International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

March 2022	Issue 50	Volume 14	Number 1	Pages 183-189

HYBRID OPERATION OF A* AND PRM ALGORITHMS FOR EFFICIENT PATH PLANNING IN AUTONOMOUS MOBILE ROBOTS

B. Uzun [1] H. Gozde [2] M.C. Taplamacioglu [1]

1. Electrical and Electronics Engineering Department, Gazi University, Ankara, Turkey
barisuzun.tr@gmail.com, taplam@gazi.edu.tr
2. Turkish Aerospace Industries, Ankara, Turkey, halukgozde@gmail.com

Abstract- These days, developing robot technology is used in many areas of human life and one of the most widely used areas is mobile robots. The most basic problems of mobile robots are their ability to calculate optimum route and find their way in short time. Nowadays, the A^* algorithm is a successful and efficient algorithm used in most robot applications. However, route calculation times are very high on large-scale maps. The PRM algorithm is a probabilistic algorithm and can calculate routes in short time even on large-scale maps. In this study, the A^* algorithm is combined with the PRM algorithm so that it could find a path in shorter times. The combined new algorithm and the classical A^* algorithm is compared in terms of path length and path generation time.

Keywords: Mobile Robot Path Planning, A^* Algorithm, PRM Algorithm.

1. INTRODUCTION

The vocabulary of robot is first used in Karel Capek's play RUR (Rossums Universal Robots) in 1921 and entered the world literature. It is sourced from Czech and means servant, slave [1]. New robots developed with developing technology are present in every part of human life. Robots are used in some situations to assist people, in some situations in order to do all their work of people and in some situations to do things that people cannot do. Particularly in space researches, robots that can resist space circumstances are mostly used [2].

Robots that can move freely are called as mobile robots. They mostly used in the carrying of humans and in unmanned missions in air-land-sea environments. With the advancing technologies, mobile robots have reached to the capability of move autonomously, that is to make their own decisions and move independently from people.

Energy spending, path generation speed and mission times are significant topics in autonomous vehicles. Several path planning algorithms have been improved in order to optimize these parameters. The aim in these algorithms is to generate the shortest path as soon as possible. Path planning algorithms can be described as algorithms that seek to find the path of the most suitable length from a starting point to a requested goal point in a map, without crashing obstacles in the map.

While calculating these algorithms, the following points should be considered.:

- The path generated does not intersect with obstacles,
- Generating smooth path without sharp turns,
- Generation of the route in the shortest possible time.

Researchers in literature are separated algorithms that used in path planning into 2 main classes [3]. In according with the situation of the environment and obstacles, and to the working logic of the path planning algorithms.

Besides path planning algorithms are divided into two classes, static and dynamic, according to the condition of the environment and obstacles. Static environments are the environments where everything but the robot on the map is stable. Dynamic environments are environments in which obstacles and other environmental components are mobile. In accordance with the working logic, path planning algorithms are divided into two as global and local.

When calculating the route with global path planning algorithms, it is assumed that the robot has complete knowledge of the environment and the positions of the obstacles. With this knowledge, the robot analyzes the whole map and calculates the most optimal path between the start and the target point before starting the motion. When creating a route with local path planning algorithms, it is assumed that the robot has either no or partial information of the environment and obstacles. If the robot only knows the start and target points, it makes the path planning with the help of the knowledge obtained from the sensors on it.

Path planning techniques are grouped under 3 main headlines in regards to working methods, maps used, types of obstacles and solution times. These are classical techniques, probabilistic techniques and heuristic techniques. Classical techniques are the oldest path planning techniques used in mobile robots. These algorithms are usually efficient in static environments where is all known, and maps to be made for global path planning. If there is an appropriate path on the map, it is highly likely that classical path planning algorithms will discover this path. The disadvantage of this methods is that the computation time is very long because it performs many mathematical processes.

Probabilistic techniques are constantly used in path planning algorithms. By assigning random nodes to appropriate areas (without obstacles etc.) in the search area, it makes path planning over these assigned nodes. The way it works is basic and generates results in a short time, even on large-scale maps. The disadvantages of this method are that the path found is a sharp and uneven road. Heuristic algorithms are algorithms that use natural phenomena for path planning. These algorithms do not warrant the certain solution, but they do warrant a solution close to the certain solution within an acceptable time according to deterministic methods [4].

The A^* algorithm is one of the most used path planning algorithms in these mentioned path planning algorithms. The heuristic function used in its calculation prevents excessive branching of the research and ensures that it moves towards the target. If there is a probable path on the map, the A^* algorithm finds this path with a high possibility. The A^* algorithm works efficiently on small scale maps however calculation time is excessive on large scale maps. This circumstance is unbearable on excessively large maps. Probabilistic path planning algorithms can produce paths in short time even on large-scale maps.

In this article, a study is presented that enables faster path generation by running the PRM algorithm, which is one of the most used probabilistic path planning algorithms, with the A^* algorithm together.

2. ROUTE CALCULATION USING A^* AND PRM-A^* ALGORITHMS

2.1. A^* Algorithm

A^* algorithm is a heuristic approach algorithm. It was first presented in 1968 by researchers at the Stanford Research Institute by Peter Hart, Nils Nilsson, and Bertram Raphael [5].

While finding the shortest path in the A^* algorithm, the total cost is calculated with the formula given below.

$$F(x) = G(x) + H(x) \tag{1}$$

The distance function $G(x)$ is the cost of the distance traveled from the starting point to a desired point on the map. It can be computed using the path found. As for the heuristic function $H(x)$ is an estimate of the cost of the distance traveled from any square to the target square. This concept is called heuristic and is nothing more than a kind of smart guess. The real distance cannot be known until the route is found since various obstacles may appear on the route (walls, water or etc.). Where $F(x)$ is the total cost.

The heuristic function provides the supreme contribution to the achievement of the A^* algorithm. There are many techniques of computing heuristic distance. Euclidean distance and Manhattan distance are some of them. The computation formulas for Euclidean and Manhattan distances are given in Table 1.

Table 1. Heuristic distance function

Heuristic Function	Equation
Euclidean	$h(x) = [(Bx\text{-}Ax)^2 + (By\text{-}Ay)^2]^{1/2}$
Manhattan	$h(x) = \lvert Bx\text{-}Ax\rvert + \lvert By\text{-}Ay\rvert$

One of the important cases when computing the heuristic distance is that the closer the heuristic distance is to the real, the more accurate and faster the A^* algorithm will efficiency. The A^* algorithm's flowchart is as shown in Figure 1.

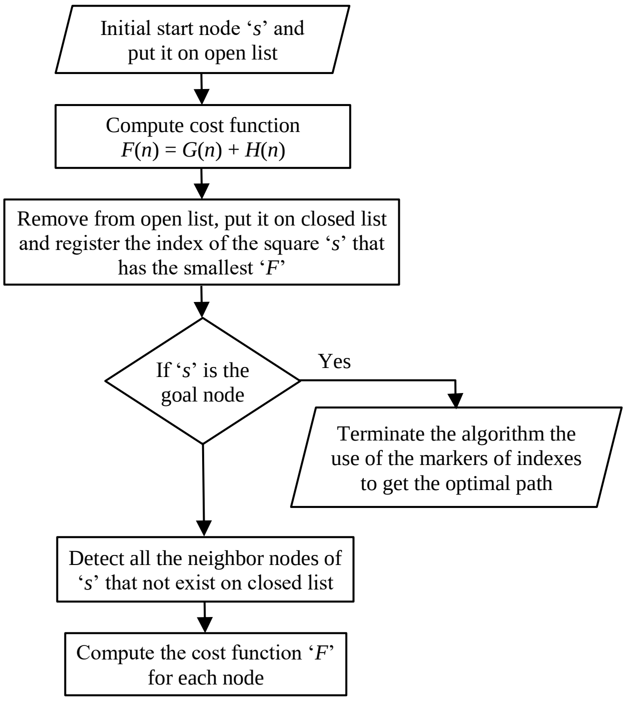


Figure 1. Flowchart of the A^* algorithm [6]

Generally, the A^* algorithm starts its calculation from the determined starting nodes by investigating the nodes around it. Open and closed lists are created to follow the review. The open list is the list where the neighboring nodes of the investigated nodes are held, and the following nodes to be investigated are selected from this list. The closed list is the list where the investigated nodes are kept. The nodes in the current study are not examined again to prevent the algorithm from the loop.

After the starting point is investigated, the lowest cost points in the open list are followed in order and continued to be investigated until arriving the goal point. The heuristic function provides that the algorithm moves towards the goal point precisely without branching. When the goal point is added to the closed list, the goal is found and the algorithm is concluded. The route is created by following the parent points from the goal point to the starting point.

2.2. PRM Algorithm

Probabilistic Roadmaps algorithm (PRM) is one of the probabilistic based shortest path planning algorithms. It is developed by Lydia E. Kavraki in 1996 [7]. The flowchart of the algorithm is shown in Figure 2.

PRM is one of the most used techniques among probabilistic techniques. It works quicker than other path planning algorithms and produces results in a shorter time.

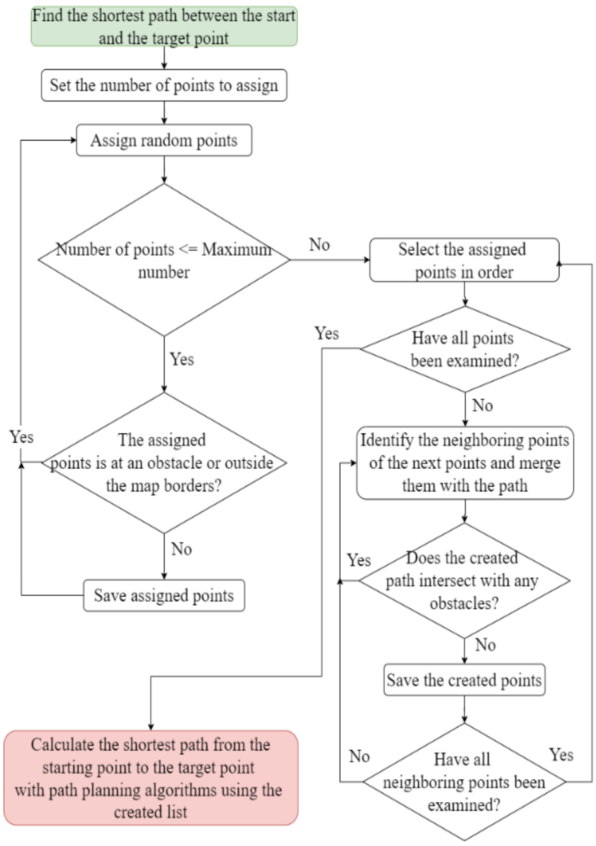


Figure 2. Flowchart of the PRM algorithm [7]

While the PRM algorithm computes the shortest path between the beginning and the goal, it proceeds in 2 stages: learning and query stages. The learning stage is the part where the PRM algorithm creates a roadmap. Road map is the map where the point assignment is made on the given map and these points are combined according to their neighborhood relations. In other words, they are more simplified versions of complex maps. This structure considerably shortens the scanning time of complex maps.

The learning phase consists of two parts. The first is the random point assignment section and the second is the section where the assigned points are combined according to their neighborhood relations.

The first part is the part where the algorithm assigns random points on the map. Randomly assigned nodes should be assigned to make it possible to reach all parts of the map. In the continuation of the algorithm, the search is done over the assigned points. If there is a connectivity problem between the assigned points (the assigned points can reach all parts of the map including the start and destination point and it is possible to connect the points with each other), the route may be difficult or even impossible to create. The result of the first part of the PRM algorithm learning phase is shown in Figure 3.

Query phase; this is the section where the shortest route is created by scanning the roadmap created during the learning phase with Dijkstra, A^*, D^* or various heuristic optimization algorithms. The result of PRM algorithm query phase is shown in Figure 5.

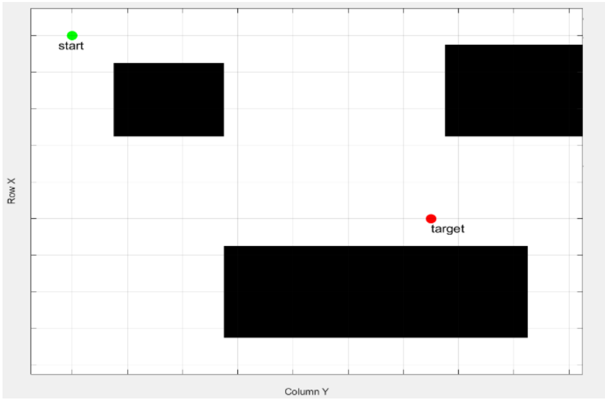


Figure 3. Result of the first part of PRM algorithm learning phase

The points assigned in the second part are connected with each other by means of paths. The purpose of merging is to know which points are adjacent to each other and to discover from which points it is possible to switch from one to another. The result of the second part of the PRM algorithm learning phase is shown in Figure 4.

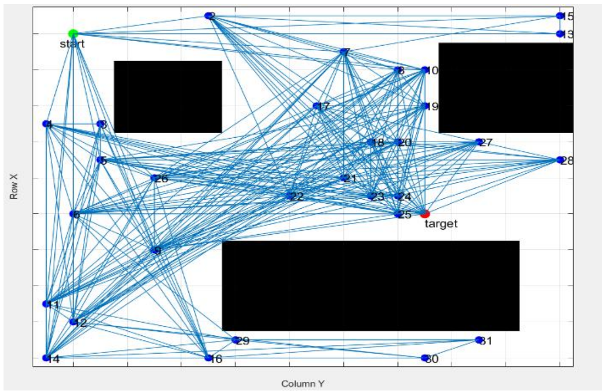


Figure 4. Result of the second part of PRM algorithm learning phase

In this study, the route will be calculated with the A^* algorithm on the road map created by the PRM algorithm. Throughout the scanning of the standard A^* algorithm, the scanning continues by computing the costs of 8 neighboring squares around a square. In the roadmap created by the PRM algorithm, randomly assigned points are examined instead of squares and neighborhoods are determined according to the connections created during the learning phase. According to this substucture, the A^* algorithm scans from the starting nodes until it finds the goal nodes and the path is created.

Since the A^* algorithm is used in the query phase of the PRM algorithm in this study, it is called PRM-A^* in the rest of the study.

3. COMPARISON OF A^* AND PRM-A^* PATH PLANNING ALGORITHMS

In this section, the A^* algorithm and the PRM-A^* algorithm is compared on two different 100×100 maps to measure their capabilities. The first of these maps is on a medium density map. The second is a more complex map than the first. The purpose of choosing such a map is to examine how both algorithms will perform in dense environments.

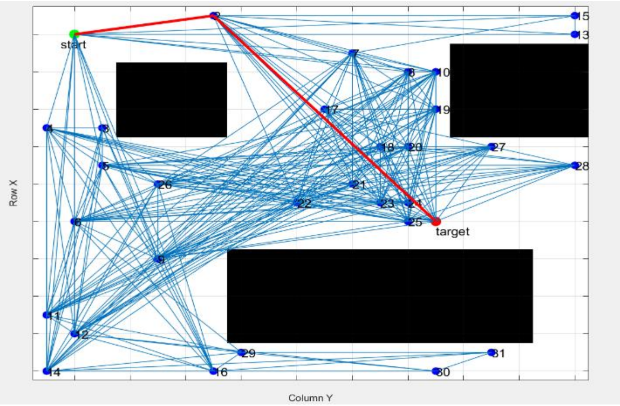


Figure 5. The result of PRM algorithm query phase

Since the PRM-A* algorithm creates a different route in each study, it was run 3 times and compared with the A* algorithm by taking the average. The A* algorithm will also be run 3 times. Since the A* algorithm finds the same route every time it works, there will be no change in the route. Only minor differences occur in the route creation process. Such a way was followed in order to have an equal comparison.

3.1. Scenario-1

In this section, the map named "T" in a study by Czech Technical University, Department of Cybernetics, Smart and Mobile Robotics Group is used by changing its dimensions [8, 9] as in Figure 6.

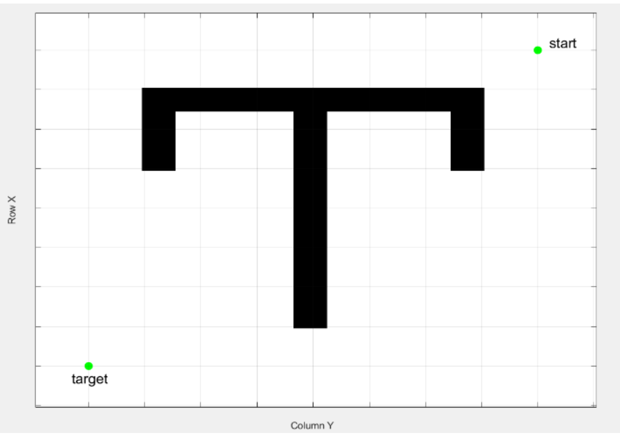


Figure 6. The map in scenario-1

The route created by the A* algorithm for the scenario-1 map is as in Figure 7. The route created by the PRM-A* algorithm for the scenario-1 map is as in Figure 8.

The length of the path found by finding the shortest path between the start point and the end point in the scenario-1 map with the A* and PRM-A* algorithms, the path creation time of the algorithm, the data found as a result of the algorithms running 3 times and their average values are shown in Table 2.

As a result of the route generation of the A* and PRM-A* algorithms in the Scenario-1 map, the bar graphs in which the routes produced by the algorithms are compared in terms of length and route creation times are shown in Figures 9 and 10.

3.1. Scenario-2

In this section, the map named "T" in a study by Czech Technical University, Department of Cybernetics, Smart and Mobile Robotics Group is used by changing its dimensions [8, 9] as in Figure 11.

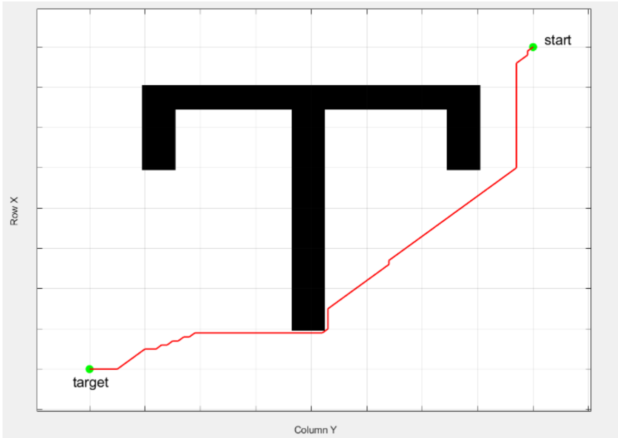
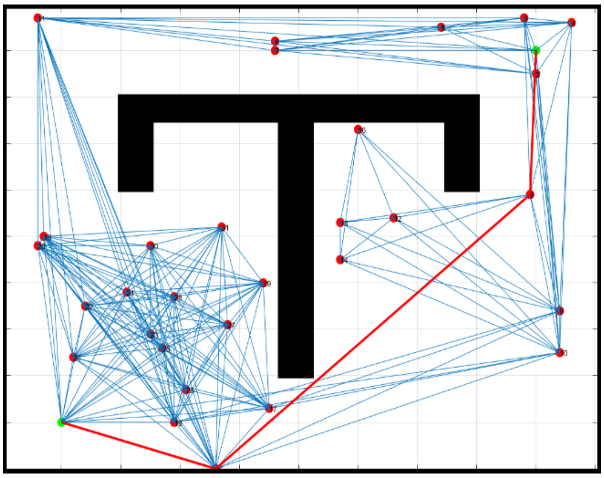
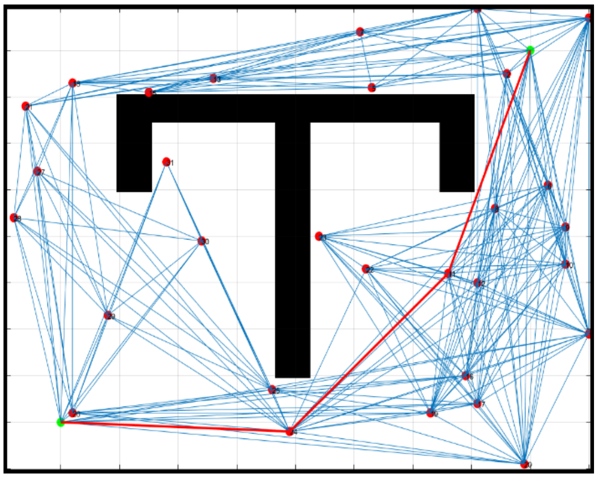


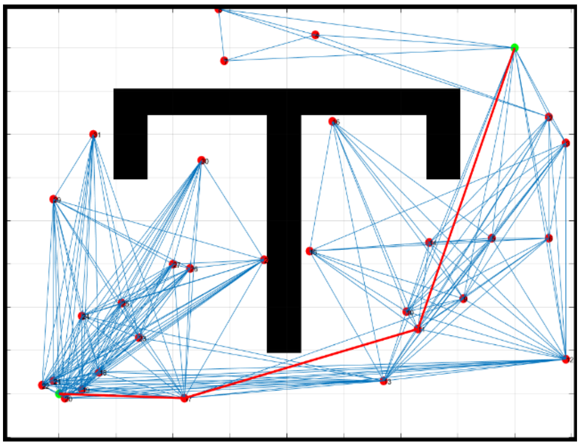
Figure 7. The route created by the A* algorithm for the scenario-1 map



(a)



(b)



(c)

Figure 8. Routes created by PRM-A* algorithm in scenario-1 map (a, b, c: 3 separate routes created by the PRM-A* algorithm in 3 separate studies)

Table 2. Performances of A* and PRM-A* algorithms in scenario-1 map

Algorithm	Test	Path Length	Time (s)	Average Path	Average Time (s)
A*	1	132.46	12.69	132.46	12.39
	2	132.46	12.52	132.46	12.39
	3	132.46	11.95	132.46	12.39
PRM-A*	1	138.18	0.91	134.62	0.96
	2	132.46	0.99	134.62	0.96
	3	133.22	0.98	134.62	0.96

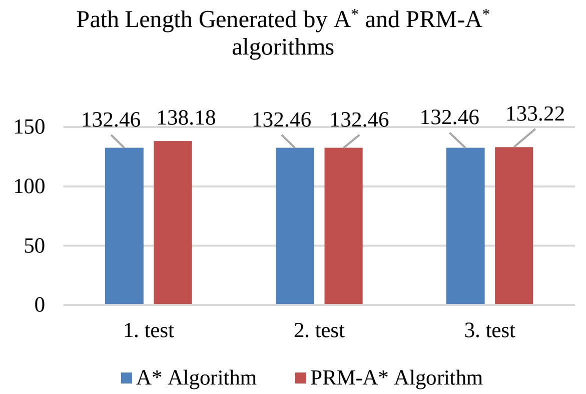


Figure 9. Comparison of path length generated by A* and PRM-A* algorithms with bar graph

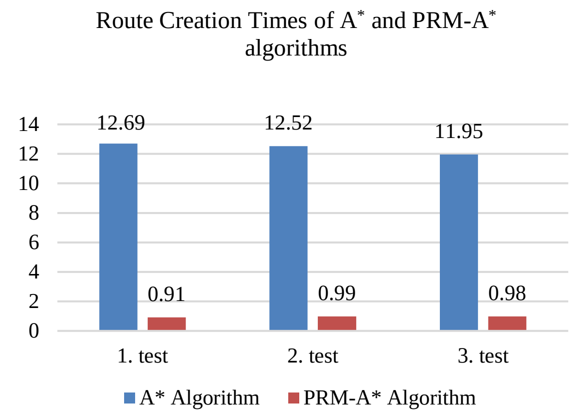


Figure 10. Comparison of path generation times of A* and PRM-A* algorithms with bar graph

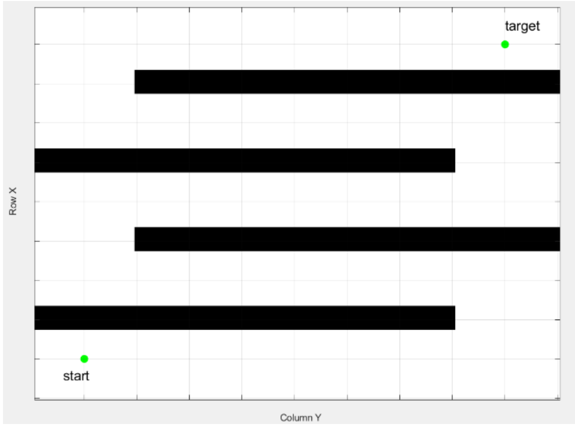


Figure 11. The map in scenario-2

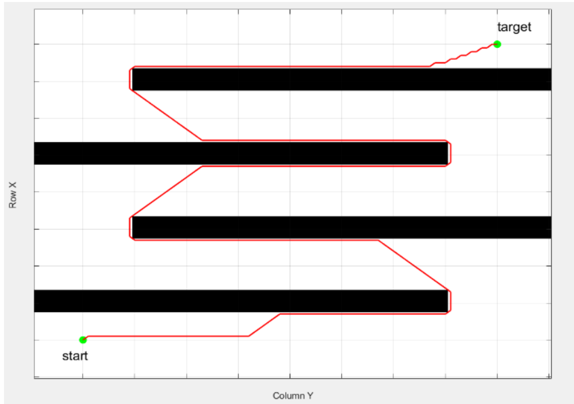
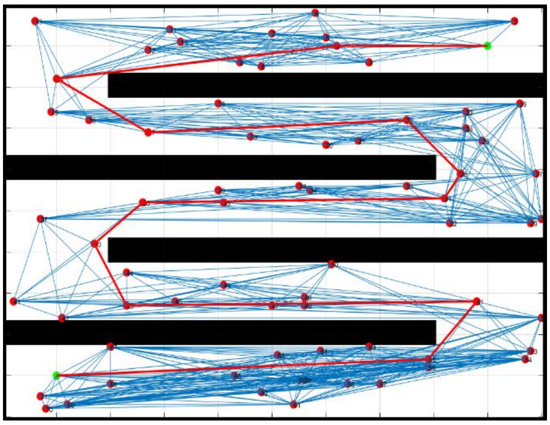
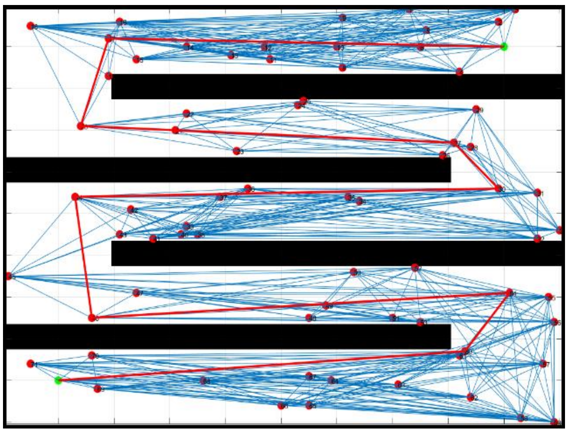


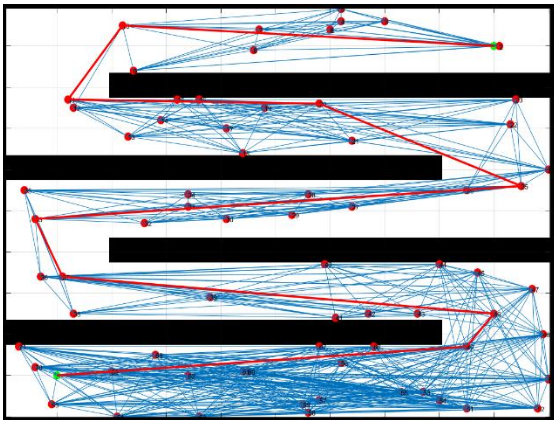
Figure 12. The route created by the A* algorithm for the scenario-2 map



(a)



(b)



(c)

Figure 13. Routes created by PRM-A* algorithm in scenario-2 map (a, b, c: 3 separate routes created by the PRM-A* algorithm in 3 separate studies)

The route created by A* algorithm for the scenario-2 map is as in Figure 12. The route created by the PRM-A* algorithm for the scenario-2 map is as in Figure 13.

The length of the path found by finding the shortest path between the start point and the end point in the scenario-2 map with the A* and PRM-A* algorithms, the path creation time of the algorithm, the data found as a result of the algorithms running 3 times and their average values are shown in Table 3.

Table 3. Performances of A* and PRM-A* algorithms in scenario-2 map

Algorithm	Test	Path Length	Time(sn)	Average Path	Average Time (sn)
A*	1	372.85	20.63	372.85	20.84
	2	372.85	20.96	372.85	20.84
	3	372.85	20.92	372.85	20.84
PRM-A*	1	409.6	1.96	432.72	2.02
	2	443.21	1.94	432.72	2.02
	3	445.34	2.17	432.72	2.02

As a result of the route generation of the A* and PRM-A* algorithms in the Scenario-1 map, the bar graphs in which the routes produced by the algorithms are compared in terms of length and route creation times are shown in Figures 14 and 15.

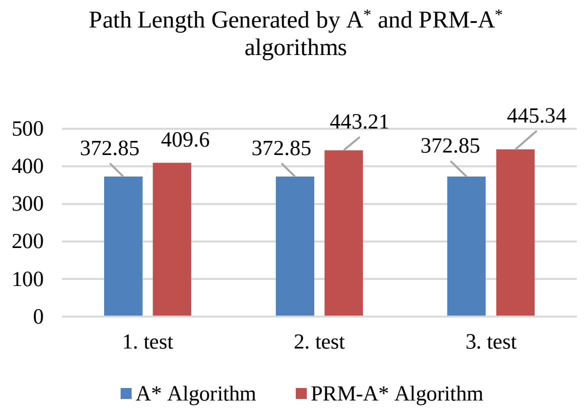


Figure 14. Comparison of path length generated by A* and PRM-A* algorithms with bar graph

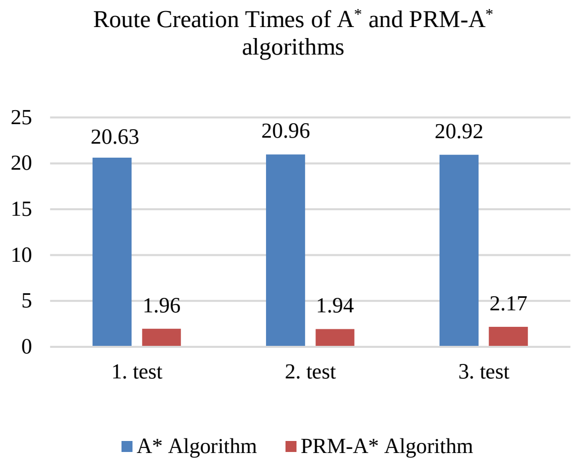


Figure 15. Comparison of path generation times of A* and PRM-A* algorithms with bar graph

4. CONCLUSIONS

The A* algorithm is a very stable and efficient algorithm. However, path generation times are very long on large maps. Therefore, the PRM algorithm has been tried to be used to solve this problem. The PRM algorithm can find a way in a very short time even on very large maps. For this reason, PRM and A* algorithm is hybridized. In order to test the results, experiments were carried out on 2 different maps.

First of all, a medium-sized map that is not very complex was used in scenario-1. In the results of this test, the PRM-A* algorithm produced results 12.9 times faster than the A* algorithm. However, it created only 1.63% longer path than the A* algorithm. As can be seen from the results, the PRM-A* algorithm can calculate a route much faster than the A* algorithm.

In order to measure the efficiency of algorithms in complex maps, a map with the same dimensions but slightly more obstacle density was chosen as the scenario-2 map. As a result of this test, the PRM-A* algorithm produced results 10.3 times faster than the A* algorithm, but created a 16.06% longer path compared to the A* algorithm. As can be seen from the results, the efficiency of the PRM-A* algorithm decreased at the point where the density of the map increased slightly. However, it was still able to create a road in a much shorter time.

The disadvantage of the PRM-A* algorithm is that the routes created usually have sharp turns. Increasing the number of points assigned for this purpose or using route smoothing algorithms will improve the route.

ACKNOWLEDGEMENT

The authors thank Merve Demirci (Ph.D. candidate) for her technical support.

REFERENCES

[1] E. Yolal, "Mobile Robot Simulators and Advanced Level Simulations", Master's Thesis, Istanbul Technical University, Turkey, January 2015.

[2] H. Zarabadipour, Z. Yaghoubi, M.A. Shoorehdeli, "Mobile Robot Control Based on Neural Network and Feedback Error Learning Approach", International Journal

on Technical and Physical Problems of Engineering (IJTPE), Issue 16, Vol. 5, No. 3, pp. 91-95, September 2013.
[3] S.A. Mnubi, "Motion Planning and Trajectory for Wheeled Mobile Robot", International Journal of Science and Research (IJSR), Issue 1, Vol. 5, pp. 1064-1068, January 2016.
[4] D. Sedighizadeh, "A Multi-Objective PSO Based Algorithm for a Vehicle Routing", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 20, Vol. 6, No. 3, pp. 38-43, September 2014.
[5] R. Dechter, J. Pearl, "Generalized Best-First Search Strategies and the Optimality of A^*", Journal of the Association for Computing Machinery, Vol. 32, No. 3, pp. 505-536, July 1985.
[6] I. Zidane, K. Ibrahim, "Wavefront and A-star Algorithms for Mobile Robot Path Planning", Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017, pp. 60-80, August 2017.
[7] L.E. Kavraki, P. Svestka, J.C. Latombe, M.H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566-580, August 1996.
[8] N. Alpkıray, "Development of Hybrid Path Planning Algorithm for Autonomous Mobile Robot Systems", Master's Thesis, Sivas Cumhuriyet University, May 2019.
[9] "http://imr.ciirc.cvut.cz/planning/maps.xml".

BIOGRAPHIES

Baris Uzun graduated from Department of Electrical and Electronics Engineering, Sakarya University, Sakarya, Turkey in 2014. Currently, he is a M.Sc. student in Electrical and Electronics Engineering Department, Gazi University, Ankara, Turkey. He works on path planning algorithms and robotics.

Haluk Gozde received the B.Sc. degree in Electrical and Electronics Engineering from Karadeniz Technical University, Trabzon, Turkey in 1997. He received the M.Sc. and the Ph.D. degrees in Electrical and Electronics Engineering from Gazi University, Ankara, Turkey in 2004 and 2010, respectively. He is an Assoc. Prof. of Electrical and Electronics Engineering since 2016. His main research area consists of power system dynamics and control, renewable energy systems, smart grid technologies, artificial intelligence-based control methods, and swarm intelligence-based optimization algorithms.

M. Cengiz Taplamacioglu graduated from Department of Electrical and Electronics Engineering, Gazi University, Ankara, Turkey. He received the M.Sc. degrees in Industrial Engineering from Gazi University and also in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey. He received his Ph.D. degree in Electrical, Electronics and Systems Engineering from University of Wales, Cardiff, UK. He is a Professor of the Electrical and Electronics Engineering since 2000. His research interests and subjects are high voltage engineering, corona discharge and modelling, electrical field computation, measurement and modelling techniques, optical HV measurement techniques, power systems control and protection, lighting techniques, renewable energy systems and smart grid applications.