

Path Planning of a Mobile Robot in Grid Space using Boundary Node Method

R. A. Saeed^a and Diego Reforgiato Recupero^b

Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy

Keywords: Robot Path Planning, Boundary Node Method, Path Enhancement Method, Autonomous Mobile Robot.

Abstract: This paper presents a new off-line path planning method for a mobile robot to generate an optimal or near-optimal collision-free path between starting and goal points in a given working environment with obstacles. In a new method called Boundary Node Method, the robot is simulated by nine-node quadrilateral element, the centroid node represents the robot's location and it moves with eight-boundary nodes in the working environments. A robot is exploring an environment with the help of the node's potential value at each location, where the potential value is calculated based on the proposed potential function. The proposed method is capable of generating the initial collision-free path for a mobile robot safely and quickly. Subsequently, an additional new method called Path Enhancement Method is used to find shortest path by reducing the overall initial path length. The simulation results indicate that this method can successfully generate an optimal or near-optimal collision-free path efficiently.

1 INTRODUCTION

Path planning has been widely applied in many robotic applications (Atzeni and Recupero, 2018) to perform various tasks that humans could not accomplish, such as nuclear facilities (Chao et al., 2018), space exploration (Shum et al., 2015), rescue mission, landmines and enemies in war field (Zhang et al., 2013), etc. Therefore, the problem of path planning has attracted many researchers' attention (Zhang et al., 2013; Kala et al., 2011). The objective of path planning is to construct a collision-free path for mobile robots to move from a starting point to destination point in a given working environment with obstacles and optimizing it with respect to some criteria, i.e. the walking distance, the walking time, the energy consumption, etc (Leena and Saju, 2014; Han and Seo, 2017; Victerpaul et al., 2017). The existing methods are mainly categorized into classical and heuristic path planning (Zhang et al., 2013; Han and Seo, 2017). The classical methods involve finding a set of defined steps to search for the solution from an initial position to a goal position, they include cell decomposition, potential field method, subgoal network and road map (LaValle, 2006). It

has been found that the classical methods have some disadvantages like high computational cost, trapping into local minima and high time complexity in high dimensions (Zhang et al., 2013; Brand et al., 2010; Leena and Saju, 2014). Therefore, many heuristic methods have been proposed, i.e. Genetic Algorithm (*GA*) (Lamini et al., 2018), Particle Swarm Optimization (*PSO*) (Zhang et al., 2013), Artificial Neural Networks (*ANNs*), Ant Colony Optimization (*ACO*) (Brand et al., 2010; Mac et al., 2016), and Fuzzy Logic (*FL*) (Brand et al., 2010). Heuristic path planning methods are computationally more efficient with better performances in term of path distance, obstacle avoidance, and elapsed time (Brand et al., 2010; Leena and Saju, 2014; Mac et al., 2016), but these algorithms are not guaranteed to provide the optimal solution (Rintanen, 2006; LaValle, 2006). The combinatorial path planning methods in continuous space can solve many path planning problem and construct optimal solution efficiently (Bi et al., 2008; LaValle, 2006; Huang and Tsai, 2011; Montiel et al., 2015; Contreras-Cruz et al., 2015; Song et al., 2016).

Additionally, many researchers made some modifications and extensions to improve the performance and to overcome the limitations of the existing methods for solving the path planning problem. For instance, authors in (Wang et al., 2018) proposed a novel learning-based multi-*RRTs* (*LM - RRT*) ap-

^a  <https://orcid.org/0000-0002-7385-0167>

^b  <https://orcid.org/0000-0001-8646-6183>

proach for robot path planning in complex environments with narrow passages. A new repulsive potential function is proposed by (Ge and Cui, 2000) to solve the problem of non-reachable goals with obstacles nearby (*GNRON*) in potential field method. The shortest path with the minimum time required for the global path planning is investigated in (Shiltagh and Jalal, 2013) by using Modified *PSO*. Authors in (Fu et al., 2018; Guruji et al., 2016; Duchoň et al., 2014) proposed an improved version of *AStar* algorithm to overcome inherent drawbacks of the original *AStar*. A new fitness function for *GAs* is proposed by (Lamini et al., 2018) for optimizing energy consumption for a mobile robot. Some optimal path planning algorithms are presented in (Jan et al., 2008) for navigating mobile robot among obstacles and weighted regions.

Many of the existing methods for robot path planning are able to find a path from an initial position to a given target, but they are either not accurate enough or their efficiency is not sufficient. Researchers have always been seeking a better solution to improve the performance of the existing path planning methods. A list of goals that researchers of several previous works have been pursuing is the following: improve the accuracy (Liu et al., 2018; Shum et al., 2015; Goyal and Nagla, 2014), improve the efficiency (Hu and Yang, 2004), increase safety (Chao et al., 2018; Shum et al., 2015; Fu et al., 2018), increase the capability (Shiltagh and Jalal, 2013), reduce the processing time (Duchoň et al., 2014; Guruji et al., 2016), overcome the non-reachable goal problem (Ge and Cui, 2000), pass through narrow passages (Jan et al., 2008).

In the literature, the problem of path planning for mobile robots has been widely discussed and researchers explored a variety of solutions. However, several important gaps and limitations remain to be addressed. For example, in several works, the computational time is still too high because of the processing a lot of unnecessary points. Moreover, the search for an optimal path might not succeed (Han and Seo, 2017). And also, there are many methods that use random operation to produce a set of solutions for each independent run. Then, in order to find the optimized solution, all of these different solutions are selected, combined and replaced. This process requires a lot of computational time. Based on the limitations and research gaps as previously explained, we investigate a new efficient method called Boundary Node Method (*BNM*) to generate a feasible path for the robot to move from starting point to ending point in the workspace without colliding with any obstacles. The *BNM* is capable of finding the path for mo-

bile robot effectively and efficiently in terms of path length and computational time, even if the complexity of the environment is increased. Moreover, a valuable benefit of this method is its simplicity and can be applied in a grid environment efficiently.

The remainder of this paper is structured as follows: Section 2 introduces the problem statement. In Section 3, a brief description of the proposed method for generating feasible path is introduced. In Section 4, the application of the proposed method is presented, and then the evaluation results are reported and discussed. Final conclusions and remarks for future research improvements are provided in Section 5.

2 PROBLEM FORMULATION

The path planning problem is formulated as follows: for a given two-dimensional (*2D*) robot working space $C \subset R^2$, the region of the working space occupied by obstacles is denoted by C_{obs} and the region of the working space free of obstacles is represented by $C_{free} = C - C_{obs}$. Each position within C_{free} is reachable by the robot. The continuous workspace is divided into square grid cells. Each grid cell of the space has integer coordinates of the form $C(x,y) \in C$, $1 \leq x \leq n$, $1 \leq y \leq m$, which either corresponds to a navigable area $C(x,y) \in C_{free}$ or a space occupied by obstacles $C(x,y) \in C_{obs}$. We assume that all information related to the workspace is known in advance, obstacles are also assumed to be fixed in their position in the workspace. Each grid cell $C(x,y)$ in C_{free} has a potential value $E(x,y) \in E$, which is calculated based on the potential function. The robot is requested to move from starting point (C_s) to goal point (C_g) in the workspace without colliding with any obstacles. It is expected that the robot reaches the final destination point safely through the shortest walking route within minimum computational time.

3 BOUNDARY NODE METHOD

In *BNM*, the robot is simulated by nine-node quadrilateral element, the centroid node represents the robot's location and it moves with eight-boundary nodes in the workspace. The robot position in the workspace is denoted by $C_r(x_r, y_r) \in C_{free}$. The robot moves forward and changes its motion direction based on the potential values and features of boundary nodes. All Way-points w visited by the robot sequentially represent the obtained initial feasible path (*IFP*). For better clarity, these way-points are connected into a continuous path, the line segment that

connects two way-points sequences is represented by $P_{l,l+1} \subset IFP$. A complete path IFP is formed by concatenation of all inter-line segments $P_{l,l+1}, 1 \leq l \leq w-1$ as follows: $IFP = [P_{1,2}, P_{2,3}, \dots, P_{w-1,w}]$. The robot's working space used in this study is divided into square grid cells, and the centre of all grid cells in the given workspace meets the following equation:

$$C = \sum_{x=1}^n \sum_{y=1}^m C(x,y) \quad (1)$$

where n and m represent the width and height of the workspace. $C(x,y)$ represents the location of the grid cells in the workspace, and the length of each grid cell is equal to 1 *unit*.

After constructing a model for the workspace, the potential value for each grid point $E(k)$ ($1 \leq k \leq N$), ($N = n \times m$) is calculated based on the new proposed potential function as illustrated in Figure 1. The potential function for several different starts and goal positions are shown in Figure 2. The potential function is used to direct the robot toward the C_g , and it has the lowest potential value at a final destination point and the potential value increases as the robot move further away. As shown in the Figure 2, the line's colour represents the potential value, i.e. the blue line corresponds to the lowest potential value and the red line corresponds to the highest potential value. The shape of the potential function and obstacles in a 2D model of the robot's workspace is presented in Figure 3. As shown in the figure, a number of static obstacles (1×1 *unit*) are distributed at different locations in the workspace. The centre of obstacle's cells is represent by Cartesian coordinates $C_{obs}(h,l)$, ($h = 1 \dots 2, l = 1 \dots O$), where $h = 1$ and $h = 2$ corresponding to x and y -coordinate, respectively, and O representing the number of obstacles.

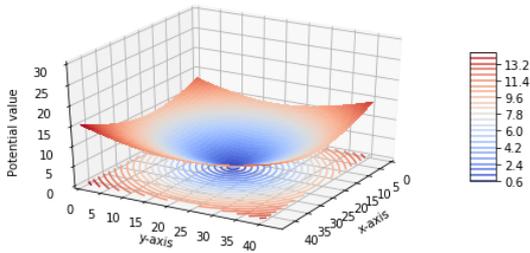


Figure 1: Potential value for each cell in the workspace in 3D view with contour plot. The size of the workspace is 44×44 , and the C_g is located (20,20).

3.1 Simulate the Robot

In this method the robot is simulated by nine-node quadrilateral element (see Figure 4a), the centroid node represents the robot's location (see Figure 4b)

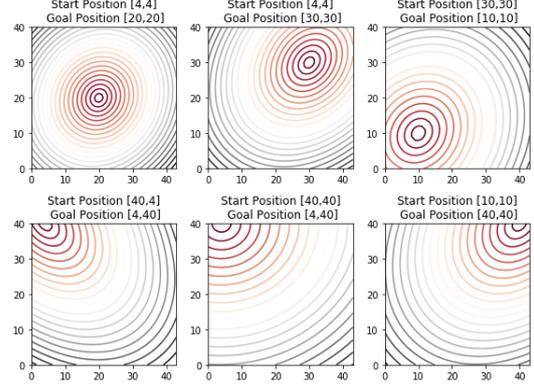


Figure 2: Potential function with considering several different start and goal Locations.

and it moves with eight-boundary nodes exploring in the workspace (see Figure 4c) to find the IFP . The nodes are denoted by $p(q)$, ($q = 1 \dots 9$), and their locations can be formulated by using Equation 2. In simulated model, the node $p(5)$ represents the robot's location and the other nodes $p(1 \rightarrow 4)$ with $p(6 \rightarrow 9)$ represent the boundary nodes which are distributed uniformly around the robot location as shown in Figure 4a. The current location of all nodes in the workspace at any iteration t is denoted by $p_1(t)$, meanwhile the robot location in the simulated model is denoted by $p_r(t) \in p_1(t)$. The set of x and y -coordinate of the robot and all boundary nodes are represented by two vectors, $x_1(t) = (x_{11}, x_{12}, \dots, x_{19})$ and $y_1(t) = (y_{11}, y_{12}, \dots, y_{19})$ respectively, $p_1(t)$ is formed by vertically concatenating $x_1(t)$ and $y_1(t)$, $p_1(t) = [x_1(t); y_1(t)]$.

$$p(q) = \begin{cases} x,y & q=5 \\ (x+v_x, y), (x, y+v_y), \\ (x-v_x, y), \\ (x, y-v_y) & q = 2, 4, 6, \text{ and } 8 \\ (x+v_x, y+v_y), (x-v_x, y+v_y), \\ (x-v_x, y-v_y), \\ (x+v_x, y-v_y) & q = 1, 3, 7, \text{ and } 9 \end{cases} \quad (2)$$

where x and y are the distance between the center of the grid cell and x and y -axis, v_x and v_y are the horizontal and vertical distance between the robot and boundary nodes, we assume that $v_x = v_y = 1$ *unit*.

In each iteration t , the robot and boundary nodes $p(q)$, ($q = 1 \dots 9$) are moved in one particular direction in the workspace. The current robot's location with boundary nodes can be described as $p_1(t) = [x_1(t); y_1(t)]$, and the new updated location of all nodes denoted by $p_2(t) = [x_2(t); y_2(t)]$. The robot

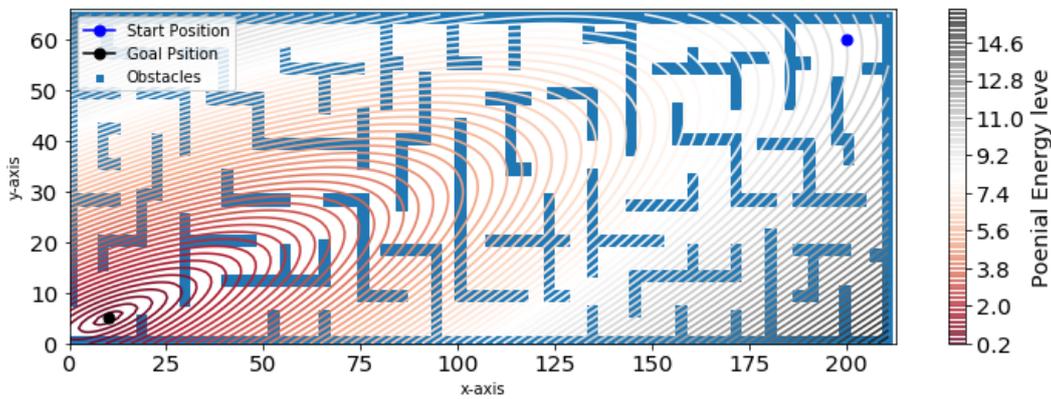


Figure 3: 2D model of the robot's workspace.

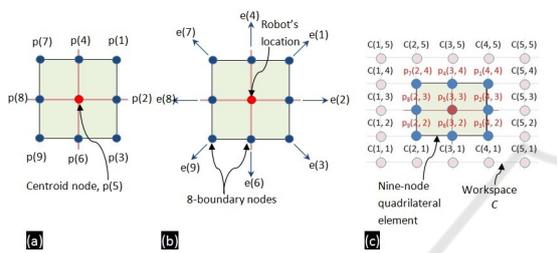


Figure 4: A nine-node quadrilateral element (a) along with its motion directions (b) and exploration location in the workspace (c).

interference the obstacles when the distance between the robot and the obstacles is less than 1 unit. Therefore, the robot and boundary nodes require to avoid obstacles and changing its moving direction by selecting a new position in the C_{free} . Characteristics of boundary nodes, their position and potential values, guides the robot toward the goal location. On the other hand, the characteristics of boundary nodes can help the robot to avoid the obstacles. To explain how the robot avoids the obstacles and changing its motion direction with the help of boundary nodes, consider a simple workspace example shown in Figure 5. As shown in the figure, the robot position $p(5)$ is represented by a red circle object and the boundary nodes $p(1 \rightarrow 4)$ and $p(6 \rightarrow 9)$ are represented by green circle object.

Initially, the robot's position $p(5)$ is coincide with C_s (see Figure 5). In the first iteration $t = 1$, the robot with boundary nodes change their positions from current position $p_1(1)$ to new position $p_2(1)$ toward the goal. In the second iteration $t = 2$, as the robot move toward the goal with boundary nodes, the nodes $p(1)$, $p(2)$, and $p(3)$ interference with the obstacles, as demonstrated in the Figure 5. Then the robot investigates the workspace to find the next position without colliding any obstacle. In this study, the direction of the robot motion could be to the left, right, forward or

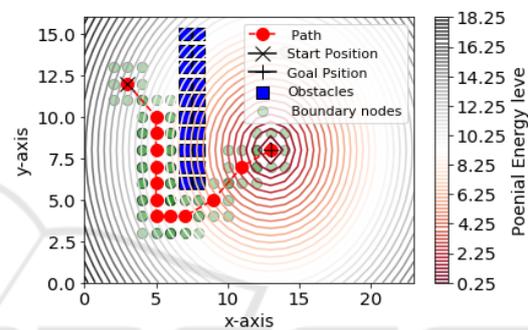


Figure 5: Demonstrates an example of path planning and obstacle avoidance for mobile robot in a static environment using BNM , where the sequence of the red circle objects represents the best solution of IFP .

backward. As shown in Figure 5, the robot can only move in the y -direction, either upward or downward direction. Changing the motion direction depends on the potential value, E . Because $E(4)$ is greater than $E(6)$, therefore the next position of the robot must be in the downward direction. The same procedure is repeated by shifting the robot to the downward direction until it passes the block of obstacles. The distance between the $p_r(t)$ and the C_g is decrease step by step until the robot reaches to the global minimum at the goal position. The simulation result of the BNM is the IFP , which is consist a set of way-points visited by the robot starting from the initial point and ending at the goal point. The BNM can generate the IFP safely and efficiently, but the path is not optimal in term of the total path length. Therefore, the PEM is introduced to reduce the number of way-points as well as the overall length of IFP .

3.2 Path Enhancement Method

This section introduces the PEM to obtain the shortest path (see Figure 6) from IFP (see Figure 5). As

shown in the figure, the way-points in the *IFP* are represented by a set of red circles object between C_s and C_g . The robot starts to move from the starting point and tracked through all the intermediate way-points until it reaches the goal point. The *PEM* method is applied to reduce the number of way-points as well as the overall length of *IFP*. The applied results show that the number of way-points between C_s and C_g has been reduced from 13 (see Figure 6) to 4 way-points (see Figure 6). The length of all line segments that connects all way-points sequentially to each other is representing the path length. The length of the shortest path U is calculated by using the following equation:

$$U = \sum_{i=1}^I (\text{sqrt}(u_{1x}(i) - u_{2x}(i))^2 + (u_{1y}(i) - u_{2y}(i))^2) \quad (3)$$

where U is the total path length, I is line segment number and $u_{1x}(i), u_{2x}(i), u_{1y}(i), u_{2y}(i)$ are represent the x and y cartesian coordinates of the left-hand and right-hand ends of the line segment $U(i)$.

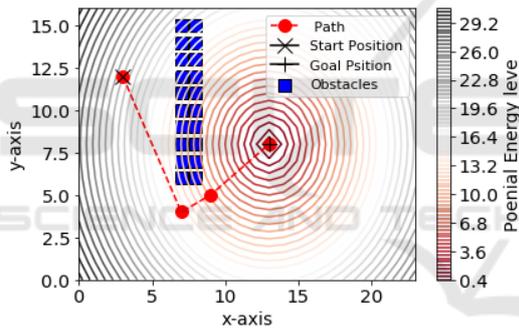


Figure 6: Constructing the shortest-path from way-points in a 2D workspace. Way-points defining the path are marked with red circles object. The red dashed line represents a shortest path found by using *BNM* and *PEM*.

4 SIMULATION

In this study, all simulations have been performed in *python*. All simulations results presented in this paper have been computed on a laptop Intel(TM) Core(TM) i5-8300H CPU, 2.3GHz, and 8GB RAM. Two different workspace scenarios with different complexity are designed. In the first scenario, several different start and goal Locations are considered. In all scenarios the size of workspace and obstacles scattered in the navigation space are same. Whereas, a complex environment with a high number of obstacles in different sizes is considered in the second scenario. The start and goal points are positioned in the free space

Table 1: The total path length of the initial and final path obtained by *BNM* and *PEM*.

Workspace No.	Total Path Length [unit]	
	<i>IFP</i>	Final Path
1	512.748	241.117
2	529.591	268.822

C_{free} . The proposed method was examined to find an optimal or near the optimal path from C_s to C_g . The simulation results of *BNM* and *PEM* are presented in subsections 4.1 and 4.2, respectively.

4.1 Simulation Results of BNM

This section presents the application results of the *BNM* for generating the *IFP* between C_s and C_g . The simulation results for all tested scenarios are shown in Figures 7 and 8. The achieved result of *IFP* is represented by a set of way-points $w(j)$, ($j = 1 \rightarrow J$). Each new point position $w(j+1)$ allocated after the current point position $w(j)$ on the path, where J represents the time in which the robot is reaching to the final destination point. In the first scenario, the way-points of the robot's motion path are represented by red circles object and for better clarity, these way-points are connected into a continuous path (see Figure 7). In the second scenario, the way-points of the robot's motion path are represented by the red solid line (see Figure 8).

From the figures, it is observed that the *IFP* allow the robot to move from C_s to C_g , and avoid obstacles successfully. As it can be seen from the results, *BNM* was able to overcome the local minima problem. The results show that *BNM* has been well applied to generate *IFP* for mobile robot and achieved important achievements. However, the path is not optimal in term of the total path length. In order to reduce the overall length of *IFP*, a new method *PEM* is introduced as explained in Section 3.2, and the application results are presented in the following subsection.

4.2 Simulation Results of PEM

This section presents the obtained results of the *PEM* to find the optimal or near-optimal path. The best obtained solutions are presented in Figures 9 and 10. Whereas the red dashed lines in Figure 9 and red solid lines in Figure 10 are represent the best-obtained solution. As shown in the figures, the proposed method can find the collision-free path that covers the least number of way-points to reach the final destination point. The computational results of the path length from the proposed method for the second scenario is provided in Table 1. As revealed in the table, the total path length from $[10, 5]$ to $[200, 60]$ and from $[200, 60]$

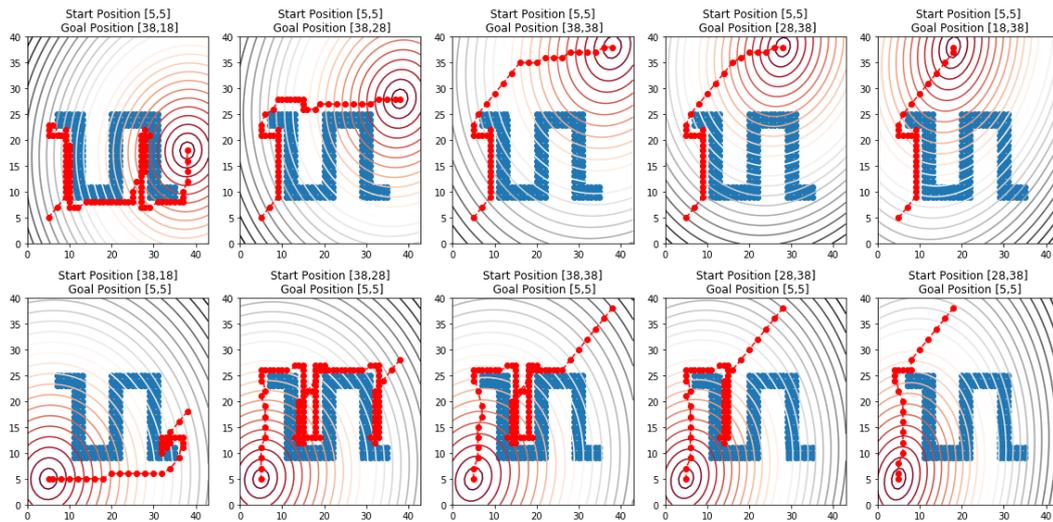


Figure 7: The IFP obtained by using *BNM* for the first scenario.

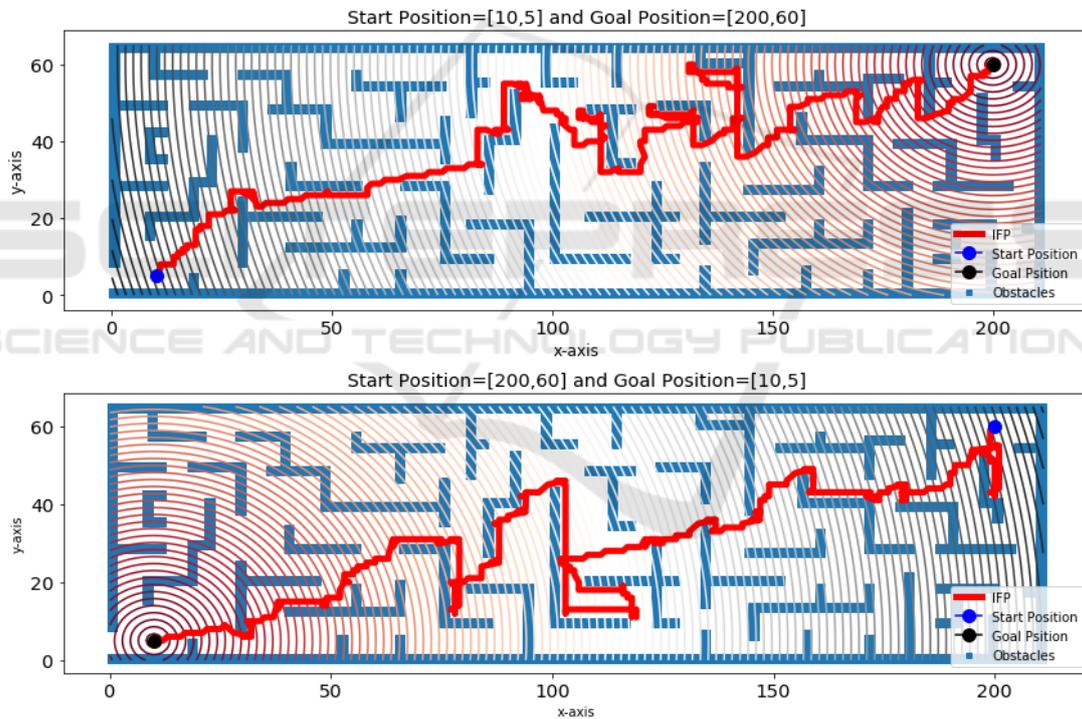


Figure 8: The IFP obtained by using *BNM* for the second scenario.

to [10,5] is significantly reduced, where the percentage of enhancement of the path length are 113% and 97% respectively.

5 CONCLUSIONS

In this paper we have introduced a new method called Boundary Node Method *BNM* for solving the path

planning problem for a mobile robot. The proposed method is applied to generate an initial collision-free path for the mobile robot to move from starting point to final destination point without colliding with any obstacles. Consequently, an additional method, Path Enhancement Method *PEM*, is used to find an optimal or near optimal collision-free path from the initial path by minimizing the overall path length. Both methods, *BNM&PEM*, are introduced to provide the

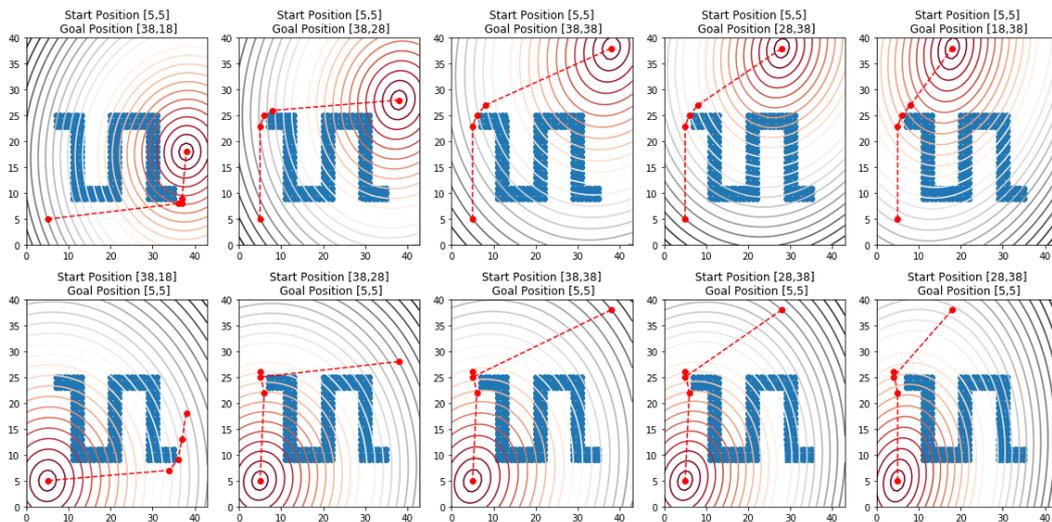


Figure 9: An optimal or near-optimal path obtained by applying *BNM&PEM* for the first scenario.

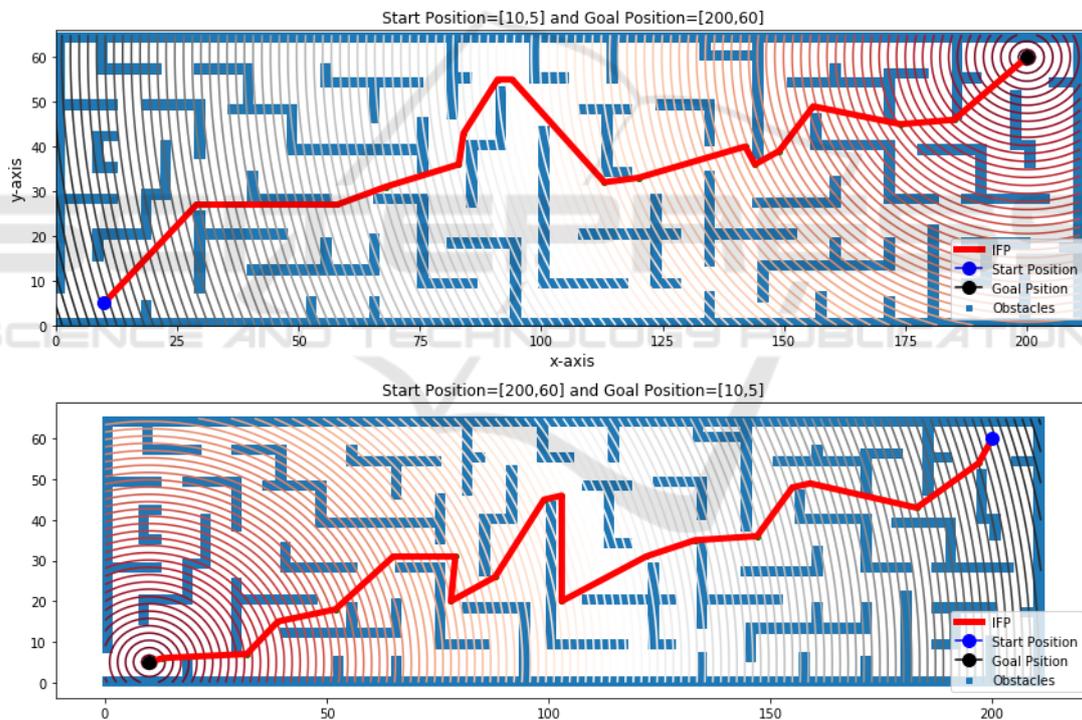


Figure 10: An optimal or near-optimal path obtained by applying *BNM&PEM* for the second scenario.

optimal path between start and goal points. Simulation results demonstrate that the *BNM&PEM* can generate the optimal or near optimal collision-free path for the mobile robot to navigate in a complicated environment, efficiently. As future work, we will deal with obstacles of different sizes and forms (irregular or arbitrary shape) and static or dynamic. Also, we will evaluate the performance of the other proposed types of the elements such as nine-node cir-

cular, seventeen-node quadrilateral, and thirteen-node octagonal. Finally, we are currently working on a validation of the proposed method on real robots, NAO from Aldebaran Robotics.

REFERENCES

- Atzeni, M. and Recuperero, D. R. (2018). Deep learning and sentiment analysis for human-robot interaction. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, pages 14–18.
- Bi, Z., Yimin, Y., and Wei, Y. (2008). Hierarchical planning approach for mobile robot navigation under the dynamic environment. In *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, pages 372–376. IEEE.
- Brand, M., Masuda, M., Wehner, N., and Yu, X.-H. (2010). Ant colony optimization algorithm for robot path planning. In *Computer Design and Applications (IC-CDA), 2010 International Conference on*, volume 3, pages V3–436. IEEE.
- Chao, N., Liu, Y.-k., Xia, H., Ayodeji, A., and Bai, L. (2018). Grid-based rrt for minimum dose walking path-planning in complex radioactive environments. *Annals of Nuclear Energy*, 115:73–82.
- Contreras-Cruz, M. A., Ayala-Ramirez, V., and Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing*, 30:319–328.
- Duchon, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., and Jurišica, L. (2014). Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*, 96:59–69.
- Fu, B., Chen, L., Zhou, Y., Zheng, D., Wei, Z., Dai, J., and Pan, H. (2018). An improved a* algorithm for the industrial robot path planning with high success rate and short length. *Robotics and Autonomous Systems*.
- Ge, S. S. and Cui, Y. J. (2000). New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, 16(5):615–620.
- Goyal, J. K. and Nagla, K. (2014). A new approach of path planning for mobile robots. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pages 863–867. IEEE.
- Guruji, A. K., Agarwal, H., and Parsediya, D. (2016). Time-efficient a* algorithm for robot path planning. *Procedia Technology*, 23:144–149.
- Han, J. and Seo, Y. (2017). Mobile robot path planning with surrounding point set and path improvement. *Applied Soft Computing*, 57:35–47.
- Hu, Y. and Yang, S. X. (2004). A knowledge based genetic algorithm for path planning of a mobile robot. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4350–4355. IEEE.
- Huang, H.-C. and Tsai, C.-C. (2011). Global path planning for autonomous robot navigation using hybrid meta-heuristic ga-pso algorithm. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 1338–1343. IEEE.
- Jan, G. E., Chang, K. Y., and Parberry, I. (2008). Optimal path planning for mobile robot navigation. *IEEE/ASME Transactions on mechatronics*, 13(4):451–460.
- Kala, R., Shukla, A., and Tiwari, R. (2011). Robotic path planning in static environment using hierarchical multi-neuron heuristic search and probability based fitness. *Neurocomputing*, 74(14-15):2314–2335.
- Lamini, C., Benhlima, S., and Elbekri, A. (2018). Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Computer Science*, 127:180–189.
- LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.
- Leena, N. and Saju, K. (2014). A survey on path planning techniques for autonomous mobile robots. *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, 8:76–79.
- Liu, H., Xu, B., Lu, D., and Zhang, G. (2018). A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm. *Applied Soft Computing*.
- Mac, T. T., Copot, C., Tran, D. T., and De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28.
- Montiel, O., Orozco-Rosas, U., and Sepúlveda, R. (2015). Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles. *Expert Systems with Applications*, 42(12):5177–5191.
- Rintanen, J. (2006). Introduction to automated planning. *Lecture notes of the AI planning course, Albert-Ludwigs-University Freiburg*.
- Shiltagh, N. A. and Jalal, L. D. (2013). Optimal path planning for intelligent mobile robot navigation using modified particle swarm optimization. *International Journal of Engineering and Advanced Technology*, 2(4):260–267.
- Shum, A., Morris, K., and Khajepour, A. (2015). Direction-dependent optimal path planning for autonomous vehicles. *Robotics and Autonomous Systems*, 70:202–214.
- Song, W., LI, H.-x., and ZHANG, Y.-n. (2016). Path planning of mobile robot based on genetic bee colony algorithm. *DEStech Transactions on Computer Science and Engineering*, 1(cmee).
- Victerpaul, P., Saravanan, D., Janakiraman, S., and Pradeep, J. (2017). Path planning of autonomous mobile robots: A survey and comparison. *Journal of Advanced Research in Dynamical and Control Systems*, 9:1535–1565.
- Wang, W., Zuo, L., and Xu, X. (2018). A learning-based multi-rrt approach for robot path planning in narrow passages. *Journal of Intelligent & Robotic Systems*, 90(1-2):81–100.
- Zhang, Y., Gong, D.-w., and Zhang, J.-h. (2013). Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*, 103:172–185.