On the Performance of Sampling-Based Optimal Motion Planners

Mohamed Elbanhawi, *IEEE Student Member,* and Milan Simic
School of Aerospace, Mechanical and Manufacturing Engineering (SAMME)
RMIT University
Melbourne, Australia
mohamed.elbenhawi@rmit.edu.au, milan@rmit.edu.au

Abstract— **Sampling based algorithms provide efficient methods of solving robot motion planning problem. The advantage of these approaches is the ease of their implementation and their computational efficiency. These algorithms are probabilistically complete i.e. they will find a solution if one exists, given a suitable run time. The drawback of sampling based planners is that there is no guarantee of the quality of their solutions. In fact, it was proven that their probability of reaching an optimal solution approaches zero. A breakthrough in sampling planning was the proposal of optimal based sampling planners. Current optimal planners are characterized with asymptotic optimality i.e. they reach an optimal solutions as time approaches infinity. Motivated by the slow convergence of optimal planners, post-processing and heuristic approach have been suggested. Due to the nature of the sampling based planners, their implementation requires tuning and selection of a large number of parameters that are often overlooked. This paper presents the performance study of an optimal planner under different parameters and heuristics. We also propose a modification in the algorithm to improve the convergence rate towards an optimal solution.**

Keywords-RRT, PRM, Sampling-Based, Optimal Planners, Motion, Planning

I. Introduction

Motion Planning is considered to be one of the basic problems in robotics. Planning, together with perception, localization and motion execution, are the fundamentals of autonomous robotics. In order to generate feasible plans for the robot to traverse, an accurate probability of the robot location in the environment and that environment topology must be developed. This is a daunting mathematical task even for the simplest of systems in a low dimension space. The complexity of the motion planning problem, referred to as the piano movers problem, has been thoroughly analysed [1]. Sampling based motion planners have been used in various applications such as digital animation, virtual reality, robot planning and computational biology [2-4]. In this paper we will focus on their use for robotics.

Several approaches have been presented to overcome the complexity of the problem. Classical methods are generally divided into Roadmaps, Decompositions and Potential Fields [5, 6]. Roadmap methods attempt to capture connectivity of the given environment. They then proceed to look for a solution using a graph search algorithm such a Dijkstra Algorithm [7] or A* algorithm [8]. Voronoi [9] Diagrams and Visibility Graphs [10] are the most commonly used roadmap methods, however others have also been suggested [11]. Cell decompositions have been proposed for robot path planning purposes. They proceed to decompose the environment into smaller cells, to capture the free space, and connect the robot through the free cells to its desired goal [12]. All mentioned classical methods perform poorly in cluttered environments and larger dimensions.

The limitations of classical approaches led to the development of heuristic based methods. For example, when Potential Field method is used artificial forces are assumed to be applied on the robot [13]. Those forces are a result of artificial attractive potential field acting towards the goal and repulsive field that is pushing object away from obstacles. Initially developed for articulated robots they have been since extended to mobile robots [14]. The main limitation for using potential fields is their tendency to get stuck in local minima in situations where attractive and repulsive forces are equivalent [15, 16].

Inspired by the use of random computational methods in solving complex mathematical problems [17, 18], random planning was introduced to the field robotics. The first use of random sampling methods in robotics was to extend potential field planners and prevent the robot from stagnating in local minima [19]. In the case of stagnation in local minima the robot would perform a random motion in order to escape. The success of random walks motivated development of an entire new field in robotics which is *sampling based motion planning*.

Probabilistic roadmap method (PRM) was proposed for both articulated and mobile car-like robots [20-22]. This method relies on the random sampling of the configuration space. All free spaces are then attempted to be connected together using a local planner. This phase is known as the learning phase, where most of the computational time is invested. The next is the query phase where start and goal configurations are defined. The planner attempt to connect both configurations to the existing roadmap and employs a graph search to find the shortest path between them. Thus this method is often used for multiple-query problems.

Rapidly-Exploring Random Trees (RRT) were proposed later, as a single-query fast exploring method [23]. This method starts exploring from the start configuration, where a tree of connected configuration is incrementally grown. Samples are only connected if the path between them is collision free and if they are in the free configuration space. RRT have been extended later on for kinodynamic planning as well [24].

It is shown that both PRM and RRT are probabilistically complete [24, 25]. This implies that the planner will successfully return a solution provided one exists. However this probabilistic completeness is guaranteed as time approaches infinity. It is a weaker notion of completeness. As it turns out, sampling based planning methods tend to generate solutions quickly however the quality of the solutions is extremely low. They provide no guarantees on the quality of the solutions that they generate. The performance of RRT and PRM have been thoroughly analysed [26, 27]. Recently, it was proven that the probability of RRT generating an optimal solution as time approaches infinity is zero [28, 29].

Driven by the gap in the quality of the solutions, a breakthrough optimal sampling based planner was put forward by Kobilarov [30]. Referred to as RRT* by its creators, this planner is characterized by asymptotic optimality i.e. it finds an optimal solutions as time approaches infinity. The drawback of RRT* is its slow convergence rate towards optimal solutions (often requiring infinite time). This research paper presents current methods employed to improve solutions for standard and optimal planners. We analyse the different parameters and attempt to introduce some of our own so that we can obtain optimal solutions at a higher rate.

This paper is organized as follows; In the Section II the main implementation of a standard RRT and its parameters are introduced. The RRT* algorithm is presented and the main differences between it and other existing algorithms are highlighted. Section III presents current methods, algorithms and heuristic used to improve the quality of solutions generated by sampling planners. Section IV presents the proposed modifications and heuristics to improve the convergence rate of the solution; we present experiments and discuss results. Finally Section V concludes the paper and presents future research directions. As far as the authors know there are no studies in the reviewed literature that attempt to experimentally adjust the parameters for optimal sampling based planning performance.

II. Primitives, Execution and Parameters

This section provides a description of the common motion planning terminologies that are related to our discussion. Moreover, both RRT and RRT* algorithms are presented and their parameters are discussed. For further details refer to [30] and [27].

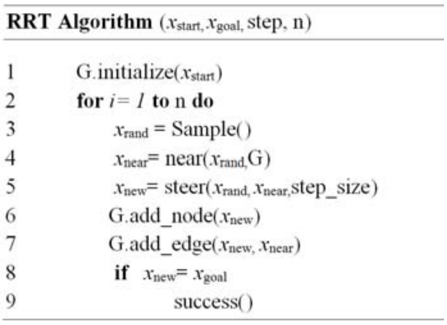
A. Motion Planning Primitives

A motion planning algorithm performs a search in the configuration space *X*. It is the assembly of all conceivable arrangements of the system. The dimensions of this space are equivalent to the degrees of freedom of the system being studied. A path is a combination or sequence of configurations in the free configuration space. The free configuration, free space X_{free}, is a component of *X* that does not lie in the obstacle configuration X_{obs}. This means that an algorithm does not require an explicit representation of the environment.

The goal of a motion-planning algorithm is to find a collision free path from a start configuration x_{start} to a goal configuration $x_{\text{goal.}}$ A path is defined as collision free if all its configurations lie in $X_{\text{free,}}$ a task that is usually delegated to a collision checker. Additionally the paths connecting those configurations must be collision free, a task delegated to the local planner.

B. RRT Algorithm

Perhaps the most commonly used single query randomized planner is the RRT algorithm. The algorithm proceeds by incrementally growing a tree from the start configuration. It is characterized by rapid exploration of the free configuration space. The growth of the RRT is given below in Fig. 1 and an example of an RRT is given in Fig. 2 The poor quality of the generated path is apparent and the need for finding methods to improve it obvious. Additionally, by increasing the runtime of the algorithm it is possible to improve the solution however the probability of it generating an optimal solution is zero [30].



```
RRT Algorithm ($x_{\text{start}}, x_{\text{goal}}$, step, n)

G.initialize($x_{\text{start}}$)
for $i = 1$ to n do
    $x_{\text{rand}}$ = Sample()
    $x_{\text{near}}$= near($x_{\text{rand}}$,G)
    $x_{\text{new}}$= steer($x_{\text{rand}}, x_{\text{near}}$,step_size)
    G.add_node($x_{\text{new}}$)
    G.add_edge($x_{\text{new}}, x_{\text{near}}$)
    if $x_{\text{new}} = x_{\text{goal}}$
        success()
```

Figure 1. RRT growing algorithm

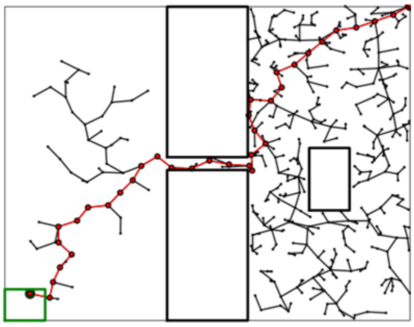


Figure 2. Example of RRT

C. RRT* Algorithm

RRT* was proposed as an algorithm to overcome the poor solutions presented by sampling based planners. This algorithm provides optimal solutions based on a predetermined cost function. It grows trees incrementally using the same method implemented by RRT. The major difference between algorithms is the rewire procedure that RRT* employs [30].

Rewiring starts by searching in the neighbourhood of the newly added configuration. The size of the neighbourhood is determined by a ball whose radius is a function of the current number of nodes in the tree. A list of near configurations X_{near} is acquired. The node that provides a shorter route towards the start of the tree is chosen as the parent for the current node. Another search is conducted to check if the current node would provide a faster route to the starting point, if so it is selected at that node's parent. This rewiring procedure ensures that the algorithm is constantly attempting to minimize the path quality. Given enough runtime the algorithm will finally reach an optimum solution.

A solution generated by the RRT* algorithm after 6000 iteration is given in Fig. 3. It can be noticed that the length of the generated path is far shorter than the one generated by the RRT* algorithm. However the number of nodes explored is rather large and so is the run time of the algorithm. The effect of the rewiring procedure on the tree structure is apparent. However, unnecessary exploration and rewiring has been wasted in unpromising areas. These areas will not improve the quality of the solution and thus growing and rewiring the tree is considered a waste of resources.

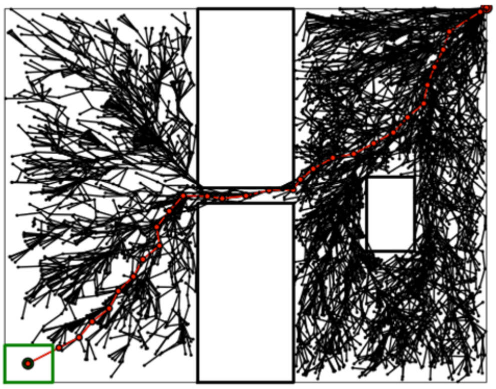


Figure 3. RRT* algorithm after 6,000 iteration

III. Relevant Approaches

There are several heuristics that are employed to speed up the search, decrease the computation and guide the growth of the tree. As far as the authors know, a study of this kind for optimal plannig have not been The proposed methods can be divided into guiding algorithms and post processing algorithms. Guiding algorithms are those that influence the sampling of the configurations and their selection. Post processing algorithms are those that optimize the solution after the search process is complete in order to generate solution of higher quality. The main algorithms for both approaches are discussed here in addition to proposed heuristics to improve the convergence rate of RRT* planner.

A. Guiding the search

It can be noticed that a large portion of the algorithm runtime is spent on sampling areas that are not promising. Sampling in those areas will not lead to finding a solution or increasing the quality of an existing solution. Sampling around obstacles [20], medial-axis [31] or a combination of both have been proposed [32] as alternatives to random sampling.

Selecting appropriate metrics has been the focus of many researchers [33-35]. In [36], it was found that a weighed Euclidian metric is yielded better solutions. However, LaValle [27] argues that choosing an appropriate distance metric proves to be a task that is as complex as the original piano mover's problem [1].

A bidirectional search method is proposed where two RRTs are grown from the start and goal configurations and once the two trees are connected the search is over [37]. Light goal biasing has been proposed where the planner attempts to connect to the goal position every n^{th} iteration, where n is an integer preferably larger than 100 [24].

It has been argued that a significant portion of the planning time is wasted on collision checking and thus delaying the collision checking was presented in [38, 39]. A naive method is checking every segment of the path for collision. Different collision checking algorithms were proposed to speed up the process [40, 41].

Urmson and Simmons [42] suggested a k-near algorithm. Once a node is sampled its parent is chosen from the nearest k configurations, where k is a positive integer. The parent is selected as the node that has the shortest path towards the root of the tree. An anytime RRT planner was presented which expanded on the k-near concepts [43]. It would generate fast suboptimal solutions and proceed to improve the solutions incrementally. The planner would only sample promising nodes. A node would be deemed promising if its heuristic length from start to finish would be lower that a predetermined value. This value is based on the current length of the tree. Optimal anytime RRT* was also proposed [29].

Some of the presented heuristics were proposed to improve the convergence rate of RRT* [44]. It was suggested to perform biasing the sampling towards existing nodes in the path, to improve the path quality, as opposed to expanding into unnecessary regions. A similar node selection algorithm given in [43] was used to select only promising nodes. The effectiveness of bidirectional search was highlighted in [24, 37] and so bidirectional RRT* was also proposed. All of these heuristics were combined and shown to improve the performance of the RRT* convergence rate towards an optimal solution [44].

B. Post Processing

Unlike methods which attempt to guide the search and improve the solution while the growth of the tree continiues, post processing methods, as their name suggest, improve the solution after the RRT growth is complete. A common method is tree pruning, in which the the number of configurations in the tree is reduced by removing redundant nodes. This is achieved by attempting to connect sucusseive nodes together until a node cannot be connected. An alternative method was suggested in which the path is divided into random segments which are then tried to be connected [45]. In the same paper, a method for moving the path towards the medial axis and removing redudand branches is also provided. RRT algorithms are run multiple times and the resulting trees are connected to create what is referred to as H-graph [46]. This method relies on the fact that suboptimal solutions will have optimal regions in them. By connecting all solutions it is shown that a graph search is capable of finding better solutions.

A post processing method was proposed to improve RRT* convergence rate based on a pruning algorithm [47]. If two configurations cannot be connected the algorithm proceeds to sample around them. It is assumed that these configurations are close to objects and thus sampling around them will improve the results. The pruning algorithm removes redundant configurations from the path towards the goal.

IV. Experiments and Results

This section is divided into two parts. In the first, the effect of the different parameters of the algorithm with respect to the quality of the solutions is given. Additionally, an extension to the rewiring procedure is proposed to improve the quality of the solution. Heuristics from literature are combined and their effects on the resluting paths are discussed. For the sake of brevity, the convergence rate of RRT* without any heurstics will not be presented as our results agree with those already discussed in [44, 47]. It must be noted that multiple experiments have been conducted. For brevity only one of each is presented. Both were run in the same environment as shown in Fig. 2. and Fig. 3. Experiments have been run with different step sizes (10, 5, 5-1.414), different goal biasing (0%, 1%, 10%), node selection heuristic to improve the path by 1% and 5% and with path rewiring.

A. Parameter Tuning

In Section IV, different heuristics and parameters used to improve the performance of general sampling based planners were introduced. In this section a number of these parameters are studied. Mainly, goal-biasing, incremental step size and node selection heuristics and values are considered. It is attempted to find the combination of parameters that improve the performance of RRT*. In this context performance refers to the quality of the solution with respect to a metric and the optimization convergence rate. Often these heuristic values are overlooked in the theoretical handling of sampling based algorithms but they are essential for their implementation [26].

Goal biasing is a heuristic in which the tree is guided towards the goal position. This percentage of iterations in which the goal connection is attempted must be maintained relativly low to maintain the randomness in planning. As expected biasing enables the planner to find a solution faster, in our case almost 1000 iterations quicker just by 1% biasing. On the other hand, when biasing is continued after the solution has been found, it tends to hinder path optimization. In the absense of biasing a 2.87% optimization rate over a 10,000 iterations can be reported, as opposed to 3.15% with 1% goal biasing.

The algorithm was run with 10% biasing to rapidly find a initial solution and then biasing was stopped to allow for path optimization. The results of the experiments are shown in Fig. 4. Guiding the tree towards the goal initial generated a quick first solution with lower costs, however the convergence rate is still low and reaching an optimal solution would possibly require an infinite number of iterations.

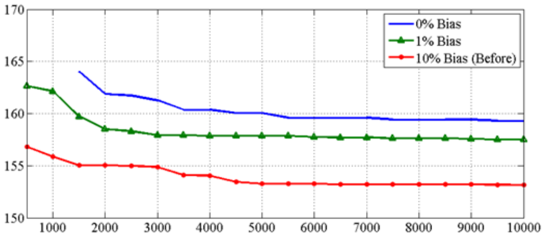


Figure 4. Effect of goal biasing on path convergence towards optimality

The distance that was set for the steering function is 5, in a 2D environment whose dimensions are 100 x 100. It is attempted to vary the step size and observe its effect on convergence rate and path quality. Large distances between nodes are expected to produce a quick first solution with low quality. On the other hand, small distances between nodes can be redundant as they would have no effect on the path quality and would only increase the computational load of the algorithm. As the result of RRT* slow convergence rate, it is attempted to generate initial paths with high quality. Fig. 5 shows multiple runs of RRT* with different step sizes. As step size increases, so do the initial solutions obtained by the planner. To overcome sampling redundant nodes, planner was run with a minimum distance of $\sqrt{2}$ in addition to a maximum distance of 5. This approach has significantly improved the quality of the solutions.

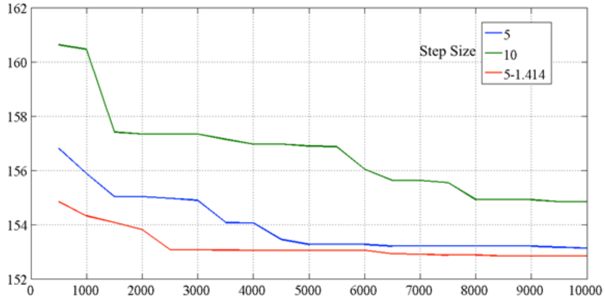


Figure 5. Effect of step size on solution convergence

In all the previous experiments nodes are only rejected if they, or their path, do not lie in the free configuration space. As a result, the number of nodes explored by the algorithm after 10,000 iterations is between 7,000 and 8,000. By analysing solutions, it can be noticed that, a large portion of these nodes are unpromising, as they will not improve the quality of the current path. See Fig. 3 for illusration. Rejecting these unpromising nodes will decrease the computational load on the planner and will allow more time for path optimization. This heuristic has no direct effect on the convergence rate of the solution. We employ the heuristic introduced in [43] for RRT planners. The lower bound cost of any node from start to goal must be less than a minimum cost, in order to consider it. Only 1,300 nodes have been explored after 10,000 iterations. The improvement from the current solution *ε* is often a percentage between 1%-5%. The total number of nodes sampled using both values is shown in Table 1.

TABLE 1. AVERAGE NUMBER OF NODES EXPLORED AFTER 10,000 ITERATIONS

Number of Sampled Nodes After 10,000 Iterations		
RRT*	ε = 1%	ε = 5%
7731	4718	4066

B. *Path Rewiring*

The slow convergence rate and the large time spent in exploring, rather than optimizing, are its main drawbacks of RRT*. We proposed an addition to the RRT* rewiring algorithm. Once the original rewiring algorithm is complete, we greedly attempt to rewire the current path. We find the *k*-nodes, in the current path, that are nearest to the recently added node. If the path through the current configuration is shorter we rewire the entire path thorugh the current configuration. This greedy heuristic is also coupled with the node selection heuristic employed in [43, 44].

The advantage of this approach is that it is constantly attempting to improve the current path. When compared to an RRT* algorithm with the same step size, *5-1.414*, and goal biasing, *10%*, using this heurstic improves the average solution obtained after 10,000 iterations and the overall convergence rate. See Table 2 and Fig. 6 for the results after 10,000 runs of RRT*, with and without path rewiring.

TABLE 2. RESULTS OF USING PATH REWIRING PROCEDURE AFTER 10,000 RUNS

Planner	Average Length	Convergence
RRT*	152.84	1.31%
RRT* - path rewire	149.25	2.74%

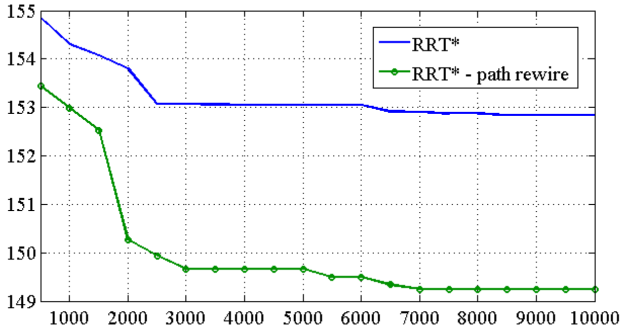


Figure 6. Effect of path rewiring on convergence rate and path length

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the main heuristics used in sampling based planning algorithms. We applied those heuristics to RRT* optimal sampling based planners. Additionally, a heuristic to improve the convergence rate of the planner has been proposed. The findings of this paper are as follows:

- Biasing the search towards the goal generates high quality initial paths.
- A large step size helps increase finding an initial solutions however the quality of that solution is low.
- Adding a minimum step size eliminates the addition of redundant nodes.
- Using a node selection/rejection heuristic reduces the number of explored nodes. However, a large improvent rate value could possibly hinder the optimization process.
- Attempting to rewire the current path every iteration improves both the quality of the solution and the convergence rate of the planner.

The presented methods must be evaluated in high dimensional configurations space and in kinodynamic planning scenarios. It is possible that findings of this research may not scale to larger dimensions, but it has to be tested and investigated further. Another possible area of inverstigation is the application of Fuzzy Logic Control (FLC) for the selection of the tuning of the planner parameters. We also plan to incorporate the robotic systems contraints, i.e. robot kinematics, in the path planning procedure. Further investigation is needed in the application of more efficient sampling methods, local planners, collision checker and their effects on the planner.

ACKNOWLEDGMENT

Elbanhawi is a recipient of the Australian Postgraduate Award (APA) and Research Training Scheme (RTS).

REFERENCES

[1] G. Sánchez and J.-C. Latombe, "On Delaying Collision Checking in PRM Planning: Application to Multi-Robot Coordination," *The International Journal of Robotics Research,* vol. 21, pp. 5-26, January 1, 2002 2002.

[2] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, *et al.*, "RRT*-SMART: A rapid convergence implementation of RRT*," *International Journal of Advanced Robotic Systems,* vol. 10, // 2013.

[3] L. Jaillet and J. M. Porta, "Path Planning Under Kinematic Constraints by Rapidly Exploring Manifolds," *Robotics, IEEE Transactions on,* vol. 29, pp. 105-117, 2013.

[4] J.-C. Latombe, "Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts," *The International Journal of Robotics Research,* vol. 18, pp. 1119-1128, November 1, 1999 1999.

[5] J.-C. Latombe, ROBOT MOTION PLANNING.: Edition en anglais: Springer, 1990.

[6] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*: Mit Press, 2011.

[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959/12/01 1959.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Systems Science and Cybernetics, IEEE Transactions on,* vol. 4, pp. 100-107, 1968.

[9] J. Canny, "A Voronoi method for the piano-movers problem," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, pp. 530-535.

[10] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai, "Visibility-polygon search and euclidean shortest paths," in *Foundations of Computer Science, 1985., 26th Annual Symposium on*, 1985, pp. 155-164.

[11] M. Elbanhawi, M. Simic, and R. Jazar, "Autonomous Mobile Robot Path Planning: A Novel Roadmap Approach," *Applied Mechanics and Materials,* vol. 373-375, pp. 246-254, 2013.

[12] J. M. Keil and J.-R. Sack, "Minimum decompositions of polygonal objects," *Computational Geometry,* pp. 197-216, 1985.

[13] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research,* vol. 5, pp. 90-98, Spr 1986.

[14] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Foundations of Computer Science, 1979., 20th Annual Symposium on*, 1979, pp. 421-427.

[15] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 1991, pp. 1398-1404 vol.2.

[16] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on,* vol. 7, pp. 278-288, 1991.

[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics,* vol. 21, pp. 1087-1092, 06/00/ 1953.

[18] N. Metropolis and S. Ulam, "The Monte Carlo Method," *Journal of the American Statistical Association,* vol. 44, pp. 335-341, 1949/09/01 1949.

[19] J. Barraquand and J.-C. Latombe, "Robot Motion Planning: A Distributed Representation Approach," *The International Journal of Robotics Research,* vol. 10, pp. 628-649, December 1, 1991 1991.

[20] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 1996, pp. 113-120 vol.1.

[21] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on,* vol. 12, pp. 566-580, 1996.

[22] P. Švestka and M. H. Overmars, "Motion Planning for Carlike Robots Using a Probabilistic Learning Approach," *The International Journal of Robotics Research,* vol. 16, pp. 119-143, April 1, 1997 1997.

[23] S. Carpin and G. Pillonetto, "Motion planning using adaptive random walks," *Robotics, IEEE Transactions on,* vol. 21, pp. 129-136, 2005.

[24] S. M. LaValle and J. J. Kuffner, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research,* vol. 20, pp. 378-400, May 1, 2001 2001.

[25] J. Barraquand, L. Kavraki, J.-C. Latombe, R. Motwani, T.-Y. Li, and P. Raghavan, "A Random Sampling Scheme for Path Planning," *The International Journal of Robotics Research,* vol. 16, pp. 759-774, December 1, 1997 1997.

[26] I. A. Sucan and L. E. Kavraki, "On the implementation of single-query sampling-based motion planners," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 2005-2011.

[27] S. LaValle, *Planning Algorithms*: Cambridge University Press, 2006.

[28] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 7681-7687.

[29] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime Motion Planning using the RRT*," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1478-1483.

[30] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research,* vol. 31, pp. 855-871, June 1, 2012 2012.

[31] Z. Qiu-Bo, P. Song-Hao, and G. Chao, "Motion planning for humanoid robot based on hybrid evolutionary algorithm," *International Journal of Advanced Robotic Systems,* vol. 7, pp. 209-216, // 2010.

[32] S. Thomas, M. Morales, T. Xinyu, and N. M. Amato, "Biasing Samplers to Improve Motion Planning Performance," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 1625-1630.

[33] H. M. Choset, Principles of Robot Motion: Theory, Algorithms, and Implementation: Prentice Hall of India, 2005.

[34] R. Geraerts and M. H. Overmars, "Sampling and node adding in probabilistic roadmap planners," *Robotics and Autonomous Systems,* vol. 54, pp. 165-173, 2/28/ 2006.

[35] R. Geraerts and M. H. Overmars, "Reachability-based analysis for Probabilistic Roadmap planners," *Robotics and Autonomous Systems,* vol. 55, pp. 824-836, 11/30/ 2007.

[36] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "Choosing good distance metrics and local planners for probabilistic roadmap methods," *Robotics and Automation, IEEE Transactions on,* vol. 16, pp. 442-447, 2000.

[37] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000, pp. 995-1001 vol.2.

[38] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 2640-2645.

[39] R. Bohlin and E. E. Kavraki, "Path planning using lazy PRM," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000, pp. 521-528 vol.1.

[40] S. M. LaValle, *Planning Algorithms*: Cambridge University Press, 2006.

[41] C. Peng, E. Frazzoli, and S. Lavalle, "Improving the Performance of Sampling-Based Motion Planning With Symmetry-Based Gap Reduction," *Robotics, IEEE Transactions on,* vol. 24, pp. 488-494, 2008.

[42] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003, pp. 1178-1183 vol.2.

[43] D. Ferguson and A. Stentz, "Anytime RRTs," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 5369-5375.

[44] S. Chakravorty and S. Kumar, "Generalized Sampling-Based Motion Planners," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 41, pp. 855-866, 2011.

[45] L.-P. Ellekilde and H. G. Petersen, "Motion planning efficient trajectories for industrial bin-picking," *The International Journal of Robotics Research,* vol. 32, pp. 991-1004, August 1, 2013 2013.

[46] B. Raveh, A. Enosh, and D. Halperin, "A Little More, a Lot Better: Improving Path Quality by a Path-Merging Algorithm," *Robotics, IEEE Transactions on,* vol. 27, pp. 365-371, 2011.

[47] R. Alterovitz, M. Branicky, and K. Goldberg, "Motion Planning Under Uncertainty for Image-guided Medical Needle Steering," *The International Journal of Robotics Research,* vol. 27, pp. 1361-1374, November 1, 2008 2008.