Path Planning in Indoor and Known Environment using BAPSO Algorithm

Mukesh Kumar Nandanwar*, Dr. Rekh Ram Janghel, Dr. A.S. Zadagaonkar*****
**CSE Department, Dr. C.V. Raman University, Bilaspur (C.G.)*
Email: mukesh.nandanwar@gmail.com
*** IT Department, National Institute of Technology, Raipur (C.G.)*
Email: janghel1310@gmail.com
**** ET&T Department, Dr. C.V. Raman University, Bilaspur, (C.G.)*
Email: arunzad28@hotmail.com

Abstract— **The path planning through BAPSO algorithm in indoor and known environments is a novel approach for robotic path planning. The BAPSO is a hybrid (BAT and PSO) algorithm. The main challenges of robotics are its automation and detection capability and the preliminary step of achieving these is through optimized path establishment, formation, flocking, and rendezvous, synchronizing and covering. Robotic Path Planning is one of the main problems that deal with computation of a collision-free path for the given robot, along with the given map on which it operates. Path establishment is possible only when the environment is known and the targets location is estimated. This work focuses on Path planning problem of coordinated multi-agents, target tracking and reaching in the indoor and known environment through obstacles avoidance. We are also showing the comparative results for this algorithm with other state of the art techniques. Parameters that asses these algorithms are no of node visit (move) and time elapsed (time).**

***Keywords*: Bio-inspired Algorithm, BAPSO, Hybrid Algorithm, Multi-Robot system, Path Planning, Target searching, Target tracking.**

I. INTRODUCTION

Every automated and intelligent robotic system uses planning to decide the motion of root and a real world. Robotics is an extremely inter-disciplinary area that takes inputs from numerous disciples and fields with varying complexities. The autonomous robots make extensive use of sensors to guide them in and enable them to understand their environment. A mobile robot exploring a known environment has no absolute frame of reference for its position, other than features it detects through its sensors [1].

Soft Computing Tools and Techniques are finding increasing application in the field of robotics. Many of the robotic problems like planning, coordination, etc. are complex problems that are now-a-days solved using the most sophisticated soft computing tools. In many real-world applications, the workspace of robots often involves various entities that robots must evade, such as a fire in the rescue mission, objects in household applications, and instruments in automated factories [2].

Path planning in a robot is the problem of devising a path or strategy which would enable the robot to move from a pre-specified source to a pre-specified goal. This can be established through the map. The generated map needs to ensure that the robot does not collide with the obstacles. The path planning would depend upon the size of the robot and the obstacles. In many cases, the robot may be assumed to be point size. This happened when we are sure that the robot would be in most of the cases able to glide through obstacles, assuming the obstacle density is not high. Another important characteristic of the path planning algorithm is its optimality and completeness. The completeness refers to the property that the algorithm is able to find a solution to the problem and return the solution in finite time, provided a solution exists.

The idea of combing two or more different algorithms into a single algorithm is known “hybrid Algorithms”. The new algorithm performing better than any of its component algorithms individually. The hybrid algorithm combines the strengths of the individual algorithms so that the resulting algorithm provides a combination of the following advantage:

- Can produce better solutions,
- And/or produce solutions in less time.
- Can effectively handle problems with larger input sizes.

Several algorithms to solve combinatorial optimization problems exist. Hybridization of some of these algorithms is intended to combine the strengths of their respective heuristic techniques into a better algorithm. This new algorithm should produce solutions more near optimal, or in less time, or both.

The algorithm which produces satisfactory results in less time can also be applied to larger problems [3]. Here I have used a hybrid algorithm i.e. BAPSO. The BAPSO algorithm is combination of BAT and Particle Swarm Optimization (PSO) algorithms.

The paper is organized in as follows. In section II, we describe the background of the problem, Section III deals with the problem formulation, Section IV we show simulation result to test the effectiveness of proposed algorithm.

II. BACKGROUND

Path planning is one of the basic and interesting functions for a mobile robot [4]. Motion planning of the robot in respect of natural is a challenging task [5] due to the various constraints and issues related to it. The most vital issues in the navigation of mobile robot, which means to find out an optimized collision free path from the start state to the goal state according to some performance merits i.e. travel time, path-length etc. The optimization in terms of time and path length and limiting constraints, especially in large sized maps of high resolution, pose serious challenges for the researcher community [6]. The Robotic path planning in the presence of the obstacle is also a critical scenario [7].

Path planning can be classified into two categories global path planning in which all the information are available to the robot and local path planning in which partially information available of the environment [8]. The environment structure might be in static or dynamic in nature [9]. There have been many algorithms for global path planning, such as artificial potential field, visibility graph, and cell decomposition etc. The potential field has been used widely due to its simple structure and easy implementation, yet it still has some shortcoming [10-11].

Exploration and mapping also be a fundamental part of mobile robots navigation. In the past, any approaches have used occupancy grid maps to represent the environment structure during the map building process. Occupancy grids, however, are based on the assumption that each cell is either occupied or free [12].

Human beings are greatly inspired by nature [13]. Nature in itself is the best example to solve problems in an efficient and effective manner. During the past few decades, researchers are trying to create computational methods that can help human to solve complex problems. This may be achieved by transferring knowledge from natural systems to engineered systems. Nature inspired computing techniques such as swarm intelligence, genetic algorithm, artificial neural network, DNA computing, membrane computing and artificial immune system have helped in solving complex problems and provide an optimum solution. The parallel, dynamic, decentralized, asynchronous and self-organizing behavior of nature inspired algorithms is best suited for soft computing applications and optimized for solving the complex problems [14]

Path planning algorithm of robots also is inspired by a social behavior of birds and fish. This types of algorithm known as Nature Inspired Algorithm (NIA). NIA Algorithms are flexible and work in changing the environment to organize and grow accordingly**.** These algorithm qualities perform in highly complex problems and can even show very good results when a problem is not properly defined. This tends to find the best available solution in every changed environment and very good decision maker. Nature inspired systems do not adapt to real world system fully in terms of scalability and performance. Systems can work well in some domain but not in other. As systems are nature inspired, then not having proper knowledge of nature can affect the design of algorithm. So for effective results no ambiguity in data is required. This needs Cluster of data. Clustering is the process of organizing data into meaningful groups, and these groups are called clusters [15]. Many specialized models of soft computing also face these problems of large data processing. This technique is restricting the amount of data that the system can serve. This achieved by clustering the data. These algorithms have the ability to self-learn, self-train, self-organize, and self-grow. They can find best optimal solutions to complex problems using simple conditions and rules of nature.

Coordination among interacting agents with a common group behavior is not only a recent engineering problem, but it is something that happens every day in nature. The interest in multi-agents systems has grown in the last decade, mostly because a group of collaborating agents can often deal with tasks that are difficult, or even impossible, to be accomplished by an individual agent. A team of agents typically provides flexibility, redundancy, and efficiency beyond what is possible with single agents. Having several agents deployed often means that they can be flexibly organized into groups performing tasks at different locations. Robustness is ensured by the numerous of the group if an agent fails the other can continue the task. Having many agents rather than one can allow performing tasks in a faster and more precise way. The multi-robot system is used for many applications due to robustness, potential to finish a task faster, varied and creative solution [16-17].

Particle Swarm Optimization (PSO) is a Natural-inspired algorithm. PSO was used as optimization method because it has interesting results in situations with local minima. It is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best-known position but, is also guided toward the best known positions in the search space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. It is used many field i.e. social modelling, computer graphics, simulation, and animation of natural swarms or flocks with artificial neural networks and evolutionary computation due to simplicity and efficiency [18-19].

Bat algorithm is a bio-inspired algorithm. It is a meta-heuristic optimization algorithm developed by Xin-She Yang in 2010. This bat algorithm is based on the echolocation behavior of microbats with varying pulse rates of emission and loudness. The idealization of the echolocation of microbats can be summarized as follows: Each virtual bat flies randomly with a velocity v_i at position (solution) x_i with a varying frequency or wavelength and loudness A_i. As it searches and finds its prey, it changes frequency, loudness and pulse emission rate r. Search is intensified by a local random walk. Selection of the best continues until certain stop criteria are met. This essentially uses a frequency-tuning technique to control the dynamic behavior of a swarm of bats, and the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters in bat algorithm [Cite].

A hybrid algorithm is an algorithm that combines two or more other algorithms that solve the same problem, either choosing one (depending on the data), or switching between them over the course of the algorithm. This is generally done to combine desired features of each, so that the overall algorithm is better than the individual components.

"Hybrid algorithm" does not refer to simply combining multiple algorithms to solve a different problem. Many algorithms can be considered as combinations of simpler pieces, but only to combining algorithms that solve the same problem, but differ in other characteristics, notably performance. So many researchers have been tried to solve problem using hybrid algorithms for find smooth path between the source and destination through A* and potential field method [20], PSOPC and UPSO [21], hybrid of Ant Colony Optimization and Genetic algorithms (ACOG) [22], hybrid of genetic algorithm and particle swarm optimization (GAPSO) [23], hybrid optimization algorithm of particle swarm optimization (PSO) and Cuckoo Search (CS) [24], hybrid genetic algorithm particle swarm optimization (HGAPSO) algorithm [25].

In this paper, we deal with the behavior of robots movement, target searching and path planning problem for multi-robot using a hybrid algorithm i.e. BAPSO. The BAPSO algorithm is combination of BAT and Particle Swarm Optimization (PSO) algorithms.

III. PROBLEM FORMULATION

For problem formulation, we have considered the known target. Known target is one kind of target tracking problem which involves reaching the target in an efficient way and with an optimized expenditure of the resources. Target tracking involves mainly avoidance of obstacles in an intelligent and explorative way so that new options get revived and recognized. So the challenge lies in searching for the various nooks of the environment and search for the target. There are some complex environments like the indoor environment where due to complex structure of walls, doors, and windows even the known target problems are tackled like unknown target problems. Gradually the various results are discussed for the indoor environment under known target schemes and the performance is evaluated and compared.

A. Assumptions

The simulation work was performed as a multi-agent system where the agents represented the robots, with all its inherent behavior and characteristics. Assumptions made in simulations are: Agents are homogeneous that is uniform in capacity, strength, size (the capacity of the unit cells) uniformity in the environment and the agents also make the fitness calculation simple and easy to compare between agents.

1. No failure of robots during path planning and exploration.
2. Speed is non-uniform but bounded and the maximum is same for all the agents.
3. There are instant breakings and no stretching happens following that.
4. Sensor range is Omni-directional and there is coordination fusion between all the sensors energy.
5. The position of the obstacle is known with reference to the co-ordinate system.
6. The obstacles can be of any shape inside a circular region as shown in Figure 1.
7. The obstacles can be adjacent but not overlapping.
8. The path planning algorithm runs until the goal has been reached.

Figure 1: Obstacles having different shapes

B. BAPSO for Robot

Hybrid BAPSO algorithm in an enhanced and modified version of particle swarm optimization and bat optimization with more control and intermediate variation through the introduction of an extra equation. The work of the extra equation is to select a certain portion of the difference in between the present position and the best position or more perfectly to say the percentage of the social influence from the best agent.

In robot agent path generation the role of hybrid BAPSO algorithm is to introduce randomness in local exploration through better frontier selection. The hybrid BAPSO algorithm is for better optimization of robot movement. The basic equations that are used in hybrid BAPSO algorithm for robot movement and their implications with respect to robotics application are discussed below.

$$f = f_{min} + (f_{max} - f_{min}) * \beta \quad (1)$$

$$dx = a * V_1 + V_1 + (p_i - x_i) * f \quad (2)$$

$$x_f = x_i + dx \quad (3)$$

Here ‘a’ is a random number, β is the weightage parameter that need to be considered for contribution for the difference for the next iteration, f_{min} and f_{max} are frequencies, which depends on range of environment. It must be kept in mind that as the movement or rather the step of movement depends on the global solution ad as any of the solution is not complete. Hence the p_i (global best) cannot be determined, but can be replaced by an equivalent. Here we have used the remaining distance as the criteria for determination of the best solution and the more near the agent is the better is its fitness. The rest of the variables have the same implications as the classical Bat algorithm, but the range depends on the search space, and partly determined and converged through experimentation.

Algorithm 1: Algorithm for Robot Movement:

1. Initialize the agents and required data structure
2. Start the particle from sources
3. If condition <= Threshold Value
   Move the agent using BAPSO
   Else
   Move using directional movement
   End of if
4. Determine fitness of the particles
5. Update the global and iteration best
6. Continue the loop until stop condition is true.

The BAPSO algorithm for mobility of Robots is described in the Algorithm-1. The condition is nothing but a threshold value for finding the next movement of an agent through BAPSO or directional movement. Directional Movement is Clustering Based Distribution Factor (CBDF) which can be used in scattering the robot agents randomly towards the various portion of the search space for efficient an opportunistic search and exploration of the target.

C. Steps for directional Movement for known target

(a) Detect the present position P_{old}

(b) Detect the known target position P_D

(c) Calculate unit vector for each direction as

$$\delta = \frac{(P_D - P_{old})}{\|P_D - P_{old}\|} \qquad (4)$$

(d) Calculate new variation for each dimension as

$$V_K = \delta_k * D \qquad (5)$$

Where $\delta_k \in \delta$ and $D \in S$ (Detection range)

$$P_{new} = P_{old} * V_k \qquad (6)$$

Where $V \in (V_1, V_2, \dots, V_k)$ and V is the variation.

IV. SIMULATION STUDIES

The problem of path planning deals with the determination of a path which navigates the robot in such a way that no collision occurs. In order to solve the problem, it is assumed that the input is already available in the form of a map. The region is traversable and may be used for the purpose of travelling. The black regions here signify the presence of obstacles. The size of the map is restricted according to the computational capability and time constraints in all the real life specific problems under consideration. Otherwise, it would become computationally impossible for the algorithm to find a result. The algorithm would generate output path that can be used by the robot for the navigation purposes. The path may be traversed using any robotic controller. This is for the execution of the steps given by the planning algorithm.

Parameters	**Values**
Map Size	500x500 pixel
Number of maps	1
Map Type	Indoor
Sensor Range	1 to 15 pixel
Target	Known
Number of Robots	3, 6, 9, 12 & 15
Start & Ending point	Any point set within map
Algorithms	BAPSO
Features	Move, Time

Table I: Experimental Parameters

The summaries of experimental parameters are showing in below Table I. The proposed Algorithm-1examine through the no of node visit (Move) and simulation time (Time) which various when the number of agents is varied. The exact time calculation will be approximately the average timing factor that is the noted time calculation divided by the number of agents.

A. Environment Description

The main challenge for engineering these systems is to design and develop distributed and adaptive algorithms that allow system entities to select the best suitable strategy / action and drive the system to the best suitable behavior according to the current state of the system and environment changes [19]. Robot path planning in known and dynamic environments is feasible for mobile robots and its main purpose is to find a collision free path for a robot from an initial position to a goal position in an environment with obstacles [20].

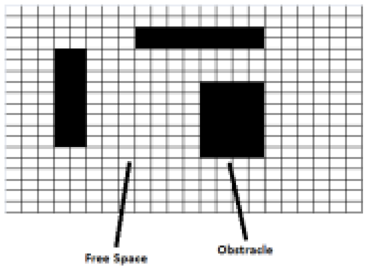

Fig 2: Environment representation

Environment or map representation is a critical task in robotics path planning. In this section we define and explain the terms that are used in the following sections. We assume the robot in question uses an occupancy-grid map representation in the exploration process (fig 2) within the map.

- **Unknown Region** is a territory that has not been covered yet by the robot's sensors.
- **Known Region** is a territory that has already been covered by the robot's sensors.
- **Open or Free Space** is known region which does not contain an obstacle.
- **Occupied space** is a known region which contains an obstacle.
- **Occupancy Grid** is a grid representation of the environment. Each cell holds a probability that represents if it is occupied.
- **Grid maps** is a discrete the environment into so-called grid cells. Each cell stores information about the area it covers.

This work uses grid map representation for environment. Where consider a 2D square map overlaid with a uniform pattern of grid points. The size of a map can be changed arbitrarily.

B. Simulation Result Analysis for BAPSO Algorithm

BAPSO Algorithm is useful for planning a path in dynamic, complex state space environment. Figure 3 is shows the results for complex environment i.e. indoor.

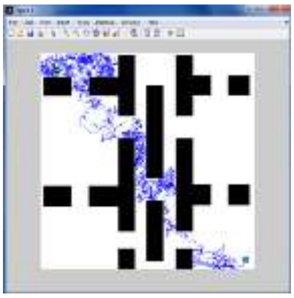

Fig. 3: Explored Area to reach known target using BAPSO Algorithm

C. Comparisons of individual Algorithms for Target searching and Tracking with varying Agents

The detection capability of the robot agents plays an important role for detection and local search movement while taking decision for the next course of movement. There is no doubt that the visibility of the robot is directly proportional to the step-size of movement and will help in movement and simultaneously in exploration denoted by the parameter coverage. However it must be mentioned that the directional capability of the robot agent in an environment depends on the distance between the obstacle and must not be too large or too small with respect to the average distance between the obstacle else it can be a wastage of the resources as the detection capability requires for hardware detection and analysis capability and the consumption of energy increases with increase in detection capability. Hence, the optimal detection capability must be justified for the robot system and must be optimized. In the simulation, we have used a range of detection ranges and have shown how the convergence rate applies for the situations. Another thing that needs to be mentioned is that the result is biased and optimized for the agents with least movements and the other parameters like time is the corresponding of that agent. Hence always the other parameters like time may not be optimized but in most of the cases, they are, however, it has been seen that agents with better timing are found in the iteration but it does not have the best moves count.

The following graphs are shown between sensor range and move, time parameters in different known environments with varying number of agents. The BAPSO algorithm is used for reaching the goal.

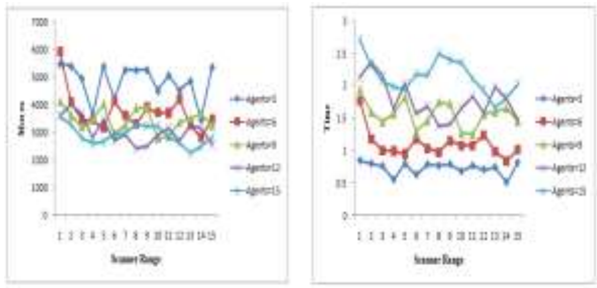

Fig: Move parameters **Fig: Time parameters**

Fig. 4: Move, Time parameters comparison for BAPSO

D. Observation

Following are observed are derived from the gradual development of simulation setup and the environment.

i. More diversified results when the social effect of the co-agents is determined from a randomly selected co-agent.
ii. For a particular environment, there is some optimal sensor range which will be sufficient.
iii. Effectively more proportion of bio-inspired computation in target tracking produces more combination of paths.
iv. Best Ratio of exploration and exploitation depends on the environment and the associated topology.
v. Quality of solution depends on the ratio of exploration and exploitation.
vi. If exploration depends on randomization and each event have equal probability of occurrence, then it may happen that after certain number of steps, the robot may end-up at the same place as it was before.
vii. Effectively more proportion of CBDF in target searching procures more combination of paths.
viii. The number of agents is increased than the number of move decreased.
ix. The number of agents is increased than the time is also increased.

V. CONCLUSIONS

In this work, we are particularly interested in the use of Nature inspired algorithm in multi-robot path planning and area exploration. We discussed BAPSO based path planning technique. Mathematical details were discussed and simulation results are shown for path planning in two different parameters i.e. moves and time in known target position. The approaches presented in this work have been tested in various simulated environment with different complexity and different team size. The results positively show the optimality of proposed algorithm as compared to other state of the art techniques.

Since the work presented here is independent of robot model, it can be easily extended to different robot models as compared to point mass model considered. It can be easily applied to consensus of robots performing task in different but interconnected environment.

REFERENCES

1. Lu F. & Milios E. (1997). Robot pose estimation in unknown environments by matching 2d range scans. Journal of Intelligent and Robotic Systems 18(3), 249-275.
2. Zhang Y., Gong D. W. & Zhang J. H. (2013). Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neuro-computing 103, 172-185.
3. Malek M., Guruswamy M., Owens H. & Pandya M. (1989). A hybrid algorithm technique. University of Texas at Austin, Department of Computer Sciences.
4. Hossain M. A. & Ferdous I. (2015). Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. Robotics and Autonomous Systems 64, 137-141.
5. Davoodi M., Panahi F., Mohades A. & Hashemi S. N. (2013). Multi-objective path planning in discrete space. Applied Soft Computing 13(1), 709-720.
6. Kala R., Shukla A. & Tiwari R. (2012). Robotic path planning using hybrid genetic algorithm particle swarm optimisation. International Journal of Information and Communication Technology 4(2-4), 89-105.
7. Gilbert E. G. & Johnson D. W. (1985). Distance functions and their application to robot path planning in the presence of obstacles. Robotics and Automation IEEE Journal of 1(1), 21-30.
8. Chakraborty J., Konar A., Chakraborty U. K., & Jain L. C. (2008, June). Distributed cooperative multi-robot path planning using differential evolution. In Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on (pp. 718-725). IEEE.
9. Oroko J. & Nyakoe G. N. (2014 December). Obstacle Avoidance and Path Planning Schemes for Autonomous Navigation of a Mobile Robot: A Review. In Proceedings of Sustainable Research and Innovation Conference (pp. 314-318).
10. Tereshko V. (2000, September). Reaction-diffusion model of a honeybee colony’s foraging behaviour. In Parallel Problem Solving from Nature PPSN VI (pp. 807-816). Springer Berlin Heidelberg.
11. Chiong R. (Ed.). (2009). Nature-inspired algorithms for optimisation (Vol. 193). Springer.
12. Stachniss C. & Burgard W. (2003 October). Mapping and exploration with mobile robots using coverage maps. In Intelligent Robots and Systems 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on (Vol. 1 pp. 467-472). IEEE.
13. Dixit M., Upadhyay N. & Silakari S. (2015). An Exhaustive Survey on Nature Inspired Optimization Algorithms. International Journal of Software Engineering & Its Applications 9(4).
14. Gupta S., Bhardwaj S. & Bhatia P. K. (2011). A reminiscent study of nature inspired computation. International Journal of Advances in Engineering & Technology 1(2), pp.117-125.
15. Dhote C. A., Thakare A. D. &Chaudhari S. M. (2013 July). Data clustering using particle swarm optimization and bee algorithm. In Computing Communications and Networking Technologies (ICCCNT) 2013 Fourth International Conference on (pp. 1-5). IEEE.
16. Nooraliei A., & Mostafa S. A. (2009). Mobile robot path planning using wavefront approach with WEFO. In International Conference on Advanced Computer Theory and Engineering (ICACTE 2009). ASME Press.
17. Nooraliei A., & Nooraliei H. (2009). Robot path planning using wavefront approach virtual wave hill.
18. Banks A., Vincent J. & Anyakoha C. (2007). A review of particle swarm optimization. Part I: background and development. Natural Computing 6(4), 467-484.
19. Ercan F. M. & Li X. (2013). Particle swarm optimization and its hybrids. International Journal of Computer and Communication Engineering 2(1), 52-55.
20. Du Z., Qu D., Xu F. & Xu D. (2007 December). A hybrid approach for mobile robot path planning in dynamic environments. In Robotics and Biomimetics 2007. ROBIO 2007. IEEE International Conference on (pp. 1058-1063). IEEE.
21. Babaee H. & Khosravi A. (2011 August). An improve PSO based hybrid Algorithms. In Management and Service Science (MASS) 2011 International Conference on (pp. 1-5). IEEE.
22. Geetha S., Chitra G. M. & Jayalakshmi V. (2011 April). Multi objective mobile robot path planning based on hybrid algorithm. In Electronics Computer Technology (ICECT) 2011, 3rd International Conference on (Vol. 6 pp. 251-255). IEEE.
23. Huang H. C. & Tsai C. C. (2011 September). Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm. In SICE Annual Conference (SICE) 2011 Proceedings of (pp. 1338-1343). IEEE.
24. Wang F., Luo L., He X. S. & Wang Y. (2011). Hybrid optimization algorithm of PSO and Cuckoo Search.
25. Kala R., Shukla A. & Tiwari R. (2012). Robotic path planning using hybrid genetic algorithm particle swarm optimisation. International Journal of Information and Communication Technology 4(2-4), 89-105.

