Path Planning Techniques for Mobile Robots

A Review

Sean Campbell, Niall O' Mahony, Anderson Carvalho, Lenka Krpalkova, Daniel Riordan, Joseph Walsh
IMaR Research Centre, Institute of Technology Tralee,
Co. Kerry, Ireland
e-mail: sean.campbell@research.ittralee.ie

Abstract—**Mobile robots have become increasingly popular in recent years, offering a wide range of applications in areas such as industry, agriculture, search and rescue and much more. This has been achieved mainly as a result of extremely active research and development work on robotic and autonomous technology. We are still faced with many challenges however in order for a robot to navigate efficiently and reliably in an environment without any human assistance. The robot should be capable of extracting the necessary information from the environment and taking the necessary action required to plan a feasible path for collision free motion to reach its goal. In this paper, we review the most commonly used path planning methodologies that have been applied for mobile robot navigation in both static and dynamic environments. We look at both global and local path planning approaches as well as classical and heuristic based techniques.**

Keywords-global path planning; local path planning; heuristic algorithms

I. INTRODUCTION

In order for a robot to navigate successfully, it must be capable of finding a path from its current position to its goal position. It must also be capable of avoiding obstacles along the way, both static and dynamic. It may also be expected that the robot should have the ability to optimize its path by calculating the shortest and safest route to its destination in order to save time and reserve energy. The task of planning a path increases in complexity the larger the robot's configuration space (C-space) becomes. The C-space represents the space that the robot may attain with respect to the external constraints. This is still a fundamental problem in mobile robotics and a great deal of research has been carried out in order to find the best solution.

Mobile robot navigation approaches have been classified as;

i) Global Path Planning
ii) Local Path Planning

In global path planning the robot has complete prior information on its environment. This includes its size, shape, orientation, current and goal locations as well as the positions of obstacles within the environment. The global path planning approach will generally use some form of optimization technique in order to maximise the efficiency and performance of the search. The main constraint however is that this approach cannot be relied upon in a dynamic environment that contains multiple moving objects along its path.

In local path planning on the other hand, the robot has little or no information on its environment and instead has to adopt a reactive approach such as the use of Genetic Algorithms (GA's) or Artificial Neural Networks (ANN's) to help it make the best decisions on how to plan its path based on its perception of the surrounding environment. The data collected from on-board sensors is crucial in determining the robots next course of action. In this paper, we review the most commonly used path planning methodologies that are being applied for mobile robot navigation in both static and dynamic environments. We look at both classical and heuristic based techniques while also highlighting the advantages and disadvantages associated. Section 2 discusses Global path planning techniques including Roadmaps, Potential Field (PF) and Cell Decomposition (CD). Section 3 looks at some of the most commonly used local path planning approaches including, Bug Algorithms and Vector Field Histogram (VFH). Section 4 examines heuristic path planning techniques including, ANN's, GA's, Fuzzy Logic (FL) and Particle Swarm Optimization (PSO).

II. GLOBAL PATH PLANNING

A. Roadmap Method

This approach involves the construction of a safe collision free network of paths from the robots initial position to the goal position. This method can be defined as a graph-searching algorithm whereby the robots C-space which is a set of achievable motions, is reduced to a network of one-dimensional lines [1]. There are various approaches to finding the most feasible path and many methods differ in how the series of pathways and nodes are defined. Some of the most notable roadmaps include the Visibility Graph and the Voronoi diagram.

In a Visibility Graph, the vertices will consist of the start, the goal and polygonal obstacles where the edges represent the edges of the obstacle and the edges linking all sets of vertices that can see each other. As a result, this method would present a minimum length path solution [2].

Visibility graphs are considered to be very useful in environments where objects will follow polygonal shapes within continuous or detached spaces [3]. A common issue

978-1-7281-5739-9/20/$31.00 ©2020 IEEE

with visibility graphs however is that the resulting paths make contact with the obstacles at the vertices and the edges which can cause potential collisions for the robot. This can make its implementation quite challenging. Voronoi diagrams on the other hand are capable of resolving this issue.

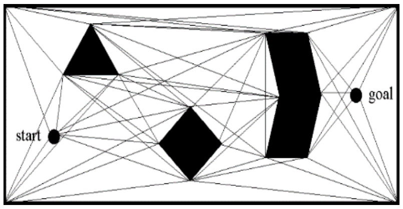

Figure 1. Visibility graph [4].

The Voronoi diagram is often utilized to calculate the proximity of points. For a given set of objects, (called sites), each Voronoi cell defines the set of points in the plane that are closer to each site than to any other site. The outlines of all the Voronoi cells form the Voronoi diagram. This approach guarantees to construct the safest path for the robot by default because the graph edges will be located at a maximum available distance from all of the nearby obstacles [5]. Such distances can only be calculated based on the capabilities and accuracy of the on-board sensors (eg LiDAR, Radar) which are used for localization. As a result, short-range sensors such as ultrasonic sensors would not be appropriate for this type of path planning approach.

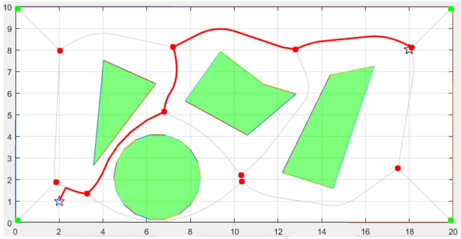

Figure 2. Voronoi diagram [6].

B. *Potential Field Method*

The PF method was first proposed by Khatib [7] to solve the task of avoiding obstacles with a mobile robot. The method involves treating the robot as a particle that is under the influence of an artificial potential field and represented as a point in the C-space. The goal of this method is to navigate the robot towards attractive fields and away from repulsive fields within its operating environment. Obstacles will be assigned as repulsive forces while the robots destination point is assigned as the positive force.

This method has proven to be very popular over the years with researchers due to its simplicity and mathematical elegance but it is only most effective in a stationary environment with stationary obstacles. Another common issue that occurs is that the robot can get trapped in a local minimum of the potential. Various solutions to this problem have been proposed such as random walks and backtracking [8]. There are many modified versions of the PF method, most of which are used as motion controllers for robots in object avoidance situations. Subramanian, George, and Thondiyath [9] proposed a multi-point PF approach for object avoidance in both static and dynamic environments. This approach observes minimal position tracking errors which tend to generate a smoother path. Valbuena and Tanner [10] proposed the use of a PF for motion planning and obstacle avoidance in non-holonomic robots. This approach allows for both the position and orientation to simultaneously converge while also minimising the online switching operation between each of the control laws.

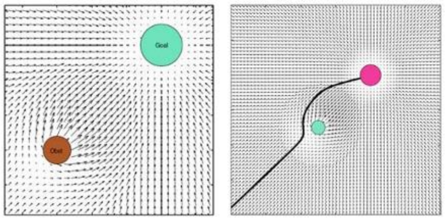

Figure 3. Potential Field with a goal and an obstacle [11].

C. *Cell Decomposition Method*

The CD approach is well known in the literature for solving both static and dynamic path planning problems. It is the representation of the robots search space (configuration space) that has been split up into cells. The relationships between each of the adjoining cells will then be computed with the goal of creating a collision free path from the start point to the goal. The cells that contain obstacles become split into two new cells and then the cells that don't contain obstacles are added to the sequence [3]. The process begins by first identifying the two cells that contain the start point and the goal point. An early example of this approach being applied to robot motion planning can be seen in [12]. A novel approximation CD method is proposed by [13]. This approach represents targets, obstacles, the sensor platform and field of view (FOV) as enclosed subsets of a Euclidean workspace. Chenghui Cai and Ferrari [14] proposed a hierarchical approximation CD method that includes time as a dimension of the world model. This approach allows dynamic obstacles to be considered as static in an extended world.

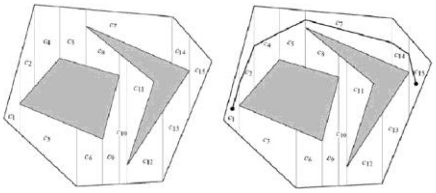

Figure 4. Cell decomposition method [15].

III. Local Path Planning

A. *Bug Algorithms*

Bug 1

In the event of the robot detecting an obstacle within its path to the goal location, this algorithm enables the robot to navigate its way by following the edge (contour) of the particular obstacle. It will continue navigating its way around until it reaches back to the point where it began following the edge – completing a full cycle. While this process is taking place, the robot is also trying to calculate the distance from the current position to the goal location in the hope of estimating the exact point on the edge line that represents the shortest distance to the goal. Once the point is stored in memory, the robot knows that this will be the point where it will depart from the obstacle when it begins its next cycle around it [16].

Bug 2

The bug 2 algorithm is a more efficient version than the bug 1 algorithm in the fact that the robot is not required to navigate a full cycle around the entire edge of the obstacle and can therefore reach its destination in less time. This can be accomplished by firstly computing an initial path from source to goal and storing the slope of the path into its 'move to goal' behaviour memory. In the event of the robot detecting an obstacle, it will automatically change behaviour to 'obstacle avoidance' and begin following the edge as it does in the bug 1 algorithm except this time it is constantly calculating the current slope line to see if it matches its stored slope line. Once both slopes become equal, the robot will then switch back to its 'move to goal' behaviour and depart from the obstacle. There are many variations and advancements to the bug algorithm such as DistBug algorithm [17] which uses a distance sensor as opposed to touch sensors. The TangentBug algorithm allows the robot to gain a selection of optimal paths by creating a local tangent graph (LTG) [18]. The CautionsBug algorithm combines LTG with spiral search approaches with the intention of ensuring competitive path selection [19].

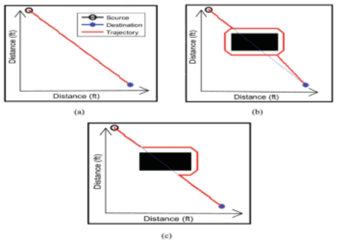

Figure 5. Paths of Bug 1 and Bug 2 algorithms (a) No obstacle (b) Bug 1 algorithm (c) Bug 2 algorithm [16].

B. *Vector Field Histogram*

This method was originally introduced by Borenstein and Koren [20] and regarded as an enhanced version of the PF method [21]. VFH is a real-time obstacle avoidance approach that allows a robot to detect and avoid unknown obstacles while also continuing to direct it towards the goal. A world model is represented by a 2-dimensional cartesian histogram grid that is constantly being updated with the data obtained from the on-board range sensors. The method then employs a 2-stage data reduction process in order to compute the required control commands for the robot. The first stage involves taking a considered fixed size subset of the 2-dimensional histogram grid around the robots location and reducing it down to a 1-dimensional polar histogram. A value representing the polar obstacle density function in that direction is contained in each sector of this polar histogram. In the second stage, the process of selecting the sector that contains the lowest polar obstacle density is computed and then the robots heading direction is subsequently aligned [22]. There are a number of variations of the VFH method that have been developed in order to improve the original algorithm. VFH+ was developed in order to help improve reliability in fast moving robot applications [23]. Another variation is VFH* which adopts the A* search heuristic method [24].

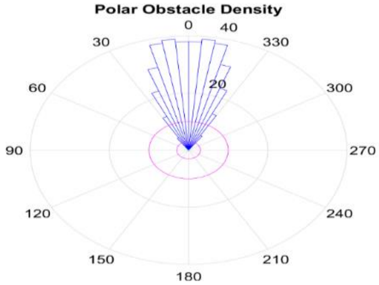

Figure 6. Vector Field Histogram [25].

IV. Heuristic Approaches

Heuristic path planning approaches have become very active in recent years compared to the above-mentioned classical approaches. The main difference lies in the potential for human-like behaviour-based learning in a heuristic based approach. The following section examines some of the most popular approaches including, ANN's, FL, GA's and PSO.

A. *Artificial Neural Networks*

ANN's are inspired by the study of the internal functionality of the human brain. They are artificial models that have the ability to learn based on current behaviour. The network is made up of neurons just like that of the human brain and each neuron represents a computational unit that can reproduce information based on experience and observation. The overall ability or 'intelligence' of the ANN is defined based on the nature and potential of each neuron and how they interconnect. ANN's are widely used in search optimization, learning, and pattern recognition problems because of their ability to provide simple and optimal solutions in complex situations [3]. A study on Self Supervised Learning (SSL) regarding a robots learning behaviour for performing a specific task has been carried out by [26]. Zou, Hou, Fu and Tan [27] suggested that many current problems regarding robot navigation in a real-world application can be addressed by using a hybrid approach that combines other artificial intelligence based methods such as knowledge based systems or FL approaches. Li and Chen [28] proposed a hybrid approach that combines an ANN with a PSO algorithm where the PSO creates a smoother trajectory for reaching the goal. One of the main disadvantages to using ANN's is that the required training data may be very large as statistically correct results need to be ensured [29]. Another common issue relates to the difficulty in minimising the error between the computed data and the required output in a supervised learning ANN.

B. *Fuzzy Logic*

The idea of FL was first proposed in 1965 by Zadeh [30]. It is based on traditional computer logic known as Boolean logic where something can be represented as ‘true’ or ‘false’, (1 or 0). The fuzzy approach offers a ‘degree of freedom’ between these two states. This approach has been used extensively for a wide range of robot path planning applications as it offers the robot the capability of gaining a heuristic knowledge of its surrounding environment. The use of a fuzzy logic controller (FLC) is well suited for the control of a robot due to its ability to make inferences even if there are uncertainties and it can assist in decision making as well as the generation of rules. A data driven fuzzy approach to creating a framework for solving a robots dynamic motion problems was developed by [31]. One of the main outcomes from this development is that it allows the robot to navigate successfully within an environment containing many dynamic obstacles and it also shows a reduction in the travel time compared to prior approaches. A new FLC for a car-like robot was designed by [32]. It is capable of providing autonomous garage parking and parallel parking by incorporating real-time image processing. A hybrid path planning algorithm for a mobile robot using a FLC and a GA is proposed by [33]. The GA is incorporated in order to carry out modifications on the input and output functions for the FLC.

C. *Genetic Algorithms*

GA’s are regarded as one of the most commonly used optimization tools for producing accurate and high-quality solutions to optimization and search problems. GA’s take inspiration from the idea of natural selection that begins without any prior knowledge of what may be the right solution. Instead it takes knowledge from the environment and then makes use of evolutionary operators such as mutation, crossover and selection to compute the best solution [34]. One of the very first uses of GA’s for a mobile robot was by [35] in 1989. It was used to try to solve the inverse kinematics problem associated with redundant robots. The requirements for applying a GA to a path-planning problem include representing the path through a suitable ‘chromosome’, creating a method of guidance along the path, finding a solution for obstacle avoidance and also, the selection of a suitable constraint definition providing method to reduce the path distance [1].

A knowledge based GA for path planning is proposed by [36], where the operators are applied to the parents depending on the heuristic knowledge from the initial generation. A GA based path planning approach using a variable length chromosome is proposed in [37]. Ghorbani, Shiry and Nodehi [38] proposed a GA based global path planner that reduces the binary strings length by converting the two-dimensional points into one-dimensional points.

D. *Particle Swarm Optimization*

PSO was first introduced by Kennedy and Eberhart in 1995 [39] as a concept for optimizing nonlinear functions. The idea was inspired by the intriguing ability of wildlife such as flocks of birds, herds of animals and schools of fish to naturally adapt to their surroundings. A socio-cognitive study investigated the idea of ‘collective intelligence’ associated with such groups. In PSO, an initial swarm represents a set of random solutions that propagate within the search space towards the optimum solution over a series of iterations based on the vast collection of data regarding the search space that has been gathered by each member (particle) of the swarm. Each particle is defined by its position and velocity within the search space and will move towards the position that represents their best personal finding (local) and the swarms best finding (global) [40]. There are many variations of PSO which are used for robot searching, obstacle avoidance and path planning applications. Min, Zhu and Zheng [41] proposed an obstacle avoidance technique in dynamic environments using multi-objective optimization through PSO. A method for solving cul-de-sac problems encountered by robots when avoiding obstacles is presented by [42] as a modified version of PSO. Pugh, Martinoli and Zhang [43] used local neighbourhood versions of PSO for multi-robot search optimization where it was found that the proposed method performs better than GA’s. A PSO based robot obstacle avoidance approach while navigating around static and dynamic objects was proposed by [44].

V. Conclusion

This paper has provided an overview on the variety of path planning methodologies used for mobile robot navigation. The paper examines both global and local path planning classifications as well as both classical and heuristic methodologies. We also discuss the advantages and disadvantages associated with each method. The classical methods discussed include, Roadmaps, PF and CD. These methods will either look to find a solution or else prove that a solution cannot be found. They may not be reliable in real-world applications due to their high computational requirements and inability to perform in dynamic situations. The heuristic-based methods discussed include ANN’s, FL, GA’s and PSO. These methods are some of the most commonly used for mobile robot navigation. In contrast to the classical methods, these algorithms do not guarantee finding a solution but in the event that they can, they will find it in much less time and computation. This makes heuristic methods much more suited to real-world applications especially in dynamic environments. The overall performance of these methods however are entirely dependent on the accuracy and reliability of the information that is being provided.

Acknowledgment

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

[1] E. Masehian and D. Sedighizadeh, “Classic and Heuristic Approaches in Robot Motion Planning A Chronological Review,” 2007.

[2] R. Illah and R. Siegwart Nourbakhsh, “Autonomous Mobile Robots Introduction to Introduction to Autonomous Mobile Robots.”

[3] A. Atyabi and D. M. W. Powers, “Review of classical and heuristic-based navigation and path planning approaches.” 2013.

[4] H. Choset, “Lecture 8.” [Online]. Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/f05/lecture/lec8.html. [Accessed: 09-Jan-2020].

[5] H. Choset and J. Burdick, "Sensor based planning. I. The generalized Voronoi graph," 2002.

[6] E. Magid, R. Lavrenov, and I. Afanasyev, "Voronoi-based trajectory optimization for UGV path planning," in *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, 2017, pp. 383–387.

[7] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings - IEEE International Conference on Robotics and Automation*, 1985, vol. 2, pp. 500–505.

[8] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," *Rob. Auton. Syst.*, vol. 54, no. 4, pp. 288–299, Apr. 2006.

[9] S. Subramanian, T. George, and A. Thondiyath, "Obstacle Avoidance Using Multi-Point Potential Field Approach for an Underactuated Flat-Fish Type AUV in Dynamic Environment," Springer, Berlin, Heidelberg, 2012, pp. 20–27.

[10] L. Valbuena and H. G. Tanner, "Hybrid Potential Field Based Control of Differential Drive Mobile Robots," *J. Intell. Robot. Syst.*, vol. 68, no. 3–4, pp. 307–322, Dec. 2012.

[11] M. A. Goodrich, "Potential Fields Tutorial." 2002.

[12] J. M. KEIL and J.-R. SACK, "Minimum Decompositions of Polygonal Objects," *Mach. Intell. Pattern Recognit.*, vol. 2, pp. 197–216, Jan. 1985.

[13] Chenghui Cai and S. Ferrari, "Information-Driven Sensor Path Planning by Approximate Cell Decomposition," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 39, no. 3, pp. 672–689, Jun. 2009.

[14] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles (mobile robot)," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 61–69, 1989.

[15] S. M. LaValle and D. Leidner, "Chapter 6: Combinatorial Motion Planning," 2006.

[16] M. Zohaib, S. M. Pasha, N. Javaid, and J. Iqbal, "Intelligent Bug Algorithm (IBA): A Novel Strategy to Navigate Mobile Robots Autonomously," *undefined*, 2013.

[17] I. Kamon and E. Rivlin, "Sensory-based motion planning with global proofs," in *IEEE Transactions on Robotics and Automation*, 1997, vol. 13, no. 6, pp. 814–822.

[18] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2002, vol. 1, pp. 429–435.

[19] E. Magid and E. Rivlin, "CAUSTIOUSBUG : a competitive algorithm for sensory-based robot navigation," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2005, vol. 3, pp. 2757–2762.

[20] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[21] A. K. Jha, "Review and Analysis of Different Methodologies used in Mobile Robot Navigation," no. 1985, pp. 1–19, 2007.

[22] M. I. Ribeiro, "Obstacle Avoidance," 2005.

[23] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proceedings - IEEE International Conference on Robotics and Automation*, 1998, vol. 2, pp. 1572–1577.

[24] I. Ulrich and J. Borenstein, "VFH *: Local obstacle avoidance with look-ahead verification," in *Proceedings-IEEE International Conference on Robotics and Automation*, 2000, vol. 3, pp. 2505–2511.

[25] Mathworks.com, "Vector Field Histogram - MATLAB &amp; Simulink," 2019. [Online]. Available: https://www.mathworks.com/help/nav/ug/vector-field-histograms.html;jsessionid=d9c9ef275c0736904316f0a5513a. [Accessed: 23-Sep-2019].

[26] K. van Hecke, G. de Croon, L. van der Maaten, D. Hennes, and D. Izzo, "Persistent self-supervised learning principle: from stereo to monocular vision for obstacle avoidance," Mar. 2016.

[27] A.-M. Zou, Z.-G. Hou, S.-Y. Fu, and M. Tan, "Neural Networks for Mobile Robot Navigation: A Survey," Springer, Berlin, Heidelberg, 2006, pp. 1218–1226.

[28] Y. Li and X. Chen, "Mobile Robot Navigation Using Particle Swarm Optimization and Adaptive NN," Springer, Berlin, Heidelberg, 2005, pp. 628–631.

[29] J. Cannady and J. Cannady, "Artificial Neural Networks for Misuse Detection," *Natl. Inf. Syst. Secur. Conf.*, pp. 443--456, 1998.

[30] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[31] M. Al-Khatib and J. J. Saade, "An efficient data-driven fuzzy approach to the motion planning problem of a mobile robot," *Fuzzy Sets Syst.*, vol. 134, no. 1, pp. 65–82, Feb. 2003.

[32] T.-H. S. Li, S.-J. Chang, and Y.-X. Chen, "Implementation of Human-Like Driving Skills by Autonomous Fuzzy Behavior Control on an FPGA-Based Car-Like Mobile Robot," *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, 2003.

[33] I. Hassanzadeh and S. M. Sadigh, "Path planning for a mobile robot using fuzzy logic controller tuned by GA," in *2009 6th International Symposium on Mechatronics and its Applications*, 2009, pp. 1–5.

[34] A. S. H. H. V Injarapu and S. K. Gawre, "A survey of autonomous mobile robot path planning approaches," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, 2017, pp. 624–628.

[35] J. K. Parker, A. R. Khoogar, and D. E. Goldberg, "Inverse kinematics of redundant robots using genetic algorithms," in *Proceedings, 1989 International Conference on Robotics and Automation*, 2003, pp. 271–276.

[36] Yanrong Hu and S. X. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 2004, pp. 4350-4355 Vol.5.

[37] Jianping Tu and S. X. Yang, "Genetic algorithm based path planning for a mobile robot," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2004, vol. 1, pp. 1221–1226.

[38] A. Ghorbani, S. Shiry, and A. Nodehi, "Using Genetic Algorithm for a Mobile Robot Path Planning," in *2009 International Conference on Future Computer and Communication*, 2009, pp. 164–166.

[39] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948.

[40] A. Stacey, M. Jancic, and I. Grundy, "Particle swarm optimization with mutation," in *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*, 2003, vol. 2, pp. 1425–1430.

[41] Hua-Qing Min, Jin-Hui Zhu, and Xi-Jing Zheng, "Obstacle avoidance with multi-objective optimization by PSO in dynamic environment," in *2005 International Conference on Machine Learning and Cybernetics*, 2005, pp. 2950-2956 Vol. 5.

[42] S. D. Hettiarachchi, "Distributed evolution for swarm robotics," 2007.

[43] J. Pugh, A. Martinoli, and Y. Zhang, "Particle swarm optimization for unsupervised robotic learning," in *Proceedings - 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, 2005, vol. 2005, pp. 95–102.

[44] C. Hu, X. Wu, Q. Liang, and Y. Wang, "Autonomous Robot Path Planning Based on Swarm Intelligence and Stream Functions," in *Evolvable Systems: From Biology to Hardware*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 277–284.