Optimal Path Forecasting of an Autonomous Mobile Robot Agent Using Breadth First Search Algorithm

M.Bala Subramanian[1], Dr.K.Sudhagar[2], G.RajaRajeswari[3]

***Abstract*— Predicting and identifying an optimal navigational forecasting for a mobile robot agent is considered to be one of the key challenges for researchers in the field of AI and robotics. A mobile robot agent participating for wide range of mission critical application will explore into an unknown environment will be challenging criteria, considering its contraints such as time, cost, energy, exploration distance etc.. When the autonomous mobile robots agent decides its action, it is necessary to plan its movement more precisely and optimally. This paper presents the use of search algorithm in optimal path forecasting with its navigation controlling mechanism of a autonomous mobile robot agent navigational systems. The autonomous mobile robot agents are modelled to work independently through its intelligence without any human intervention. This paper aims on studying the movement details of an autonomous mobile robot agent participating in dynamically changing environment using Breadth First Search (BFS) algorithm. The system evaluation is validated using Graphical User Interfcae (GUI) based test bed for robots called Robosim and the efficiceny of the system is measured via simulation resutls through a defined complex arena. Simulation results proven that the applying the BFS algorithm in a unknown envirnoment explores much faster then heuristic based other path palnning algorithms**

***Index Term*— Autonomous Mobile Robot Agent, Path Planning, BFS algorithm, A* algorithm, RoboSim**

I. INTRODUCTION

Autonomous Mobile robots (AMR) are physical agent that move and interact continuously while participating in a static or dynamic environment [15]. They are designed for wide range of mission critical applications, in which most of the applications are highly depend on its movement from the Pre-defined Source position to the known or unknown target position. The Robot Agent (RA) navigation and path planning involves getting a RA to automatically determine how to move while avoiding collision with obstacles. The ability to explore and planning its own motion into a known or unknown environment is considered to be an important capability of a truly autonomous robot [1]. When a AMRA is in goal discovery, there are high chances for path obstacles before achieving its target position and without reacting to the unforeseen events; it will never reaches its target location. There were numerous algorithms proposed and investigated to handle this fundamental path finding problem [2, 3, 4, 5, 6, 7, 15, 16] . RA navigation system, does not know the path that RA to be follow. In order to achieve and obtain the target position, the RA should update its current location details along with the environmental information that it gains while in exploration process. So, that the navigation control system will perform an optimal path input feeding mechanism. There are several algorithms used to obtain a path in such environments. Over last few decades, there were several number of algorithms were proposed by many researchers for path planning in static and dynamics environment. The characterization of algorithms is made with respect to the length of generated path and with respect to the time spent to obtain it. These two criteria are independent of each other. The shortest path does not necessarily imply the minimum time, or vice versa. The graph search algorithm, are the most known solutions for this problem. These algorithm uses directed or undirected graph trees. In addition these algorithms were used in most of the computer based simulated or online games and in GPS systems for to identifying shortest and lower cost path [1][4].

II. LITERATURE REVIEW

With the latest advancement on Artificial Intellignce (AI), RA is deployed in the task of searching and rescuing operation in an uncertain or dynamic environment. The searching effficiency is considered to be high priority task in this mission critical application. A typical RA navigation system includes a high intelligence path planning module which determine appropirate path for a RA based on the current map and the corresponding obstacle avoidance algorithm which will determines a suitable direction of motion exploration based on the obtained data. Path planning consideres a model or map of the environment to detrermine the geometric path point for the RA to track from its initial starting postion to the target or goal position.

San et-al proposed a sensor based path planner methodology to handle both local and global path planning [3]. Qualid et-al proposed a methoid using DVFF approach bsedon D* algorithm, [4 ,5, 6, 7]. The common path planning method such as genetic algorithm [7], A* algorithm [8], particle swarm optimazation [9], D * search algorithm, Moore algorithm were discussed. shi-gang, HUi, Wang, Li yang made a simulation study on application of both the D* and A* algorithm in intelligent trasnfer system as well as robotic path planning [8]. Tolga,Sezgin made an experimental study by implementing various algorithm, in identifying the optimal path for a mobile robots on a grid based map [11]. The AMRA are initially positioned at the center of the objective environment and control agents start to work without having any preliminary information about the environment. It will depend on the composing process. According to its current position and following the defined exploration algorithm the AMRA starts its exploration process In parallel and independently of what this agent is doing, the agent that avoids obstacles calculates other velocity values according to infrared sensor values that are included in the navigation control architecture[2]. This paper aimed at

AMRA in uncertain environment using the BFS algorithm can realize AMRA to precisely turn away from dynamic obstacles, and if discovering new obstacle in the original path then the AMRA redesign path, until it achieve its target.

III. Problem Definition

The multi-robot path planning problem is defined as follows: given a set of n AMRA in k-dimensional work space, each with an initial starting configuration and a desired target configuration, determine the path each AMRA should take to reach its goal, while avoiding collisions with obstacles and other AMRA in the workspace [12]. The collision free path planning for a AMRA among static obstacles are formally considered as: Let 'm' be a rigid AMRA in a static workspace Ws = Rk, where k=2 or k=3. The workspace is populated with obstacles. A configuration q is a complete specification of the location of every point on the AMRA geometry. The configuration space C represents the set of all the possible configurations of M with respect to Ws. Let O ⊂ WS represent the region within the workspace populated by obstacles. Let the closed set M(q) ⊂ WS denote the set of points occupied by the AMRA when it is in the configuration q ∈ C. Then, the C-space obstacle region, Cobs is defined as:

$$C_{obs} = \{q \in C | M(q) \cap O \neq \Theta\} \quad (1)$$

The set of configurations that avoid collision is:

$$C_{free} = C/C_{obs} \quad (2)$$

A free path between two obstacle-free configurations c_{init} and c_{goal} is a continuous map and is represented as $T[0, 1] \rightarrow C_{init}$. Such that $T(0) = c_{init}$ and $T(1) = c_{goal.}$ For a team of m AMRAs, define a state space that considers the configurations of all the AMRAs simultaneously:

$$X = C_1 \times C_2 \times \cdot \cdot \cdot \times Cm \quad (3)$$

Note that the dimension of X is N, where $N = \sum_{i=1}^{m} dim(C_i)$. The *C-space* obstacle region must now be redefined as a combination of the configurations leading to an AMRA-obstacle collision, together with the configurations leading to AMRA-AMRA collision. The subset of X corresponding to AMRA Mi in collision with the obstacle region, O is

$$X^{i}_{obs} = \{x \in X | M^{i}(q^{i}) \cap O \neq \Theta\} \quad (4)$$

The subset of X corresponding to AMRA Ai in collision with AMRA Mj is

$$X^{ij}_{obs} = \{x \in X | \mathbf{M}^{i}(\boldsymbol{q}i) \cap \boldsymbol{M}^{j}(\boldsymbol{q}^{i}) \neq \Theta\} \quad (5)$$

The obstacle region in X is then defined as the combination of Equations 1 and 2, resulting in

$$X_{obs} = (U^{m}_{i=1} X^{i}_{obs}) \cup (U_{i,j,i\neq j} X^{i}_{obs}) \quad (6)$$

With these definitions, the planning processes for multi-RA systems treats *X* the same as *C*, and X_{obs} the same as C_{obs} ,where C_{init} represents the starting configurations of all the RAs, and C_{goal} represents the desired goal configurations of all the AMRA. The AMRA should obtain the collision-free configuration connecting the initial and the target configurations.

IV. Breadth First search (BFS) Algorithm

BFS algorithm works with the method branching from the starting cell to the neighbour cells (just traversable cells), (untraversable cells and cells out of boundaries are discarded) until the goal cell is found [13]. It is a strategy for searching in a graph when search is limited to essentially two operations: (a) visit and inspect a node of a graph; (b) gain access to visit the nodes that neighbour the currently visited node. The BFS algorithm begins at a root node and inspects all the neighbouring nodes. Then for each of those neighbour nodes in turn, it inspects their neighbour nodes which were unvisited, and so on. In this traversal process, every node in the graph, we assign the direction to each edge from discoverer S to discovered D. The discoverer source point 'S' is considered as a 'master/parent' of 'subordinate/child' D, since each node has exactly one master/parent node, except root node.

```
BFS(G, s)
    for each vertex u ∈ V [G]− {s} do
        state[u] = "undiscovered"
        p[u] = nil, i.e. no master/parent is in the BFS tree
    state[s] = "discovered"
    p[s] = nil
    Q = {s}
    while Q  = ∅ do
        u = dequeue[Q]
        process vertex u as desired
        for each v ∈ Adj[u] do
            process edge (u, v) as desired
            if state[v] = "undiscovered" then
                state[v] = "discovered"
                p[v] = u
                enqueue[Q, v]
        state[u] = "processed"
```

A. Search Process

The algorithm uses a queue data structure to store intermediate results as it traverses the graph as follows:

a) Define the Map Matrix

b) Define the Source and Target goal location details and load the map matrix..

c) Enqueue the root node

d) De-queue a node and examine it

e) If the element sought is found in this node, quit the search and return a result.

f) Otherwise enqueue any successors (the direct child nodes) that have not yet been discovered.

g) If the queue is empty, every node on the graph has been examined – quit the search and return "not found".

h) If the queue is not empty, repeat from Step 2.

V. Simulation Results

A large amount of simulation software is available for AMRA systems, and it is already being used extensively. The majority of the AMRA simulation tools focus on the motion of the robotic manipulator in different environments. As the motion simulation has a central role in all simulation systems they all include the kinematics or dynamic models of robot manipulators [10]. Which type of models will be used depends on the objective of the simulation system. All these simulators are with the goal of creating an artificial robot environment, as real as possible, considered to be the test bed for the implementing the mobile robotic algorithms. The effectiveness of the above defined approach is tested and evaluated in RoboSim simulation environment, which shows potential results in the reality and nonexistence of anonymous obstacles.

This paper presents a algorithm based on cell branching method, called Breadth First Search (BFS) algorithm for an AMRA on a grid map based dynamic environment using RoboSim Simulator tool [17]. RoboSim is a Java based robotics simulation utility that is easy to understand some basic concepts of movement and path planning algorithms. RoboSim comes with various modules, which enables the researchers by learning about the path planning and smoothing concepts.

The working environment is defined based on the input grid size. Using this pre-defined GRID world, the BFS algorithm constructs a path between the starting point and the goal point. It is shown that, when a AMRA start executing its current navigational plan and it continuously sense the obstacles for next movement. Once the obstacles are sensed the AMRA sends the signal to its associated master agents and the newly constructed path is passed through communication channel to the AMRA. The input for this simulation is defined as grid coordinates. Fig. 1 defines a grid map of 7 X 7 cells, using World Builder of RoboSim and the starting node of the AMRA is spotted with red color and the black node represent obstacles and node with green spot shows as target point. The AMRA motion analysis or path trace details are shown in Fig. 2, 3, 4.

Fig. 1. Grid Map

Fig. 2. Exploration of BFS algorithm without diagonal motion

It is assumed that, the initial position of a AMRA is set to be as (2,1), defined with an path obstacles in (2,2), (2,4), (4, 2), (4,5), (5,3), (5,8), (6,5), (7,1), (7,5) and defined thee target point as (7,7). The input details are passed to the AMRA. . Fig. 2, 3, 4 shows the behaviour movement or exploration aspects of a AMRA.

The AMRA movements are defined with its initial position and the movements from initial to target positions are captured along with the path trace details for different scenarios, under different parametric function. Fig. 2 shows the exploration of AMRA applied with BFS algorithm, and without allowing any diagonal motion. Fig. 3 uses exploration of AMRA applied with BFS algorithm, allowed with diagonal motion and Fig 4 shows exploration of A* algorithm for comparative purpose

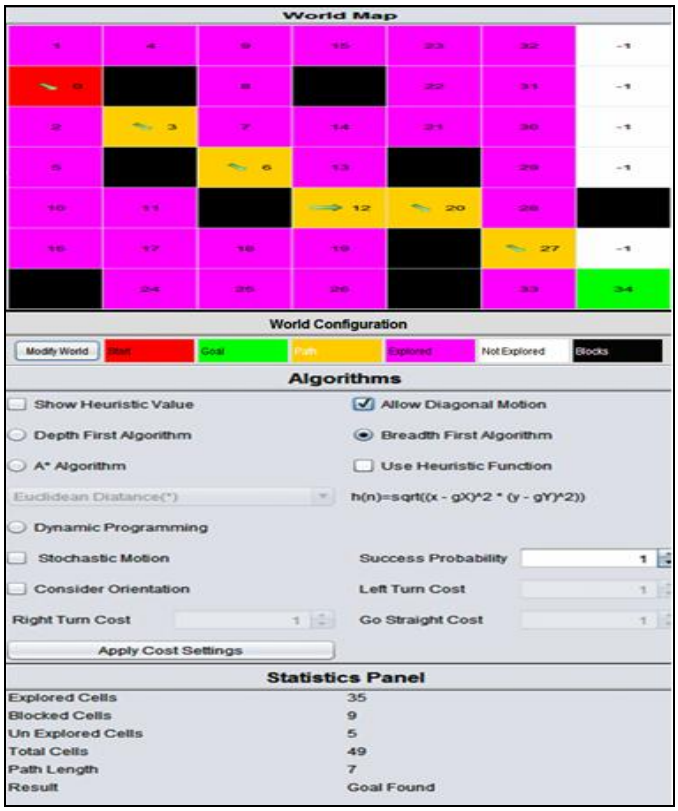

Fig. 3. Exploration of BFS algorithm with diagonal motion

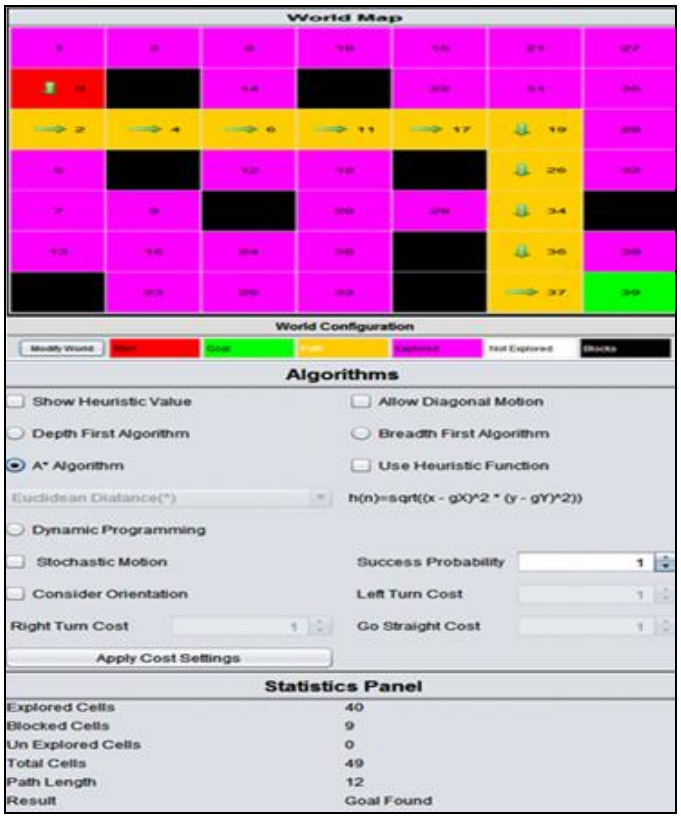

Fig. 4. Exploration of A* algorithm without heuristic values

TABLE I.
Comparison of BFS algorithm with A* (A Star)

Statistics	Explored Cells (M^2)	Blocked Cells (M^2)	Unexplored Cells (M^2)	Total Cells (M^2)	Path Length (M^2)
BFS algorithm (Nodiagonal motion)	**40**	**9**	**0**	**49**	**12**
BFS algorithm diagonal motion	**35**	**9**	**5**	**49**	**7**
A Star (A *) algorithm	**40**	**9**	**0**	**49**	**12**

Table I, shows the functional behavioural exploration comparison details on BFS algorithm applied and allowed with diagonal motions which will be heuristics for a AMRA. Fig. 5 shows the graphical representation of the above comparison of BFS algorithm with A* algorithm.

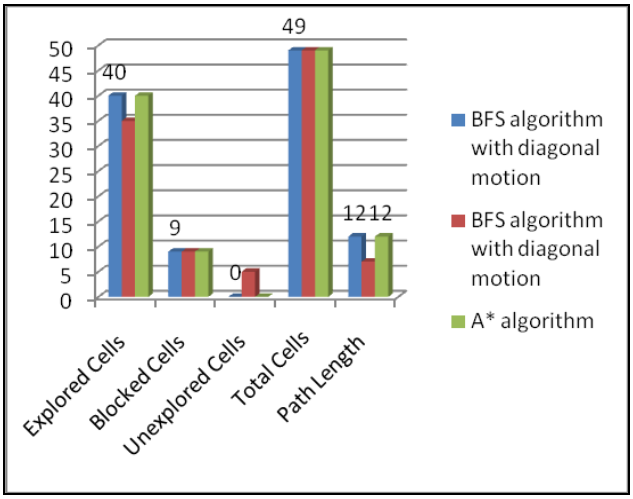

Fig. 5. Graphical ouput comparison of BFS algorithm with A* algorithm.

It has been observed that, When A* algorithm computes f(n), the square of distance will be much higher than the cost g(n) and it will end up with an overestimating heuristic. Also after applying different heuristic value for path calculation it has been observed that the AMRA explores as optimal as possible, using BFS algorithm depends on heuristic selection.

VI. Conclusion

This paper exhibits a comprehensive study on BFS algorithm, by performing various simulation experiments. To realize the efficiency of the BFS algorithm, it is simulated via Java based simulation tool called RoboSim. The experimental results shows that BFS algorithm can realize the path planning under the dynamic environment very optimal and successfully avoid the obstacles, and reach to its target point. However, when applying different heuristic function, it has been observed that the results are getting differed. This algorithm will be further investigated and will be deployed in real time AMRA, participating in mission critical application.

REFERENCES

[1] B.Frank, C.Stachniss, N.Abdo and W.Burgard, "Efficient Motion Planning for Manipulation Robots in Environments with Deformable Objects," IEEE International conference on Intelligent Robots and Systems (IROS), pages 2180-2185

[2] A.Oualid, Karim and Redouane, "A Sensor Based Navigation Algorithm for a Mobile Robot using the DVFF Approach," International Journal of Advanced Robotic Systems,Vol 6, No.2(2009) ISSN 1729-8806, pp. 97-108

[3] Garrido, Moreno, Blanco & Jurewicz, "Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching," International Journal of Robotics and Automation (IJRA), Vol. 2: Issue (1): 2011

[4] S. Koenig and M. Likhachev, "Fast Replanning for Navigation in Unknown Terrain," IEEE Transactions on Robotics, Vol. 21, pp. 354-363, 2005.

[5] A. Poncela, C. Urdiales, E. J. Perez and F. Sandoval, "A New Efficiency Weighted Strategy for Continuous Human/Robot Cooperation in Navigation," IEEE Transaction .on Systems, Man and Cybernetics Part A: Systems and Humans, Vol. 39, pp. 486-500, 2009.

[6] Shu-Yun Chung and Han-Pang Huang, "Predictive Navigation by Understanding Human Motion Pattern," International Journal of Advanced Robotic Systems, Vol.8, No.1, 2011, pp 52-64

[7] I.K.Nikolos, K.P. Valavanis "Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation," IEEE Transaction on Syetems, Man, and Cybernetics, Vol.33, No.6,DEC 2003

[8] Shi-Gang, Hui, Yang, "A Simulation Study of A-star Algorithm for Robot Path Planning, " 16th International Conference on Mechatronics Technology, Oct 16, 2012,Tianjin,China pp – 506-509

[9] Q.Tang, J.Y.Wang, Z.Q.Zhu, "The simulation study of PSO based 3-D vehicle route planning for low attitude penetration, " Journal of System Simulation, vol. 16, no. 9, pp. 2033-2036, 2004

[10] Leon Zlajpah, "Robot Simulation for Control Design," Journal of intelligent and Robotic Systems, Vol No 32, 219-234, 2001.

[11] P.Lester. A* Path finding for Beginners: http://www.policyalmanac.org/games/aStarTutorial.htm

[12] Lynne E.Parker, "Path Planning and Motion coordination in Multiple Mobile Robot Teams," in Encyclopedia of Complexity and System Science, Springer,2009.

[13] B. Stout, " Smart Moves :Intelligent Pathfinding ", Game Developer , October 1996 www.gamasutra.com/features/19970801/pathfinding.html

[14] S. M. LaValle, Motion planning: The essentials. IEEE Robotics and Automation Society Magazine, 18(1):79--89, 2011

[15] M.BalaSubramanian, Dr.K.Sudhagar, G.RajaRajeswari, "A Study on seamless information sharing between robots in identifying the optimal path: An agent based approach", *Proc. of Int. Conf. on Advances in Communication, Network, and Computing, Elsevier, CNC,2014,INDIA*

[16] M.BalaSubramanian, Dr.K.Sudhagar, G.RajaRajeswari, "Intelligent Path Planning of Mobile Robot Agent by Using Breadth First Search Algorithm", *International Conference on Innovations in Engineering and Technology, (ICIET'14) , INDIA*

[17] http://softplanet.com/RoboSim

AUTHOR'S INFORMATION

[1]Research Scholar (Ph D), Department of Computer Science and Engineering, Bharath University, Chennai, INDIA

[2]Professor and Head, Department of Mechanical Engineering, Agni College of Technology, Chennai, INDIA.

[3]Assistant Professor, Department of Information Technology, Rrase College of Engineering, Chennai, INDIA.

M.Bala Subramanian received his Bachelor of Technology (B.Tech) Information Technology, from Anna University, India in 2005. He obtained his Master of Technology (M.Tech) degree from PRIST University, Thanjavur, INDIA in 2011. He is currently pursuing his Doctor of Philosophy in the Department of Computer Science and Engineering, Bharath University, Chennai. He is active member in various professional bodies like ISTE, ACEEE. His area of interest is in the field of Robotics science and engineering. He has published many papers in national and international conferences, various national and international Journals. He is currently working as System Development Specialist in VDSI, India and research scholar in Bharath University, Chennai, India.

Dr. K.Sudhagar received his Doctor of Philosophy from Anna University, Chennai, India in 2010. He is an active member in various professional bodies like MIE (I), MISTE, MIIE and MIIPE. His interest is in the field of Alternate fuels, Robotics Science and Engineering, and Artificial Intelligent Engineering. He has Published more than 30 Publication that appears in International and National Journals. He is currently working as Professor and Head in Department of Mechanical Engineering, Agni college of Technology, Chennai, India.

G.Rajarajeswari, received his Masters of Science from Madras University, Chennai and Masters of Technology (M.Tech) Computer Science and Engineering, PRIST University, Thanjavur, INDIA 2011. She is active member in various professional bodies like ISTE. Her area of interest is in Data mining, Operating System, Robotic Science and Engineering. She has published many papers in various reputed journals and conferences. She is currently working as Assistant Professor in Department of Information Technology, Rrase College of Engineering, Chennai, India.