World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:8, No:10, 2014

Comparison of GSA, SA and PSO Based Intelligent Controllers for Path Planning of Mobile Robot in Unknown Environment

P. K. Panigrahi, Saradindu Ghosh, Dayal R. Parhi

Abstract**—**Now-a-days autonomous mobile robots have found applications in diverse fields. An autonomous robot system must be able to behave in an intelligent manner to deal with complex and changing environment. This work proposes the performance of path planning and navigation of autonomous mobile robot using Gravitational Search Algorithm (GSA), Simulated Annealing (SA) and Particle Swarm optimization (PSO) based intelligent controllers in an unstructured environment. The approach not only finds a valid collision free path but also optimal one. The main aim of the work is to minimize the length of the path and duration of travel from a starting point to a target while moving in an unknown environment with obstacles without collision. Finally, a comparison is made between the three controllers, it is found that the path length and time duration made by the robot using GSA is better than SA and PSO based controllers for the same work.

Keywords**—**Autonomous Mobile Robot, Gravitational Search Algorithm, Particle Swarm Optimization, Simulated Annealing Algorithm.

I. Introduction

AUTONOMOUS mobile robot finds applications in various fields like, rescue search in different space environments, helping handicapped people, hazardous areas of plant etc. The robot should navigate through a collision free path and optimization criteria's can be applied to it for shortest path. Classical techniques for path planning like, roadmap, cell decomposition, potential fields and mathematical programming have various disadvantages, for example, high time complexity in large problem space and trapping in local optimum. These drawbacks can be overcomed by different heuristic and meta-heuristic algorithms like, Simulated Annealing, Genetic Algorithm and Ant Colony Optimization, in path planning problem. Swarm intelligence can be used to solve optimization and cooperative problems of various fields like, in computer networks, mobile robotics and cooperative and decentralized control. Swarm intelligence (artificial intelligence), in nature, may be composed of three main principles: evaluation comparing and imitation. Evaluation is based on analysis of what is positive or negative in nature, attractive or repulsive. Comparison is the basis of living beings to evaluate themselves taking other beings as a standard. Imitation is an effective form of learning. This paper explores the use of different optimization methods namely, GSA, SA and PSO for navigation of mobile robot in unknown complex environment.

P.K.Panigrahi is with the Department of Electrical Engineering, Padmanava College of Engineering, Rourkela, Odisha, India (email: pratapkumarpanigrahi@gmail.com

Saradindu Ghoshis with the Department of Electrical Engineering, National Institute of Technology, Durgapur, West Bengal, India (email: sghosh.ee@lgmail.com).

D.R.Padhi is with the Department of Mechanical Engineering, National Institute of Technology, Rourkela, India (e-mail:dayalparhi@yahoo.com).

II. Related Work

Navigation of mobile robot can be resolved by various methods; starting from conventional techniques like, road map to artificial intelligence techniques, such as, artificial neural network, fuzzy logic and potential field methods. However, these methods have some drawbacks in the process of convergence properties, long execution time etc. in complex environments. In order to overcome these disadvantages researchers have been used evolutionary and heuristic algorithm such as, Genetic Algorithm (GA), Differential Evolution (DE), Particle swarm optimization (PSO) and Simulated Annealing (SA). This section presents review of various optimization techniques based on swarm intelligence. Lin et al. [1] used swarm intelligence and chaotic dynamics to find an optimal path for autonomous mobile robot. They have proposed an approach for path planning of robot, combining artificial bee colony optimization with chaos. The work of Panigrahi et al. [2] is based on a comparison of path planning of mobile robot using Mamdani and Sugeno based fuzzy controller. The efficiency of both the algorithms is obtained using static obstacles in the environment. Panigrahi et al. [3] have implemented a Radial Basis Function (RBF) based intelligent controller for path planning of mobile robot and hence to avoid static obstacles in unknown environment. The performance of the algorithm is verified using MATLAB simulations. Castillo et al. [4] have described the use of Genetic Algorithm (GA) and Multi Objective Optimization Algorithm (MOOA) for path planning of mobile robot. Their work shows that both conventional GA and MOOA are effective tools for solving path-planning problem. Venayagamoorthy et al. [5] have presented two new strategies for navigation of robots inspired by natural swarms like, flock of birds. In the first method they used fuzzy logic embedded with PSO while in the second method swarm intelligence in fuzzy logic controllers was utilized. They have concluded that both the methods are reliable on convergence and are more efficient than non-swarm oriented techniques. Yarmohamdi et al. [6] have used PSO as optimization algorithm for finding optimal path for mobile robot in dynamic environment. They

proposed an approach which works as follows: Robot moves on direct path from start to goal until collision with obstacle occur. When the robot encounters the obstacle, it rotates either left or right. Moaner et al. [7] have proposed Random Particle Optimization Algorithm (RPOA) for navigation of mobile robot in dynamic environment. The algorithm is inspired by bacteria foraging technique. They have used different scenarios like, fixed obstacles and target, random moving obstacle and fixed target, random moving obstacle and target. The work shows that RPOA gives better optimal solution than artificial potential field methods. Ahmadzadeh et al. [8] used PSO for finding optimal path for mobile robot in an unknown environment, where obstacles can be fixed or movable. Firstly, the navigation problem is treated as optimization problem and then PSO is used to solve it. The work of Parhi et al. [9] was based on swarm intelligence principles for co-operative behavior of multiple autonomous mobile robots. They have used ant colony optimization for finding optimum path for mobile robot. They concluded that larger size group of robots is better than a smaller size group of robots. Sahoo et al. [10] presents a navigation method for mobile robot using Honey Bee Mating Optimization (HBMO) algorithm. It is a swarm based approach used to find optimized path for multi-robot environment. They concluded that this method outperforms PSO and differential evolutionary (DE) algorithm. Shiltagh et al. [11] investigated the application of Modified Particle Swarm Optimization (MPSO) for finding shortest feasible path of a mobile robot. The proposed algorithm uses the map of the environment and expressed in grid model then creates an optimal and near optimal collision free path. It is verified that the evolutionary or heuristic algorithms are more efficient than classical algorithms for the same problem. The work of Sierakwski et al. [12] was based on forage theory of a bacteria colony. It is an optimization algorithm and used to find optimum path for mobile robot to reach a goal from start position. This approach mimics the behavior of bacteria like foraging and avoiding noxious substances. Masehian et al. [13] presented two PSO-based path planning algorithm. The first algorithm is a combination of PSO, used as global planner and probabilistic roadmap (PRM) used to perform local planning task. The second algorithm is a hybrid of the new or negative PSO (NPSO) and PRM. Both the algorithms are compared with classic PRM method to give the conclusion that new algorithm is faster than the classic method in terms of path length. Wei [14] introduced simulated annealing (SA) combined with Powell algorithm for better convergence and short computation time in path planning and global optimal problem. Curkovic and Jerbic [15] have used Honey Bee Optimization) (HBO) algorithm to solve benchmark Diophantine problem and problem of path planning. In comparison with GA, HBO performed better for Diophantine problem. They have used fuzzy fitness function for evaluation of trajectories of mobile robot. Parvez et al. [16] have used behavior-based robot controller to test in maze-like indoor environment. Parhi et al. [17] have used Wavelet Neural Network (WNN) based intelligent controller for path planning of mobile robot to avoid static obstacles during navigation from one starting place to another target place. The neural network uses different mother wavelets as activation function and is trained using a set of training pairs. To verify the algorithm MATLAB simulations are conducted in different static obstacles environment. Yazadi et al. [18] have proposed a new heuristic algorithm to find multiple solutions in multimodal problems. In their work a new technique, namely Niche GSA (NGSA) is introduced for multimodal optimization. The results are compared with niching algorithms. Ghose [19] has described the difference between GA and SA algorithm for different optimization problems. In this paper he has presented the advantages of SA over GA. Bayar et al. [20] have proposed a reinforcement learning based RBF neural network for solution of orientation of mobile robot during navigation from one place to another place with static obstacles in the environment. The effectiveness the technique is tested using MATLAB graphical results. Soman et al. [21] have suggested a new algorithm based on RBF network hybridized with Particle Swarm Optimization technique for different applications. The effectiveness of the same is compared with conventional type back propagation algorithm.

III. Configuration of Mobile Robot

Fig. 1 (a) is the typical example of a non-holonomic four wheeled differential driven mobile robot. It consists of two rear and two front wheels which are controlled by four D.C geared motors. The rear wheels and front wheels are mounted on two common shafts / axis of robot chassis. Since it is a differential driven mobile robot, therefore, the two left and two right side driving wheels can be independently controlled by D.C. motors.

ICC is the instantaneous center of curvature, ϕ is the steering angle, d is the front to rear axle distance, θ is the robot heading, R is the radius of curvature, w is the angular velocity, v_L and v_r are the ground linear velocity of left and right rear wheels, r is the radius of each wheel.

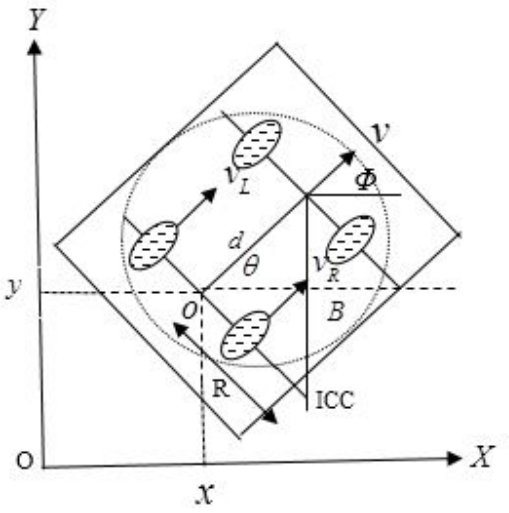

Fig. 1 (a) Four wheeled mobile robot configuration

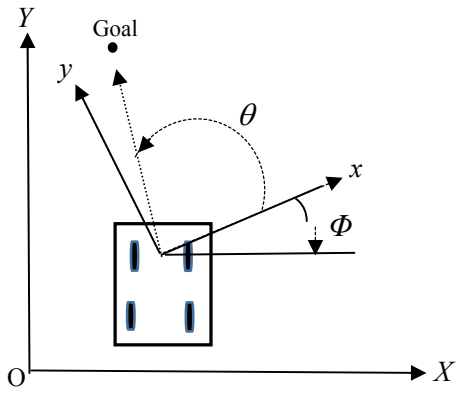

Fig. 1 (b) Robot pose in global coordinates

A. Assumptions

- The instantaneous center of curvature must lie coincident with the axis of wheel in contact with the ground; circular motion is exhibited by each wheel around that point.
- No slippage: Traveled distance is assumed to be equal to $\theta \times r$
- Lateral slippage is neglected.

B. Analysis

The robot has 3 degrees of freedom and (x, y) (position), and θ is the heading or orientation; the triplet (x, y, θ) is the pose of the robot in the plane.

$$R = d\tan\left(\frac{\pi}{2}-\alpha\right)$$

$$w = \frac{V}{B} = \frac{V}{(d^2+R^2)^{1/2}} = \frac{V}{\left(d^2+\left[d\tan\left(\frac{\pi}{2}-\alpha\right)\right]^2\right)^{1/2}} \quad (1)$$

$$\theta = \int_0^t w + \theta_0 = \int_0^t \frac{V}{\left(d^2+\left[d\tan\left(\frac{\pi}{2}-\alpha\right)\right]^2\right)^{1/2}} d\tau + \theta_0$$

$$V_x = V\cos\theta$$

$$V_y = V\sin\theta \quad (2)$$

$$x = \int_0^t V_x = \int_0^t V\cos\theta\, d\tau + x_0$$

$$y = \int_0^t V_y = \int_0^t V\sin\theta\, d\tau + y_0$$

Each steering operation occurs when the robot reaches an obstacle.

IV. Objectives

The main objectives of the research are as follows:

a. To apply GSA, SA and PSO algorithms for robot path planning in global static environment (Different complexities of obstacles).
b. To evaluate performances of three algorithms in terms of path (distance) and computation time.
c. To identify the robust path planning algorithms which can produce optimal path that satisfies optimization criteria and to reduce path cost with less computation time.

To solve this problem the choice of fitness function / objective function is a fundamental issue for the solution of intelligent controller of a mobile robot. There are several types of objective functions have been proposed in the literature. In this proposed work, we have obtained the objective function by developing around 300 heuristic data sets from different location of obstacles in the environment to obtain collision free optimal path while moving from a starting point to a target with respect to distance between the goal and robot and distance between nearest obstacles and robot. The fitness function used in the work is:

$$fit = C_1 + C_2 \times LOD + C_3 \times FOD + C_4 \times ROD \quad (3)$$

Where $C_1 = 42.439, C_2 = 2.753, C_3 = 2.581$and $C_4 = -4.676$

The fitness function has 4 constants, i.e. C_1, C_2, C_3 andC_4 which are obtained using MATLAB program out of which three constants C_2, C_3 and C_4 are coefficient of three distances (LOD, FOD and ROD) and fourth constant C_1 is independent of distance variable.

V. Gravitational Search Algorithm (GSA)

Gravitational Search Algorithm [13]-[15] is an optimization algorithm derived by Newton's laws of gravity and mass interaction. Here, agents are considered as objects having different masses. The movement of all agents globally towards the agent with heavier masses is due to the gravitational attraction force acting between them. The heavy masses are taken as good solutions of the problem. Each agent has four specifications: inertia mass, its position, active gravitational mass and passive gravitational mass. The position of agent in specified dimensions corresponds to a solution of the problem. The inertia mass reflect the agent's resistance to change its state of motion on application of force. Agents with large inertia mass move slowly whereas agents with small inertia mass move rapidly. Active gravitational mass measures the strength of gravitational field due to a particular object. Passive gravitational mass measures the strength of an agent's interaction with gravitational field. In GSA, position of the agent is considered as solution of the problem. Gravitational and inertia masses are determined by fitness function of the problem. The positions of the agents are updated at each iteration and best value for the corresponding agent is recorded. The algorithm terminates when maximum iteration value is reached and the final iteration value is taken as global solution of that problem.

GSA algorithm has following steps:

A. Initialization of the Agents

Let us consider a system with N agents whose positions are randomly initialized within the search interval.

$$X_j = \left(x_j^{\,1}, x_j^{\,2}, x_j^{\,3} \ldots . x_j^{\,d}\right) \text{ for } j = 1, 2, \ldots N$$

x_j^d is the position of the j^{th} in d^{th} dimension.

B. Fitness Evaluation of All Agents

Calculate fitness of all best and worst agents at each iteration

$$\text{best}(t) = \min fit_i(t),\ i \in \{1,2,3\ldots N\}$$
$$\text{worst(t)} = \max fit_i(t),\ i \in \{1,2,3\ldots N\}$$

where $fit_i(t)$ is the fitness of i^{th} agent at the iteration t, best (t) and worst (t) are called best and worst fitness of all agents at iteration t.

C. Computation of Gravitational Constant G (t)

Gravitational constant G at iteration t is calculated by:

$$G(t) = G_0 e^{(-\beta \frac{t}{T})} \quad (4)$$

where G_0 is the initial value of gravitational constant, β is a constant, t is current iteration, T is total number of iteration.

D. Calculation of the Mass of the Agents

Following equations are used to calculate gravitational and inertia masses for each agent at iteration t.

$$M_{aj} = M_{pj} = M_{ij} \quad j = 1,2,3\ldots N$$
$$m_j(t) = \frac{fit_j(t) - worst_j(t)}{fit_j(t) - worst(t)} \quad (5)$$
$$M_j(t) = \frac{m_j(t)}{\sum_{i=1}^{N} m_i(t)} \quad (6)$$

M_{aj} is the j^{th} agent's active gravitational mass, M_{pj} is the j^{th} agent's passive gravitational mass, M_{ij} is j^{th} agent's inertial mass of , $M_j(t)$ is j^{th} agent's mass at iteration t.

E. Calculation of Total Force

The total force acting on the j^{th} agent ($F_j(t)$) is calculated by:

$$F_j^d(t) = \sum_{i \in k_{best}, j \neq i} \text{rand}_i F_{ji}^d(t) \quad (7)$$

where $rand_i$ is called random number between [0, 1], K_{best} is the best fitness value of first K agents, $F_{ji}^d(t)$ is the force acting on agent 'j' from agent 'i' at d^{th} dimension and t^{th} iteration. It is calculated as

$$F_{ji}^d(t) = G(t) \frac{M_{pj} \times M_{ai}}{R_{ji}(t) + \in} (x_i^d(t) - x_j^d(t)) \quad (8)$$

where $R_{ji}(t)$ is called Euclidian distance between the agents 'j' and 'i' at iteration t, $G(t)$ is gravitational constant at iteration t, $\in$ is a small constant.

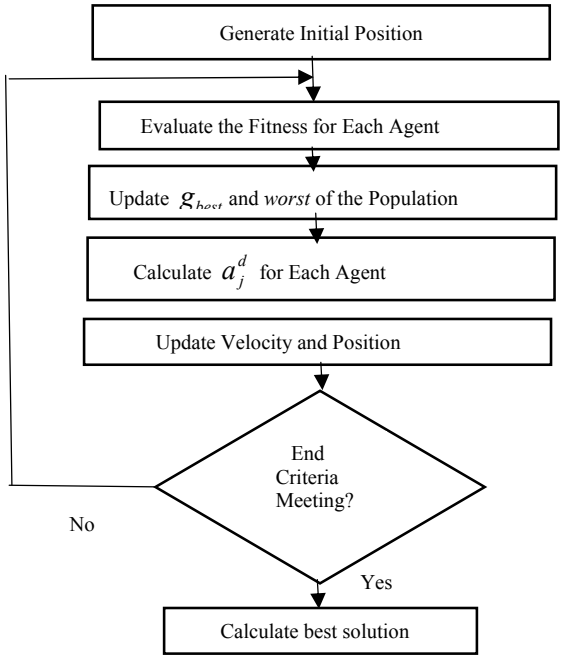

Fig. 2 Flow Chart of Gravitational Search Algorithm

F. Update Velocity and Positions of the Agents

The agent's velocity and position at next iteration (t+1) is calculated as follows:

$$V_j^d(t+1) = rand_j \times V_j^d(t) + a_j^d(t)$$
$$x_j^d(t+1) = x_j^d(t) + V_j^d(t+1) \quad (9)$$
$$a_j^d(t) = \frac{F_{ji}(t)}{M_j(t)}$$

i.e. $a_j^d(t)$ is the acceleration of j^{th}agent at iteration $t$in d^{th} dimension, $rand_j$ is the random number between interval [0,1].

Steps B-F are repeated until the maximum limit of iterations reach. Best fitness computed at final iteration is returned as global fitness of the problem and the positions of the corresponding agentat specified dimensions is the global solution of that problem. During simulation the value of different parameters are β =20, G_0 (initial value of gravitational constant) =100, N (number of agents) = 50, Maximum iteration = 20.

VI. Simulated Annealing Algorithm (SAA)

Simulated annealing is an optimization method, which resembles the annealing process of metals. When a hot metal's temperature goes down in slower rate the crystal formed is well structured with minimum possible energy. In this work we have used simulated annealing process to find an optimized path for autonomous mobile robot in a static environment. Simulated annealing mimics are the annealing process of metal. At high temperature, the atoms of metals move freely with respect to each other. But, as the temperature goes down the atoms start to get ordered and forms crystal

depends on cooling rate i.e., if the process is fast, crystals may not be formed and it will reach to a state having higher energy state. So, the temperature is reduced at slow rate which is known as annealing process in metallurgy. The algorithm starts with an initial point with a high temperature T. A second point is selected randomly near to the initial point. Then the difference in the function values (ΔE) at these two points are calculated. If the second point has function value smaller, the point is accepted; otherwise the point is accepted by considering the probability of exp (− Δ E/T). The above steps constitute one iteration of the simulated annealing process. In the next step, a new point is selected randomly in the neighborhood of the current point and then the algorithm is used to accept or reject the point. To obtain the thermal equilibrium at every temperature, a number of points (n) are tested with the algorithm at a particular temperature, before reducing the temperature. The above algorithm process is completed when a sufficiently small temperature is determined. The efficiency of the final solution and the convergence speed of algorithms are based on simulated annealing process and depend on the selection of k and the initial temperature also with the design of the cooling process schedule . The temperature is taken initially high value so that the probability of accepting uphill moves are close to 1, and then it is slowly decreased towards frozen according to a cooling process schedule.

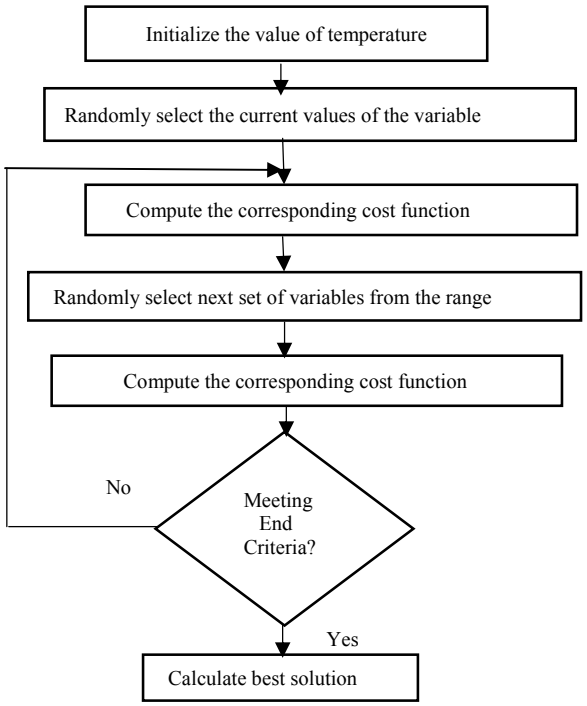

Fig. 3 Flow Chart of Simulated Annealing Algorithm

Different steps involved in simulated annealing process are explained below:

Step 1. A high temperature T is set as initial temperature and a cooling schedule is specified.

Step 2. Randomly choose an initial point X_i and find E (X_i) where E (.) is the fitness function.

Step 3. Generate a new point X_{i+1} and calculate E (X_{i+1}).

Step 4. If E (X_{i+1}) < E (X_i) accepts new solution i.e. E (X_{i+1}).

Step 5. If E (X_{i+1}) > E (X_i) then accepts new solution with a probability ($e^{-\Delta/T}$) where

$$\Delta = E(Xi + 1) - E(Xi) \quad (10)$$

Step 6. Reduce temperature according to cooling schedule and repeat step 3 to 5 until freezing temperature is not reached.

While applying SA in mobile robot path planning, simulated annealing algorithm starts with current solution at initial temperature. At each temperature the algorithm iterates i-times. Proper cooling scheme is important for the performance of SA. The proper annealing process is related with the initial temperature, iteration of each temperature, temperature decrement coefficient and stopping criteria. In this proposed work maximum temperature is taken as 100 degree, for each temperature, 20 test points / iterations are considered to reach thermal equilibrium.

VII. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) algorithm [6, 8] is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. It optimizes a problem by taking a population of candidate solutions. It moves these particles around in the work-space by considering two mathematical formulae with respect to the position and velocity of the particle. The movement of the particle is affected by its local best known position. It is also guided towards the known best positions in the work-space by other particles. Therefore, it is obvious that the swarm is expected to move towards the best solutions.

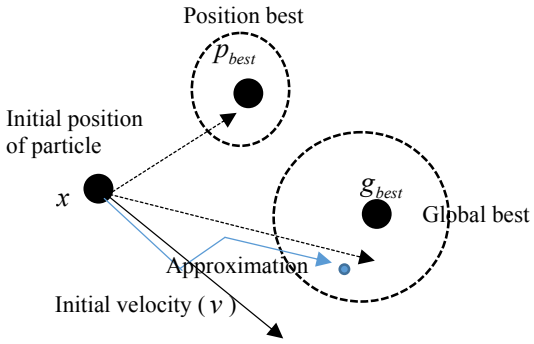

Fig. 4 Basic view of PSO

Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. Each particle in the workspace is robot which gives information of its coordinates in the problem space and is associated with the best solution called fitness. The fitness value is called p_{best} . Another "best" value that is obtained by any particle in the neighbor's location of the particle is called

l_{best}. When a particle takes all the population as its topological neighbors, the best value is called global best and is called g_{best}. Consider Swarm of particles which are flying in the given work space and is searching for optimum

Position vector..... $X_{pi}(t)$

Velocity vector..... $V_{pi}(t)$

During the process of movement, each particle occupy its individual p-best and social knowledge g-best i.e. p_{best} and g_{best} of its best neighbor.

The velocity update formula i.e.

$$V_{pi}(t+1) = \mu . V_{pi}(t) + K_i \times rand \times p_{best}(t) - X_{pi}(t) + K_2 \times rand \times g_{best(t)} - X_{pi}(t) \quad (11)$$

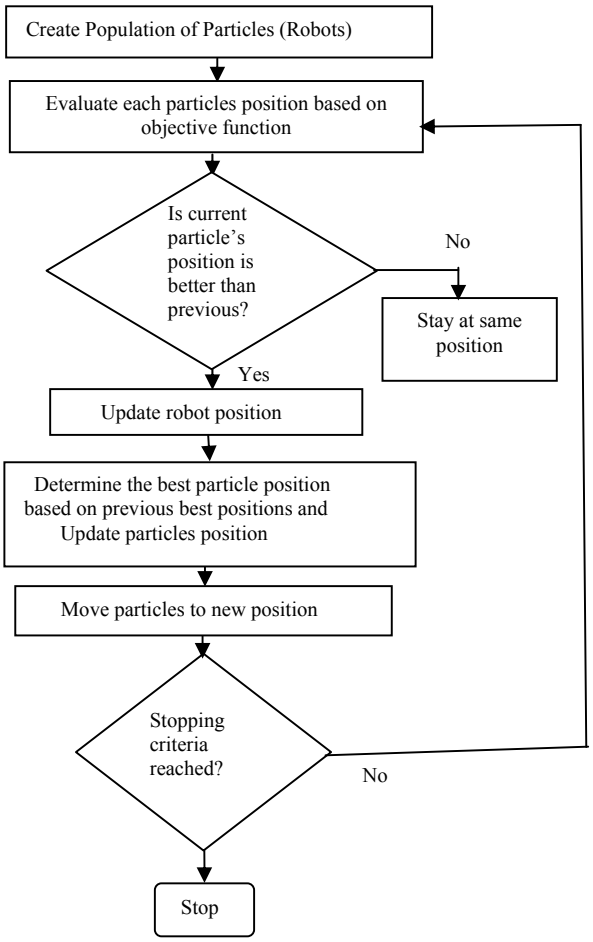

Fig. 5 Flow Chart of Particle Swarm Optimization Algorithm

The μ is the inertia weight that controls the exploration and exploitation of the search space. K_1 and K_2 are the cognition and social components constants which changes the velocity of a particle towards the p_{best} and g_{best}, rand is a random number between 0 and 1. K_1 and K_2 values are set to 1.03.

Now, the position update formula is

$$X_{pi}(t+1) = X_{pi}(t) + V_{pi}(t+1) \quad (12)$$

The above process is repeated for each and every particle considered in the computation till the best optimal solution is obtained.

The basic operation of PSO is given by,

Step 1. *First of all initialize the swarm in the work space*

Step 2. *Calculate the individual particle's fitness*

Step 3. *Update the values of* g_{best}, p_{best} *and velocity*

Step 4. *Shift each particle in the work space to a new position*

Step 5. *Go to step 2, and repeat the process until Convergence/ stopping condition is achieved*

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.

The simulation parameters are $C_1 = C_2 = 1.03$, Population size N=33, Number of iterations =50, $\mu = 0.03$.

VIII. Simulation Results and Discussion

The present research work deals with the performance of navigation of mobile robot using GSA, SAA and PSO algorithms in different complex unknown environments. The environment used for path planning of mobile robot is 500×500 square unit. The simulation work is based on two behaviors, such as obstacle avoidance and target seeking in MATLAB environment with Intel Pentium IV, Dual core, 1.8GHZ processor. Figs.6 (a)-(c) are the simulation results of mobile robot in the L- shaped obstacle environment where as Figs.7 (a)-(c) shows the results of path made by the robot in presence of I-shaped and square sized obstacle environment for GSA, SA and PSO respectively. Each scenario the three proposed algorithms are applied, the performance is considered on the basis of smoothness of trajectory, path length and time taken to reach the target. During simulation mode the robot is allowed to navigate from (450, 50) position to the target position (50,450). The obstacle avoidance behavior is activated when obstacle distance from the robot is less than the minimum threshold value which is pre-defined in the algorithm. It can be observed from different simulation results that if there is no obstacle or obstacles away from the robot, then it navigates in a straight path otherwise makes curved path, i.e. either a right or a left turn to avoid collision with obstacles. A comparison in terms of time duration of the path in seconds is depicted in Table I corresponding to the Figs. 6 (a)-(c) and Fig.7 (a)-(c) respectively. From the TableI, it is evident that the navigational path made by the GSA algorithm in Figs.6 (a) and 7 (a) are smooth and time duration is smaller than SAA and PSO. Thus, the GSA algorithm is faster than the two other methods.

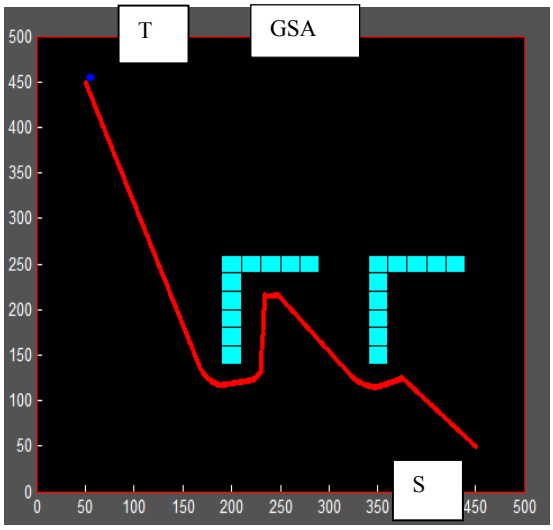

Fig. 6 (a) Robot Start Point (450, 50), Target (50,450)

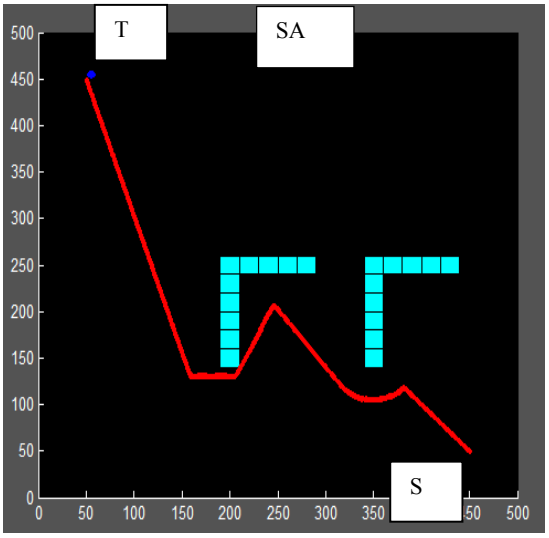

Fig. 6 (b) Robot Start Point (450, 50), Target (50,450)

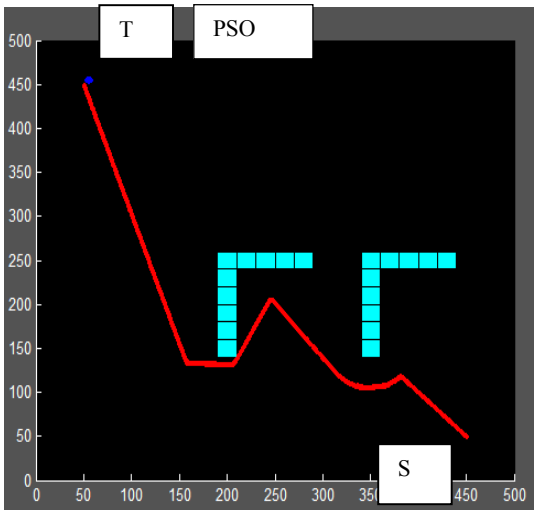

Fig. 6 (c) Robot Start Point (450, 50), Target (50,450)

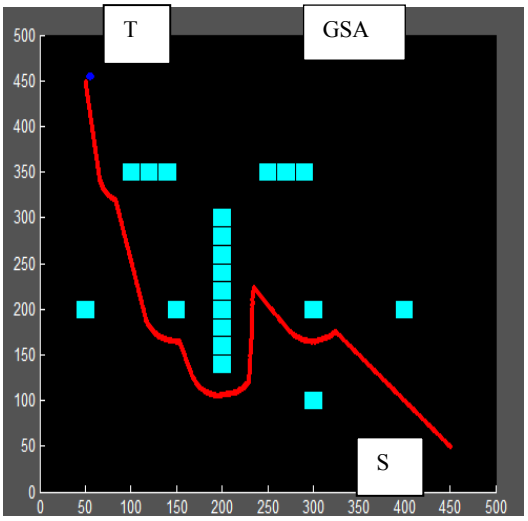

Fig. 7 (a) Robot Start Point (450, 50), Target (50,450)

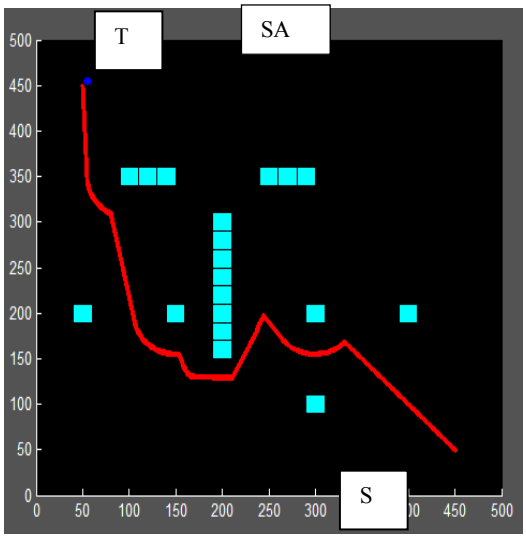

Fig. 7 (b) Robot Start Point (450, 50), Target (50,450)

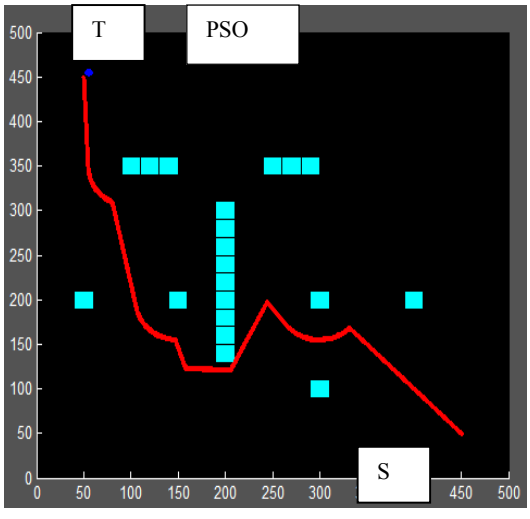

Fig. 7 (c) Robot Start Point (450, 50), Target (50,450)

TABLE I
ROBOT PATH TIME OF DIFFERENT ALGORITHMS

Figure Number	Starting Point	Target Point	Name of the Algorithm	Time duration of path in seconds	Number of Iterations
6(a)	(450,50)	(50,450)	GSA	190	20
6(b)	(450,50)	(50,450)	SA	278	20
6(c)	(450,50)	(50,450)	PSO	299	33
7(a)	(450,50)	(50,450)	GSA	273	20
7(b)	(450,50)	(50,450)	SA	389	20
7(c)	(450,50)	(50,450)	PSO	445	33

IX. EXPERIMENTAL SET UP

In In order to validate the simulation results, experimental analysis has been carried out for a differential drive mobile robot. The robot is equipped with the following while performing the experiments in several real environments.

1. The Arduino Mega 2560 is a μC board based on the ATmega2560. It contain 54 digital I/O pins, 16 analog inputs, 4 UARTs , a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. For example, the pins on the Mega can be used as an input or output, using program mode are pinMode (left1, OUTPUT), digitalWrite(left1, L1) etc.
2. The GPS controlled autonomous mobile robot uses a GPS receiver module to capture the GPS signal and to determine the vehicle's current location. The robot direction of movement is controlled by an Arduino Mega 2560 microcontroller which is interfaced with a LCD, a magnetic compass, sensor and DC motors. The compass and sensor determines the vehicle direction by continuously providing measurement of heading angle. The microcontroller drives the DC motors to move the vehicle to the target coordinates. Obstacle detection and avoidance are achieved by four ultrasonic sensors to measure the distance between the vehicle and the obstacles in the left, right and front directions (LOD, FOD and ROD). The designed GPS autonomous vehicle is able to navigate itself independently from one starting point to a user-prescribed target location using GPS-location data.
3. Arduino GPS shield is GPS module bread out board for Global positioning system receiver with SD interface. It is easy to use for recording the position data into SD card.
4. Arduino 3 axis Magnetic Compass (HMC 5883L) measures the earth's magnetic field in three axes. The 3-axis compass module measures field in thee directions labeled X, Y and Z. It is used for auto navigation. It is interfaced using I2C communication which determines the direction of the robot.
5. Ultrasonic sensor (HC-SR04) detects the distance of the closest obstacle in left, right and front of the sensor (from 0.5mt to 3mt). It works by sending out a burst of ultrasound and listening for the echo when it bounces off of an object. The Arduino board sends a short pulse to trigger the detection, then listens for a pulse on the same pin using pulse command pinMode (ultraSoundSignal, INPUT).The duration of this second pulse is equal to the time taken by the ultrasound to travel to the object and back to the sensor. Using the speed of sound, the time is converted into distance.
6. Position encoder is used for determining position and velocity of the wheel. It works on external interrupts and LM324 is used for comparison.
7. LCD display unit.

A. Robot Activities

- Request GPS co-ordinates from target
- Get target GPS coordinates
- Get self GPS co-ordinates
- Calculate distance between self and target
- Calculate angle/heading
- Correct heading by use of magnetic compass

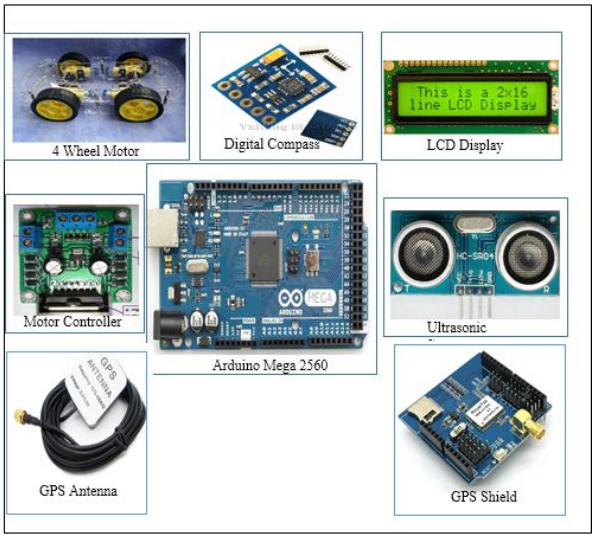

Fig. 8 Mobile Robot System Components

- Move in the heading direction till distance = 0
- If obstacle is detected (i.e. safe zone <30cm), avoid obstacle till it is not detected (i.e. Safe zone > 30cm)
- Again calculate heading/angle and keep moving till distance is 0.

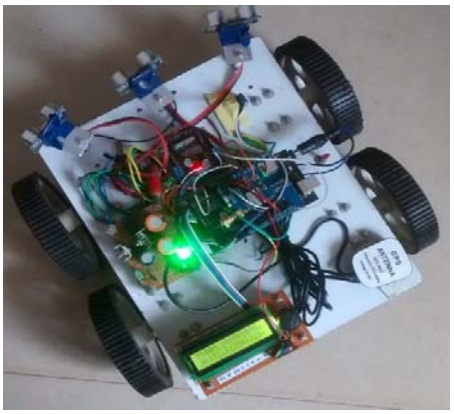

Fig. 9 (a) Mobile Robot with GPS Antenna, Magnetic compass and LCD Unit

Fig. 9 (b) Testing set up of Arduino Mega 2560

B. Target Activities

- Wait till receiving request from robot
- Upon receiving request
- Send target co-ordinates to robot
- Keep waiting till next request

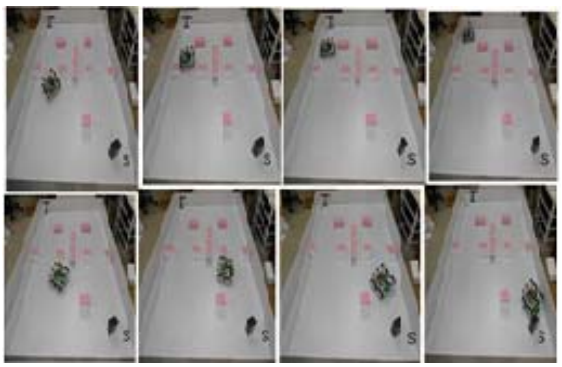

Fig. 10 (a) Test case I for the environment in Fig. 6 (a)

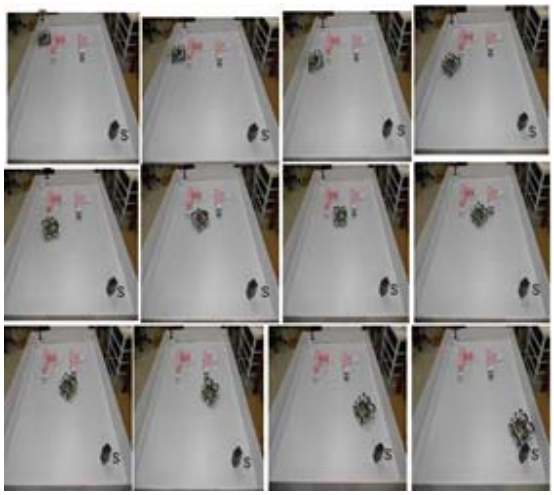

Fig. 10 (b) Test case II for the environment in Fig. 7 (a)

As noticed by the simulation results presented in Fig. 6 and Fig.7, it was concluded that the best trajectory with less time is achieved by GSA with given fitness function. In the experimental mode, the experiments are conducted using GSA algorithm with similar environments. Using GPS system the different target point coordinates are provided to the Arduino Mega 2560 micro controller. Then robot is allowed to move from the similar start point to target point, during navigation different positions of mobile robot and its path are presented in Fig.10 (a), (b) and Fig.11 (a), (b). It is observed that the mobile robot has successfully navigated from starting point to target point making collision free path.

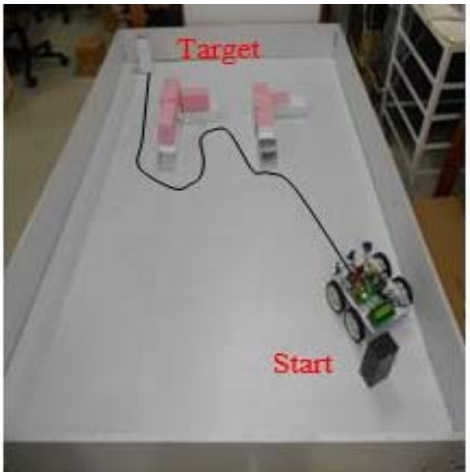

Fig. 11 (a) Path for test case I of the environment in Fig. 6 (a)

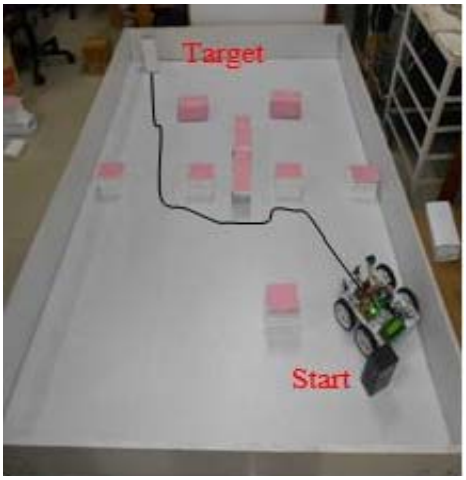

Fig. 11 (b) Path for test case II of the environment in Fig. 7 (a)

X. Conclusions

There exists many optimization approaches for path planning of mobile robot in unknown environment containing static obstacles. Considering the results cited in this paper it may be concluded that Gravitational Search Algorithm (GSA) is better than other two algorithms, because it gives better solution in both simulation mode using MATLAB and case studies also requires less time to execute and path length is shortest. The results of these simulations and experiments are very encouraging hence GSA has good contribution to the path planning of mobile robot in unknown environment having static obstacles in different positions.

REFERENCES

[1] J. H. Lin and Li-Ren Huang, "Chaotic Bee Swarm Optimization Algorithm for Path Planning of Mobile Robots," Proceedings of 10th WSEAS International Conference on Evolutionary Computing, pp. 84-89, Wisconsin, USA, March 23, 2009.

[2] P K Panigrahi, S Ghosh and D R Parhi, "A Comparison of Mamdani and Sugeno Based Fuzzy Controller for Mobile Robot to Avoid Static Obstacles", 5th International Elsevier Conference Electronics and Computer Science (IEMCON), pp.226-231, Aug 28-31, Kolkata, 2014.

[3] P K Panigrahi, S Ghosh and D R Parhi, "A Novel Intelligent Mobile Robot Navigation Technique for avoiding Obstacles Using RBF Neural Network", IEEE International Conference on Control, Instrumentation, Energy and Communication (CIEC), pp. 51-56 , Dec 31-Feb 2, Kolkata, 2014.

[4] O.Castillo, L.Trujillo and P. Melin, "Multiple Objective Genetic Algorithms for Path Planning Optimization in Autonomous Mobile Robot," Springer International Journal of Soft Computing, vol. 11, pp. 269-279, 2007.

[5] G. K. Venayagamoorthy, L. L. Grant and S. Doctor, "Collective Robotic Search Using Hybrid Techniques: Fuzzy Logic and Swarm Intelligence inspired by Nature", Journal of Engineering Applications of Artificial Intelligence, vol. 22, pp. 431-441, 2009.

[6] M. Yarmohamadi, H. H. S. Javadi and H. Erfani, "Improvement of Robot Path Planning Using Particle Swarm Optimization in Dynamic Environments with Mobile Obstacles and Target", Journal of Advanced Studies in Biology, vol. 3, no.1, pp. 43-53, 2011.

[7] B. Mohajer, K. Kiani, E. Sameiei and M. Sharifi, "A New Online Random Particles Optimization Algorithm for Mobile Robot Path planning in Dynamic environments", Hindwai Journal of Mathematical Problems in Engineering, vol. 2, pp. 1-9, 2013.

[8] S. Ahmadzadeh and M. Ghanavati, "Navigation of Mobile Robot using the Particle Swarm Optimization", Journal of Academic and Applied Studies (JAAS), vol. 2, pp. 32-38, 2012.

[9] D. R. Parhi, J. K. Pothal and M. K. Singh, "Navigation of Multiple Mobile Robots using Swarm Intelligence", IEEE conference on Nature and Biological Inspired Computing, pp. 1145-1149, Coimbatore, Dec 9-11, 2009.

[10] R. R. Sahoo, P. Rakshit, Md T. Haider, S. Swarnalipi, B. K. Balabantaray and S. Mohapatra, "Navigational Path Planning of Multi-Robot using Honey Bee Mating Optimization Algorithm (HBMO)", International Journal of Computer Applications, vol. 27, no.11, August 2011.

[11] N. A. Shiltagh and L. D. Jalal, "Optimal Path Planning for Intelligent Mobile Robot Navigation using Modified Particle Swarm Optimization", International Journal of Engineering and Advanced Technology, vol. 2, Issue - 4, pp. 260-267, April 2013.

[12] C. A. Sierakowski and L. D.S. Coelho, "Path Planning Optimization for Mobile Robots Based on Bacteria Colony Approach", Springer Book Series of Applied Soft Computing Technologies: The challenge of complexity, vol. 34, pp.187-198, 2006.

[13] E. Masehian and D. Sedighizadeh, "Multi-Objective PSO and NPSO based Algorithms for Robot Path Planning", Journal of Advances in Electrical and Computer Engineering, vol. 10, no.4, pp. 69-76, 2010.

[14] Xianmin Wei, "Robot Path Planning Based on Simulated Annealing and Artificial Neural networks", Journal of Applied Sciences, Engineering and Technology, vol. 5(23), pp.5384-5390, 2006.

[15] P. Curkovic and B. Jerbic, "Honey Bees Optimization Algorithm Applied to Path Planning Problem", International Journal of Simulation Model, vol. 6, pp. 154-164, 2007.

[16] W. Parvez and S. Dhar, "Path Planning Optimization Using Genetic Algorithm –A literature review", International Journal of Computational Engineering Research, vol. 3, Issue 4, pp. 23-28, 2013.

[17] D R Parhi, P K Panigrahi and S Ghosh," Wavelet Neural Network Based Intelligent Mobile Robotic Agent Operating in an Unknown Environment", International Journal of Artificial Intelligence and Computational Research, vol no.6, Issue no 1, pp. 77-83,2014.

[18] S. Yazdani, H. Nezamabadi-pour and S. Kamyab, "A Gravitational Search Algorithm for Multimodal Optimization", Elsevier Journal of Swarm and Evolutionary Computation, vol. 14, pp.1-14, 2014.

[19] T Ghose, "Optimization Technique and an Introduction to Genetic Algorithms and Simulated Annealing", Proceedings of International workshop on Soft Computing and Systems, pp.1-19, August, Mesra, 2002.

[20] G Bayar, E I Konukseven and A BugraKoku," Control of Differentially Driven Mobile Robot Using Radial Basis Function Based Neural Networks", WSEAS Transactions on Systems and Control, vol-3, issue 12, Dec 2008.

[21] S Noman , S M Shamsuddin and A E Hassanien," Hybrid Learning Enhancement of RBF Network with Particle Swarm Optimization", Foundations of Computational Intelligence , vol. 1,pp. 381-397,2009.

P.K.Panigrahi. received the Bachelor degree in Electrical Engineering from Institution of Engineers (India). M Tech. degree from Birla Institute of Technology, Mesra, Ranchi, India in Electronics Communication Engineering with specialization Control and Instrumentation.
Since 2009, he has been an Assistant Professor with Electrical Engineering Department, Padmanava College of Engineering, Rourkela, India. He has more than 17 years teaching experience in his field. His research interests include Navigational Control of Mobile Robot using Artificial Intelligence.

Saradindu Ghosh received his Ph.D. in Electrical Engineering from Indian Institute of Technology , Kharagpur, India. He has 27 years of experience in teaching and research, and presently he is Professor in the Department of Electrical Engineering, National Institute of Technology,Durgapur, India.He has published more than 60 papers in International Journals and Conferences.

Dayal R. Parhi received his first Ph.D. in Mobile Robotics from Cardiff School of Engineering, UK and second Ph.D. in Vibration from Sambalpur University, Odisha, India. He has gone to CMU, USA for advance research in the field of Robotics. He has 21 years of research and teaching experience in his fields. He has published more than 150 papers in international journals and conferences in the field of mobile robotics and vibration. He has written chapters for two Books. Presently, he is engaged in Mobile Robot Navigation research, and is in the post of professor in the Department of Mechanical Engineering, National Institute of Technology, Rourkela, India.