2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress

An Experimental Study for Optimal Homotopy Property of Motion Planning Algorithms

Yang Liu
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
qwertyliuyang@buaa.edu.cn

Jiantao Zhang
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
ZhangJiantao@buaa.edu.cn

Yizhi Zeng
Science and Technology on Avionics Integration Laboratory of Automation
Shanghai, China
1551324116@qq.com

Haonan Yao
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
Yaohn@buaa.edu.cn

Jiankang Li
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
1755425790@qq.com

Jing Zhang
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
zhangjing2013@buaa.edu.cn

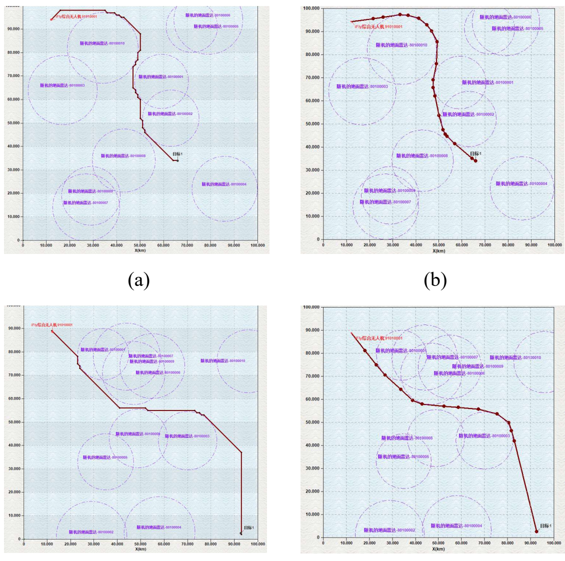
***Abstract*—Path Planning Problem is very hard for its huge search space. In this paper, we quote the definition of homotopy in mathematics to describe the right "direction" of the optimal path, which is a necessary condition for the optimal paths under the holonomic condition. Based on the concept of homotopy, optimal homotopy property is proposed for paths. Numerical experiments involving RRT*, sPRM, RRT, PSO, A* and Jump Point Search (JPS) are carried out. Although A* and JPS are gird space optimal, these do not satisfy optimal Homotopy property.**

Keywords—Motion planning, RRT*, A*, Jump Point Search(JPS), Homotopy

INTRODUCTION

Motion planning aims to find a collision-free path that connects an initial state to a goal state. However, finding optimal paths in continuous environments with polyhedral obstacles is NP-hard [1][2]. Despite the existence of complete algorithms and their proof, their complexity makes them unsuitable for practical applications [3]. So instead of focusing on the exact solution, the researchers have associated the approximate one.

These approximate algorithms can be classified into three categories: graph-based algorithms, sample-based algorithms and computational intelligence based algorithms. The graph-based algorithm discretizes the continuous space first, and then to calculate the path according to the preprocessed graph. Many methods are designed for dividing scenarios and obtaining these graphs, from the visual graphs[3], the vonoroi graph[3] and the cell decomposition[12] that were proposed early, to the NavMesh[13] and the uniform grid that most widely used. There are plenty of algorithms for solving optimal paths in these graphs, including A*[3], D* and the Jump Point Search algorithm[10][13]. These algorithms guarantee their optimality and completeness. As for sample-based algorithms, they usually begin from the start state node and gradually approach the goal state according to elaborate designed random sample rules. Such algorithms can avoid the memory overhead for describing the shape of the obstacle zone. With simple collision detection, these algorithms perform very fast. Further, some algorithms are guaranteed asymptotic optimality, such as RRT*[5], sPRM[5], FMT*[14] and BIT*[15]. Computational intelligent based algorithms generate feasible paths by random constructing path candidates and iteratively evolution these candidates according to the rules. Unfortunately, these methods have no relevant literature to guarantee their completeness.

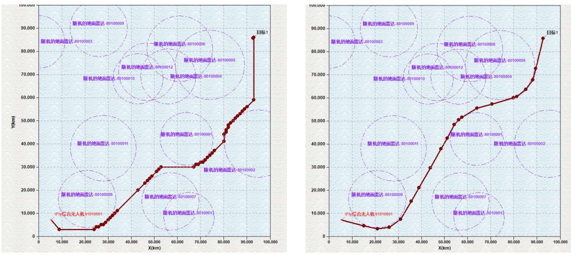


(a) (b) (c) (d)

Fig. 1. Consistent path results with the same direction. JPS*(left) vs RRT* (right)

Among algorithms introduced above, some of them offer optimality guarantees including A*, JPS, RRT* and sPRM. Intuitively, for the same scenario, these algorithms with optimality guarantees should obtain a group of similar paths. Here the similar implies these paths go through the same direction. For most scenarios, this is true. (See Fig. 3) However, cases studies indicate that these path generated by different optimal algorithms do not come up to the expectations. To be exact, for seldom scene, paths calculated by A* and JPS with gird space are inconsistent with that calculated by RRT* and sPRM. These paths have different topology structure, which implies that these paths go through different directions. (See Fig. 2)

978-1-7281-3024-8/19/$31.00 ©2019 IEEE
DOI 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00133



(a) (b)

Fig. 2. Inconsistent path results with different directions. JPS*(left) vs RRT* (right)

Based upon the finding, we start to explore why these optimal paths obtained by different algorithms belong to different topology structure (See Fig. 2). In order to characterize the difference in topology, path homotopy [6][7] is introduced to describe the topology structure of the path. If a feasible path, with start and target state fixed, can be continuously deformed into the other without intersecting any obstacle, such a deformation being called a homotopy between these two paths. Based on the homotopy, if a planning algorithm's result could be deformed into the optimal path, then we define the algorithm has the optimal homotopy property. Worth noting that here the optimal means the shortest feasible path in $\mathbb{R}^2$. So, the research question arises:

- Which motion planning algorithm holds the optimal homotopy property?

We use this property to analyze a wide range of path planning algorithms. Numerical experiments involving RRT, RRT*, sPRM, A*, JPS and PSO based algorithm are provided to verify the correctness of the necessary condition. Surprisingly, gird-based optimal algorithms, JPS and A*, do not satisfy the optimal homotopy property.

The outline of this paper is listed as follows. In Section 2, we propose the mathematical statement of the problem. In Section 3, optimal homotopy property is presented and its correctness is proofed. Then numerical experiment setting comes in the following section, including the mainline of the experiment and parameter configuration in detail. Results and the related discussion are provided and concluding remarks are stated in Section 5. And the conclusion comes in Section 6.

PRELIMINARY

Problem Formulation

Considering Karaman's definition of the holonomic problem [5], feasible and optimal path are formalized as follows.

Let $C \doteq [0,x] \times [0,y]$ be the configuration space, where $x, y \in \mathbb{R}^+$ and $C \subset \mathbb{R}^2$. For convenience, configuration space is short for scene. Let C_{obs} be the obstacle region, which is an open set, and denote the obstacle-free space as $C_{free} \doteq cl(C \backslash C_{obs})$, where $cl(\cdot)$ represent the closure of a set. The initial condition c_{init} belongs to C_{free}, so does the goal condition that $c_{goal} \in C_{free}$. A path planning problem is defined by a triplet$(C_{free}, c_{init}, c_{goal})$.

Let $\sigma : [0,1] \to \mathbb{R}^2$ be a sequence of states as a feasible path, where $\sigma(0) = c_{init}$, $\sigma(1) = c_{goal}$ and $\sigma(r) \in C_{free}$ for all $r \in [0,1]$.

Let $\sigma^* : [0,1] \to \mathbb{R}^2$ be an optimal path if it is a feasible path such that $\sigma^* = \arg\min\{s(\sigma) : \sigma \in \Sigma\}$ where $\Sigma$is the set of all feasible paths and $s(\cdot) : \Sigma \to \mathbb{R}_{\geq 0}$ is a chosen cost function.

Homotopy

A homotopy between two continuous functions f and g from a topological space X to a topological space Y is defined to be a continuous function $H : X \times [0,1] \to Y$ from the product of the space X with the unit interval $[0,1]$ to Y such that, if $x \in X$ then $H(x,0) = f(x)$ and $H(x,1) = g(x)$ [4]. As Figure 1 shows, the two bold paths shown below are homotopic relative to their endpoints. The changing light trace represents one possible homotopy class.

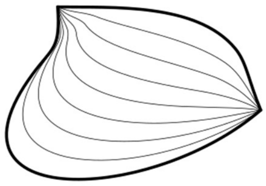


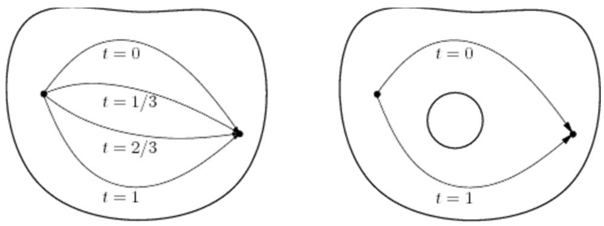
Fig. 3. An example of Homotopic relationship

Homotopy Classes of Feasible Paths

If a feasible path, c_{init} and c_{goal} fixed, can be continuously deformed into the other without intersecting any obstacle, such a deformation being called a homotopy between these two paths. And these paths are called homotopic or belong to the same homotopy class. Further, a set of all homotopically equivalent paths (i.e., homotopic paths) constitute a homotopy class, denoted the homotopy class of a path σ as $[\sigma]$[7]. Here 0 gives an example of homotopy classes.

Optimal Homotopy Class

For an optimal path, denoted as σ^*. There always exists a homotopy class contains σ^*. We define this homotopy class as the optimal class, denoted as $[\sigma^*]$. (See Fig. 4)



(a) (b)

Fig. 4. (a) Homotopy continuously warps one path into another. So they belong to same homotopy class. (b) The image of the path cannot be continuously warped over a hole in $\mathbb{R}^2$, because it causes a discontinuity. In this case, the two paths are not homotopic. And these path belongs to

Assumptions

In this work, the proposed algorithm is mainly based on the following assumptions.

Assumption 1: This article discusses the planning problem under holonomic conditions, which include but not limited to. So robots can be regarded as particles without differential constrains.

Assumption 2: For any scenario, there always exists a feasible path. So does the optimal one. For narrative convenience, we suppose there is only one optimal path in $\mathbb{R}^2$.

OPTIMAL HOMOTOPY PROPERTY FOR OPTIMAL PATH

In this section, we introduce the optimal homotopy property, which is based on the relationship between optimal homotopy class and the optimal path. Then a simple method is presented to check whether two paths belong to the same homotopy.

Optimal Homotopy Property

Definition 1:

If a planning algorithm's result σ could be deformed into σ^*, then we define the path σ has the optimal homotopy property.

Asymptotic optimality algorithms such as RRT* and sPRM hold the optimal homotopy property. Otherwise, the result of asymptotic optimality algorithms σ must belong to the other homotopy class that satisfies $[\sigma] \neq [\sigma^*]$, which is contrary to Theory 1. Moreover, here is the proof as follows.

Lemma 1: For any path σ, if $\sigma \in [\sigma]$, then $\sigma \notin [\sigma']$ with $\forall[\sigma'] \neq [\sigma]$.

Proof:

Suppose $[\sigma]$ and $[\sigma']$ are 2 different homotopy classes. If $\sigma \in [\sigma]$ and $\sigma \in [\sigma']$. Then with $\forall\sigma_1 \in [\sigma]$, σ_1 could deform into σ for $\sigma \in [\sigma]$. On the other hand, with $\forall\sigma_2 \in [\sigma']$, σ could deform into σ_2, for $\sigma \in [\sigma']$. So for any σ_1 could deform into σ then deform into σ_2, which implies $[\sigma]$ and $[\sigma']$ are the same homotopy class. It contradicts with supposed condition. Q.E.D

Theory 1: A homotopy class $[\sigma]$ is an optimal homotopy class iff $\sigma^* \in [\sigma]$.

Proof:

According to Assumption 2, suppose there is only one optimal path in $\mathbb{R}^2$.

Sufficient condition:

According to the definition of the optimal homotopy class we know that $\sigma^* \in [\sigma^*]$. On the other hand, if $\sigma^* \in [\sigma]$ is satisfied, then according to the lemma 1, $[\sigma]$ and $[\sigma^*]$ is the same homotopy class. Sufficient condition satisfies.

Necessary condition:

According to the definiton of the optimal homotopy class, for the optimal path σ^*, there must exist the $[\sigma^*]$ that contains σ^*.

Homotopy Class Check between different paths

According to the definition of the homotopy, if one path could deform into the other, then these two paths are homotopic. However, the deformation function is unnecessary to calculate for only the homotopy relationship between these two is required.

It is worth noting that if there is no obstacle in the close set consisting of two feasible paths, then these two path are homotopic. So, for any two feasible paths, if there is any obstacle's center located in the close set, then these two paths are not homotopic.

EXPERIMENT SETUP

In this section, we managed to answer which algorithm's path result belongs to the $[\sigma^*]$. Obviously, asymptotic optimality algorithms such as RRT* and sPRM hold the optimal homotopy property. However, there are two more subquestions:

- Is there an algorithm that it cannot obtain the optimal or asymptotic optimal path but it holds the optimal homotopy property?
- Is there an algorithm that it can obtain the optimal path to some degree but it does not satisfy the optimal homotopy property?

To answer these questions, our experiment selects several algorithms. They are RRT*, sPRM, RRT, PSO, A* and JPS. Each algorithm performs 2500 times experiments in different scenarios. Several statistic attributes are collected, including the accuracy and the path length. The average accuracy means the percentage of feasible paths that homotopic to the shortest path obtained by RRT* in the same scenario. Results of RRT* and sPRM are regarded as the standard data for its asymptotic optimality, which is utilized to be the base of homotopy class comparison. Then RRT, PSO, A* and JPS are the test candidates to check whether they satisfy the optimal homotopy property.

Here 25 randomly generated scenarios are prepared for the experiment. All scenarios are indexed from 1 to 25. The size of all scenarios is set as $100km \times 100km$, with 8 to 15 obstacles randomly with the proper size. RRT, PSO, A* and JPS are regarded as the test algorithms, and the best result of RRT* is the optimal solution for each scenario. Considering different parameters influence, several parameter setting is prepared for test candidates. So each algorithm runs 100 times for each parameter set in the above 25 scenarios. Except that A* and JPS runs only one time for each parameter and scenario because path results of them are determined once the scenario and the parameter are set.

Given the raw data, we calculate the success rate and the accuracy. The success rate represents the percentage of feasible path calculated by an algorithm. The average accuracy means the percentage of feasible paths that homotopic to the shortest path obtained by RRT* (It is globally optimal at the same time.) in the same scenario.

Here is the introduction of parameter settings for different test candidates.

For RRT*, the implementation is derived from Karaman's work[5]. The anytime version is able to obtain an asymptotic optimality result. We do not test the performance of different parameter settings, because under sufficient sampling condition, the best path that RRT* obtained gets closer to the optimal one. We implement sufficient running time and the result is credible. As for sPRM, we got statistics that are consistent with the asymptotic optimality.

For RRT, The implemented algorithm is derived from [3] with consistent biased sampling strategy. The probability of biased sampling is proposed, which is set as 0.7, 0.3 and 0. The step length λ is set to $5km$, $10km$ and $15km$, respectively. And the best average accuracy among them is selected from TABLE I. . According to the best result of the TABLE I. , algorithm for comparison with the step length λ is $5km$ and the biased sampling parameter is 0.7. Besides, anytime version of RRT is implemented as a new candidate with the same best parameter setting.

For A* and JPS, we only consider the effect of sampling gird's size, which has been analyzed in detail in LaValle's work[3]. Here we select the highest resolution among 0.5, 1 and 2 for the following discussion (See TABLE II.).

For PSO, we tried to adjust parameters such as particle dimensions, iteration times and so on. Unfortunately, path results hardly feasible for complicated scenario, because the algorithm has no guarantee to completeness, which is confirmed by experiment. So we just select the best parameter scheme for the experiment.

It's worth noting that the best result for each algorithm means the highest accuracy that the algorithm result obtained with the parameter. For RRT, the best result comes from the parameter settings with step length $\lambda = 5km$ and 0.7 as the probability of biased sampling (See TABLE I.). For PSO, the dimension of particle is set to 10, which means the path consists of 10 nodes. The number of particle is 40. The maximum number of iterations is 2000. For A* and JPS, the gird is $0.5km \times 0.5km$ square. As for RRT*, step length λ equals to $5km$, and $k_{RRT*} = 2\lambda$.

TABLE I. AVERAGE ACCURACY FOR RRT WITH DIFFERENT CONFIGURATION SETTINGS

Parameter	$\lambda = 5km$	$\lambda = 10km$	$\lambda = 15km$
biased sampling probability 0.7	62.04%	60.88%	60.48%
biased sampling probability 0.3	56.88%	55.16%	57.32%
biased sampling probability 0	50.8%	51.4%	50.88%

TABLE II. AVERAGE ACCURACY FOR A* AND JPS WITH DIFFERENT CONFIGURATION SETTINGS

Parameter	$Size = 0.5km$	$Size = 1km$	$Size = 2km$
Accuracy	96%	96%	92%

The experimental hardware platform is Intel I5 4300 with 4G DDR3 1600MHz RAM in windows 7. All candidate algorithms run in the self-developed motion planning algorithm platform. The platform includes four main algorithm libraries: task assignment, motion planning, path tracking and obstacle modeling. The task assignment library contains solution algorithms for multiple types of assignment problems, such as multi-traveler problems. The motion planning library includes a series of algorithms such as Voronoi Graph, RRT[3], RRT*[5][8], PRM[5], Bi-level base planning[9], A*, JPS[10] and so on. These algorithms cover different operating modes, including anytime mode and real-time mode. The path tracking library includes classic path tracking algorithms, such as pure pursuit[11] and Stanley method[11]. For obstacle modeling library, several different model based obstacle are prepared for the scenario generation. In addition, in order to facilitate the implementation of automated code testing and the requirement to generate the training set for machine learning based planning algorithms, the platform implements automated batch operations and collects data based on custom scripts. Finally, in order to verify the practicability of the algorithm, the platform provides a communication interface with the hardware, realizing real-time interaction between the execution process of the relevant algorithm and the real UAV.

RESULTS ANALYSIS

Data Result Analysis

TABLE III. AVERAGE ACCURACY AND SUCCESSFUL RATE

Algorithm	*RRT**	*sPRM*	*RRT*	*RRT anytime*	*A**	*JPS*	*PSO*
Success Rate	100%	99.28%	100%	100%	100%	100%	63.68%
Average Accuracy	100%	96.00%	62%	95.28%	96%	96%	40.12%

Successful Rate represents the percentage of feasible path calculated by an algorithm.

For RRT*, we implement sufficient running time and the result is credible. All simulations pass the test. As for sPRM, the statistics are consistent with the asymptotic optimality. For failed simulation scenes, we believed that the algorithm lacks the sampling nodes and sampling strategy for low dispersion area.

As for other algorithms, something interesting happened.

1) There may exist an algorithm that it cannot obtain the optimal or asymptotic optimal path but it holds the optimal homotopy property.

RRT that it cannot obtain the optimal or asymptotic optimal path. It cannot hold the optimal homotopy property. However, RRT anytime version does. It is proved that RRT is not asymptotically optimal[5]. But the experiment result seems that its average accuracy is high enough to suppose RRT anytime hold the optimal homotopy property with sufficient sampling. Which requires confirmation of further experiment.

2) There exists an algorithm that it can obtain the optimal path in some degree but it doesn't satisfy the optimal homotopy property.

A* and JPS are able to almost match the $[\sigma^*]$ in the experiment. For most scenario, the homotopy class of the shortest path in the gird is consistent with that of the optimal path. In some extreme cases (See 0) the resulting path is inconsistent with the result obtained by RRT*. And low-dispersion sampling seems to be the reason. However, according to our experiment result, A* and JPS remain to persist its previous conclusion of path homotopy even though the grid size is shrunk to $0.1km \times 0.1km$. A* and JPS with octile distance as the heursitic can't hold the optimal homotopy property. The proof will narrative in the following part. (See Theory 2)

Last, the PSO base algorithm cannot find it out. Although its optimal solution can still match the $[\sigma^*]$, and seldom path length of the best solutions is shorter than that of RRT or A*, it cannot be proved to be probabilistic completeness theoretically. The statistics confirmed that the PSO based algorithm is not completeness. In summary, the PSO based algorithm cannot find the homotopy set of the optimal path.

Proof Part

Lemma 2: Octile distance is a norm.

Proof:

The octile distance is $\max(|x|,|y|)+(\sqrt{2}-1)\min(|x|,|y|)$, denoted as L_{octile}

For any 2D vector u and v,

$L_{octile}(u)+L_{octile}(v)$
$=\max(|x_u|,|y_u|)+\max(|x_v|,|y_v|)+(\sqrt{2}-1)(\min(|x_u|,|y_u|)+\min(|x_v|,|y_v|))$
$>=\max(|x_u+x_v|,|y_u+y_v|)+(\sqrt{2}-1)(\min(|x_u+x_v|,|y_u+y_v|))$
$=L_{octile}(u+v)$

So $L_{octile}(u+v)<=L_{octile}(u)+L_{octile}(v)$ is proofed.

$L_{octile}(au)$
$=\max(|ax_u|,|ay_u|)+(\sqrt{2}-1)\min(|ax_u|,|ay_u|)$
$=|a|\max(|x_u|,|y_u|)+a(\sqrt{2}-1)\min(|x_u|,|y_u|)$
$=|a|L_{octile}(u)$

So $L_{octile}(av)=aL_{octile}(v)$ is proofed

$L_{octile}(u)=0$
$\rightarrow\max(|x_u|,|y_u|)+(\sqrt{2}-1)\min(|x_u|,|y_u|)=0$
$\rightarrow max(|x_u|,|y_u|)=0\ \ \&\ \ \min(|x_u|,|y_u|)=0$
$\rightarrow|x_u|=0\ \ \&\ \ |y_u|=0$
$\rightarrow u=0$

So $L_{octile}(u)=0\rightarrow u=0$ is proofed

Based on 3 property proofed above, octile distance is a norm.

Theory 2 A* and JPS with octile distance can't hold the optimal homotopy property for any scenario.

Proof:

If A* and JPS hold the optimal homotopy property, then for any scenario we have $\sigma^*_{grid}\in[\sigma^*]$. Suppose the grid size is sufficiently small. It could be deduced that for the optimal path σ^* with L2 norm (Euclid distance), the distance of σ^* with octile distance is still the optimal with octile norm. However, for equivalence of norms, there is no guarantee to keep the order relation. That is, there exists vector u and v, if $|u|_L<|v|_L$, then $|u|_{L'}>|v|_{L'}$. As for L_2 and L_{octile}, here is a counterexample:

Suppose $u=(1.1,0)$ and $v=(\sqrt{3}/2,1/2)$

$|u|_{L_2}=1.1$,$|v|_{L_2}=1$,

But, $|u|_{L_{octile}}=1.1$,$|v|_{L_{octile}}=1.11$

So A* and JPS can't always hold the optimal homotopy property for any scenario.

CONCLUSIONS

Based on the homotopy class, this paper reveal the fact that from the view of topology, the asymptotic optimality for RRT* and the optimality for A* and JPS are different in $\mathbb{R}^2$. It implies that even though sampling gird's size are shrunk tiny enough, these 2 categories still obtained results with different topology. The reason is that these 2 categories use different norm to compare the path distance. Equivalent norm cannot guarantee that the order relations are unchanged under different norms. So for the same scenario with same start state and goal state, even though the optimal path under octile distance belongs to some homotopy class, the optimal path under Euclid distance may belong to other homotopy class.

We think this paper just gives a preliminary exploration and discussion of the homotopy set for planning problem. We believe with proper strategy, homotopy can be used as a pre-processor to achieve pruning of the search space. On the other hand, from the perspective of testing, this requirement can verify the correctness of the algorithm implementation by constructing metamorphic relation. Since test oracles are hard to generate for planning in $\mathbb{R}^2$.

REFERENCES

[1] Canny, John, and John Reif. "New lower bound techniques for robot motion planning problems." In 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), pp. 49-60. IEEE, 1987..

[2] Fredman, Michael L., and Robert Endre Tarjan. "Fibonacci heaps and their uses in improved network optimization algorithms." Journal of the ACM (JACM) 34, no. 3 (1987): 596-615.

[3] LaValle, Steven M. Planning algorithms. Cambridge university press, 2006.

[4] Hazewinkel, Michiel, ed. Encyclopaedia of Mathematics: Volume 6: Subject Index—Author Index. Springer Science & Business Media, 2013.

[5] Karaman, Sertac, and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning." The international journal of robotics research 30, no. 7 (2011): 846-894.

[6] Ivan, Vladimir, Dmitry Zarubin, Marc Toussaint, Taku Komura, and Sethu Vijayakumar. "Topology-based representations for motion planning and generalization in dynamic environments with interactions." The International Journal of Robotics Research 32, no. 9-10 (2013): 1151-1163.

[7] Bhattacharya, S., Kim, S., Heidarsson, H., Sukhatme, G. S., & Kumar, V. (2015). A topological approach to using cables to separate and manipulate sets of objects. The International Journal of Robotics Research, 34(6), 799-815.

[8] Karaman, Sertac, Matthew R. Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. "Anytime motion planning using the RRT." In 2011 IEEE International Conference on Robotics and Automation, pp. 1478-1483. IEEE, 2011.

[9] Liu, Wei, Zheng Zheng, and Kai-Yuan Cai. "Bi-level programming based real-time path planning for unmanned aerial vehicles." Knowledge-Based Systems 44 (2013): 34-47.

[10] Harabor, Daniel Damir, and Alban Grastien. "Improving jump point search." In Twenty-Fourth International Conference on Automated Planning and Scheduling. 2014.

[11] Snider, Jarrod M. "Automatic steering methods for autonomous automobile path tracking." Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08 (2009).

[12] Lingelbach, Frank. "Path planning using probabilistic cell decomposition." In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 1, pp. 467-472. IEEE, 2004.

[13] Rabin, Steven. "Game AI Pro 2: Collected Wisdom of Game AI Professionals." (2015).

[14] Janson, Lucas, Edward Schmerling, Ashley Clark, and Marco Pavone. "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions." The International journal of robotics research 34, no. 7 (2015): 883-921.

[15] Gammell, Jonathan D., Siddhartha S. Srinivasa, and Timothy D. Barfoot. "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs." In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3067-3074. IEEE, 2015.