A Novel Offline Path Planning Method

Pejman Kamkarian
Department of Electrical and Computer Engineering,
Southern Illinois University, Carbondale, IL 62901, USA

Henry Hexmoor
Department of Computer Science,
Southern Illinois University, Carbondale, IL 62901, USA

***Abstract*— This paper investigates a novel offline path planner for single point robots in cluttered environments. In order to achieve the best results in building shortest collision free trajectory lengths from initial to the goal configuration, we considered a multi-layer solution in the form of a unique algorithm that works as a unit on workspace elements such as obstacles and determines proper trajectories. In addition, we have employed multiple parameters in our path planner to increase its level of flexibility to be able to maneuver on different scenarios related to the workspace components layouts. This versatility has increased our planner capabilities to override constraints related to diversity of environmental specifications as well as adjustment to the robot equipment constraints.**

Keywords*— *Navigation, Path planning, Rapidly Optimizing Mapper, Robot trajectory builder

I. INTRODUCTION

An artificial intelligent point robot by default refers to a single mechanical device simplified to be represented by a point in space, which is able to maneuver through using proper equipment such as wheels in an environment called a workspace. In addition, a point robot is usually supplied with a predefined discipline, which enables it to operate in the workspace along with its related components independently. A workspace is usually bounded in a limited boundary area that consists of an initial and goal configurations as well as obstacles with various shapes and sizes located on different regions of the environment. One of the most important concerns for employing mobile robots is the ability to achieve the assigned tasks successfully without collisions. Unlike an online planner, an offline path planner outputs trajectories in an environment that is semi-static. The environment is composed of obstacles that remain stationary for long durations (e.g., as in a ship yard) and the robot path is expected to be traversed frequently. Therefore, path length and robot safety are of higher priority than impromptu path adjustments. In order to fulfill performing tasks flawlessly, a robot, as a key feature of its capability, has to possess a viable strategy to enable it to move among workspace objects while avoiding collisions. In other terms, the mobile robot has to be able to build an optimal collision less trajectory from the initial to a prescribed goal configuration. A path planner for a point robot, hence, contemplates a viable trajectory, which is constructed in a form of a procedure that is responsible to control collisions while the robot is pursuing its goal. The more vigorous path planner that a single point robot has in terms of accuracy and planning, the higher rate of performance to reach the goal successfully in terms of collision avoidance and safety. In recent decades, the vital need of the planner construction for a robot has attracted several groups of researchers to work on developing and optimizing planners for purposes of achieving better and more accurate results to build trajectories for moving robots from a start point to the goal configuration successfully. Comparing the early planners with the later ones manifests drastic upgrades on both the accuracy and safety of path planners. Researchers have used a variety of different criteria and rules to construct robots path planners. At the early stages of developing path planners for offline robots, [2], and [15] independently proposed planners inspired by electromagnetic fields. The potential field planner is constructed based on considerations of virtual attractive and repulsive forces among start and goal configurations along with the repulsive forces of obstacles that are present in the workspace. The *Probabilistic Roadmaps* method as another example of path planners was mainly introduced in [14]. It is categorized as a motion planning method that is built based on constructing a network of vertices located in the available free spaces in the workspace. They have adopted *Dijkstra's* algorithm, [8], as a tool to analyze and refine the shortest trajectory from initial to the goal configuration. Some path planners are more focused on building a safe trajectory. For example, the *Voronoi* diagrams as a solution for planning have been studied and developed by several researchers: [12], [16], [17], [19], [20], [23]. It basically works based on a set of vertices in the environment and builds the ideal trajectory considering the middle distances between workspace and obstacles as well as the distances between obstacles themselves. Employing this strategy allows robots equipped with lower accuracy rate of sensor detections to move between obstacles and reach their goals with the highest safety possibilities. While the earlier papers proposing the principle solutions of novel path planners, there is usually a departed change with later related approaches in terms of addressing issues and covering them. Therefore, later approaches achieved better results for building optimized trajectories. For instance, the local minima as one of the major disadvantages for using *Potential Field* planning method was addressed in [5], and [11]. Another issue that was resolved is the problem of using a single attraction point in workspace which leads to having difficulties with producing the resultant forces in environments including several closely located obstacles is reported in [18]. In order to fix the mentioned problems, [3], and [9] have proposed modifications and upgrades on the potential field's originally constructed formulation by using different functions such as harmonic functions: [7], [10], [24].

In order to reach higher optimization levels for path planners in terms of reliability, security, and trajectory length, several research articles from a variety of different groups of researchers have focused on incorporating many different approaches in form of hybrid solutions, with the sole intention of constructing more powerful planners with higher performances. Hybrid planning approaches are generally built based on a mixture of the key features that are adopting from different path planners in form of unique solutions with the purpose of treating problems that are involved with planners in calculating and constructing the ideal trajectories: [1], [4], [6], [13], [21], [22], [25].

This research article introduces an offline path planner for single point robots that incorporate multi-layer strategies with the purpose of minimizing the trajectory length between initial to the goal configuration with respect to maintaining trajectory safety and saving the needed hardware resources to analyze the environment and achieve the ultimate trajectory in a reasonable time. Since the method rapidly optimizes for the path we coined it *Rapidly Optimizing Mapper* (ROM). In the next section we briefly explain the main parameters and components that are needed to construct our planner along with the key feature roles for each followed by illustrations of our novel offline path planner in the subsequent section.

II. ALGORIHMIC FORMULATION FOR ROM

In order to build an efficient path planner which is able to build collision less trajectories for a mobile robot in a variety of different workspaces that are furnished to the planner with various obstacle arrangements, successfully, considering salient elements that play vital roles in robot navigation toward goal is essential. We categorized the most important elements that are necessary to be taken into account, while fabricating our adequate path planner in terms of building a collision free and optimal trajectory, into two parts. Safety concerns and optimality consideration are our two major deliberations in terms of the shortest possible length between initial to the goal configurations. We will express these concepts in more details later in this section. We considered a three phase algorithm for our path planner in general, including: *processing the workspace*, *graph conversion of workspace*, and *optimal trajectory determination*. Figure 1 capitulates our planner phases along with steps that belong to each phase.

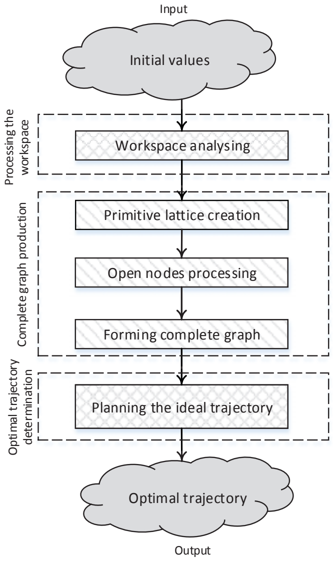

Figure 1. The general phases of our path planner

The planner algorithm computes the optimal trajectory based on the primitive adjustments for the related values that will be given at the beginning of the process of planning. These values empower the planner algorithm to consider the optimal trajectory based on the workspace and robot specifications. In other terms, equipping robots with different types of sensors having various detection capacities lead us to consider proper measurements in determining the trajectory to allow the robot to pursue a collision avoidance trajectory toward goal, successfully. The main objective for the first phase of our planner is to analyze the workspace elements such as obstacles with the purpose of determining and scanning perimeters of obstacles to locate their boundary edges and to provide necessary data for the next phase of planner. As it was discussed earlier in this section, in order for the planner to accomplish its task to build a flawless trajectory, we considered defining attributes that are directly incorporated in defining a safe and reliable path toward the goal. The safety property relies on considering proper boundaries around obstacles in regards to the robot's sensor equipment strength in terms of sensitivity to detecting and recognizing surrounding objects and also accuracy of detecting obstacle edges along with the robot's ability to maneuver in workspace via considering a proper distance to allow the mobile robot to adjust its direction toward following the computed trajectory. To fulfill the safety criterion, we defined *Standoff Distance* (SD) measurement. SD is an adjustable variable that is empirically determined based on the robot's sensor equipment specifications. In other words, the path planner algorithm computes the trajectory with consideration of the vertical length of SD value. The more mobile robot has poor obstacle edge detection as well as low degree of accuracy for detecting objects, the larger value for SD are applicable. An offline path planner by default analyzes the workspace and computes the proper trajectory prior to robot movement toward goal. Our planner builds the trajectory based on considering an abstract straight line from the initial to the goal configuration at each cycle of analyzing the workspace. In case where planner is confronted with crossing obstacles by the mentioned virtual straight line toward goal, the planner algorithm enters the phase of analyzing encountered obstacle accordingly. To fulfil obstacle inspection stage, the planner uses the *Degree of Traverse* (DOT) concept. The DOT value, quantifies the rate of obstacle surface scanning sensitivity. The DOT value regulation depends on the workspace size along with obstacle primitives as well as the maneuvering skills rate of mobile robot. In some certain situations, the robot may need to move toward obstacle surfaces. When it comes to the skills of the robot in terms of safely maneuvering through obstacle primitive movements, considering a proper distance around obstacles as secure boundaries is essential. These boundaries have to be large enough to allow the robot to adjust its path toward the determined trajectory without collisions. The *degree of surface traversal* (DOST) is the measurement that we considered to achieve this objective. The value for the DOST that is acceptable will vary depending on the distance that robot needs to adjust its direction successfully. The lower rate of accuracy for a robot to change its direction corresponds with considering higher value for DOST. The algorithm steps of the workspace processing unit of the planner is illustrated with the following five step procedure:

1. The algorithm sets primitive variables related to the SD, DOT, and DOST, along with the start and goal configurations,
2. It then considers a virtual straight line from start point to the goal configuration,
3. The first obstacle that intersects with the virtual line in step 2 in at least one hit point will be categorized as a roadblock obstacle,
4. For roadblock obstacles, the planner algorithm uses the analysis phase to scan the surface of the roadblock obstacles with the purpose of determining obstacle side edge nodes. In order to fulfil this aim, the planner adopts the value determined for the DOT at the initialization phase. The initial angle that planner considers for obstacle surface examination is 0. At the beginning of the process, the algorithm inspects virtual paths in both sides of the obstacle hit point for the value resulting from accumulated of the latest angle and the DOT value. If at least one of the virtual rays intersects the same obstacle in at least one hit point, the planner repeats the process of the roadblock obstacle surface scanning. In order to obtain the roadblock obstacle side edge nodes, the planner, at the beginning, considers the first virtual paths from both sides of the primitive hit point that shares no hit points with the same obstacle. It then reaches to the unsecure roadblock side edge nodes, which are calculated based on the nearest vertical distance from the surface of the obstacle. The planner appraises the roadblock side edge nodes by accumulating the unsecure roadblock side edge nodes distance and the SD value adjusted at the initiate phase of the planner. The last step of this phase of planner consists of considering the newly calculated roadblock side edge nodes as the new initial configurations that will be considered by the planner to route the trajectory from those points toward the goal, and
5. The process of analyzing workspace obstacles ends with reaching to the goal configuration. This unit of the planner supply the needed data for the next unit in order to revise and to form the complete graph. The information produced at this unit includes the calculated roadblock side edge node points along with the initial start and goal configurations.

As it was outlined in the planner first unit procedure, the planner analyzes the workspace and recognizes and hence, selects those obstacles that are blocking the abstract straight paths from initial point toward goal configuration to be processed at the second unit of the planner algorithm. The major task of the second phase of our planner, as recently expressed, is to use the data produced in the first phase with the sole purpose of constructing a completed graph based on forming a lattice of nodes located on the edges of roadblock obstacles starting from primitive initial point and ending to the goal configuration. To fulfil the second stage of our planner, we conceived of the following three general attributes that the planner benefits to route an accurate and reliable complete graph toward goal.

Node visibility: Node visibility is the primitive condition that is met where pairs of nodes located on either side edges of roadblock obstacles or on their surfaces respectively, when a straight ray crossing from both nodes does not intersect any obstacles in the workspace. In other words, two nodes are considered to be visible, if there is a possibility to connect them through a straight line without intersecting any obstacles.

Visible pathways: A visible pathway consists of a group of visible nodes located consecutively on the same path. The following equation illustrates the expressed condition:

$$\begin{aligned}\forall \Big[\big((\mathrm{n_i}, \mathrm{n_j}, \mathrm{n_k}) \in \mathrm{VN_s}\big) \&\&\big(\mathrm{VN_s} \in \mathrm{W_{O_s}}\big)\&\&\big(\mathrm{n_i n_j} \in \mathrm{C_x}\big)\&\&\big(\mathrm{n_i n_k} \\ \in \mathrm{C_y}\big)\&\&\big(\mathrm{n_j n_k} \in \mathrm{C_z}\big)\Big], \text{if } \Bigg[\Bigg(\bigg(C_x \cap \bigcup_{f=1}^{p} W_{O_f}\bigg) \\ = \emptyset\Bigg) \&\& \Bigg(\bigg(C_y \cap \bigcup_{f=1}^{p} W_{O_f}\bigg) \\ = \emptyset\Bigg) \&\& \Bigg(\bigg(C_z \cap \bigcup_{f=1}^{p} W_{O_f}\bigg) = \emptyset\Bigg)\Bigg], then\ VP_t \\ = \{\mathrm{n_i}, \mathrm{n_j}, \mathrm{n_k}\} \&\& (\mathrm{n_i}, \mathrm{n_k}) \in SVP_t \qquad (1)\end{aligned}$$

In equation 1, $\mathrm{VN_s}$ indicates the visible nodes that belong to the obstacle $\mathrm{W_{O_s}}$. $\mathrm{C_x}$, $\mathrm{C_y}$, and $\mathrm{C_z}$ are illustrating connections between $\mathrm{n_i}$ and $\mathrm{n_j}$, $\mathrm{n_j}$ and $\mathrm{n_k}$ *, and* $\mathrm{n_i}$ *and* $\mathrm{n_k}$, roadblock obstacle side edge nodes of $\mathrm{W_{O_s}}$, respectively. VP_t indicates the t^{th} path segment, which consists of $\mathrm{n_i}$ and $\mathrm{n_j}$ as its side points, SVP_t.

Uncompleted node: The first phase of the planner task is to determine the roadblock obstacles side edge nodes. One of the major tasks of the second phase of the planner is to recognize roadblock obstacles side edge nodes belonging to the same obstacle that are not connected to one another through at least one direction toward the surface of the roadblock obstacle respectively.

The steps of the second phase of our planner is illustrated in the following four step procedure:

1. Identifying roadblock obstacles side edge nodes connection,
2. Eliminating uncompleted nodes generated by the analysis phase of the planner by connecting them together toward both directions of the surface of the obstacle with the vertical length of SD,
3. Simplification of nodes forming visible pathways by disposing the nodes located in between both sides of visible paths, and
4. Adjusting distances between pairs of nodes for the resultant graph which will be constructed from exploiting the previous steps of this phase of planner algorithm.

The second phase of our planner constructs a complete graph based on considering available paths from initial to goal configuration. This graph will be used in the subsequent phase of the planner algorithm with the purpose of recognizing optimal trajectory. In order to increase the performance of our planner in terms of building proper trajectory, we have benefitted from the application of the *Dijkstra* algorithm for the third phase of our planner. The main function of the last phase of our planner is to analyze the complete graph and hence, determining the shortest path toward goal configuration using the *Dijsktra* algorithm that we assume is common knowledge.

The next section is dedicated to validation of the functionality of our planner performance to determine optimal trajectories in workspaces consisting of various arrangements for initial and goal configurations along with obstacles specifications such as sizes, shapes, and locations.

III. VERIFICATION AND VALIDATION OF ROM

In order to validate the performance of our planner, we considered four different exemplar scenarios. These scenarios in diverse forms of various workspaces along with the results obtained from applying our planner to each are indicated in the following three case studies:

Case study 1:
For our first workspace map, we considered locating a variety of different obstacles in terms of shapes, connectivity, and locations in the form of crowded obstacle arrangements scenario to assess our planner performance on building trajectory in complicated situations. As shown in Figure 2, some obstacles are connected to each other and hence, form larger obstacles while other obstacles are isolated from one another. The size of the workspace is 100 by 100 units. In order to maximize the number of roadblock obstacles from start to the goal configurations, we considered the start point at the top left corner of the map at (70, 15) point and the goal configuration at the bottom right corner of the workspace at (75, 88) coordinate point. The first workspace map along with the start and goal configurations are illustrated in the following figure 2.

Figure 2. The sample map of the workspace for the case study 1

Figure 3 reflects results from applying our path planner to the related workspace map.

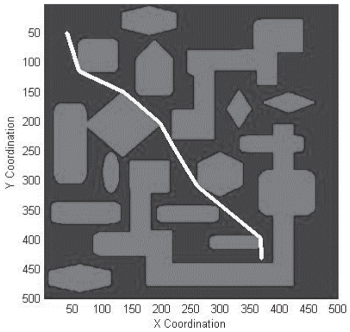

Figure 3. The first sample map of the workspace for the case study 1

The optimal pathway in the form of a collision less trajectory is shown in bright path starting from start location and ending at the goal configuration. Figure 3 reveals that our path planner is able to build a collision free trajectory in crowded workspaces in terms of obstacle numbers, sizes, shapes and arrangements, successfully.

Case study 2:
The second workspace consists of obstacles connected to each other to form a maze pattern shown in Figure 4.

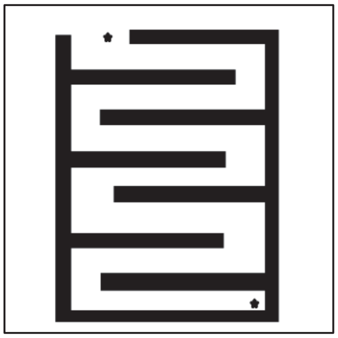

Figure 4. The second pattern of the workspace map for the case study 2

The following figure 5 shows the optimal path resulting from our planner operation on the related workspace illustrated in figure 4.

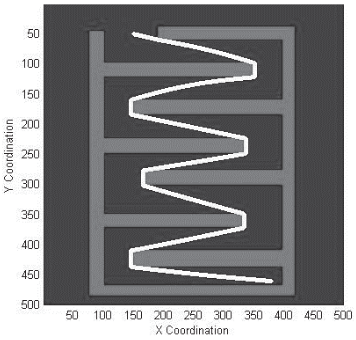

Figure 5. The first sample map of the workspace for the case study 2

As it shown in the figure 5, our planner is able to route the optimal trajectory benefiting different techniques in the form of a unique algorithm to calculate and refine the shortest path possible. This is done by adopting the simplifying skills to keep the side visible nodes and eliminate other nodes that are located in between. As a result, the final constructed graph consists of paths that are either in the form of straight lines that have the minimum distance between pairs of nodes or located on the surface of roadblock obstacles.

Case study 3:
The final considered case study is fabricated based on a workspace including spiral shapes obstacles. We used this scenario with the sole purpose of evaluating our planner skills to route shortest collision less trajectory toward the surface of the roadblock obstacles. The following figure 6 is the showcase of the third environment including obstacles, start, and goal points.

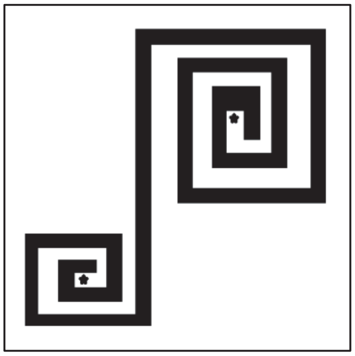

Figure 6. The third pattern of the workspace for the case study 3

Our path planner generates the optimal trajectory in the form of the shortest collision free path from start into goal configurations as indicated in the following figure 7.

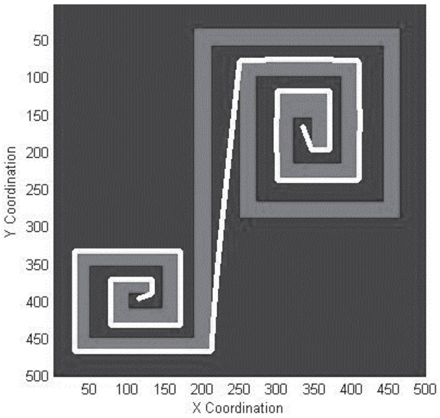

Figure 7. The first sample map of the workspace for the case study 3

The optimal trajectory is marked as a bright pathway crossing from the surface of the spiral obstacle with vertical distance with the *Standoff Distance* value. Our planner considered a portion of the optimal trajectory to be on the surface of the roadblock obstacle as indicated in figure 7. This is because the virtual ray crossing from roadblock obstacle side edge nodes were continuously intersecting the same obstacle and hence, a roadblock obstacle surface scanning with the length of DOST enforced by the planner with the purpose of obtaining new side edge nodes.

The results obtained from applying our planner to all considered three cases revealed that our path planner is able to route optimal trajectory for all cases flawlessly. In addition, our planner is able to build the path without restriction to specific workspace arrangements. The way we fabricated our planner enables it to consider all possible workarounds to eliminate constraints, which elevate its performance to perform in any scenarios to analyze and refine trajectories from initial to the goal configurations.

IV. CONCLUSIONS

A novel offline path planner for single point robots has been presented that we dubbed ROM. ROM is able to plan a collision free route from start point to the goal configuration. The planner benefits a series of different methods and parameters to analyze the workspace and compute the shortest collision free trajectory toward goal. We increased the performance of the planner algorithm dramatically by considering techniques that overlook irrelevant obstacles from the workspace. The strategy of focusing on certain obstacles instead of all available obstacles in the environment leads to operation on fewer numbers of obstacles, and hence, reduces the required number of operations to analyze the workspace and to regulate the trajectory from start into goal configuration, more efficiently.

V. FUTURE WORK

Thus far, we focused on building ROM for moving offline robots that is able to analyze the workspace and construct an optimal trajectory in terms of computing the shortest length from initial points to the goal configuration, successfully. In addition, in regards to the safety of the processed path, our proposed planner is able to plan a collision free route from start to the goal configuration. In order to accomplish the security aspect of determining the optimal trajectory, our path planner benefits from use of several parameters that we developed for our planner, as expressed in the previous sections. Throughout this research, we appraised our novel path planner performance by applying it on many prototypically complex situations with different workspace objects arrangements. Our planner revealed that it is able to route optimal collision avoidance trajectories in all cases, flawlessly. As the next phase of our planner performance evaluation in future work, we plan to assess our planner performance by comparing it with other offline path planners such as *Potential Field* and *Rapidly-exploring Random Trees* path planners. Our intent is to apply our planner along with other offline planners in different scenarios with the sole purpose of acknowledging the abilities of our path planner on operating workspace objects such as obstacles in terms of analyzing the environment as well as constructing the ideal collision less trajectories.

REFERENCES

[1] Abinaya, S., Hemanth Kumar, V., Srinivasa Karthic, P., Tamilselvi, D., and Mercy Shalinie, S., “Hybrid Genetic Algorithm Approach for Mobile Robot Path Planning” in Journal of Advances in Natural and Applied Sciences, Vol.8, No.17, Special 2014, pp.41-47, 2014.

[2] Andrews, J.R. and Hogan, N., "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator." Control of Manufacturing Processes and Robotic Systems, Eds. Hardt, D. E. and Book, W., ASME, Boston, pp.243-251, 1983.

[3] Arámbula Cosío F. and Padilla Castañeda M.A., “Autonomous Robot Navigation using Adaptive Potential Fields” in Mathematical and Computer Modelling, Vol.40, Issue 9-10, pp.1141-1156, 2004.

[4] Bashra K., and Oleiwi and Hubert Roth, “Modified Genetic Algorithm Based on A* Algorithm of Multi Objective Optimization for Path Planning” in Journal of Automation and Control Engineering, Vol.2, No.4, pp.357-362, 2014.

[5] Borenstein J. and Koren Y., “The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots” in IEEE Transactions on Robotics and Automation 7, pp.278-288, 1991.

[6] Chaari, I., Koubaa, A., Trigui, S., Bennaceur, H., Ammar, A., and Al-Shalfan, K., “SmartPATH: An Efficient Hybrid ACO-GA Algorithm for Solving the Global Path Planning Problem of Mobile Robots” in International Jounal of Advanced Robotics Systems, pp.1-15, 2014.

[7] Connolly C.I., Burns J.B., and Weiss R., “Path Planning using Laplace's Equation” in Proceedings of the lEEE Conference On Robotics and Automation, Cincinnati, OH, pp.2102-2106, 1990.

[8] Dijkstra, E.W., "A Note on Two Problems in Connexion with Graphs" in Numerische Mathematik 1, pp.269–271, 1959.

[9] Ge S.S. and Cui Y.J., “New Potential Functions for Mobile Robot Path Planning” in IEEE Transactions on Robotics and Automation, Vol.16, No.5, pp.615-620, 2000.

[10] Guldner J., Utkin V., and Hashimoto H., “Robot Obstacle Avoidance in N-Dimensional Space using Planar Harmonic Artificial Potential Fields” in Journal of Dynamic Systems, Measurement, and Control, Vol.119, pp.160-166, 1997.

[11] Grefenstette J. and Schultz A.C., “An Evolutionary Approach to Learning in Robots” in Proceedings of the Machine Learning Workshop on Robot Learning, International Conference on Robot Learning, New Brunswick, N.J., pp.65-72, 1994.

[12] Hwang, F.K., “An O(nlogn) Algorithm for Rectilinear Minimal Spanning Trees” in the journal of association of computer, pp.177-182, 1979.

[13] Ju, M., Wang, S., and Guo, J., “Path Planning using a Hybrid Evolutionary Algorithm Based on Tree Structure Encoding” in The Scientific World Journal, Vol.2014, Article ID 746260, 8 pages, 2014.

[14] Kavraki, L.E., Svestka, P., Latombe, J.C., and Overmars, M.H., "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", IEEE Transactions on Robotics and Automation Vol.12, No.4, pp.566–580, 1996.

[15] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots" in IEEE International Conference on Robotics and Automation, St. Louis, Missouri, pp.500-505, 1985.

[16] Kirkpatrick, D.G., “Efficient Computation on Continuous Skeletons” in Proceedings of the 20th IEEE Symposium on Foundations of Computer Science, San Juan, pp.18-27, 1979.

[17] Klein, R., “Abstract Voronoi Diagrams and Their Applications (Extended Abstract)” in H. Noltemeier (ed.), Proceedings Computational Geometry and its Applications (CG "88), Wfirzburg, Lecture Notes in Computer Science, Vol.333, Springer-erlag, Berlin, pp.148-157, 1988.

[18] Koren Y., and Borenstein J., “Potential Field Methods and Their Inherent Limitations for mobile robot navigation” in Proceedings of the IEEE International Conference on Robotics and Automation, pp.1398-1404, 1991.

[19] Lee, D.T. and Drysdale, R.L. III, “Generalization of Voronoi diagrams in the plane” in Socity for Industrial and Applied Mathematics journal of computing, Vol.10, Issue.1, pp.73-87, 1981.

[20] Lee, D.T. and Wong, C.K., “Voronoi Diagrams in $L_1(L_\infty)$ Metrics with 2-Dimensional Storage Applications” in Socity for Industrial and Applied Mathematics Journal of Computing, Vol.9, Issue.1, pp.200-211, 1980.

[21] Loo, C.K., Rajeswari, M., Wong, E.K., and Rao, M.V.C., “Mobile Robot Path Planning using Hybrid Genetic Algorithm and Traversability Vectors Method” in Intelligent Automation and Soft Computing, Vol.10, No.1, pp.51-64, 2004.

[22] McFetridge, L. and Ibrahim, M.Y., “New Technique of Mobile Robot Navigation using a Hybrid Adaptive Fuzzy-Potential Field Approach” in Computers and Industrial Engineering, Vol.35, Issues (3–4), pp.471–474, 1998.

[23] Shamos, M.I. and Holy, D., “Closest-Point Problems” in Proceedings of the 16th IEEE Symposium on Foundations of Computer Science, Berkeley, CA, pp.151-162, 1975.

[24] Utkin V.I., Drakunov S., Hashimoto H., and Harashima F., “Robot Path Obstacle Avoidance Control via Sliding Mode Approach” in Proceedings of the lEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka, Japan, pp.1287-1290, 1991.

[25] Yao., Z. and Ren, Z., “Path Planning for Coalmine Rescue Robot based on Hybrid Adaptive Artificial Fish Swarm Algorithm” in International Journal of Control and Automation, Vol.7, No.8, pp.1-12, 2014.