International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 03 - Issue 01 || January 2017 || PP. 21-27

Application of A* and GA for Path Planning of Mobile Robots: A Comparative Study

Deepanjali Mundari[1] , Madhusmita Panda[2]
[1]*(Electronics & Telecommunication, VSSUT ,Burla, India)*
[2]*(, Electronics & Telecommunication, VSSUT ,Burla, India)*

Abstract: In this paper there is a comparative study of different algorithms for path planning of a mobile robot to successfully reach a target in a known environment. The algorithms are A*Algorithm and Genetic Algorithm (GA) respectively which allows a mobile robot to navigate through static obstacles and find the path in order to reach target without collision. Here the mobile robot concerned is an AUV. The above strategy is designed in a 2-D grid map form of known environment and static obstacles. When the mission is executed it is necessary to plan an optimal or feasible path for itself avoiding obstructions on its way and minimise a cost such as computational time and path distance. The algorithms are implemented in Matlab and the results are compared.
Keywords: Path Planning, A*Algorithm, Genetic Algorithm, Mobile robot, AUV

I. Introduction

Path planning is an essential component for navigation of mobile robotic vehicles. Autonomous mobile robots widely used in civilian and military applications. Finding a collision-free path between a starting and target locations is the main aim of path planning. The planned path is the optimum path between the source and target. The planned path is usually decomposed into line segments between ordered sub-goals or way points. In the navigation phase, the robot follows those line segments toward the target. The navigation environment is usually represented as configuration space. Depending on the surrounding environment and the running conditions, the optimality criterion for the path is determined. For example, in most of indoor navigation environments, the optimum path is the safest one. Safest path considers the path which is as away from the surrounding obstacles and avoids collision. Outdoor navigation recommends the shortest path more. Path planning is a very crucial task as the planned path should be a complete, smooth, feasible, obstacle free and most importantly should have minimum cost in terms of path length and computational time. In our study both the algorithms are verified considering autonomous underwater vehicle (AUV) as a mobile robot.

II. Path Planning

Many methods were proposed in the literature to address the path planning problem. Potential field methods use the physics of electrical potentials as a heuristic to guide the search for a path [1]. Movements of the robot are governed by a potential field which is usually comprised of two components, an attractive potential drawing the robot towards the goal and a repulsive potential pushing the robot away from obstacles. The main drawback with these methods is their susceptibility to local minima. The idea behind roadmap approaches is to reduce the map to a network of one dimensional curves. If start and goal points are linked to this network, then path planning becomes a graph searching problem. The key issue is the method used to construct the roadmap. A visibility graph is constructed by considering all the vertices of the obstacles and the start and goal points as the graph nodes [2]. The graph links are the line segments which connect two nodes without intersecting any obstacle. This method becomes complex in more than two dimensions. A Voronoi diagram is the set of points that are equidistant from two or more obstacles [3]. The result is a roadmap where the edges stay away from the obstacles. This method was extended for path planning in arbitrary dimensions. It has not been widely used as a practical solution but has been a tool for analysing the complexity of motion planning. Probabilistic Path Planning (PPP) is a general planning scheme. Here a local operator is used to locally link positions in a feasible way for the robot avoiding obstacles and respecting kinematic constraints for example[7]. PPP is probabilistically complete and allows one to deal with high dimensional configuration spaces. Nonetheless this method may fail to find a solution when the environment presents singularities such as the 'narrow passage' problem and it is heavy to implement for real-time applications in dynamic environments. Then Classical grid-search algorithms were introduced which assume that the environment is sampled on a uniform grid. The key issue is then to use a suitable search algorithm to find an optimal path for a particular criterion, usually defined by a metric. The metric defines the distance to be the cost-to-go for a specific robot moving in an environment described in the cost function. The Grid-search algorithms are very popular in Artificial Intelligence (AI) for

planning paths. Once the distance map is built until the goal point the optimal path is the one which follows the steepest descent from the goal to the start point. It is equivalent to say that we solve a functional minimization problem. The overall method is robust as no local minima are exhibited during the exploration process. Some of the grid search algorithms are: -BFS (Breadth-first search) [4]. It explores the grid points in order of their distance from the start point. DFS (Depth-first search) [5] is very similar to BFS. DFS method tends to focus the search directly to the goal point instead of developing a front around the start point. One of the hybrid search is the A* algorithm , which is probably the most popular hybrid search algorithm in artificial intelligence. It combines features of BFS and DFS to efficiently compute acceptable solutions. A major advantage of the A* algorithm compared to the other methods is that A* is guaranteed to give the optimum path. The above algorithm can be categorised as deterministic approaches. There are probabilistic approaches other than deterministic ones.
Genetic Algorithm (GA) method is one of the probabilistic search technique based on the principles of biological evolution, natural selection, and genetic recombination. GA generates a population of solutions. And such solutions give offspring solutions in the next generation. The solutions in the population improve over many generations until the best solution is obtained. However, GA have been accepted slowly for research problems because crossing two feasible solutions does not, in many cases, result in a feasible solution as an offspring. The other disadvantage of GA is that although it can generate solutions to a path planning problem, it cannot guarantee that the solution is optimal, i.e. it can converge to a local, rather than a global, minimum [6].
In the previous methods, there were many drawbacks like the path obtained were not feasible, limited to small maps, converges to the local minima, so they fail to give optimum path. They are not used practically. However, deterministic method like A*Algorithm and probabilistic method like Genetic Algorithm were able to give optimum path.

III. A* ALGORITHM

A* Algorithm evaluates iteratively the moving cost from the current cell to one of its neighbours through a defined cost function. This function f is obtained summing of 2 terms:

$$f=g+h \qquad (1)$$

(i) *g*-This parameter is proportional to actual distance from current cell to the evaluated cell.
(ii) *h*-This parameter is proportional to heuristic estimate distance from evaluated cell to the goal.

g value is zero from the starting cell, it increases while the algorithm expands successive cells(i.e. at each step the algorithm sum the moving cost from the starting cell to the current one, the distance from the current cell to one of its neighbours).To enforce convergence ,the *h* value has to be admissible and *h* function has to be consistence. The value of the cell has not to be overestimate the evaluated distance from goal [9].
Defining closed list and open list in graph search algorithm is that the open list collects the nodes expanded along the graph search. At each step the algorithm evaluates the nodes surrounding the current node putting them into open list and sorting the list with respect to the cost function value. The first element of the sorted list is moved in the closed list. This contains the best neighbour of the node expanded at each step. This list contains the best neighbour of the node expanded at each step. These nodes are removed from the open list and never evaluated again and the path is build with nodes coming from the list.
The following pseudo-code describes how the 2-D A* algorithm works .

```
1) Put the start node on the OPEN list(O) and calculate the cost function f (n). {h (n) = 0, g(n) is the distance between the goal and the start position, {f(n) = g(n)}.
2) Remove from the OPEN list the node with the smallest function $f$ cost where $f=g+h$ and put it on the CLOSED list(C). This is the node n.
3) If n is the goal node then terminate the algorithm and use the pointers to obtain the solution path. Otherwise, continue.
4) Determine all the successor nodes of n and compute the cost function for each successor not on CLOSED list.
5) Associate with each successor not on OPEN or CLOSED list the cost calculated and put these on the OPEN list, placing pointers to n (n is the parent node).
6)Associate with any successors already on OPEN the smaller of the cost values just calculated and the previous cost value.
7) Go to step 2.
```

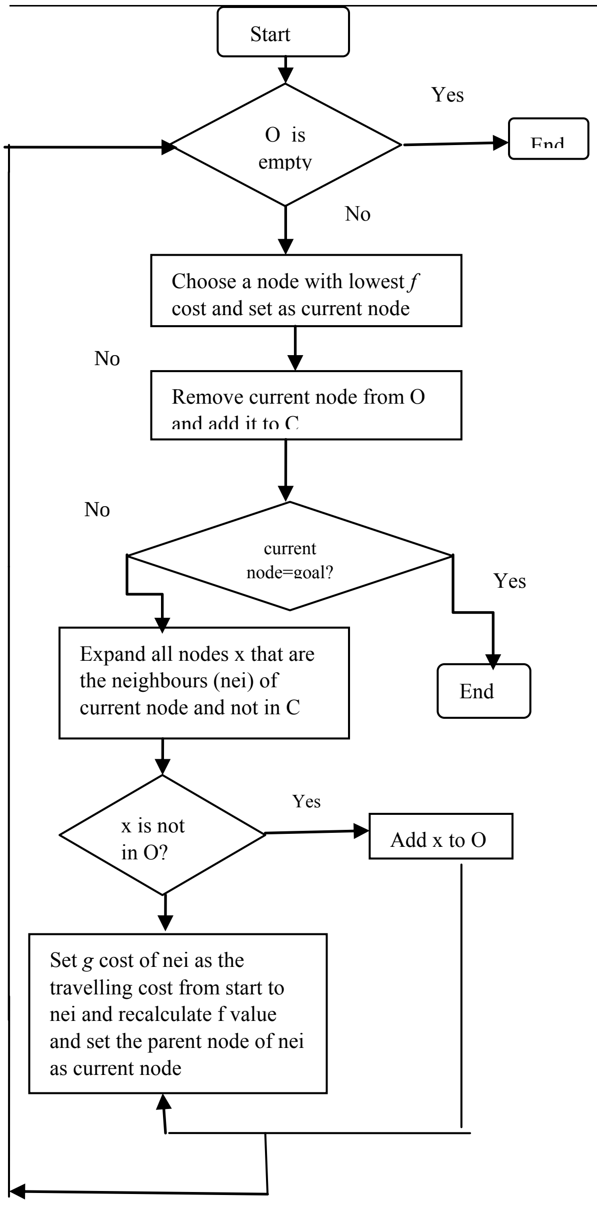
Start
O is empty
Yes
End
No
Choose a node with lowest *f* cost and set as current node
No
Remove current node from O and add it to C
No
current node=goal?
Yes
End
Expand all nodes x that are the neighbours (nei) of current node and not in C
Yes
x is not in O?
Add x to O
Set *g* cost of nei as the travelling cost from start to nei and recalculate f value and set the parent node of nei as current node

Fig. 1 Flow diagram of A* Algorithm to find optimal path

IV. GENETIC ALGORITHM

Genetic Algorithm (GA) introduced by John Holland in 1975 is an evolutionary method that takes advantage from operators such as natural selection, crossover, and mutation. GA is successfully applied to problems such as the classical travelling salesman problem, flow-shop optimization, and job shop scheduling in which the aim is to either optimize or find the best solution out of a number of possibilities. The inherently parallel search characteristic of the GA makes it attractive for developing near-optimal solutions.

The following steps describes how Genetic Algorithm works [9].
1) Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. The population is generated randomly, covering the entire range of possible solutions i.e. the search space.
2) Selection is the operation of ranking chromosomes based on their fitness and sub-selecting high ranked chromosomes for generating new population. Here, chromosomes represent shorter or smoother paths, whereas fitness refers to chromosomes distances from goal.

3)Crossover is the operation used for generating new population from the sub-selected chromosomes by selection operator. Typically, the crossover operator divides the sub-selected chromosomes into two parts and exchanges their parts with each other in order to generate the new population.
4)Mutation is the operation in which one or a group of chromosomes are chosen to be fully or partially randomized. It is typical to use mutation operator whenever the population is converged toward local optimum, trapped between obstacles, or the performance is not improved for certain number of generations due to the lack of genetic diversity.
5) This process is repeated until a termination condition has been reached. Common terminating conditions are:
I. when a solution is found that satisfies minimum criteria
ii. fixed number of generations is reached
iii. allocated budget (computation time/money) is reached
iv. the highest ranking solution's fitness is reached.

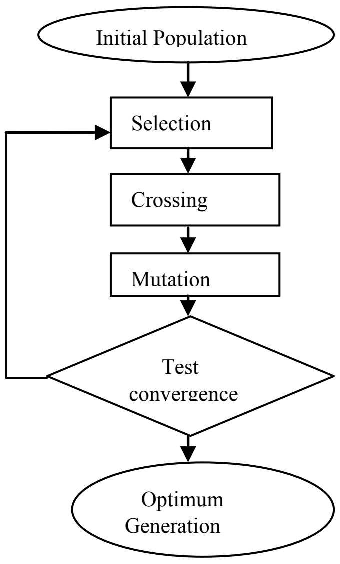

Fig. 2 Flow diagram of Genetic Algorithm to find optimal path

V. PROPOSED METHODOLOGY

Here we tried to present a comparative study of A*Algorithm and of Genetic Algorithm applied in the field of AUV path planning respectively which allows the AUV to navigate through static obstacles and find shortest path in order to reach target without collision. These algorithms provide the robot the possibility to move from the initial position(source) to final position (target). The above strategy is designed in a 2-D grid map form of known environment and static obstacles. When the mission is executed it is necessary to plan an optimal or feasible path for itself avoiding obstructions in its way and minimise a cost such as computational time and path distance. It must take the robot toward its target.

VI. SIMULATION RESULTS

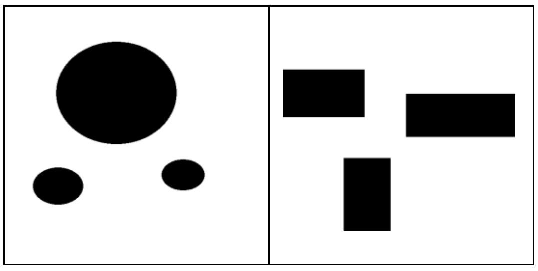

Fig. 3 Map 1 and Map 2

Map1 and Map2 are taken as the risks maps. Here, black regions are considered as obstacles, so whenever the robot comes near the risk area, it has to give a penalty. These maps form the grid map which make computation easy. The source and destination points are given and there is a connection matrix which act as a robot that travels from source to destination to obtain optimum path which is obstacle free, feasible, complete and smooth. The originals maps taken are 500*500 pixels. This size is reduced to 100*100 grid. The starting and the destination point is [10,10] and [490,490] respectively. Here, the connection matrix or the robot was given as [1 1 1;1 2 1;1 1 1] .We assume, 2 is the current position of the robot and 1 are the 8 possible moves that allow robot to move in all directions. A*Algorithm was then run on the grid maps twice to get optimum path. For Genetic Algorithm method same maps of 500*500 pixels were used. Some variables were set up ie Number of points in solutions is 3, Population Size is 50 and Number of generations is 10. Splines were used to smoothen the path. Then, GA was run and got terminated after reaching its stopping criteria of crossing maximum number of generations and the optimum paths were found.

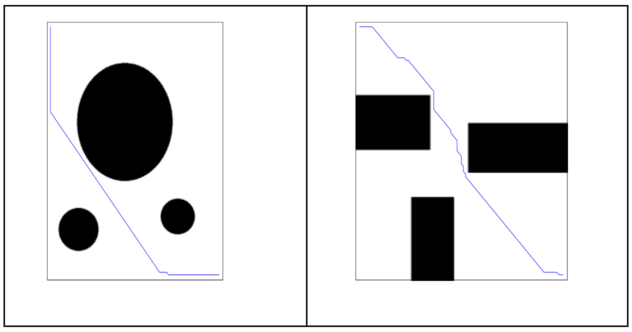

Fig. 4 Optimal path obtained by A*Algorithm on Map 1 and Map2

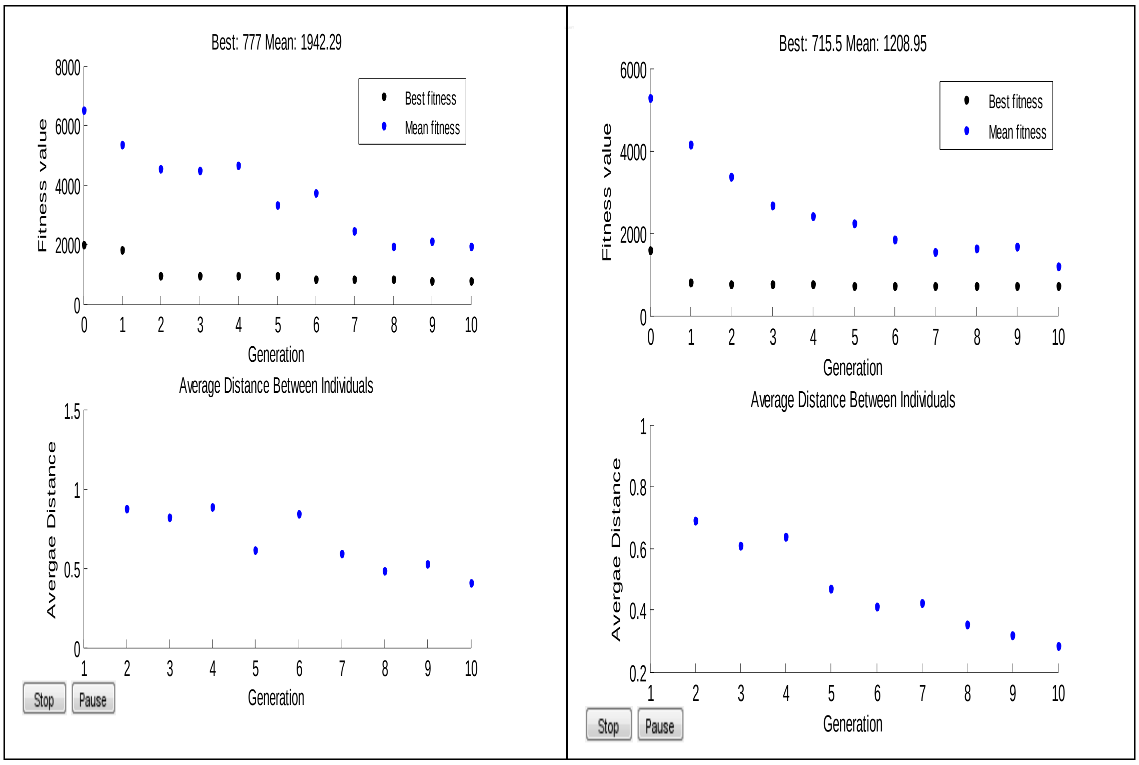

Fig. 5 Graph of fitness value and average distance of individual solution with respect to number of generations by Genetic Algorithm on Map 1 and Map2

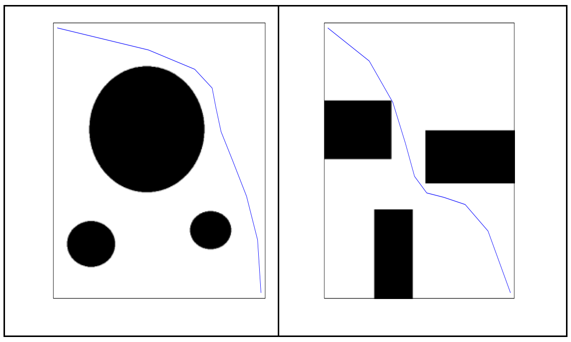

Fig. 6 Optimal path obtained by Genetic Algorithm on Map 1 and Map2

TABLE 1- Comparision of Processing time and Path Length of A*Algorithm and Genetic Algorithm on 2-D map1 and map2 respectively.

Method of Path Planning	Run no	Processing Time(in secs)	Path Length (in pixels)
A*Algorithm(Map1)	1	80.1	775.4
	2	91	775.4
Genetic Algorithm(Map1)	1	52.8	777
	2	37.5	754
A*Algorithm(Map2)	1	95.3	731
	2	44.1	731
Genetic Algorithm(Map2)	1	34.8	740
	2	29.9	745

VII. Discussion

The performance of A* and GA search algorithms to find optimal path of a mobile robot such as an AUV in 2-D problem space has been compared successfully. The simulation result shows that A*Algorithm provide accurate solutions as same solutions are obtained when run multiple times on each map respectively.Unlike A*algorithm,GA gives less accurate solutions ,sometimes the paths crossed the obstacles which is undesirable.The comparisions of computational time show that GA takes less time to find optimal path than A* algorithm with the help of genetic operators.
Since accuracy is considered most important in path planning so A*algorithm is preferred more than GA for the problem.The proposed A*algorithm can also be used in 3-D path planning for Autonomous Underwater Vechiles which would be interesting and challenging.

References

[1]. J. Barraquand, J. C. Latombe and B. Langlois, "Numerical Potential Field Techniques for Robot Path Planning," *IEEE Transaction on System,Man and Cybernetics*, vol. 22, no. 2, March,1992.

[2]. Y. H. Lui and S. Arimoto, "Computation of the Tangent Graph of Polygonal obstacles by Moving line processing," *IEEE Transaction on Robotics and Automation*, vol. 10, no. 6, Dec,1994.

[3]. O. Takahashi and R. J. Schilling, "Motion Planning in a plane using generalised Voronoi Diagrams," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, April,1987

[4]. R. Zhou and E. A. Hansen, "Breadth First Heuristic Search," *Elsevier, Artificial Intelligence*, pp. 385-408, 2006.

[5]. R. E.Korf, "Depth First Iterative Deepening :An optimal Admissible Tree Search," *Elsevier*, vol. Artificial Intelligence 27, pp. 97-109, 1985.

[6]. Zhang et al, "An edge Detection Technique using Genetic Algorithm based Optimization," *Pergamon*, vol. Pattern Recongnition 27, no. 9, pp. 1159-1180, 1994.

[7]. L. E. Kavraki, P. Svestka and Latombe, "Probabilistic Roadmaps for Path Planning in High Dimension Configuration Spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, August ,1996.

[8]. S. M. Lavalle, "Rapidly exploring random trees:Anew tool for Path Planning," *Computer Science Dept of Lowa University* , pp. 98-111, Oct 1998.

[9]. A. R. Soltani et al, "Path planning in construction sites: performance evaluationof the Dijkstra, A*, and GA search algorithms," *Elsevier Advanced Engineering Informatics*, vol. 16, pp. 291–303, 2002.

[10]. H. Cao, N. E. Brener and S. S. Iyengar, "3-D large grid Route Planner for autonomous underwater vehicle," *International Journal of Intelligent Computing and Cybernetics*, vl. 2, no. 3, pp. 455-476, 2009.

[11]. L. D. Fillpis, G. Guglieri and F. Quagliotti, "Path Planning Strategies for UAVs in 3D environment," *Springer*, 16 August,2011.

[12]. A. Atyabi and D. M. Powees, "Review of classical and heuristic based navigation and path planning approaches," *International Journal of Advancement in Computing Technology(IJACT)*, vol. 5, no. 14, 2013.

[13]. T. Lee, H. Kim, H. Chung and Y. Bang, "Energy Efficient Path Planning for a Marine Surface Vehicle considering heading angles," *Elsevier*, no. 107, pp. 118-131, 2015.

[14]. Z. Zeng, L. Lian, K. Sammut and A. Lammas, "A survey on Path Planning for persistent autonomy of autonomous underwater vehicles," *Elsevier*, vol. 110, pp. 303-313, 8 Nov ,2015.