The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Goal Seeking for Robots in Unknown Environments

V.R. Jisha and Debasish Ghose

Abstract— **We consider the problem of goal seeking by robots in unknown environments. We present a frontier based algorithm for finding a route to a goal in a fully unknown environment, where information about the goal region (GR), the region where the goal is most likely to be located, is available. Our algorithm efficiently chooses the best candidate frontier cell, which is on the boundary between explored space and unexplored space, having the maximum "goal seeking index", to reach the goal in minimal number of moves. Modification of the algorithm is also proposed to further reduce the number of moves toward the goal. The algorithm has been tested extensively in simulation runs and results demonstrate that the algorithm effectively directs the robot to the goal and completes the search task in minimal number of moves in bounded as well as unbounded environments. The algorithm is shown to perform as well as a state of the art agent centered search algorithm RTAA*, in cluttered environments if exact location of the goal is known at the beginning of the mission and is shown to perform better in uncluttered environments.**

I. Introduction

For search and rescue missions in hazardous environments or in a situation where nuclear leakage or forest fire occurs, the exact location of the disaster will not be known *a priori*. However, an approximate idea about the area in which the disaster might be located is likely to be known from other information sources. This is a practical scenario and is different from when there is no information about the location of the target point. The robot or the search and rescue agent starts from an initial location and reaches the location of interest through a completely unknown terrain. Thus, the problem is to find a route from a start position to a goal region (GR), where the goal is most likely to be located, through an unknown intervening area which is cluttered with unknown obstacles. The robot can only sense the surrounding area within the range of its sensors.

When the robot starts navigating in an unknown terrain, it knows only what it sees from where it is situated. Yamauchi [1] introduced the concept of frontiers and defined them to be the boundary between the open space and unexplored space. The main idea is that, to gain the most new information about the world, the robot should move to one of the frontier cells.

Most methods of path planning in completely unknown environment use cell decomposition methods, which partition the world into grids, for the representation of search space. Grid based heuristic search algorithm like A* [3] searches all possible routes from a starting point until it finds the shortest path to a goal. Some grid based approaches plan an initial path using all known information, making assumptions about those parts of the environment that are unknown [4]. As the robot acquires new information about the environment, the assumptions are updated with correct information, and the path is replanned [5]. In [6] the algorithm plans the optimal path to goal using all known information and replans from the current state whenever new or conflicting information becomes available. Real time search methods in [7] and [8] are closest to our method in terms of the framework used.

In all the methods mentioned above for goal seeking, the exact goal location needs to be known at the start of the search task. These methods do not reduce multiple traversals through the same cell. We introduce a frontier based algorithm for successfully reaching the goal, given the probable location of the GR. Information about the probable location of the GR can be given to the robot at any point of time from the beginning of the mission. Till date the concept of frontiers have been successfully implemented for autonomous exploration in unknown environments [1], [2]. Our algorithm focusses on a proper choice of the frontier cell which ultimately reduces the number of moves to reach the goal. As this algorithm is frontier based, route planning in unknown environments ensures reducing multiple traversals, that is, the robot will avoid going back again to an already explored region. In a previous paper we have presented some preliminary work based on this idea [9]. The present paper contains improved versions of the algorithm with more detailed comparison results.

II. Problem Description

A robot, equipped with sensing and localization capabilities, starts searching for a goal in an unknown environment. The only information available to the robot is the goal region (GR), where a goal is likely to be present. But the exact location of the goal is not known *a priori*. The objective is to find a route from the start to the GR. Once the robot reaches the GR exploration of that region is done, till the goal is found. If at the start of the mission, the information about the location of GR is not available, the robot will start its search in exploration mode and when GR information becomes available it will switch to goal seeking mode. Our approach uses occupancy grid maps [10] to represent the environment. The underlying framework used is to partition the terrain into identical hexagonal cells and to store in each

V.R. Jisha is a Lecturer in the Department of Electrical Engineering, College of Engineering, Trivandrum, Kerala, 695 016, India and is presently pursuing doctoral studies at the Guidance, Control, and Decision Systems Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, 560012, India. jisha@aero.iisc.ernet.in

Debasish Ghose is a Professor in the Guidance, Control, and Decision Systems Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, 560012, India. dghose@aero.iisc.ernet.in

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

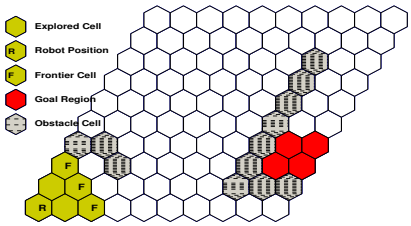

Fig. 1. A 10×10 grid with robot position, frontier cells and goal region

cell the probability Po_{xy} which denotes the probability that the cell is occupied by an obstacle.

Given the discrete terrain model, we also discretize the robot motion. More specifically it can move to any of the six neighbours. It is assumed to have a radial sensor of two cell depth, that is, all the cells within two cell sensor radius from the current robot position are updated (if unobstructed by obstacles). Given a sequence of sensor measurements, corresponding positions of robot and the prior probability of occupancy, the posterior probability Po_{xy} can be found [10].

Initially, all the cells are set to the prior probability of occupancy, 0.5. While the robot is moving through the environment, cell values are updated [11] as soon as it has been intercepted by a sensor beam. An occupancy value near zero corresponds to a free cell. An occupancy value near one indicates that the cell is occupied. So all the unexplored cells will have occupancy probability equal to prior probability. The robot is assumed to have perfect localization capabilities.

III. Goal Directed Route Planning Strategy

A. Basic Goal Seeking (BGS) Algorithm

We assume that the possible location of GR is available to the robot at the beginning of the mission and there are no obstacles in GR. The Basic Goal Seeking (BGS) algorithm tries to find a series of intermediate target points for the robot from the starting position till the goal is reached. By performing a 360^0 scan at these points the robot can update the cells within the sensor range, with Po_{xy} calculated from the sensor measurements. These group of cells are referred to as explored cells and the collection of explored cells is known to the robot at that moment. From this set of explored cells, frontier cells which are on the boundary between explored and unexplored cells are identified. A frontier cell is an already explored cell which is an immediate neighbor of an unexplored cell (See Fig. 1). BGS chooses the best fit frontier cell, that is, the cell with maximum "goal seeking index", among these and a route is found from the current location to the frontier cell at each iteration.

In the following subsections we describe how to compute the cost of reaching a frontier cell from the current robot position, how the "goal seeking index" is calculated and how the next target positions for the robot is chosen.

1) Evaluating Cost: We have to find out the cost of reaching the current frontier cells to compute the optimal path from the current position of the robot to all the frontier cells. Consider a cell (x,y), which represents the x^{th} cell in the horizontal direction and y^{th} cell in the upward direction in the hexagonal occupancy grid map. The cost for traversing a cell (x,y) is proportional to its occupancy value Po_{xy}. It is computed using the algorithm in [2], modified for the hexagonal discretization instead of the square grid pattern used in [2]. The algorithm is as follows:

1) The grid cell that contains the robot is initialized with a value 0 and all others with ∞.
2) For determining the cost of reaching the current frontier cells, update all cells (x,y) as

$$C_{xy} = \min\{C_{x+\Delta x,y+\Delta y} + Po_{xy}\}$$

where, $C_{x+\Delta x,y+\Delta y}$ is the cost associated with immediate neighbours, Po_{xy} is the probability that the cell is occupied by an obstacle ($Po_{xy} \in [0, Occ_{max}]$) and Occ_{max} is the maximum occupancy probability value of a grid cell the robot is allowed to traverse. The second step is repeated until convergence. Then each C_{xy} represents the cumulative cost of reaching from the current position of the robot to (x,y). From this we can find a minimum cost path from the robot position to frontier cells, by steepest descent in cost value starting at (x,y).

2) Evaluating "Goal Seeking Index": The main task of the BGS algorithm is to select a frontier cell such that the robot will be able to reach the GR in minimal number of moves. For this we need to find a "Goal Seeking Index", denoted by G_s for each frontier cell which is calculated as follows:

Goal Region GR is a collection of cells where the goal is likely to be present. Once we know the GR, it is possible to choose a representative cell, from GR. This cell could be the centroid cell of GR. We denote that cell as CG. Let G_D be a measure which gives an indication of distance between current frontier cell and CG. As the distance between the frontier cell and CG decreases, the value of G_D increases.

We use a variable g to make the robot stay and search inside GR, once it arrives there. The only information available is that the goal is present in GR. So, once the robot reaches GR, it is necessary to search GR till the goal is found. So g is assigned values as : $g = 0$, $\forall$ frontier cells $\notin$ GR and $g > 0$, $\forall$ frontier cells $\in$ GR.

Suppose at a particular step, the robot is inside GR and there are two candidate frontier cells available, one inside and one outside GR. Because $g > 0$ for frontier cells inside GR, the one inside GR will be chosen.

The "Goal Seeking Index" for each frontier cell is calculated using (1) where K_1, K_2 are weighting constants and C is the cost of reaching the frontier cell from current position.

$$G_s = K_1 G_D - K_2 C + g \tag{1}$$

To determine appropriate target points for the robot, Goal Seeking Index G_s, of the current frontier cells are calculated. K_1, K_2 are nonnegative weighting constants and by changing the weights the importance of each component is varied. $K_1 > 0 \Longrightarrow$ goal seeking behaviour and $K_1 = 0 \Longrightarrow$ exploratory behaviour of the algorithm. (i) $K_1 > 0$, $\forall$ frontier

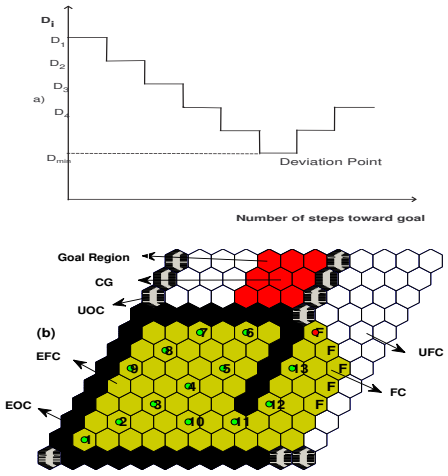

Fig. 2. (a) Distance D between robot and CG of GR as the robot moves toward the goal and deviation point is the move at which obstacle cells are encountered, preventing motion directly toward the goal (b) D decreases from position 1 to 6 and shows an increasing trend from there onwards (**EOC**-Explored obstacle cell, **UOC**-Unexplored obstacle cell, **EFC**-Explored free cell, **UFC**-Unexplored free cell, **FC**-Frontier cell, **GR**-Goal Region)

cells $\notin$ GR (ii) $K_1 = 0$, $\forall$ frontier cells $\in$ GR (iii) $K_2 > 0$, $\forall$ frontier cells.

Before reaching GR, the value of $g = 0$, and hence the selection of frontier cell is based on values of C and G_D only, resulting in directed exploration toward GR. Once the robot reaches GR, K_1 is made equal to zero, which results in simple exploratory behaviour of the algorithm in that region. In simple exploration mode, the minimum cost frontier cell from current position is chosen. Once the robot reaches GR, the search will be confined only to that region till a goal is reached. This is because $g > 0$ in GR and hence more weightage will be given to the frontier cells in GR.

3) The BGS Algorithm:

1) While the goal is not reached, identify all frontier cells that are within the current sensing region. If no such cell exists, go to step 4.
2) Determine the cost C to reach the current frontier cells.
3) Choose a frontier cell with maximum "Goal Seeking Index" G_s. If multiple cells exist then choose any one.
4) If no such frontier cells exist within the current sensing region, then pick the closest frontier cell outside the current sensing region as the next target.
5) If no frontier cells exist then all accessible area has been covered and the goal is not present in the area.

Even though goal seeking in an unknown environment cannot claim optimality, the total distance traveled to reach the goal is reduced considerably by this algorithm over the time taken by an exploratory search. Our algorithm efficiently chooses the best candidate frontier cell, in an attempt to find a route from start to goal in minimal number of steps. The algorithm tries to search for a goal by doing directed exploration toward the goal.

If there are no obstacles on its way, the robot will be making a move toward GR. So, as time passes the distance between the present robot position and CG of GR decreases, until it reaches the GR. The algorithm performs well if small obstacles are scattered in the environment. But if the environment contains walls or wall like obstacles (obstacles kept in orderly manner such that wall like obstacle is created), the performance of the algorithm may deteriorate. We refer to the distance between robot position and CG as D. But as it encounters an obstacle, D will start increasing as shown in Fig. 2(a). The minimum distance achieved in this process is referred to as D_{min}. This is illustrated in Fig. 2(b). The robot starts from position 1 and the distance D decreases till it reaches position 6. From there onwards D keeps increasing because the robot encountered a series of obstacle cells like a wall at that point. The distance D at position 6 is referred to as D_{min}. In such a situation, more moves are required to reach the goal and hence the total route length from start to goal will increase. Hence the total time for reaching the goal will also increase. In order to account for this situation, a modified algorithm is proposed below.

B. Modified Goal Seeking (MGS) Algorithm

The modification in the BGS is mainly in the form of choosing a frontier cell more effectively.

1) While the goal is not reached, identify all frontier cells that are within the current sensing region. If no such cell exists, go to step 6.
2) Determine the cost to reach the current frontier cells.
3) Determine the distance D from the current robot position to the CG. If $D > D_{min}$, go to step 5.
4) Choose the frontier cell such that "Goal Seeking Index" is maximum (The chosen frontier cell may or may not be near an obstacle cell). Go to step 6.
5) Choose the frontier cell near to the obstacle such that "Goal Seeking Index" is maximum (The chosen frontier cell must be near an obstacle cell).
6) If no frontier cell exists within the current sensing region, then pick the frontier cell outside the current sensing region and near the obstacle. If multiple cells exist then choose the one nearest to the CG.
7) If no frontier cells exist then all accessible area has been covered and the goal is not present in the area.

The condition at step 6, that is, the situation with no frontier cell in the current sensing region, is referred to as a "trap" situation. In such a situation a frontier cell has to be chosen outside the current sensing region.

Fig. 3a illustrates the robot in a "trap" situation. At position 15, there are no frontier cells in the current sensing region and the frontier cells at this position are f1, f2, ...f9 which are outside the current sensing region. There are three frontier cells f1, f2 and f9 near the obstacle and the one near the CG is to be chosen according to step 6. Hence the robot chooses the frontier cell f9. The path to the frontier cell outside the current sensing region is also found in the same way as described in the previous section. Unlike BGS, MGS chooses frontier cell near the obstacle and to the CG in a trap situation. Choosing frontier cell near the obstacle will drive the robot to the exit of the trap, and the one near

CG will direct the robot toward the CG. This will reduce the unnecessary exploration around the robot position and robot will reach a frontier cell near CG.

IV. SIMULATIONS

All experiments are performed in 25×25 cell environments. The start position of the robot is the same in all experiments. The location of the GR is chosen randomly. The coordinates of the goal and obstacles are fixed. The robot knows its own and GR's coordinates. A total of 50 such environments, both cluttered (in which small obstacles are randomly placed and wall like obstacles are not present) as well as uncluttered (in which wall like obstacles are present), are simulated with obstacles of the same type, but the obstacles were placed at different position and orientations randomly in each environment. Note that both cluttered and uncluttered environments are unknown. The parameters used for simulations are: K_1=1; K_2=1; G_D=N-D_{fcg} (N should be a large positive number, which is the largest possible distance between any cell in the grid and CG in the environments used for simulations (N=25) and D_{fcg} is the distance in the hexagonal grid between the frontier cell and the CG); g=25, $\forall$ frontier cells $\in$ GR and Occ_{max}=0.1.

A. Basic Goal Seeking

1) When GR information is available to the robot at the beginning of the mission: Simulations are done for cluttered as well as uncluttered environments, as shown in Fig. 3b and 3c. The robot is assumed to have two cell depth radial view sensor. The robot start position in all the cases is assumed to be extreme left corner of the unknown area. The series of frontier cells, chosen from the start position 1 to the final step when robot reaches the goal, is shown in the Fig. 3b and 3c. From the extensive simulations it is seen that the BGS algorithm performs well and is reaching the goal successfully. Goal reachability is assured, because in the worst case exploration of the entire region including GR is done, that is, till frontier cells are exhausted. Hence robot will reach the goal if it is present in the GR.

2) When GR information is made available to the robot at some intermediate time during the mission: For all experiments discussed previously, it is assumed that the information about the probable location of the GR and boundary limits are available to the robot at the beginning of the mission. If only the information of boundary limits are available, the robot will start its search in an exploratory mode ($K_1 = 0$ in (1)) and it can switch to goal seeking mode when information becomes available. This situation is illustrated in Fig. 4a. In this case the GR information becomes available at position 20 and the algorithm switches from exploratory mode to BGS by making $K_1 > 0$. The case when GR information is not available to the robot at all, is illustrated in Fig. 4b. In this case also goal reachability is assured because, in the worst case exploration of the entire region is done, that is, till frontier cells are exhausted.

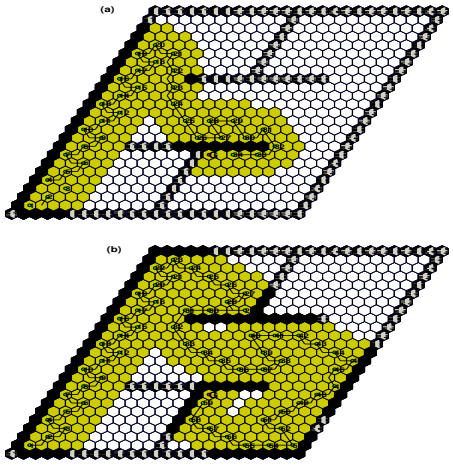

Fig. 4. Searching for a goal via (a) exploration and BGS (GR information becomes available to the robot only at position 20 and switches from exploratory mode to BGS at that position) (b) simple exploration, in one of the sample environments

3) BGS performance when boundary is not specified: In order to evaluate the performance of BGS, if only GR information is available to the robot at the start of the mission and boundary is not specified, simulations are done in larger grids as shown in Fig. 5a. The robot successfully reaches the goal without deviating from the search area because of the directed motion toward the goal.

As mentioned previously, D, the distance between robot and CG, monotonically decreases till an obstacle is encountered. Depending on the type of obstacle, the change in D may or may not be monotonic. It will show a worst case behaviour (D mononotonically increasing) only when a series of obstacle cells like a wall is encountered. In this situation also goal reachability is assured, because, there exist unexplored cells near the GR because the GR is not reached; there exist explored cells near GR but are separated by the wall; hence in the boundary between explored and unexplored cells there exist frontier cells (away from GR as well as near GR). Even though the robot chooses frontier cells away from the GR, at some point in time the robot will choose the ones near GR according to (1). And hence the robot will ultimately reach GR after zigzag motion around GR (unless the wall is of infinite length).

B. Modified Goal Seeking

Simulations are also done in similar environments using MGS. It is found that the total number of moves to reach the goal is considerably reduced if, in a "trap" situation, a frontier cell is chosen according to Step 6 of MGS algorithm. It also avoids zigzag motion when a wall like obstacle is encountered. Hence, the total route length from start to goal is considerably reduced compared to BGS and the time taken to reach the goal is also less. But in cluttered environments, where wall like obstacle are not present, the performance of MGS will be the same as that of BGS. Fig. 5c and 5d shows a comparison between BGS and MGS algorithms with the above mentioned strategy in one of the uncluttered environments. Performance is also evaluated for MGS, when

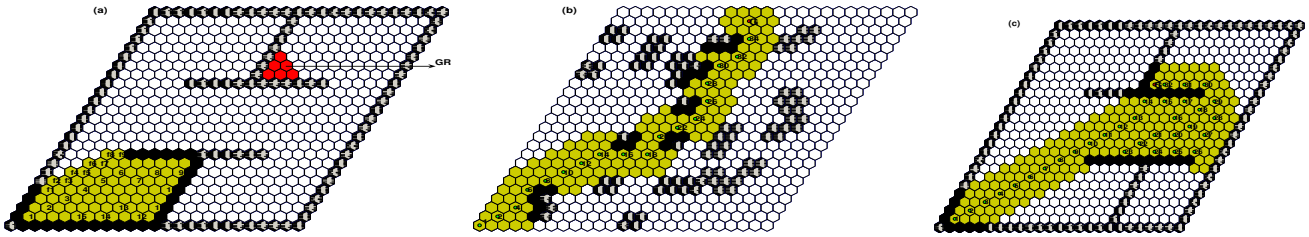

Fig. 3. (a)The robot in "trap" situation at position 15 and the frontiercells f1-f9 are outside the current sensing region (b) Illustration of BGS in one of the cluttered environments (c) Illustration of BGS in one of the uncluttered environments

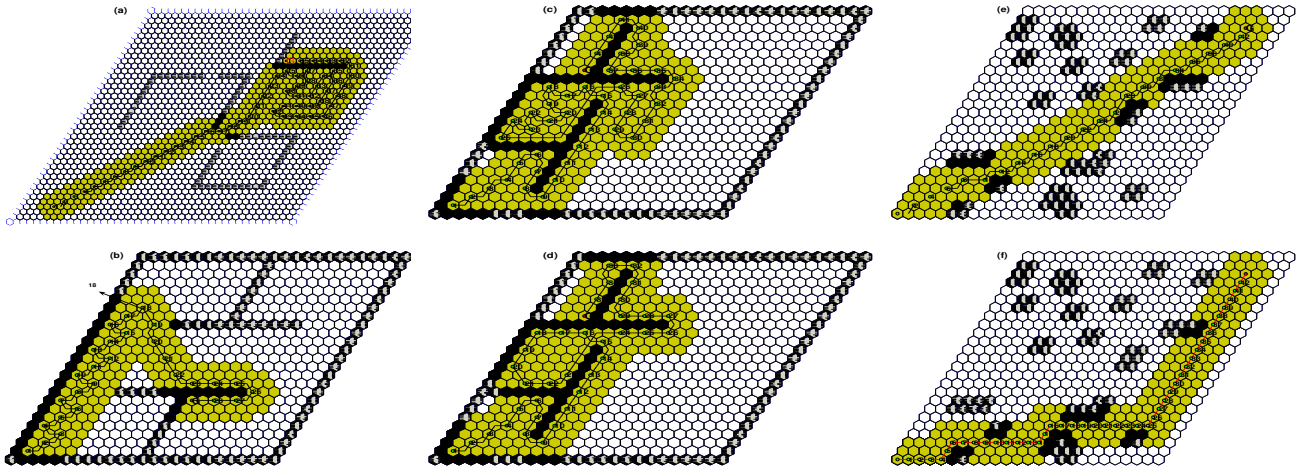

Fig. 5. (a)BGS performance in an environment without specifying the boundary (b)Exploration and MGS (GR information becomes available only at position 18 and switches from exploratory mode to MGS at that position)(c) BGS in one of the sample environment requires 97 steps to reach the goal (d) MGS requires only 75 steps to reach the goal (e) BGS and MGS in one of the cluttered environments (f) RTAA* with LA=4 in the same environment)

GR information is made available to the robot at some intermediate time during the mission (Fig.5b) and when performance boundary is not specified. In this case also goal reachability is assured as walls are not of infinite length.

C. Competitiveness of the algorithm

One of the main requirements of the navigation strategy in unknown environments has been its competitiveness. A strategy for working with incomplete information is called competitive if it solves each problem instance at cost not exceeding the cost of an optimal solution (with full information available), times a constant [12]. The algorithms were tested again in the similar 25×25 environments to evaluate the competitive factor. The optimal path for all the environments are found under the assumption that the environment is known *a priori*. The results are shown in Fig.7.

D. Comparison with RTAA*

Simulations are done in similar 25×25 environments to compare the performance (in terms of the number of steps to reach the goal) of BGS and MGS with one of the state of the art real time heuristic search algorithm, RTAA* [8]. This real time heuristic search method is able to choose its local search space in a fine grained way and it updates the heuristic values of all states in the local search spaces. For comparison purpose, GR in BGS and MGS algorithms are shrunk to the exact goal location and is made available to the robot at the beginning of the mission.

The comparison in one of the cluttered environments is shown in Fig. 5e-f. In this case the number of steps required to reach the goal, for all the algorithms is the same. In almost all experiments in cluttered environments, BGS and MGS perform as efficiently as RTAA*. But in uncluttered environments, because of the presence of wall like obstacles, RTAA* performance is not as good as in cluttered environments. It can be seen from Fig.6 that the number of steps required by BGS, to reach the goal is 62, and that for MGS is 48, while RTAA* takes 87 steps. In the 25 uncluttered environments considered for simulation, RTAA* takes more steps to reach the goal than BGS, when wall like obstacles are present. Performance of MGS is even better in a majority of the uncluttered environments.

V. Results and Discussion

Table I gives the performances of the algorithms in cluttered environments. It can be seen that BGS and MGS performs as well as RTAA*.

Table II gives the comparison of the algorithms in uncluttered environments. Simple exploration is also done in these environments to know the worst case behavior of the algorithm without the information of GR known *a priori*. Simple exploration strategy takes more number of moves to

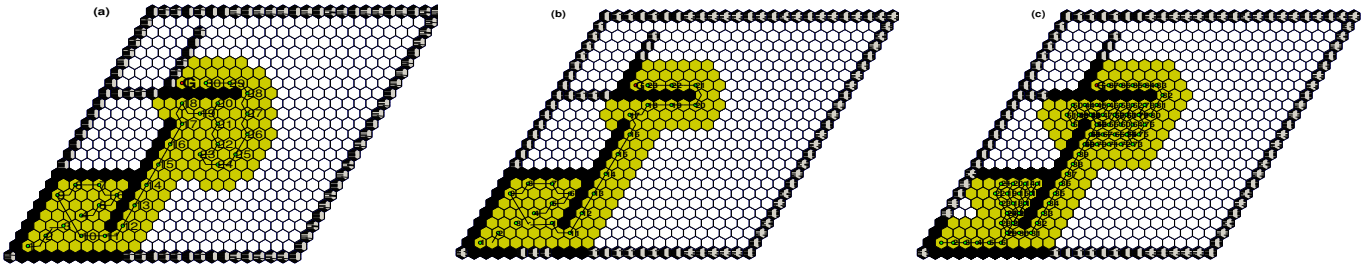

Fig. 6. Performance of (a) BGS in one of the uncluttered environments (b) MGS in the same environment (c) RTAA* with LA=4 in the same environment

TABLE I

PERFORMANCE COMPARISON: BGS, MGS , RTAA* WITH LOOK-AHEAD 4 & OPTIMAL PATH ALGORITHMS IN CLUTTERED ENVIRONMENTS

Number of moves to reach the goal	Number of Experiments		
	BGS, MGS	RTAA-LA4	Optimal
≤30	4	4	4
≤40	15	15	18
≤ 50	25	25	25

TABLE II

PERFORMANCE COMPARISON: EXPLORATION, BGS, MGS, RTAA* (LA 2,4 & 6) & OPTIMAL ALGORITHMS IN UNCLUTTERED ENVIRONMENTS

No. of moves	Number of Experiments						
	EXP	BGS	MGS	RTAA*			Opt
				LA (2)	LA (4)	LA (6)	
≤50	1	5	9	4	4	4	25
≤100	8	15	18	8	10	11	-
≤ 150	13	20	25	11	17	16	-
≤ 200	18	23	-	14	21	20	-
≤ 250	24	24	-	17	23	23	-
≤300	25	25	-	19	24	24	-
≤400	-	-	-	23	25	25	-
≤500	-	-	-	25	-	-	-

reach the goal and hence the route length from start to goal is more in a majority of the simulation environments. BGS performs better compared to simple exploration strategy, because of the directed exploration toward the goal till obstacle is found. Due to local minima in these type of uncluttered environments, RTAA* algorithms do not perform well, as in cluttered environments.

The performance of BGS in uncluttered environments is slightly better than RTAA* with different look-ahead because it is capable of escaping faster from local minima present in these type of environments. In RTAA* the heuristics update of the corresponding states are done till it comes out from local minima. But BGS will avoid going to the already explored cells in these situations and will be able to come out in a better way in the type of environments where local minima are occurring. MGS performs even better than BGS in uncluttered environments because of proper choice of frontier cells when a wall is encountered and also in trap situations. The performance comparison between BGS, MGS and RTAA* with different look-ahead in terms of competitiveness is illustrated in Fig.7 and competitive ratio is comparatively less for MGS, than for BGS and RTAA* .

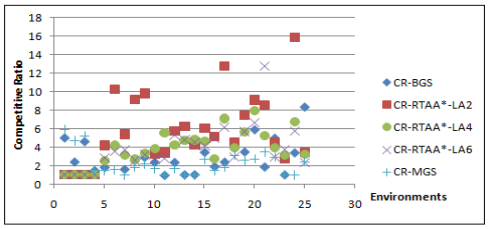

Fig. 7. Competitive Ratio Comparison of BGS and MGS with RTAA* of look-ahead 2,4 and 6 in uncluttered environments

VI. CONCLUSIONS

In this paper we presented an algorithm for goal seeking and exploration in an unknown environment. The algorithm was tested in extensive simulation runs. They demonstrate that the algorithm is able to seek for a goal in unknown environments with minimal moves from start to goal. Comparison with standard RTAA* algorithm shows the proposed algorithm performs better in several ways.

REFERENCES

[1] B. Yamauchi, Frontier based approach for autonoumous exploration, *In Proc. IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, July 1997, pp 146-151.

[2] W. Burgard, M. Moors, C. Stachniss, Coordinated multi-mobot exploration, *IEEE Trans. on Robotics*, Vol.21, No.3, 2005, pp 376-386.

[3] P. Hart, N. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp 100-107.

[4] S. Koenig, Y. Smirnov, Sensor based planning with freespace assumption, *Proc. of the IEEE Int. Conference on Robotics and Automation*, 1996, pp 3540-3545.

[5] S.Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja , K. Schwehr, Recent progress in local and global traversability for planetary rovers, *Proc. of the IEEE Int. Conference on Robotics and Automation*, April 2000, Vol. 2, pp 1194-1200.

[6] A. Stenz, Map based strategies for robot navigation in unknown environments, *Proc. of AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, March 1996, pp 110-116.

[7] R. E. Korf, Real time heuristic search, *Artificial Intelligence*, 1990,pp 189-211.

[8] S. Koenig, M. Likhachev, Real time adaptive A*, *Proc. of Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Hakodate, Japan, May 2006, pp 281-288.

[9] V.R. Jisha, D. Ghose, Goal directed route planning for robots in unknown environments, *Proc. of the Int. Conference and Exhibition on Aerospace Engineering*, Bangalore, India, May 2009, pp 1309-1318.

[10] H. Moravec, A. Elfes, High resolution maps from wide angle sonar, *Proc.of IEEE Int. Conference on Robotics and Automation*, Durham, England, March 1985, Vol. 2, pp 116-121.

[11] R. R. Murphy, Introduction to AI Robotics, Prentice Hall, India, 2005, pp 376-434.

[12] C. Icking, R. Klein, Competitive strategies for autonomous systems, *Modelling and Planning for Sensor Based Intelligent Robot Systems*, World Scientific, 1995, Vol. 21, pp 23-40.