehi, (2009). "Using Genetic Algorithm for a Mobile Robot Path Planning", IEEE International Conference on Future Computer and Communication, 164-166, Kuala Lumpar, Malaysia

Ganganath N. and C.T Cheng, (2013) "A 2-Dimensional ACO-Based Path Planner for Off-Line Robot Path Planning," 2013 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov, 302–307, Beijing, China

Harshini B. and K. Ramji, (2015). "Navigation of Mobile Robot in the Presence of Static Obstacles of Various Shapes," International Journal of Scientific and Research Publications, vol. 5, issue. 2, 2–6,

He, Y., Q., Zeng, J. Liu, G. Xu, and X. Deng, (2013) "Path planning for indoor UAV based on Ant Colony Optimization," 25th Chinese Control Decis. Conf., 2919–2923. Guiyang, China

Joshi, G. (2014) "Review of Genetic Algorithm : An Optimization Technique", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 4, 802–805.

Jun J., H. and Z. Q. Z. Qingbao, (2010). "Multi-objective Mobile Robot Path Planning Based on Improved Genetic Algorithm," International Conference on Intelligent Computation Technology and Automation, vol. 2, 752-756. Changsha, China.

Leena N. and K. K. Saju, (2014). "A survey on path planning techniques for autonomous mobile robots," International Conference on Advances in Engineering & Technology, 76–79. Singapore.

Li, B., L. Liu, Q. Zhang, D. Lv, Y. Zhang, J. Zhang, X. Shi, and A. Bezier, (2014) "Path planning based on firefly algorithm and Bezier curve *", Proceeding of the IEEE International Conference on Information and Automation, 630–633, Hailar, China.

Masehian E. and D. Sedighizadeh, (2010) "A multi-objective PSO-based algorithm for robot path planning," Proc. IEEE Int. Conf. Ind. Technol. 465–470. Viña del Mar - Valparaíso, Chile.

Mohanraj, T., S. Arunkumar, M. Raghunath, M. Anand, (2014) "Mobile robot path planning using ant colony optimization," International Journal of Research in Engineering and Technology, Vol. 03 Issue: 11, 1–6.

Purcaru, C., R. E. Precup, D. Iercan, L. O. Fedorovici, and R. C. David, (2013) "Hybrid PSO-GSA robot path planning algorithm in static environments with danger zones," 17th Int. Conf. Syst. Theory, Control Comput. ICSTCC, 434–439. Sinaia, Romania.

Talukder, S. (2011) "Mathematical Modelling and Applications of Particle Swarm Optimization", Masters's Thesis, School of Engineering at Blekinge Institute of Technology, Karlskrona, Sweden.

Vermeulen, G. N. and J. P. (2014) "Obstacle Avoidance Algorithm for a mobile robot", Bachelor thesis, Technische Universiteit Eindhoven, Netherlands

Wang Y. and X. Yu, (2012). "Research for the Robot Path Planning Control Strategy Based on the Immune Particle Swarm Optimization Algorithm," International Conference on Intelligent System Design Engineering Application, 724–727. Changsha, Hunan, China

Yun, S. C., V. Ganapathy, and L. O. Chong, (2010) "Improved Genetic Algorithms based Optimum Path Planning for Mobile Robot", 11th Int. Conf. Control, Automation, Robotics and Vision, Singapore, 7–10.

Zhou F. and L. E Gon, (2013) "An Improved Path Planning for Mobile Robots," International Conference on Information Science and Cloud Computing Companion,589–594. Guangzhou, China.