Target Seeking Behaviour of an Intelligent Mobile Robot Using Advanced Particle Swarm Optimization

B.B.V.L. Deepak, Dayal R. Parhi

***Abstract*— the present research work aims to develop two different motion planners for an autonomous mobile robot. The developed motion planners are inspired by particle swarm optimization and are useful for generating feasible paths by the robot within its unknown environment. Each motion planner works on its own fitness function and each fitness function are modelled based on the robot sensory information. Path analysis results showed that both motion planners are generating collision free paths and reaches its destination within its workspace. Finally, comparison has been done in between the developed PSO based motion planners.**

***Index Terms*— autonomous mobile robot, motion planning, obstacle avoidance, target seeking, particle swarm optimization.**

I. Introduction

Mobile robot navigation is an important issue in the field of robotics, which includes the generation of optimal collision free trajectories within its work space and finally reaches its destination position. Based on this issue the path planning is categorized into two types namely global path planning and local path planning. In first category, the robot generates the path from its initial position to final position in its known static environments. In second category, robot generates path trajectories within its unknown environments.

From last decades a large amount of researches have been dedicated to solve mobile robot path planning problem and various techniques such as artificial potential field, visibility graphs and cell decomposition method etc. have been proposed. Due to simple structure of potential field [1], it is widely used for robot motion planning, but this technique may face local minima problem, which leads to robot trap situation within its environments. Visibility graph [2] needs more control accuracy because its search path efficiency is low as explained in [3]. In cell decomposition method [4] work space is divided into number of parts called cells, which are predefined in size and shape. This method is not working for real time path planning and can be applied when the environment of the robot is known.

B.B.V.L. Deepak is with the Department of Industrial Design, National Institute of Technology, Rourkela, India, (phone: 0661-246-2514; e-mail: bbv@nitrkl.ac.in).

Dayak R. Parhi, is with Department of Mechanical Engineering, National Institute of Technology, Rourkela, India, (e-mail: DRKPARHI@nitrkl.ac.in).

An evolutionary computational method has been developed by Kennedy and Eberhart [5] named as Particle Swarm Optimization (PSO), which was motivated by social behaviour of bird flocking or fish schooling. This method has been successfully implemented to solve many engineering problems because of its extraordinary features like proximity, quality, diverse response, stability and adaptability. As explained by Hassan et al. [6], PSO is more efficient in computational view (uses less number of function evaluations) than the Genetic Algorithm. Many authors have applied PSO for solving scientific problems such as unknown parameters estimation in nonlinear systems [7], Bioinformatics [8], Machine Learning [9], job-shop scheduling [10] and constrained optimization problem [11] etc., due to its effectiveness and faster response.

Path planning of a mobile robot can be considered as a multi objective optimization problem because it includes generation of trajectories from its source position to destination with less distance/time traversal and avoiding obstacles within its known/unknown environments. In past, researchers have been applied PSO for solving mobile robot path planning problem. It has been proved in [12], for optimal navigation of mobile sensors, the time taken by convergence using PSO is ten times faster than time taken by fuzzy logic. PSO is considered by Zhang and Li [13] for motion planning of a robot, when the work space is having the obstacles of generalised polygons. But their execution may not generate best possible paths in all situations. To overcome this intricacy Qin et al. [3] have applied PSO integrated with mutation operator. But this approach requires lot of work to fine-tune the controlling parameters of PSO. Maschian and Sedighizer [14] have recently proposed a novel computational method based on PSO, to design a motion planner of an autonomous mobile robot. They have performed a large amount of work for adjusting the controlling parameters of PSO and obtaining optimal paths between two consecutive robot positions. Even though, their work requires an adaptive algorithm to obtain safest path in robotic environments.

The above described algorithms are useful only for known environments, but their methodologies can't be operated for unknown/partly known environments. A PSO based computational method has been outlined by Derr and Mannic [15], for motion planning of robots in noisy environments, but this methodology increase the robot search time in finding its target. PSO have been applied in [16], [17] for obstacle avoidance in automated robotic search, but they didn't summarise results for various environmental situations. Lu and Gang [18] applied PSO for generating best path of an autonomous mobile robot in unknown environment. However their implementation lacks in adjusting the controlling guidelines of their developed fitness function to improve the efficiency of system architecture.

The present researches work deals with the development of two efficient PSO based system architectures for solving mobile robot path planning problem. Motion planning of an autonomous mobile robot is considered here as a multi objective constrained optimization problem. Two new fitness functions have been selected (one for each motion planner) to solve this optimization problem. It means, controlling parameters are adjusted to get optimal trajectories within the robotic environment. Finally, simulation results are showed to verify the feasibility of the proposed methods for generating optimal trajectories in its maze environments.

II. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a computational methodology, introduced by Kennedy [5]. PSO was inspired from the natural system bird flocking, animal herds and fish schools etc. The special features observed in those social behaviours are: producing impressive, collision free and synchronized moves. When the birds are flocking for gathering food, they will move according to the sharing of information among the population. The population considered is swarm and its individuals are particles. So a swarm in PSO can be identified as a set $S = \{p_1, p_2, p_3, \ldots, p_r\}$. Where $p_1, p_2, p_3, \ldots, p_r$ are 'r' number of individuals in the swarm.

As mentioned above, each particle in swarm is moving in its search space. It means each individual is having its own position and velocity as follows:

Set of particles position= $\{d_1, d_2, d_3, \ldots, d_r\}$.

And set of particles velocity= $\{v_1, v_2, v_3, \ldots, v_r\}$.

And each particle is having its position best value (X_{pbest}). Based on the communicated information among the swarm, the particles will approach to one global best position. The particle which is having the greatest fittest is treated as the global best position (X_{gbest}). The particles in the swarm are mutually shared their experience and they will approximate to one global best position, ever visited by all particles as shown in Fig.1. This process will be continued until they reach to their destination (next/food source).

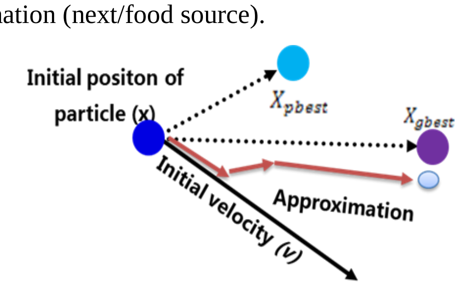

Fig.1 Basic structure of PSO for global best approximation

While the swarm is moving, the velocity and position of each particle are updating as follows:

$$v_i(z+1) = v_i(z) + C_1 * rand1 * (X_{pbest} - x_i) + C_2 * rand2 * (X_{gbest} - x_i) \quad (1)$$

$$\text{and } x_i(z+1) = x_i + v_i(z+1) \quad (2)$$

Where z= iteration number,

rand1 & *rand2* are random variables,

and C_1 & C_2 are cognitive and social parameters.

III. Implementation

In order to approximate the swarm to its best position, particles in the swarm are communicated their information for determining the best fitness valued particle (X_{gbest}). In the same way, the present research work aims for developing fitness function to select the best position of the mobile robot within its sensing range for further movement.

When the robot is moving within its search space for target seeking, it is not necessary to implement any learning strategy for its iterative motion, if the robot is not sensing any obstructions in its path. But it is very difficult task for an autonomous mobile robot, to generate collision free trajectories when it is sensing any obstacles within its sensing range.

For suppose, robot has sensed a definite number of obstacles (S_{ob}) within its sensing range, then the robot can identify the nearest obstacle according to reflected radiation intensity from the observed obstacles. Consider the mobile robot is represented as a point $(robotx, roboty)$ in X,Y – plane and centres of sensed obstacles be (obx_i, oby_i) for $1 \le i \le S_{ob}$. Then the distance between the robot and sensed obstacles can be obtained from equation (3):

$$(\lambda_{R-Ob})_i = \sqrt{(robotx - obx_i)^2 + (roboty - oby_i)^2} \quad \text{for } 1 \le i \le S_{ob} \quad (3)$$

From the obtained 'S_{ob}' number of distance values, the obstacle which is having minimum λ_{R-Ob} can be treated as a nearest obstacle. Once the robot finds the nearest obstacle (NOb) within its sensing range, robot will generate a random population around it within the sensing range. So one fitness function (*F*) is required to estimate the fitness values of each particle in the swarm for further movement of the robot.

A. Fitness function(s) generation

As explained in previous section it is necessary to find fitness value for each particle in the swarm. For this purpose a new fitness function has to be modelled by satisfying the following conditions.

1. The fitness of particle should maintain the minimal distance from the robot's target, it means the fitness function is directly proportional to the distance between the particle and robot destination.

$$\Rightarrow F_i \propto (\lambda_{P_i - T}) \quad \text{for } 1 \le i \le p \quad (4)$$

Where λ_{P_i-T} indicates the distance between i[th] particle and target position.

2. The fitness of particle should maintain the maximal distance from the nearest obstacle, it means the fitness function is indirectly proportional to distance between the individual/particle and nearest obstacle.

$$\Rightarrow \; F_i \propto \left({}^{1}\!/_{\lambda_{P_i-NOb}} \right) \qquad \text{for} 1 \leq i \leq p \qquad (5)$$

Where λ_{P_i-NOb} indicates the distance between i[th] particle and nearest obstacle.

From the above mentioned conditions shown by Eq.(4) and (5), the required fitness function can be generated as represented either Eq.(6) or Eq.(7).

$$F_i = W_1 * \left({}^{\lambda_{P_i-T}}\!/_{\lambda_{P_i-NOb}} \right) \qquad \text{for} 1 \leq i \leq p \qquad (6)$$

$$F_1 = K_1 * (\lambda_{P_i-T}) + K_2 * \left({}^{1}\!/_{\lambda_{P_i-NOb}} \right) \qquad \text{for} 1 \leq i \leq p \quad (7)$$

By observing Eqs.(6)& (7), the particle which is having the minimum fitness value can be treated as X_{gbest}, because that particle (X_{gbest}) is away from nearest obstacle and close to the goal position.

The possible minimum value of the fitness using Eq. (6) is

$$F_{min} = \left({}^{min(\lambda_{P_i-T})}\!/_{max(\lambda_{P_i-NOb})} \right) \qquad (8)$$

Therefore the final fitness function can be transformed from Eqs. (6) & (8):

$$F_{final} = |F_{min} - F_i|$$

$$\Rightarrow F_2 = \left| \left({}^{min(\lambda_{P_i-T})}\!/_{max(\lambda_{P_i-NOb})} \right) - W_1 * \left({}^{\lambda_{P_i-T}}\!/_{\lambda_{P_i-NOb}} \right) \right| \qquad (9)$$

The selection of X_{gbest} will be continued for several cycles until the robot is away from the obstacle or it reaches to its destination. The algorithm for PSO based mobile robot navigation is as follows:

Step 1: Initialize robot initial and final positions.
Step 2: Robot moves until any obstacles or its target position in front it.
Step 3: If robot is senses obstacles, activate PSO.
Step3: Initialise positions and velocities for each individual.
Step 4: Obtain position best of each particle and global best of the swarm from Eq. (7) & (9).
Step 5: Find out new positions and velocities of each particle by using Eq. (1) & (2)
Step 6: Repeat steps 3, 4 and 5 until the robot is away from the sensed obstacles.
Step 7: Repeat step 2 until robot reaches its destination.

III. Numerical Results

PSO is giving better results from the previous work [16-17], at their controlling parameters, $C_1 \& C_2 > 1$ and $rand1 = rand2 = 1$. So for easy calculation, experiments have been performed at $C_1 = C_2 = 2$ and $rand1 = rand2 = 1$.Once the robot senses any obstacle in front it, it will form a swarm of 80 particles randomly within its sensing range as shown in Fig. 4.

B. Results for first Fitness function

From Eq.(9), it can be noticed that the particle considering as X_{gbest} would have the minimum fitness value. The controlling parameter W_1 in Eq. (9) indicates the closeness of the particle to the robot target and distance away from the nearest obstacle. So X_{gbest} can be obtained by minimizing the fitness function as shown in Eq.(9). Low value of W_1 indicates the particle is far away from the nearest obstacle and high value of W_1 indicates the particle is close to the robot's target. So it is required to adjust the controlling parameter W_1 of fitness function in order to obtain optimal fitness value.

A large amount of experiments as tabulated in Table I have been performed in order to tune the parameter W_1. Statistical results showed that the values of W_1 greater than '1.6' are not obtaining collision free paths as shown in Fig.2. From Fig.3, it can be noticed that the robot is facing trap situation for small values of $W_1 \leq 0.3$.

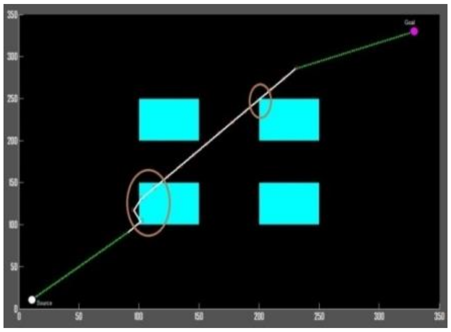

Fig.2. Robot motion for W_1= 2

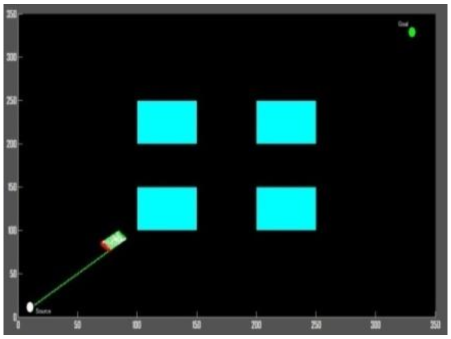

Fig.3. Robot motion for W_1= 0.1

Later experiments have been conducted when $W_1$is varying from 0.3 to 1.6. Results showed that the mobile robot is obtaining optimal trajectories when it sensing obstacles in front of it, at the value W_1=1.3. Robot path generated in green colour indicates the robot motion when it is not facing any obstacles; yellow coloured points around the mobile robot symbolize the random particle distribution within its sensing range and small red coloured circles correspond to the global best positions obtained by calculating fitness value of each individual.

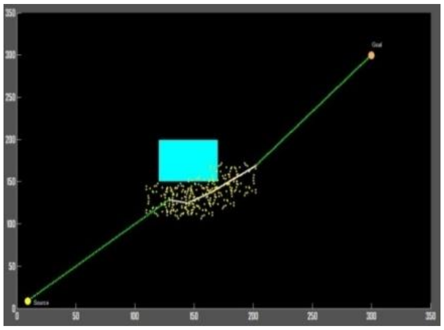

Fig.4. Robot collision free path for W_1= 1.3

TABLE I EXPERIMENTAL RESULTS FOR W_1

W_1	Robot travelled distance (cm)	Collision free path(Yes/No)
0.1	560.4	No (Indefinite motion)
0.2	555.2	No (Indefinite motion)
0.3	552.4	No (Indefinite motion)
0.4	548.8	No
0.5	548.6	No
0.6	546.4	No
0.7	545.0	No
0.8	542.8	No
0.9	542.2	No
1.0	542.2	No
1.1	541.6	No
1.2	540.8	No
1.3	**538.2 (Min)**	**No**
1.4	538.6	No
1.5	538.6	No
1.6	535.2	Yes
1.7	533.8	Yes
1.8	530.4	Yes
1.9	528.6	Yes
2	525.2	Yes

C. Results for second Fitness function

A large amount of experiments have been performed in order to adjust the parameters $K_1$and K_2. Statistical results showed that the values of K_1 greater than one and the values of K_2 less than 100 are not obtaining collision free paths.Later experiments have been conducted when $K_1$is varying from 0.1 to 0.9 and $K_2$is varying from 100 to 1000. Results showed that the mobile robot is obtaining optimal trajectories when it sensing obstacles in front of it, at the values of K_1=0.5 and K_2= 900.

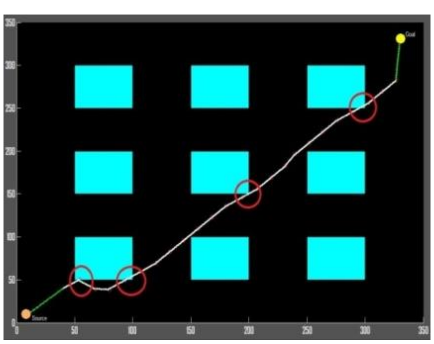

Fig.5. Path for K_1= 0.1 &K_2= 100

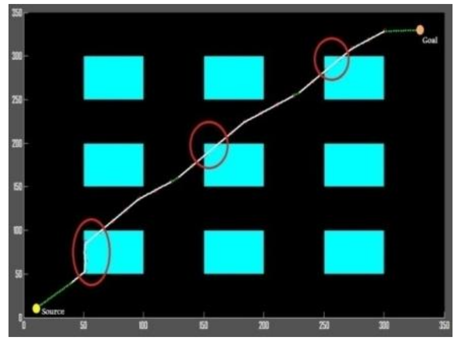

Fig.6. Path for K_1= 2&K_2= 1000

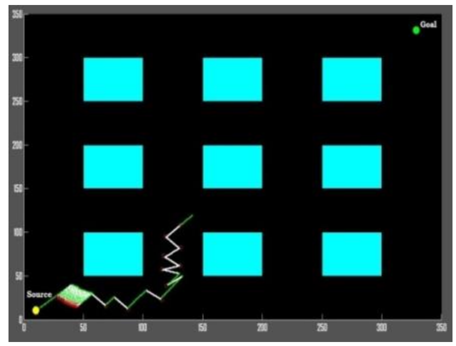

Fig.7. Path for K_1= 0.1 &K_2 = 1000

TABLE II EXPERIMENTAL RESULTS FOR K_1&K_2

K_1	K_2	Robot travelled distance (cm)	Collision free path(Yes/No)
0.1	150	545.2	Yes
	300	625.2	Yes
	450	613.2	Yes
	600	553.2	Yes
	750	553.2	Yes
	900	553.2	Yes
0.3	150	466.4	No
	300	488.4	No
	450	545.2	Yes
	600	580	Yes
	750	661.2	Yes
	900	625.2	Yes
0.5	150	460.8	No
	300	477.6	No
	450	481.6	No
	600	497.6	No
	750	545.2	Yes
	900	**538**	**Yes (Min.)**
0.7	150	460	No
	300	466.4	No
	450	481.6	No
	600	488	No
	750	484.4	No
	900	499.6	No
0.9	150	457.2	No
	300	460.8	No
	450	466.4	No
	600	481.6	No
	750	488	No
	900	488.4	No

D. Comparison of 1st fitness based motion planner and 2nd fitness based motion planner

From the above analysis results from Table I & II, motion planner with first fitness function is generating most favourable paths at W_1 = 1.3 and motion planner with second fitness function is generating most favourable paths at K_1 = 0.5 and K_2 = 900.

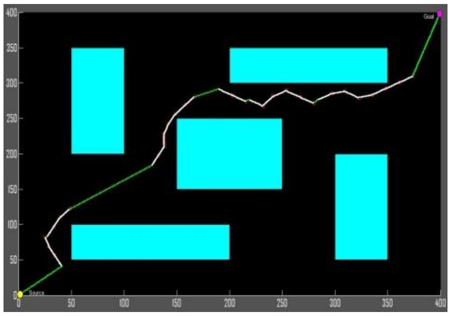

Fig.8(a) path by second motion planner

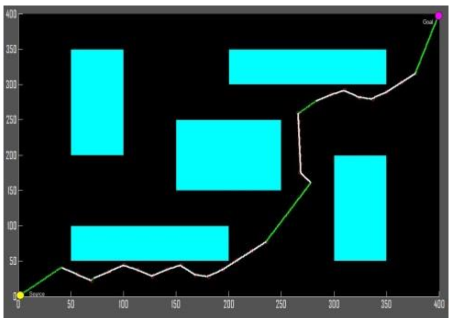

Fig. 8(b) path by first motion planner

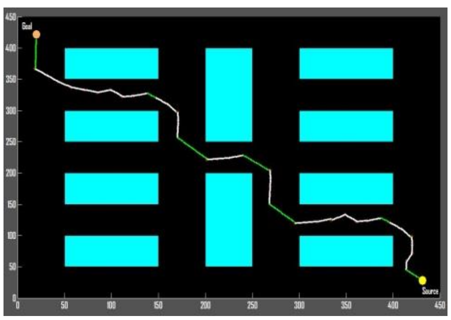

Fig. 9(a) path by second motion planner

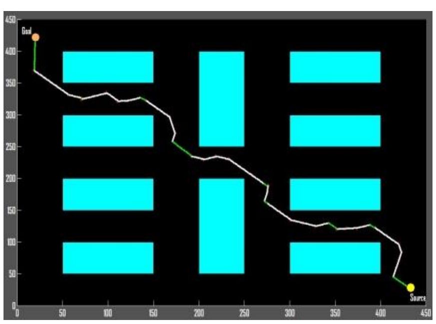

Fig. 9(b) path by first motion planner

TABLE III PATH ANALYSIS RESULTS FOR SCENARIOS, FIGS. 8 & 9

Scenario	Source	Goal	PSO iterations	Normal iterations	Distance travelled (cm)
8(a) [first fitness]	(0,0)	(400,400)	341	173	***755.2***
8(b) [second fitness]	(0,0)	(400,400)	377	162	776.4
9(a)[first fitness]	(430,30)	(20,420)	452	109	***760.4***
9(b)[second fitness]	(430,30)	(20,420)	400	150	780

Path analysis results from Table III, it is concluded that path generated by first motion planner is obtaining shortest path as compared to first motion planner.

IV. CONCLUSION

In the present research work, two new efficient PSO based algorithms have been addressed for solving mobile robot navigation problem. The main objective of the proposed methodologies was to generate shortest and safest paths within unknown robotic environments. With PSO based motion planner, robot generates collision free trajectories by finding global best position within its sensing range. Thereby, the robot reaches to global best position in sequential. This process will continue until the robot reaches to its destination. From the path analysis results, it is concluded that the performance of the first (PSO) fitness based motion planner is giving better results than second fitness (PSO) based motion planner. As future work, for the current PSO based motion planners it is necessary to apply an adaptive mechanism in order to obtain better results.

REFERENCES

[1] Park, M.G., and Lee, M.C., Experimental Evaluation of Robot Path Planning by Artificial Potential Field Approach with Simulated Annealing, Proceedings of 41st SICE Annual Conference, August 2002, DOI: 10.1109/SICE.2002.1195739, pp. 2190-2195

[2] Oommen, B.J., S. Sitharama, I., Nageswara, S.V.R and Kashyap, R.L., Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case, IEEE Journal of Robotics and Automation 1997, 3(6), pp. 672-681.

[3] Qin, Y.Q., Sun, D.B., Li, M., and Cen, Y.G., Path planning for mobile robot using the Particle swarm optimization with Mutation operator, Proceedings of Third International Conference on Machine Laming and Cybernetics, August 2004, DOI: 10.1109/ICMLC.2004.1382219, pp. 2473-2478.

[4] Glavaški, D., Volf M., and Bonković, M., Robot motion planning using exact cell decomposition and potential field methods, Proceedings of 9th WSEAS international conference on simulation, modelling and optimization, Budapest, Hungary, 2009, pp.126-131.

[5] Kennedy, J., and Eberhart, R., Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, Nov/Dec 1995, vol.4, DOI: 10.1109/ICNN.1995.488968, pp.1942-1948

[6] Hassan, R., Cohanim, B., Weck, O.D., and Venter G., A comparison of particle swarm optimization and the genetic algorithm. Proceedings of 1st AIAA multidisciplinary design optimization specialist conference. DOI. AIAA-2005-1897, Austin, TX, 2005.

[7] Alireza1 ALFI, PSO with Adaptive Mutation and Inertia Weight and Its Application in Parameter Estimation of Dynamic Systems, Acta Automatica Sinica, 2011, 37(5), pp. 541-549.

[8] Zhang, Y., Xuan, J., Benildo, G.D.L.R, Clarke, R., and Habtom, W.R., Reverse engineering module networks by PSO-RNN hybrid modelling. Proceedings of international Conference on Bioinformatics & Computational Biology. Las Vegas, USA, 2008, pp. 1-18

[9] Wu, Q., Car assembly line fault diagnosis based on robust wavelet SVC and PSO, Expert Systems with Applications 2010, 37(7), pp. 5423-5429.

[10] Sha, D.Y., and Lin, H.H., A multi-objective PSO for job-shop scheduling problems, Expert Systems with Applications 2010, 37(2), pp. 1065-1070.

[11] Yiqing, L., Xigang, Y., and Yongjian L., An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints, Computers & Chemical Engineering 2007, 31(3), pp. 153-162.

[12] Venayagamoorthy, G.K., and Doctor, S., Navigation of mobile sensors using PSO and embedded PSO in a fuzzy logic controller, Proceedings of 39th IEEE Conference IAS Annual Meeting Industry Applications, vol.2, DOI: 10.1109/IAS.2004.1348565, October 2004, pp. 1200 – 1206.

[13] Zhang, Q., and Li, S., A Global Path Planning Approach Based on Particle Swarm Optimization for a Mobile Robot, Proceedings of 7th WSEAS International Conference on Robotics, Control & Manufacturing Technology, Hangzhou, China, 2007, ISBN: 111-222-5555-66-7, pp. 263-267.

[14] Masehian, E., and Sedighizadeh, D., Multi-Objective PSO- and NPSO-based Algorithms for Robot Path Planning, Advances in Electrical and Computer Engineering 2010, 10(4), 69-76

[15] Derr, K., and Manic, M., Multi-Robot, Multi-Robot, Multi-Target Particle Swarm Optimization Search in Noisy Wireless Environments, In Proceedings of Human System Interactions'09, Catania, Italy, May 2009, DOI: 10.1109/HSI.2009.5090958, pp. 81-86

[16] Doctor, S., Venayagamoorthy, G.K., and Gudise, V.G., Optimal PSO for Collective Robotic Search Applications, Proceedings of Congress on Evolutionary Computation, Vol. 2, June 2004, DOI: 10.1109/CEC.2004.1331059, pp.1390-1395

[17] Smith, L., Venayagamoorthy, G.K., and Holloway P.G., Obstacle Avoidance in Collective Robotic Search Using Particle Swarm Optimization, Proceedings of IEEE Swarm Intelligence Symposium, Indianapolis, USA, May 2006.

[18] Lu, L., and Gong, D., Robot Path Planning in Unknown Environments Using Particle Swarm Optimization, Proceedings of Fourth International Conference on Natural Computation, vol.4, Jinan, China, DOI: 10.1109/ICNC.2008.923, October.2008, pp. 422-426.

[19] Yong, B.Q, Ming, L.S, Yan S.W. and Jin, A.M. A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments, Proceedings of International Conference on Artificial Intelligence and Computational Intelligence, November.2009, pp.257 - 261.

[20] Mester, G. and Rodić, A. Sensor-Based Intelligent Mobile Robot Navigation in Unknown Environments, International Journal of Electrical and Computer Engineering Systems, Vol. 1, No. 2, 2010, pp. 55-62.