Presentation of algorithm for a full proximate flat surface optimum padding included obstacles with use routing algorithms

Parisa Padidar[1],AmirReza Estakhrian haghieghie[2]*
1- department of computer, sepidan Branch, Islamic Azad University, sepidan, Iran
2- department of computer, sepidan Branch, Islamic Azad University, sepidan, Iran
Email: parisapadidar1394@gmail.com , amirestakhrian@iausepidan.ac.ir

Abstract

This article proposes a new planning navigation strategy for use with mobile terrestrial robots. The proposed algorithm is applicable in any point of the areas for tasks such as cleaning the floors of building, mowing and clearing mined areas. The strategy of this algorithm is analyzed and checked in conditions where the environment and the obstacles are known. There are various routing algorithms such as A, Genetic, Dijkstra, antcolony and etc, but any of the main algorithms is not singly the optimum one and all of them have some disadvantages. The proposed algorithm reduces the number of required repetitions to passing the routes by a combination of good characteristics of the main algorithms; after each movement event, the robot passing through all eight directions of source node according to specified fitness function, traverses the intended area at each repetition that the cost of each node is the distance of that node from the source. In this article, some problems presented and finally, the proposed algorithm in comparison with the algorithm of Roomba made by iRobot Corporation, has been checked by the observation and test method and their response has been obtained. In the path of achieving to the response by using the presented algorithm, some pseudo-codes have been designed in the C# environment.*

Keywords: Hybrid algorithm, Area traverse, Optimum traverse, Roomba.

1- Introduction

The evolution of robotics research in the last fifteen years has had an important influence on human activities in many sectors of society. In the case of mobile robots, the main objective is to develop autonomous intelligent robots that are able to effectively plan their movements in static or dynamic environments that are unknown. Methods involving Artificial Intelligence (AI) are amongst the techniques that can be used for this purpose, and the navigation strategies based on Artificial Intelligence (AI) have shown a high level of precision in terms of avoidance of obstacles and achievement to the optimum route. In the area of AI, many

Traverse techniques such as fuzzy logic, artificial neural networks (ANNs), genetic algorithms (GAs), or combinations of these procedures have been used [1].

A* algorithm is a heuristic based algorithm widely used in Artificial Intelligence. It follows path having lowest known heuristic costs. Drawback of A* is requiring a large amount of CPU and memory resources [2]. Ant colony algorithm is based on behavior of ants. In ant colony algorithm nodes are traversed in random fashion initially, and cost of each node in the path is updated. The path with maximum use time by ants is considered to be optimal [3]. The main drawback of this algorithm is that the number of required iterations to find an optimum path is so much. Techniques based on GAs have been widely used due to their robustness, and in most cases have been able to provide optimum routes in both static and dynamic environments. However, for dynamic environments, global planning strategies need to use external observation and devices in order to transmit the current state of the environment to the robot at a suitable rate. Navigation strategy using GA have been presented by employing global planning where the individuals (or chromosomes) are consistent of all possible routes between the initial and final points. GA's drawback is that you cannot be sure of response times for the constant optimization, considering works [4], [5], [6], [7], [8], [9]. Our proposed algorithm by exploiting the good properties of ant colony, A*, GA, Flood Fill algorithms, developed a new algorithm that is more efficient than the original algorithm. The proposed algorithm introduces a hybrid approach with the passage of all eight directions of source node according to specified fitness function. This algorithm is an optimized hybrid approach [10] that can be used for any problem that needs to be applied to find the optimal route.

2- Studies' Review

2.1. Genetic Algorithm

Genetic algorithm is used to find approximate optimal solutions. It is inspired by evolutionary biology such as inheritance, mutation, crossover and selection [11]. Advantages of this algorithm are so useful to solve problems with multiple solutions and many inputs. Some of the disadvantages of GA are the special optimization problems that cannot be solved due to weak known fitness function, and through which you cannot assure constant optimization response times. In GA the entire population is improving, but this could not be true for an individual within this population [12].

2.2. Heuristic Function

Heuristic maps' function describes problem state as a number which represents the degree of desirability. Heuristic function has different errors in different states. It plays a vital role in optimization problem [13].

2.3. Depth-First-Search

DFS is used as Last-In-First-Out stack and is a recursive algorithm. It is simple to be implemented. But the major problem with DFS is that it requires large computing power and runtime increases exponentially, for a small increase in map size [13].

2.4. Breadth-First-Search

BFS is used as First-In-First-Out at the queue. It is used when space is not a problem and few solutions may exist and at least one has the shortest path. It works poorly when all solutions have long path lengths or there exists some heuristic function. It has large space complexity [13].

2.5. A* Algorithm

A* combines features related to uniform-cost search and heuristic search. It is a BFS in which the cost associated with each node is calculated using admissible heuristics [14].

For graph traversal, it follows a path with lowest known heuristic cost. The time complexity of this algorithm depends on used heuristic. Since A* has the BFS drawback, it needs a large memory to save all of the open lists [13].

2.6. Ant Colony Optimization

ACO is meta-heuristic algorithm based on the behavior of real ants [15]. While traversing to destination, each ant deposits a chemical substance called pheromone. Each ant traverses in random fashion but when it encounters pheromone trail it has to decide whether to follow it or not. If an ant follows the same path, the amount of pheromone deposition increases. Thus, the path mostly followed by an ant has maximum pheromone deposition. An ant using shortest path to destination will reach source fast, as shortest path will have twice more pheromone deposition than others [15]. If there is more than one optimal path, then ACO cannot decide to choose optimal path [16].

2.7. Flood-Fill Algorithm

Robot maze problems are considered as important fields of robotics and are based on decision making algorithm [17]. It requires complete analysis of workspace or proper maze planning [18]. Flood fill algorithms and modified flood fills are widely used for robot maze problem [19]. Flood fill algorithm assigns the value to each node representing the distance of that node from centre [14]. The flood fill algorithm floods the maze when a mouse reaches new cell or node. Thus, it requires high cost updates [14]. The flooding is avoided in modified flood fill [19].

3- Proposed Algorithm

The proposed system starts with source node and search in all eight directions of source node according to specified fitness functions. This process is continued until destination node is reached. In the process cost of each node, passing equivalent to number of iterations required to reach that node is stored. The hybrid navigation strategy of path production consists of M displacement in the first generation of the population in order to reach the final goal.

$$P=(X , Y) \tag{1}$$

There is a local target at any M displacement of the robot moves.

$$P^{ol}(M)=(X^{ol}(M) , Y^{ol}(M)) \tag{2}$$

Local target selection P^{ol} (M) exists in any event of displacement robot using the algorithms and the current position with regard to neighbors who have the lowest cost with a common border, and all positions PR (M) are visited and stored in vector.

P^R : Location that can be expressed as follows:

$$P^R=[P^R(0), P^R(1), P^R(2),\dots P^R(m)] \tag{3}$$

The algorithm terminates when all nodes are visited freely.

3.1. Working

The working of algorithm is illustrated by using examples. Optimum and full padding of a 4*4 proximate flat surface without obstacles. An environment is divided into a number of locations, and each location has a specified amount as it has been shown in Figure 1.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig.1. Flat surface 4*4

- **Statement of the problem**

Goal: Complete traverse of free cells

The starting point (source): The cell number 1

As it has been shown in Figure 1-a, cells initially are valued toward the source.

start	1	2	3
1	1	2	3
2	2	2	3
3	3	3	3

Fig.1-a. cells valuation

First step: First all of the source neighbors (initial population) are traversed. The cost of each node is equal to the number of iterations required to reach there.

In the second step: the next nearest neighbor's surface to the source is traversed. The method for selecting the first node among nodes with similar costs is thought as the fitness function.

First, fitness function selects cells from among free cells with similar costs having common borders and fewer cells in neighbor. This process will continue until the end surface.

In general, traversing cells is done according to cells' number of Figure 1 as follows regarding an order from left to right:

5-6-2-3-7-11-10-9-13-14-15-16-12-8-4

- **Statement of the problem:**

Goal: Complete traverse of free cells

The starting point (source): The cell number 11

2	2	2	2
2	1	1	1
2	1	start	1
2	1	1	1

Fig.1-b. cells valuation

In general, traversing cells is done according to cells' number of Figure 1, as follows regarding an order from left to right:

16-15-14-10-6-7-8-12-8-4-3-2-1-5-9-13

As you can see, in the without obstacles' environment with any starting point, the whole surface gets traversed in the most possible optimal way.

- **Statement of the problem:**

Goal: The optimal and complete traverse of free cells without collision to obstacles

The starting point (source): The cell number 13

Scattered blocked cells: 5, 7, 10, 16

The symbol of blocked points: 999

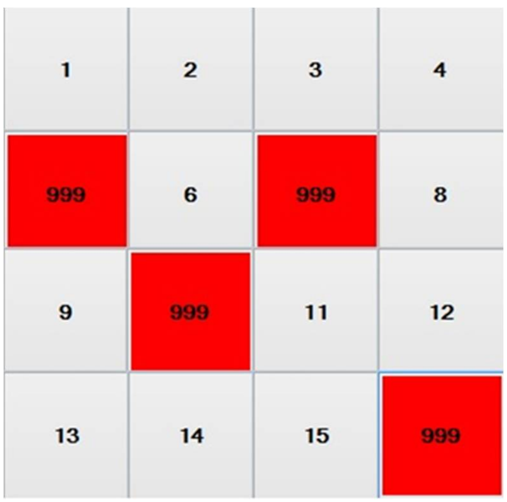


Fig.2. Flat surface 4*4 with obstacles

As can be seen in Figure 2, a setting has been divided to some locations, and obstacles are marked with red color. Mobile robots can traverse the free zones.

The traversing environment of the robot is called the chart of safe map. The inputs of the problem are size of map network in Figure 2, starting point and the obstacles' cells. The best solution is a free choice of path from the starting point to the goal and based on the whole route it gets minimized.

3.2. Block Nodes and Clean Nodes

Block nodes and clean nodes are different. The block node gets disconnected from population, but the clean node will be ignored. So it is possible for the clean node to traverse again. Block cells have been shown in Figure 2.

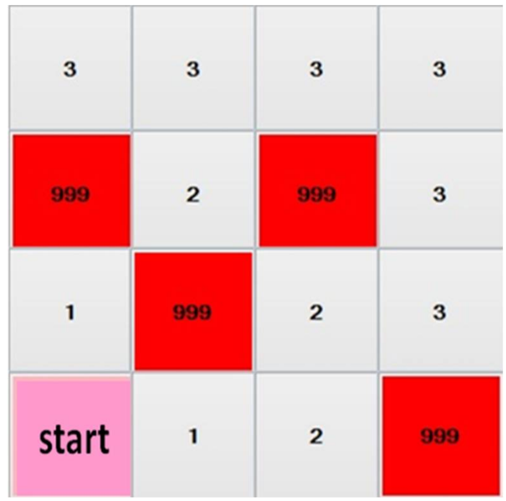


Fig.2-a. cells' valuation

In Figure 2-a, valuation of cells has been shown according to the starting point:

In general, traversing cells is done according to cells' number of Figure 2 as follows regarding an order from left to right:

9-14-15-11-6-1-2-3-4-8-12

- **Statement of the problem:**

Goal: The optimal and complete traverse of free cells without collision to obstacles

The starting point (source): The cell number 16

Scattered blocked cells: 2, 7, 11, 15

The symbol of blocked points: 999

Block cells have been shown in Figure 3:

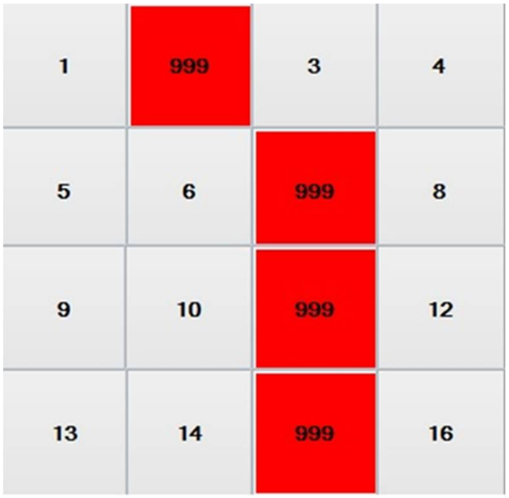


Fig.3. Flat surface 4*4 with obstacles

In Figure 3-a, valuation of cells has been shown according to the start point:

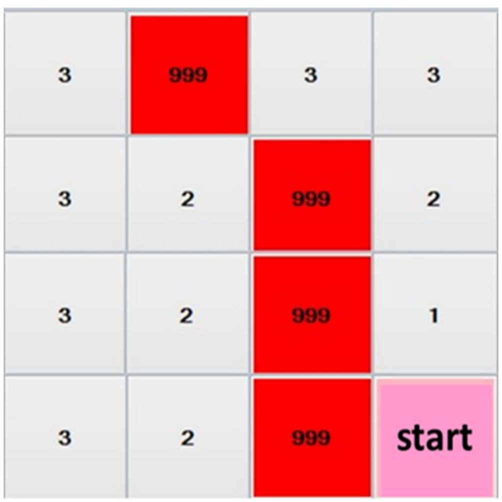


Fig.3-a. cells' valuation

In general, traversing cells is done according to cells' number of Figure 3, as follows regarding an order from left to right:

12-8-4-3-6-10-14-13-9-5-1

3.3. Flowchart

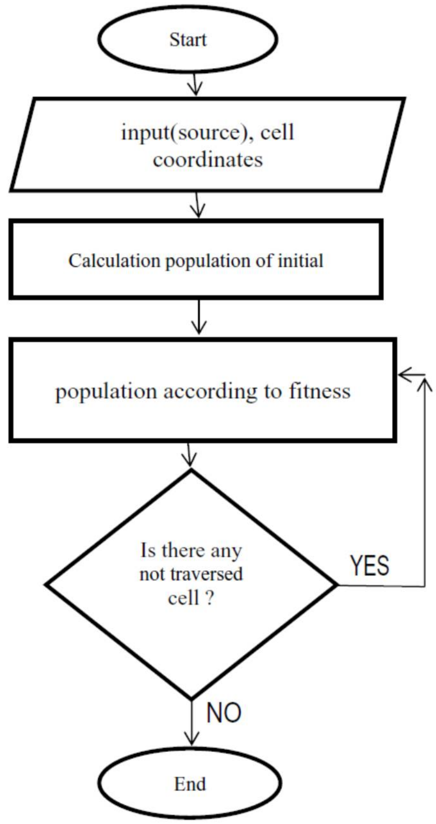
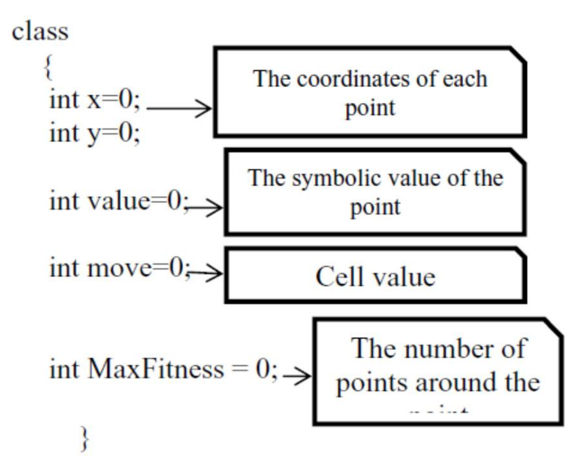


Fig.4. Flowchart for map exploring

3.4. Structure of Elucidation Algorithms



4- Conclusions

The proposed algorithm was implemented in C # and it had an optimal performance by reducing the number of required iterations to traverse the entire path and reduce time complexity of algorithm in environments with obstacles and without obstacles. Also the proposed algorithm was implemented on the Roomba robot vacuum cleaner (made by iRobot Corporation). And the performance of the proposed algorithm in comparison with the Roomba algorithm indicated a huge reduction in traverse time and depreciation of hardware with the same local conditions.

In some cases, such as increasing the number of empty neighboring cells around the starting point of the robot movement, Roomba algorithm had a better performance in comparison with the proposed algorithm. By using the proposed algorithm for perfect traverse of proximate flat surfaces included obstacles, optimum and practical results have been obtained in most situations.

Acknowledgment

We sincerely thank all those who have helped us to accomplish this research somehow.

References

[1]Ansari A., Sayyed M., Ratlamwala K, Shaikh P., AN OPTIMIZED HYBRID APPROACH FOR PATH FINDING., International Journal in Foundations of Computer Science & Technology (IJFCST), Vol.5, No.2 (2015).

[2]Aravindh S., Michae G., hybrid of ant colony optimization and genetic algorithm for shortest path., Journal of Global Research in Computer Science,volume 3, no. 1 (2012).

[3]A´tila V F., deOliveira M., Marcelo A., Fernandes C., Dynamic Planning Navigation Technique Optimized with Genetic Algorithm., Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol (2014).

[4]Chaari I., Koubaa A, Bennaceur H, Trigui S., Al-Shalfan K., smartpath: A hybrid aco-ga algorithm for robot path planning., in Evolutionary Computation (CEC), IEEE Congress on, pp. 1–8 (2012).

[5] Intelligent Control Techniques in Mechatronics Geneticalgorithm., http://www.ro.feri.uni-mb.si.

[6]Ismail A ., ALTaharwa S., AlWeshah M., A mobile robot path planning using genetic algorithm in static environment," Journal of Computer Science, vol. 4, no. 4, pp.341–344 (2008).

[7]Khantanapoka K., Chinnasarn, K., Pathfinding of 2D & 3D game real-time strategy with depth direction A* algorithm for multi-layer., IEEE (2009).

[8] Kaur D., Mundra PS., Ant Colony Optimization: A Technique Used for Finding Shortest Path., International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 5 (2012).

[9]Law G., Quantitative Comparison of Flood Fill and Modified Flood Fill Algorithms., International Journal of Computer Theory and Engineering, Vol. 5,No.3 (2013).

[10]Mishra S., Bande P., Maze solving algorithms for micro mouse., in Proc. IEEE International Conference on Signal Image Technology and Internet Based Systems, Bali, pp. 86-93 (2008).

[11]Mohanraj T., Arunkumar S., Raghunath M., Anand M., MOBILE ROBOT PATH PLANNING USING ANT COLONY OPTIMIZATION"., International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308 (2014).

[12]Poli R., William B. Langdon., Nicholas F. McPhee, John R. Koza., A Field Guide to Genetic Programming", page no-9 (2008).

[13]Reddy S K., Artificial Intelligence Algorithms., IOSR Journal Of Computer Engineering (IOSRJCE), volume 6, issue3 (2012).

[14] Samadi M., Othman M F,. Global path planning for autonomous mobile robot using genetic algorithm., in Signal-Image Technology Internet-

Based Systems (SITIS), International Conference on, Dec 2013, pp. 726–730 (2013).
[15]Sharma M., Robeonics K., Algorithms for Micro-mouse., in Proc. International Conf. on Future Computer and Communication., Kuala Lumpur, pp. 581-585 (2009).
[16]Stout B., Smart Moves: Intelligent Pathfinding.,Game Developer Magazine, page no.-5 (1997).
[17]Tuncer A ., Yildirim M,. Dynamic path planning of mobile robots with improved genetic algorithm., Computers& Electrical Engineering, vol. 38, no. 6, pp. 1564 –1572 (2012).
[18]Yongnian Z., Lifang Z., Yongping L., An improved genetic algorithm for mobile robotic path planning., in Control and Decision Conference (CCDC), 2012 24th Chinese, May, pp. 3255–3260 (2012).
[19]Zhang L., Min H., Wei H., Huang H., Global path planning for mobile robot based on a*; algorithm and genetic algorithm., in Robotics and Biomimetics (ROBIO), IEEE International Conference on, Dec 2012, pp.1795–1799 (2012).