



Review

Motion planning for robotics: A review for sampling-based planners

Liding Zhang^{a,*}, Kuanqi Cai^{b,1}, Zewei Sun^a, Zhenshan Bing^a, Chaoqun Wang^c,
Luis Figueredo^{b,d}, Sami Haddadin^b, Alois Knoll^a

^a School of Computation, Information and Technology (CIT), Technical University of Munich, Munich 85748, Germany

^b Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich, Munich 80992, Germany

^c School of Control Science and Engineering, Shandong University, Jinan 250061, China

^d School of Computer Science, University of Nottingham, Nottingham 315199, United Kingdom

ARTICLE INFO

Article history:

Received 1 October 2024

Revised 16 November 2024

Accepted 10 December 2024

Available online 4 January 2025

Keywords:

Robotics

Motion planning

Sampling-based algorithms

ABSTRACT

Recent advancements in robotics have transformed industries such as manufacturing, logistics, surgery, and planetary exploration. A key challenge is developing efficient motion planning algorithms that allow robots to navigate complex environments while avoiding collisions and optimizing metrics like path length, sweep area, execution time, and energy consumption. Among the available algorithms, sampling-based methods have gained the most traction in both research and industry due to their ability to handle complex environments, explore free space, and offer probabilistic completeness along with other formal guarantees. Despite their widespread application, significant challenges still remain. To advance future planning algorithms, it is essential to review the current state-of-the-art solutions and their limitations. In this context, this work aims to shed light on these challenges and assess the development and applicability of sampling-based methods. Furthermore, we aim to provide an in-depth analysis of the design and evaluation of ten of the most popular planners across various scenarios. Our findings highlight the strides made in sampling-based methods while underscoring persistent challenges. This work offers an overview of the important ongoing research in robotic motion planning.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, robotics technology has rapidly advanced across various industries, including manufacturing [1–3], logistics [4–6], robotic surgery [7–9], and planetary exploration [10–13], bringing profound changes. Among the challenges in robotics, developing efficient and effective motion planning algorithms that help robots navigate complex environments, avoid obstacles, and complete tasks with minimal energy consumption and time is a critical task. The core objective of motion planning is to find the optimal path from the starting point to the target location, while considering various constraints such as dynamic environments and non-holonomic motion restrictions. This issue has become a central research topic in the field of robotics.

Fig. 1 shows the state-of-the-art and most common algorithms related to solving the motion planning problem. Motion planning can be divided into global motion planning and local

motion planning. Spatial partitioning approach within configuration space [29], graph-based search algorithms, sampling-based methods, Bio-inspired and meta-heuristic optimization methods belong to global motion planning. Methods like Artificial Potential Field (APF) [26] and Dynamic Window Approach (DWA) [30] belong to local motion planning for real-time and dynamic obstacle avoidance. Graph-based algorithms like Dijkstra [14], wavefront [31], A* [15] and D* [16] are resolution-complete but are computationally expensive for high dimensional complex problems. Bio-inspired and meta-heuristic optimization methods, such as Genetic Algorithm [17], Particle Swarm Optimization [25], and Ant Colony Optimization [32], are well-suited for solving multi-objective optimization problems. However, Like many other evolutionary methods, such as Simulated Annealing [18], Artificial Neural Networks [24], often face issues like getting stuck in local optima and having high computational costs. Additionally, they are highly sensitive to the size of the search space and the problem's data representation scheme. The APF method's advantage is its real-time adaptability for obstacle avoidance, while its drawbacks include susceptibility to local minima and difficulty in navigating between closely spaced obstacles. Among the various methods developed to address this challenge, sampling-based algorithms have emerged as a powerful and versatile approach, particularly suitable for

* Corresponding author.

E-mail address: liding.zhang@tum.de (L. Zhang).

¹ The two authors contributed equally to this work.

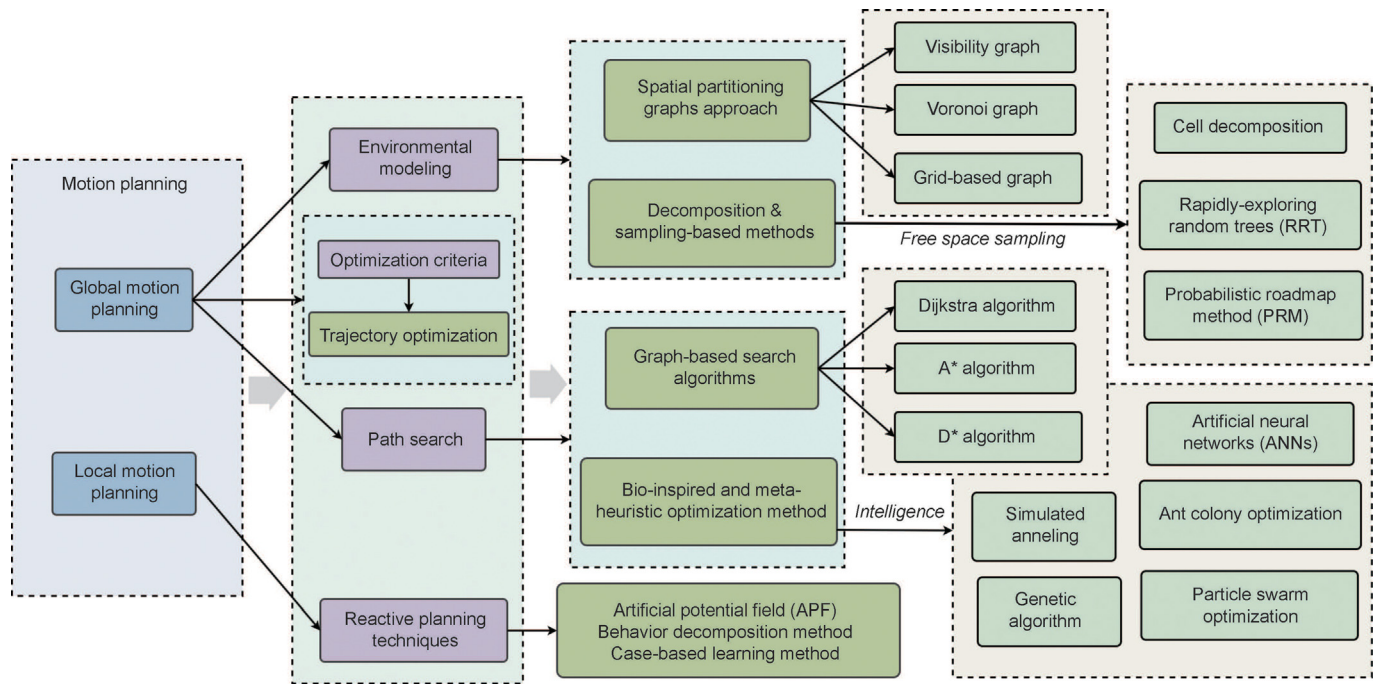


Fig. 1. Overview of the most common motion planning techniques in robotics research. The diagram presents various methods for robotic motion planning, primarily divided into global and local planning. Global motion planning builds a comprehensive representation of the environment through environmental modeling and uses optimization criteria for trajectory optimization to find the optimal path. Path search methods include graph-based search algorithms (such as Dijkstra [14], A* [15], and D* [16]) and bio-inspired and meta-heuristic optimization methods (such as genetic algorithms [17] and simulated annealing [18]). It also employs decomposition and sampling methods (like RRT [19] and PRM [20]) for free space sampling. Additionally, the diagram includes specialized techniques, such as spatial partitioning graphs (visibility graph [21], Voronoi graph [22], and grid-based graph [23]) for environment modeling, and intelligence-driven methods (such as neural networks [24] and particle swarm optimization [25]) that incorporate AI strategies to handle complex and uncertain environments. In contrast, local motion planning focuses on reactive planning techniques that allow adaptive obstacle avoidance and real-time response to dynamic changes, using methods like artificial potential fields [26], behavior decomposition [27], and case-based learning [28].

high-dimensional and complex environments, as illustrated in Fig. 2.

Sampling-based algorithms, such as the Probabilistic Roadmap (PRM; [20]), Expansive Space Trees (EST) [37], and Rapidly-exploring Random Trees (RRT; [19]), have gained main attention due to their ability to efficiently handle the computational complexity associated with planning in high-dimensional spaces. These algorithms work by randomly sampling the configuration space of the robot and incrementally building a graph or tree that represents feasible paths. This approach contrasts with traditional deterministic methods, which often struggle with the curse of dimensionality and require comprehensive environment modeling. Additionally, these methods are highly adaptable and can be easily extended to accommodate various constraints and optimization criteria, making them suitable for a broad spectrum of robotic applications. The flexibility and efficiency of sampling-based approaches have led to extensive research and numerous enhancements, resulting in a rich body of literature that continues to evolve.

In recent decades, researchers in sampling-based motion planning have published numerous comprehensive review papers. An early exploration of the field's history, documented in the review by Latombe [38], discusses the advancements in motion planning, from the initial stages of basic collision detection to the current ability to handle multi-degree-of-freedom robots in complex environments, with non-robotic applications emerging as a key driver for future research. Lindemann and LaValle [39] review the development of sampling-based motion planning during the early 2000's. The survey by Tsianos et al. [40] focus on the application of sampling-based motion planning in realistic settings. Elbanhawi and Simic [41] provide a comprehensive overview of various planners, along with a collection of common

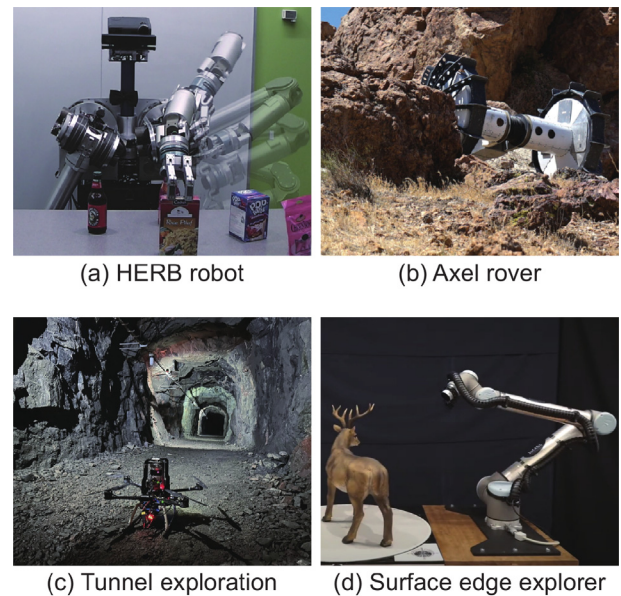


Fig. 2. Illustrates the diverse applications of sampling-based planners across multiple domains, highlighting their versatility. (a) Manipulation of high-dimensional spaces by the HERB robot. Reprinted with permission from [33]. Copyright 2020, licensed under CC BY, Sage. (b) Navigation of the Axel rover (UGV) over uneven terrain. Reprinted with permission from [34]. Copyright 2020, IEEE. (c) UAV exploration within constrained tunnels utilizing tree-based planning methods. Reprinted with permission from [35]. Copyright 2024, IEEE. (d) 3D construction planning in the context of the Next Best View (NBV) problem. Reprinted from [36]. Copyright 2024, licensed under CC BY, Sage.

primitives applicable to a wide range of algorithms. Furthermore, Noreen et al. [42] offer a detailed review of RRT* based path planning approaches, Kingston et al. [43] provide an overview and discuss the problem of motion planning with constraints, while Cai et al. [44] focus on mobile robot motion planning in dynamic environments, Sanchez et al. [45] focus on autonomous mobile robots. Respectively, Gammell and Strub [46] present an extensive overview of asymptotically optimal planners. Subsequently, Orthey et al. [47] provide an excellent review of sampling-based motion planning. Recently, the paper [48] reviews the research on RRT-based improved algorithms from 2021 to 2023, including theoretical improvements and application implementations. With the rapid advancement of sampling-based methods, conducting a survey on the current developments in sampling-based planners is highly valuable and relevant. By systematically reviewing the existing literature and identifying key trends and developments, this survey aims to serve as a valuable resource for researchers and practitioners seeking to understand and advance the capabilities of sampling-based motion planning.

This survey provides a comprehensive overview of sampling-based motion planning algorithms, analyzing their applications and discussing their limitations. Unlike previous reviews, it distinguishes itself by conducting an in-depth analysis of the RRT* framework, detailing various improvements made to its components. These enhancements cover key aspects such as sampling strategies, nearest neighbor search, tree expansion, collision checking, heuristic guidance, and rewiring optimization. The survey also examines how these advancements improve planning efficiency, convergence speed, and path quality. Additionally, systematic evaluations of these improved algorithms in diverse experimental scenarios highlight their strengths and limitations in different applications. By highlighting advances in sampling-based methods and identifying ongoing challenges, this survey aims to serve as a valuable resource for researchers and practitioners seeking a deeper understanding of the current state and future directions in sampling-based planning. Section 2 introduces the motion planning problem and fundamental concepts. In Section 3, we delve into a detailed analysis of traditional sampling-based motion planning algorithms. Section 4 explores the various applications and limitations of these algorithms. In Section 5, we evaluate the performance of ten popular planners across both simulated random scenarios and manipulation tasks in different dimensions. Finally, in Section 6 we make a conclusion.

2. Problem definition

2.1. Definition of planning problem

Firstly, we define the path-related parameters based on the formal definition provided by Karaman and Frazzoli [49]. Let $\mathcal{X} = (0, 1)^d$ represent the configuration space, where $d \in \mathbb{N}$ and $d \geq 2$. Define \mathcal{X}_{not} as the obstacle region, such that $\mathcal{X} \setminus \mathcal{X}_{\text{not}}$ is an open set. The obstacle-free space is denoted as $\mathcal{X}_{\text{free}} = \text{cl}(\mathcal{X} \setminus \mathcal{X}_{\text{not}})$, where $\text{cl}(\cdot)$ denotes the closure of a set. The initial condition x_{start} is an element of $\mathcal{X}_{\text{free}}$, and the goal region $\mathcal{X}_{\text{goal}}$ is an open subset of $\mathcal{X}_{\text{free}}$. A path planning problem is thus defined by the triplet $(\mathcal{X}_{\text{free}}, x_{\text{start}}, \mathcal{X}_{\text{goal}})$.

Consider a function $\sigma : [0, 1] \rightarrow \mathbb{R}^d$ that represents a path in a d -dimensional configuration space. A path σ is considered valid if it is continuous and its total variation, denoted by $TV(\sigma)$, is bounded. The collection of all such valid paths is denoted by Σ .

The total variation for a path σ , that is, $TV(\sigma)$, is defined as the supremum of the sum of Euclidean distances between successive

points along the path. That is

$$TV(\sigma) = \sup_{\{n \in \mathbb{N}, 0 = \tau_0 < \tau_1 < \dots < \tau_n = 1\}} \sum_{i=1}^n \|\sigma(\tau_i) - \sigma(\tau_{i-1})\| \quad (1)$$

When $TV(\sigma) < \infty$, it indicates that the path σ has bounded variation, making it a valid path belonging to the set Σ .

Definition 1 (Path). A function $\sigma : [0, 1] \rightarrow \mathbb{R}^d$ of bounded variation is defined as follows:

- **Path:** It is termed a path if the function σ is continuous.
- **Collision-free path:** A path is called collision-free if it satisfies $\sigma(\tau) \in \mathcal{X}_{\text{free}}$ for every $\tau \in [0, 1]$, where $\mathcal{X}_{\text{free}}$ represents the space free of obstacles.
- **Feasible path:** A path is considered feasible if it is collision-free, starts at the initial point x_{start} , and ends within the goal region or its closure, i.e., $\sigma(1) \in \text{cl}(\mathcal{X}_{\text{goal}})$.

Secondly, the concept of path-planning problems in robotics and automation can be broadly classified into two primary categories [46]:

(1) **Feasible Path Planning:** This type of path planning focuses on identifying a route that effectively moves from a start point to a specified goal, ensuring that the path is achievable. For instance, the path must avoid obstacles or comply with constraints in the environment. There are often multiple solutions to a feasible path planning problem, but the primary objective is directly to reach the goal without concern for the path's quality or efficiency.

Definition 2 (Cost Function). For two paths $\sigma_1, \sigma_2 \in \Sigma$ such that the endpoint of σ_1 matches the starting point of σ_2 ($\sigma_1(1) = \sigma_2(0)$) [49], their concatenated path $(\sigma_1 | \sigma_2)$ is defined as

$$(\sigma_1 | \sigma_2)(t) = \begin{cases} \sigma_1(2t), & \text{if } t \in [0, \frac{1}{2}] \\ \sigma_2(2t - 1), & \text{if } t \in (\frac{1}{2}, 1] \end{cases} \quad (2)$$

The cost of a concatenated path $\sigma = (\sigma_1 | \sigma_2)$ is always greater than or equal to the maximum cost of its individual segments:

$$c(\sigma) \geq \max\{c(\sigma_1), c(\sigma_2)\} \quad (3)$$

The cost of a path σ is upper-bounded by a constant multiple of its total variation $TV(\sigma)$:

$$c(\sigma) \leq k \cdot TV(\sigma) \quad (4)$$

where $k \in [0, \infty)$ is a scaling constant. A path has zero cost only if it is trivial, meaning the path remains at the same state throughout:

$$c(\sigma) = 0 \iff \forall t \in [0, 1], \sigma(t) = \sigma(0) \quad (5)$$

Definition 3 (The Feasible Path Planning Problem). Given a path planning problem $(\mathcal{X}_{\text{free}}, x_{\text{start}}, \mathcal{X}_{\text{goal}})$, the feasible path planning problem aims to find any valid path from the start to the goal, according to a predefined rule, or to report failure if no such path exists:

$$\sigma \in \{\sigma_0 \in \Sigma \mid \sigma_0(0) = x_{\text{start}}, \sigma_0(1) \in \mathcal{X}_{\text{goal}}\} \quad (6)$$

(2) **Optimal Path Planning:** This type of path planning goes beyond simply finding a valid route by focusing on identifying the most efficient path according to specific criteria. The objective might be to minimize factors such as distance traveled, time taken, energy usage, or other relevant task-specific factors. Consequently, the optimal path planning problem seeks to determine the best solution among all feasible options.

Definition 4 (*The Optimal Path Planning Problem*). Given a path planning problem $(\mathcal{X}_{\text{free}}, x_{\text{start}}, \mathcal{X}_{\text{goal}})$ and a cost function $c: \Sigma \rightarrow \mathbb{R}_{\geq 0}$, the task is to find a feasible path σ^* such that

$$c(\sigma^*) = \min\{c(\sigma) : \sigma \text{ is feasible}\} \quad (7)$$

$$\sigma^* = \arg \min_{\sigma \in \Sigma} \{c(\sigma) \mid \sigma(0) = x_{\text{start}}, \sigma(1) \in \mathcal{X}_{\text{goal}}, \forall t \in [0, 1], \sigma(t) \in \mathcal{X}_{\text{free}}\} \quad (8)$$

where $\mathbb{R}_{\geq 0}$ represents the set of non-negative real numbers. The cost of this optimal path is denoted as c^* .

2.2. Analysis of sampling-based planning

Probabilistic Completeness is an important property for sampling-based path planning algorithms, which means that if there exists a feasible path from the starting position to the target position, the probability that the algorithm finds this path approaches 1 as the number of samples increases infinitely.

Definition 5 (*Probabilistic Completeness*). An algorithm is said to be *probabilistically complete* if the probability of finding a feasible path, assuming one exists, converges to 1 as the number of samples q goes to infinity. Mathematically, this is represented as

$$\liminf_{q \rightarrow \infty} P(\Sigma_q \neq \emptyset) = 1 \quad (9)$$

where Σ_q is the set of feasible paths found using q samples, and P denotes probability

Definition 6 (*Asymptotic Optimality Definition*). An algorithm is said to be asymptotically optimal if, for any positive number ϵ , the following holds:

$$\lim_{q \rightarrow \infty} P(c(\pi_q) - c^* < \epsilon) = 1 \quad (10)$$

where $c(\pi_q)$ is the cost of the path found with q samples, c^* is the cost of the optimal path, and P denotes probability.

3. Optimal path planning with sampling-based methods

3.1. Sampling-based planner framework

The Rapidly-exploring Random Tree (RRT) [19] algorithm, introduced by LaValle, is an incremental, random sampling algorithm designed for high-dimensional space problems involving algebraic constraints (resulting from obstacles) and differential constraints (arising from non-holonomic and dynamic environments). The RRT algorithm's key advantage is that it does not require a system model or geometric partitioning of the search area. It offers high coverage of the search space, enabling extensive exploration of unknown regions. However, the RRT algorithm also has a significant computational cost. Due to the randomness of the sampling process, the resulting path is often feasible but not optimal. To address this, several RRT variants aim to optimize the generated path. One notable variant, the Rapidly-exploring Random Tree star-based planner (RRT*) [49], was introduced by S. Karaman and E. Frazzoli in 2011. The primary difference between RRT and RRT* is that RRT* includes a process of searching the neighboring nodes of each newly generated node to select a parent node with the lowest cost. Additionally, it involves a rewiring process to further reduce the path cost. This approach represents a breakthrough for optimal path planning in high-dimensional spaces. RRT* is asymptotically optimal, meaning it will converge to the optimal solution if given sufficient computation time.

To find the optimal path in the planning task, the Rapidly-exploring Random Tree star-based planner (RRT*) [49] is widely used. The workflow of RRT* is shown in Algorithm 1. First, the

Algorithm 1: Rapidly Exploring Random Tree Star (RRT*)

Input : Start state x_{start} , goal region $\mathcal{X}_{\text{goal}}$
Output: Feasible tree \mathcal{T}

- 1 $V \leftarrow x_{\text{start}}, E \leftarrow \emptyset, \mathcal{T} = (V, E) \leftarrow \text{RRT}^*(x_{\text{start}})$
- 2 **for** $i \leftarrow 1$ **to** N $\triangleright N$ represents iteration times **do**
- 3 $x_{\text{rand}} \leftarrow \text{sampling}(i)$
- 4 $x_{\text{nearest}} \leftarrow \text{nearest}(\mathcal{T}, x_{\text{rand}})$
- 5 $x_{\text{new}} \leftarrow \text{steer}(x_{\text{nearest}}, x_{\text{rand}})$
- 6 **if** $\text{collisionFree}(x_{\text{nearest}}, x_{\text{new}})$ **then**
- 7 $x_{\text{near}} \leftarrow \text{near}(\mathcal{T}, x_{\text{new}}, k \text{ or } r)$ $\triangleright k$ represents
k-nearest neighbor strategy, r represents r-nearest
neighbor strategy
- 8 $x_{\text{min}} \leftarrow \arg \min_{x_{\text{near}} \in \mathcal{X}_{\text{near}}} (\text{cost}(x_{\text{near}}) + \|x_{\text{near}} - x_{\text{new}}\|)$
- 9 \triangleright heuristic for minimal path cost, $\|\cdot\|$ denote as L^2
norm
- 10 $\mathcal{T} \leftarrow \text{extend}(\mathcal{T}, x_{\text{new}})$
- 11 $\text{rewire}(\mathcal{T}, x_{\text{near}}, x_{\text{min}}, x_{\text{new}})$
- 12 **if** $x_{\text{new}} \in \mathcal{X}_{\text{goal}}$ **then**
- 13 **return** \mathcal{T}

14 **return failure**

search tree \mathcal{T} is initialized with the root node x_{start} . Random sample points x_{rand} are generated through the **Sampling**(\cdot) function within the search space. Then, the **Nearest**(\cdot) function identifies the nearest node x_{near} in the tree \mathcal{T} to the sample point x_{rand} . Next, the **Steer**(\cdot) function moves from x_{near} towards x_{rand} by a defined step size to obtain a new node x_{new} and creates the edge $(x_{\text{near}}, x_{\text{new}})$. After that, the **CollisionFree**(\cdot) method checks whether this edge collides with any obstacles or boundaries in the map. If no collision is detected, the tree is expanded by adding x_{new} to the set of vertices and the edge $(x_{\text{near}}, x_{\text{new}})$ to the set of edges. If a collision is detected, the newly generated node is discarded, and the algorithm retries the sampling process. Once x_{new} is added, the algorithm performs an **Extend**(\cdot) operation by searching within a defined neighborhood around x_{new} to identify a parent vertex that minimizes the overall cost from the initial state. A **Heuristic**(\cdot) method is employed at this stage to estimate the cost-to-go from x_{new} to the goal, allowing the algorithm to prioritize more promising nodes and improve convergence towards the optimal solution. Subsequently, a **Rewire**(\cdot) step is conducted by checking neighboring vertices to see if their connection through x_{new} reduces their path cost. If so, their parent vertices are updated accordingly to reflect the more efficient path.

As the number of iterations increases, the sampling-based algorithm progressively generates paths that approach the optimal solution. Ultimately, the algorithm outputs a path that is refined with each iteration, ensuring it is close to the shortest path.

Researchers have proposed various strategies to improve different components of the sampling-based planner framework, including the **Sampling**(\cdot) function, **Nearest**(\cdot) function, **Steer**(\cdot) function, **CollisionFree**(\cdot) method, **Extend**(\cdot) operation, **Heuristic**(\cdot) method and **Rewire**(\cdot) step. Fig. 3 provides a general overview of these technological advancements, which will be further detailed in the following sections.

3.2. Advanced sampling approaches

The **Sampling**(\cdot) function generates a sequence of sample points within the search space. Sampling functions are classified as **Informed Set**, **Learning-based**, **Adaptive Sampling**, **Smart Biased Sampling** and **Kinodynamic**.

Informed Set, such as Informed RRT* [33,50] (Fig. 4 (a)), which restricts the node sampling range by creating a hyper-ellipsoid

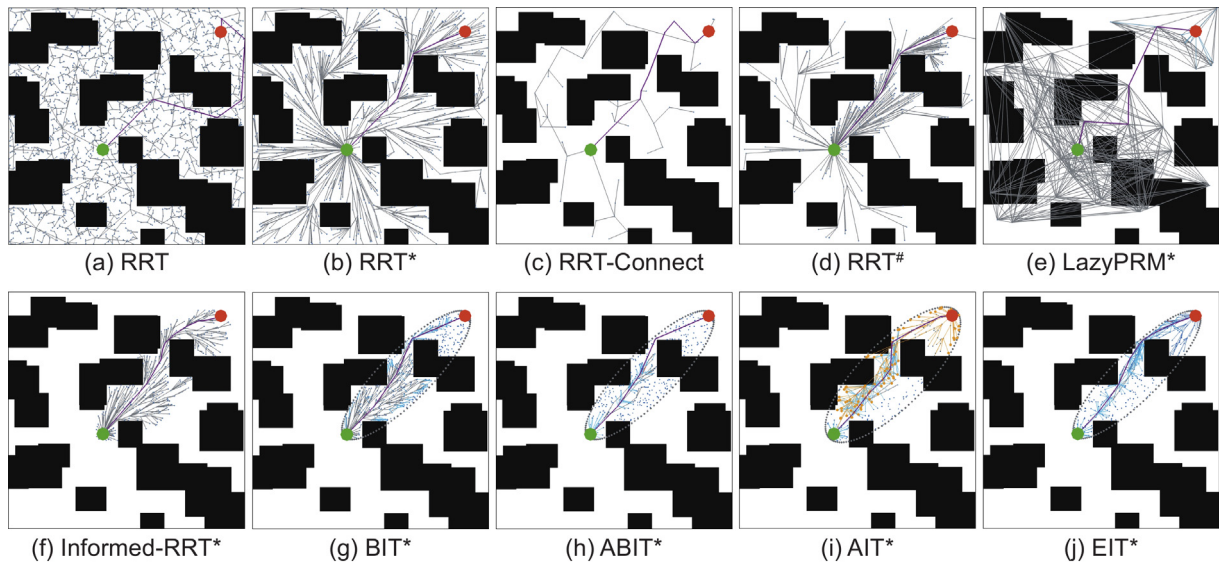


Fig. 3. The 2D representation of the search trees constructed by RRT, RRT*, RRTConnect, RRT#, lazyPRM*, Informed RRT*, BIT*, ABIT*, AIT*, and EIT* to find an initial solution when optimizing path length (a–j). The start and goal are represented by a green dot and a red dot, respectively. Sampled states are shown as small black dots. Obstacles in the state space are indicated by black rectangles. The initial solutions are depicted in purple. (a) RRT randomly explores the state space, fully evaluating many unnecessary edges for both objectives. (b) RRT* similarly evaluates many edges that do not contribute to the initial solution for both path length and clearance. (c) RRT-Connect is more efficient, connecting start and goal faster, but still evaluates redundant edges. (d) RRT# reduces redundant edge evaluation using graph refinement. (e) lazyPRM* avoids full evaluation by deferring edge checking until necessary. (f) Informed RRT* refines its exploration by focusing on promising regions. (g) BIT* ranks edges by potential solution cost, reducing unnecessary evaluations. (h) ABIT* improves BIT* by incorporating more efficient search heuristics. (i) AIT* utilizes problem-specific heuristics to further reduce redundant evaluations. (j) EIT* combines cost and effort heuristics, making it robust even when problem-specific heuristics are unavailable.

subset [56] for sampling and conducts direct sampling within this subset. By constraining the algorithm's sampling region, this subset increases the probability of selecting relevant nodes, thereby boosting the algorithm's efficiency. BIT* [57,58] builds on Informed RRT*, with batch sampling in the hyper-ellipsoid subset, this leads to a further enhancement in the algorithm's efficiency. Respectively, Ding et al. [51] propose the expanding path RRT* (EP-RRT*) (Fig. 4 (b) based on heuristic sampling in the path expansion area. Another paper [59] proposes a cylinder-based informed rapid exploration random tree (Cyl-iRRT*) path planning algorithm, which biases the sampling of new candidate states into an admissible cylindrical subset around the centerline from start to goal positions via rejection sampling to quickly find the homotopy optimal path in a 3-D environment.

Learning-based Sampling, such as Neural RRT* [52] (Fig. 4 (c) and Neural Informed RRT* [60], guides the sampling direction by training neural networks, which are particularly effective in navigating narrow passages and complex environments. Another paper [61] uses a conditional variational auto-encoder (CVAE) to learn sampling distributions, Molina et al. [62] propose to use convolutional neural networks (CNNs) to identify critical regions for robot planning so that the sampling process can be biased to these regions.

Adaptive Sampling, as demonstrated in methods like the Rapidly-Exploring Adaptive Sampling Tree* (RAST*) [63], Gaussian Mixture Regression Rapidly exploring Random Tree* (GMR-RRT*) [53] (Fig. 4 (d) and the similar works such as [64,65], leverages an adaptive sampling approach to prioritize information-dense regions. This strategy effectively reduces localization uncertainty by concentrating on areas with higher information gain.

Smart Biased Sampling, like RRT*-SMART [54] (Fig. 4 (e)), mRRT*-Smart [66] and RRT*SMART-A* [67] employs intelligent sampling around key nodes from an initial path, accelerating convergence.

Kinodynamics, Boeuf et al. [68] present an incremental state-space sampling technique to avoid generating local trajectories

that violate kinodynamic constraints. Other papers prioritize generating samples that maximize clearance, meaning the distance between the robot and its surroundings. This is typically done by sampling a feasible state and taking random steps to further increase the clearance [55,69] (Fig. 4 (f)).

3.3. Nearest neighbor methods

The **Nearest**(\cdot) function refers to a key step in the sampling-based algorithms where the nearest node in the tree is found for a randomly sampled point and used as the base to expand the tree towards that point. An important component in this process is the definition of the distance metric between nodes, which directly impacts both the efficiency and accuracy of the algorithm [70]. This distance metric can vary depending on the requirements of the environment and constraints. For example, when non-holonomic constraints are present, factors such as directionality and acceleration may need to be incorporated into the distance metric [71]. Traditional nearest-neighbor searches are computationally expensive due to their linear search nature. Techniques like KD trees [72] address these challenges by employing advanced data structures and algorithms. **KD trees** partition the space recursively, enabling efficient multi-dimensional searches. This method improves the search process compared to a linear search, with an average time complexity of logarithmic order, making it highly suitable for large, high-dimensional datasets.

In contrast, the **k-nearest neighbor (k-NN) strategy** in sampling-based algorithms can also be implemented using formulas that adaptively determine the number of neighbors k for each node based on the number of samples q and the dimensionality of the space n . The number of neighbors $k(q)$ is calculated as

$$k(q) := \eta e^{\left(1 + \frac{1}{n}\right) \log(q)} \quad (11)$$

where $\eta > 1$ is a tuning parameter. This strategy ensures that the tree expands efficiently while maintaining good coverage of the space. Pan et al. [73] propose a novel approach for rapid

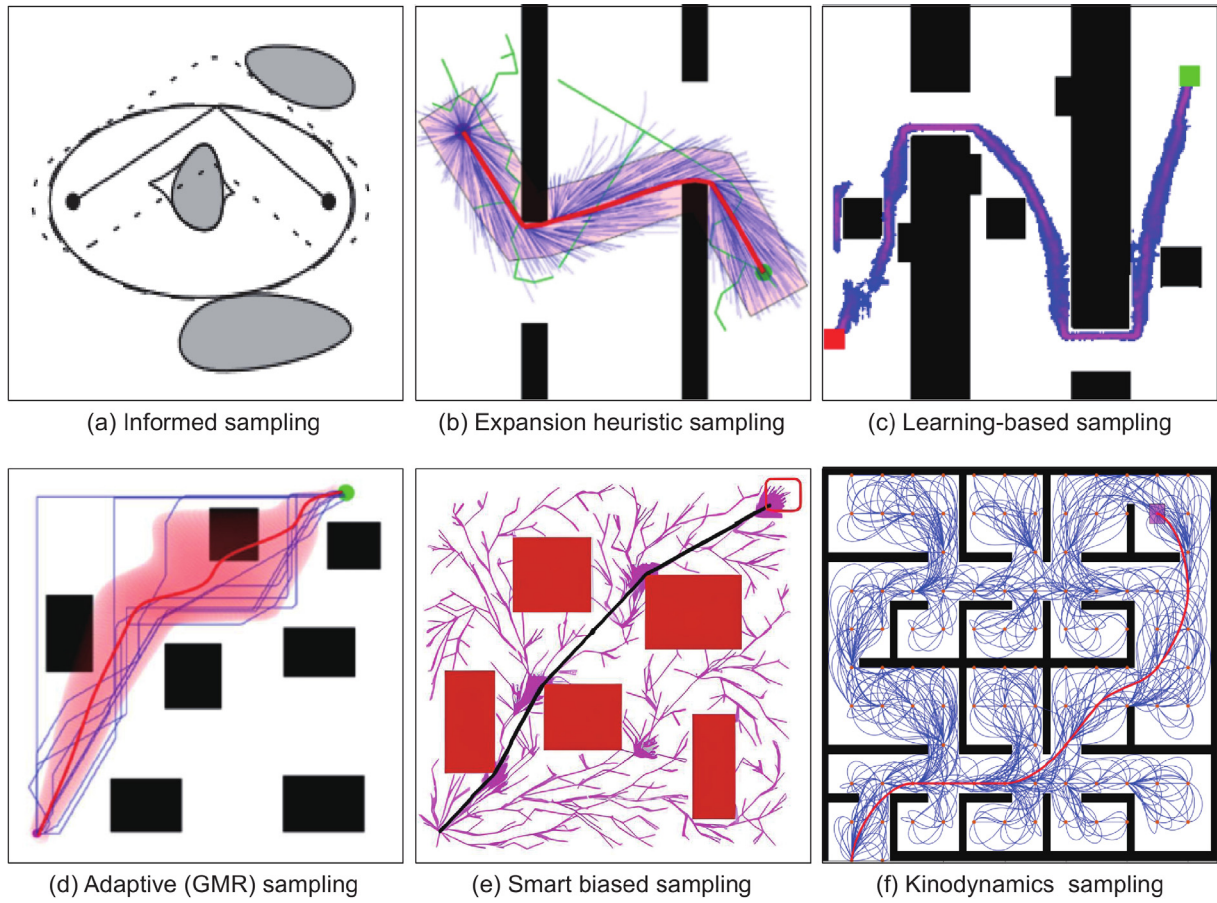


Fig. 4. (a) Informed RRT* enhances the standard RRT* by focusing sampling in an ellipsoidal region that contains the optimal path, thereby improving convergence speed and solution quality. Reprinted with permission from [50]. Copyright 2018, IEEE. (b) EP-RRT* expands the initial path to define an “expansion area”, where heuristic sampling is performed to refine the path and converge to an optimal solution more efficiently. Reprinted with permission from [51]. Copyright 2023, Elsevier B.V. (c) Neural-RRT* combines RRT* with learning-based features (e.g., neural networks) and optimizations for sampling prediction and convergence time reduction, resulting in rapid pathfinding in cluttered environments. Reprinted with permission from [52]. Copyright 2020, IEEE. (d) GMR-RRT* incorporates demonstrated trajectories into a Gaussian Mixture Model to define exploration-worthy regions. Reprinted with permission from [53]. Copyright 2022, IEEE. (e) RRT*-Smart utilizes the triangular inequality to improve paths and employs intelligent sampling techniques for the path edges. Reprinted from Nasir et al. [54]. Copyright 2013, licensed under CC BY 3.0, licensee InTech. (f) This method presents pre-computed kinodynamic motion primitives for sampling and efficient path planning. Reprinted with permission from [55]. Copyright 2019, Springer Science Business Media, LLC, part of Springer Nature.

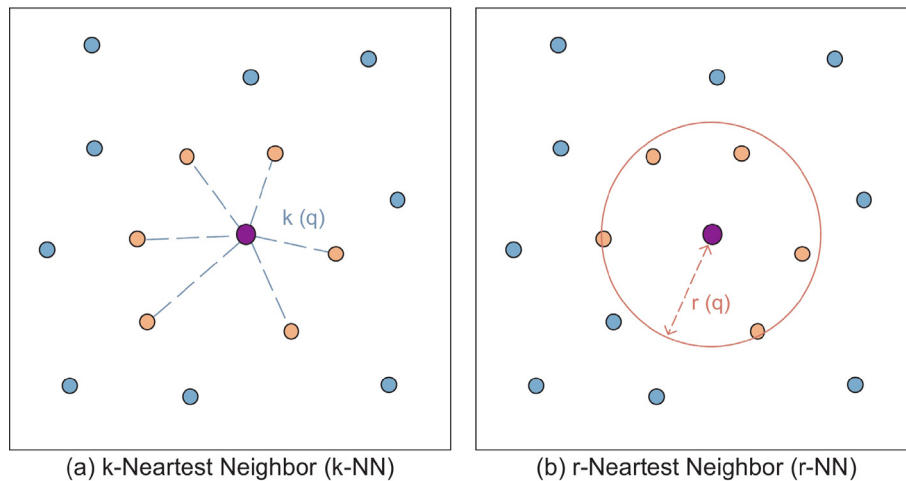


Fig. 5. Two nearest neighbor strategies. (a) illustrates the k -nearest neighbors approach, which uses a constant number of neighbors $k(q)$ as defined in Eq. (11). This algorithm identifies the closest k points to a query point q based on a specified distance metric. (b) depicts the r -nearest neighbors strategy, where the radius $r(q)$, as defined in Eq. (12), dynamically changes based on the current number of samples per batch. This approach finds all neighbors within a radius r around the query point q .

probabilistic collision checking aimed at boosting the efficiency of sampling-based motion planning. In this method, the k-NN strategy is employed to identify the closest prior query sample to the new query configuration. The results demonstrate that this integrated approach enhances the RRT-based path planner by speeding up local pathfinding and optimizing the search sequence on the roadmap. These findings have been validated on both rigid and articulated robots (see Fig. 5).

Respectively, the **r-nearest neighbor (r-NN) strategy** used in the r-disc (i.e., $r(q)$ as connection radius) to connect nearest nodes is computed by:

$$r(q) := 2\eta \left(1 + \frac{1}{n}\right)^{\frac{1}{n}} \left(\frac{\min\{\lambda(\mathcal{X}), \lambda(\hat{\mathcal{X}})\}}{\lambda(B_{1,n})}\right)^{\frac{1}{n}} \left(\frac{\log(q)}{q}\right)^{\frac{1}{n}} \quad (12)$$

where $\lambda(\cdot)$ denotes the Lebesgue measure, and $B_{1,n}$ is the n -dimensional unit ball, $\hat{\mathcal{X}}$ is defined as the informed set, this providing a way to scale the connection radius according to the sample distribution and the space's properties.

In addition to the aforementioned r-disc strategy, Zhang et al. [74] proposed an elliptical nearest neighbor method based on Coulomb's law and invalid vertices in obstacles, Kleinbort et al. [75,76] demonstrate that the r-disc approach can achieve superior performance compared to the k-nearest variant, though the connection radius $r(q)$ must be calibrated according to the characteristics of the state space. Properly tuning the connection radius for specific applications can improve search efficiency. Furthermore, faster-decreasing radii are presented by Janson et al. [77, 78] and Tsao et al. [79], enabling the r-disc strategy to perform even better when handling varying dimensions and sampling densities.

3.4. Tree-based expansion strategies

Based on the step size constraint, from the nearest node, the **Steer**(\cdot) function takes a small step in the direction of the random point, generating a new node, and creates the edge. Wang et al. [80,81] present variant step size RRT, which adaptively changes the step size of the tree according to the location of obstacles. Similarly, another paper [82] changes the step size adaptively according to the density of obstacles. Subsequently, Yang et al. [83] present a novel Variable Step Size (VSS) strategy based on RRT*. The VSS strategy adapts the expansion step size dynamically by considering both the direction of the vertex and the target point within the random tree, with the goal of accelerating the approach towards the target point. In another paper [84] the gradient descent method is employed to adapt the step size, causing it to gradually decrease as the goal region is approached. Additionally, Li et al. [85] present a gravity adaptive step size strategy.

To quickly find a feasible path, the bidirectional expansion strategy for steering was developed. Bidirectional RRT, where two trees grow simultaneously from the start and goal points, seeking to connect with each other, as illustrated in Fig. 6.

Initial approach, like RRT-Connect is introduced by Kuffner and LaValle [86], establishes the foundational method of constructing two trees from start and goal points and connecting them, thereby greatly improving search efficiency. However, RRT-connect is not asymptotically optimal like RRT*. To solve this, Akgun et al. [87] present Bidirectional RRT* (B-RRT*), incorporate an existing bi-directional approach to search which decreases the time to find an initial path. Similarly, Jordan et al. [88] present Optimal B-RRT*, an asymptotically optimal bidirectional method that achieves an improved convergence rate by incorporating several heuristic techniques. While Qureshi et al. [89] introduce a new variant called Intelligent Bidirectional-RRT* (IB-RRT*) which is an improved variant of the optimal RRT* and bidirectional version of RRT* (B-RRT*) algorithms. IB-RRT* utilizes the

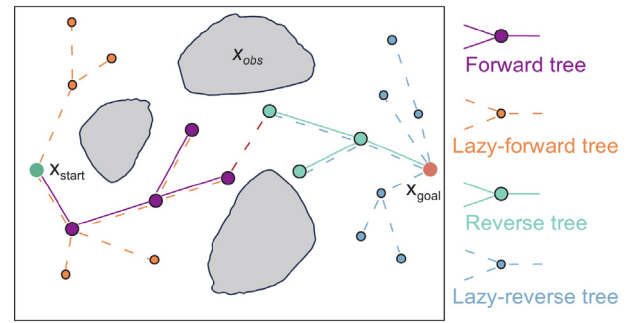


Fig. 6. The figure illustrates bidirectional search from both the start and goal states, showing four trees: the forward tree in purple, the lazy-forward tree in orange, the reverse tree in light green, and the lazy-reverse tree in blue. Bidirectional search reduces pathfinding time, while the lazy search in both the forward and reverse trees performs sparse collision checking, postponing the computationally expensive full collision check. Further details will be provided in Section 3.5.

bidirectional trees approach and introduces an intelligent sample insertion heuristic for fast convergence to the optimal path solution using uniform sampling heuristics. Furthermore, Tahir et al. [90] propose the Potentially Guided Intelligent Bi-directional RRT* (PIB-RRT*) and Potentially Guided Bi-directional RRT* (PB-RRT*), which are extensions of Bi-directional RRT* (B-RRT*) and Intelligent Bi-directional RRT* (IB-RRT*). These methods greatly enhance the convergence rate and utilize memory more efficiently in cluttered environments. Additionally, the BI²RRT* algorithm [91] builds upon the Informed RRT* [33] by implementing a bidirectional search strategy, which helps in finding initial solutions faster and allows more time for refinement. Subsequently, Yi et al. [92] combine homotopy topological spaces with bidirectional trees in the Homotopy-aware RRT*, increasing the probability of finding optimal paths by exploring different topological spaces. Moreover, Lin et al. [93] propose the Bidirectional Homotopy-Guided RRT (BH-RRT), which uses obstacle contour information to guide tree growth, improving success rates. Wang et al. [94] introduced the Bidirectional-Unidirectional RRT Extend Function, which switches from bidirectional to unidirectional search to overcome complex boundary problems, enhancing search efficiency. Another paper [95] present a goal-biased bidirectional RRT with curve smoothing, connecting tree parts with Bezier curves to meet kinematic constraints, thus achieving higher success rates and shorter search times. Recently, Peng et al. [96] develop the Improved Bidirectional RRT*, utilizing an artificial potential field to reduce randomness and inflection points, resulting in shorter and more efficient paths.

3.5. Collision check methods

After the **Steer**(\cdot) function creates the edge, the **Collision-Free**(\cdot) method involves verifying whether the path between two nodes intersects with any obstacles. This basic approach, as detailed in LaValle and Kuffner's 2001 seminal paper on Rapidly-exploring Random Trees (RRT) [97], involves checking the line segment between the current node and the new node for collisions. Over time, various methods have been developed to enhance the efficiency of collision checking.

Lazy Collision Checking, such as Lazy PRM [98] and Fuzzy PRM [99], both initially assume the path is collision-free and perform collision checks only when necessary. If a collision is later detected, the path is re-evaluated and corrected. However, neither Lazy PRM nor Fuzzy PRM guarantees solution quality. To solve this, Hauser et al. [100] present two novel motion planners, Lazy-PRM* and Lazy-RRG*, they are almost-surely asymptotically

optimal algorithms that grow a network of feasible vertices connected by edges. Flexible Informed Trees (FIT*) [56] improve performance by using adaptive batch sizes. Dense batches are used initially to find a solution, followed by sparse batches for optimization, reducing costly operations like edge validations. Edges are not immediately checked for collision, but rather are checked only when a better path to the goal is found. This strategy avoids checking the vast majority of edges that have no chance of being on an optimal path. Another paper [101], Adaptive Lazy Collision Checking delays collision checking in regions likely to be free of obstacles while checking early in other regions to reduce optimistic thrashing, enhancing the planner's performance in complex environments. Most recently, Neural Network Collision Checking has been introduced by Kew et al. [102] through ClearanceNet, a neural network-based heuristic that predicts collision clearance, facilitating parallel RRT processing and significantly accelerating the collision checking process.

3.6. Tree extension techniques

After x_{new} is added, the algorithm executes an **Extend**(\cdot) operation to determine the parent node of x_{new} , the node that results in the lowest cost from the starting point to x_{new} is selected as its parent node.

LaValle and Kuffner's paper "Rapidly-exploring Random Trees: Progress and Prospects" [97] introduce the basic principles of the standard RRT algorithm and the Extend Tree process, laying the foundation for subsequent advancements. FMT* [103] (Fast Marching Tree) leverages the Fast Marching Method (FMM) to enhance the efficiency of RRT in high-dimensional spaces by integrating efficient sampling and searching strategies. Specifically, the FMT* algorithm uses FMM to quickly evaluate the quality of nodes when building the tree structure, accelerating the convergence of the path planning process. This makes FMT* particularly effective in complex, high-dimensional path planning problems. Subsequently, Qureshi et al. [104] introduce the P-RRT*, which integrates Artificial Potential Fields (APF) into the RRT* framework to provide a more directed exploration and faster convergence. Another strategy is Quick-RRT* [105], which uses triangular inequality for parent node selection. Extending these approaches, Li et al. [106] propose the PQ-RRT*, which combines the strengths of P-RRT* and Quick-RRT*, further enhancing the convergence speed by improving both the sampling strategy and the optimization procedures used in the tree expansion process. Moreover, Liao et al. [107] present F-RRT*, which improves path cost by generating a parent node for the random point rather than choosing one from the existing vertices. Similarly, another paper [108] introduces Fast-RRT*. In order to achieve a path with a lower cost compared to the RRT* algorithm, the ancestors of the nearest node are taken into account up to the initial state when selecting a parent for the new node. Next, Armstrong et al. [109] introduce the Assisting Metric RRT* (AM-RRT*). This algorithm incorporates an assisting metric that combines Euclidean distance with a novel metric to optimize tree growth, extension, and rewiring, thereby improving coverage and path quality. This approach not only considers distance but also factors such as path smoothness or traversability, resulting in higher quality paths while maintaining efficient search performance. Respectively, Wanga et al. [110] present an evaluation function for growth points based on the adaptive resolution octree map, which guides the generation of RRT paths towards a more intentional extension strategy. Additionally, the algorithm reselects parent nodes and candidate nodes, followed by a rewiring process. As a result, the improved RRT algorithm eliminates redundant bifurcations in the growth tree, reduces the number of sampling instances, and significantly improves growth efficiency.

3.7. Heuristic-guided exploration methods

While the tree is extending, the **Heuristic**(\cdot) method is used to estimate the remaining cost from x_{new} to the goal, enabling the algorithm to prioritize nodes that appear more promising and accelerate convergence towards an optimal solution.

Heuristic-guided sampling enhances the efficiency of path planning algorithms by biasing the extending process towards regions more likely to yield high-quality paths [111]. The Heuristically Guided RRT (hRRT) algorithm [112] proposed by Urmson and Simmons utilizes a heuristic cost function to evaluate the quality of different states, thereby optimizing the sampling process. This algorithm has improved performance but does not provide any guarantee on the quality of its solution. To solve this, Gammell et al. [33,50] present Informed RRT* enhances RRT* by incorporating an admissible cost heuristic, ensuring that only states capable of improving the current solution are considered. This approach accelerates the convergence rate of RRT* while preserving its asymptotic optimality with high probability. Following this, Gammell et al. [57,58] develop the Batch Informed Trees (BIT*), which compactly groups states into an implicit random geometric graph (RGG) [113], employing step-wise search similar to Lifelong Planning A* (LPA*) [114], based on expected solution quality. This approach combines heuristic-guided search with batch sampling and heuristic sorting of the search process, ensuring rapid convergence to high-quality solutions in both low- and high-dimensional spaces. Next, the Adaptively Informed Trees (AIT*) [34], improves upon BIT* by utilizing the same progressively denser RGG approximation, but it employs an asymmetric bidirectional search. This search calculates and leverages a more accurate cost heuristic tailored to each specific RGG approximation. Then, the Advanced BIT* (ABIT*) [115] further improves performance by integrating advanced graph-search techniques, such as heuristic inflation and search truncation. Furthermore, the Greedy BIT* (GBIT*) [116] extends ABIT* by incorporating a greedy search policy inspired by RRT-Connect [86], which accelerates the discovery of initial solutions and enhances convergence speed, while maintaining the asymptotic optimality of the solution. Respectively, Effort Informed Trees (EIT*) [117] builds upon AIT* by utilizing problem-specific information in a way that takes advantage of informative admissible cost heuristics when available, but still performs effectively in their absence. It achieves this by incorporating additional types of problem-specific data, such as the computational effort required to validate a path. This generalization extends asymptotically optimal informed path planning algorithms to a wider range of problems, including those without effective a priori cost heuristics. Moreover, Hartmann et al. [118] present Effort Informed Roadmaps (EIRM*), EIRM* extends the EIT* approach to efficiently handle multiquery problems by actively reusing computational effort and managing graph size over multiple planning queries, thereby improving both speed and efficiency in complex environments. Subsequently, Li et al. [119] propose Symmetrical Bidirectional Optimal Path Planning with Adaptive Heuristic (BiAIT*), in contrast to AIT*, BiAIT* utilizes a symmetric bidirectional search for both heuristic calculation and space exploration. This method allows BiAIT* to find an initial solution faster than AIT* and to update the heuristic more efficiently when a collision occurs.

3.8. Various rewiring for tree optimization

Subsequently, a **Rewire**(\cdot) step is performed by evaluating neighboring vertices to determine if connecting through x_{new} would lower their overall path cost. If a more efficient route is found, the parent vertices of those neighboring nodes are updated accordingly to reflect the improved path.

Karaman and Frazzoli [49] are the first to present the idea of using the rewiring method to ensure asymptotic optimality. Building on top of the RRT* presented in [49], RRT-sharp (RRT[#]) [120,121] employs a more efficient rewiring cascade that not only propagates reduced cost-to-goal information throughout the graph but also adjusts connections within local neighborhoods when even lower cost-to-goal values are achievable. Moreover, RRT^X [122] introduces a rewiring strategy to adapt to changes in dynamic environments. This algorithm quickly recalculates paths when the environment changes for dynamic obstacles. Subsequently, Quick-RRT* [105] enhances the rewiring procedure by incorporating the ancestry of nearby vertices, instead of only checking if a new node can provide a better connection for its direct neighbors. This improves the path optimization process by generating straighter paths and reducing detours. Respectively, The Real-Time RRT* (RT-RRT*) algorithm [123], based on RRT* [49] and Informed RRT* [33], introduces an online tree rewiring strategy for real-time path planning in dynamic environments. This method enables the tree to quickly adapt to changes, maintaining near-optimal paths without rebuilding the entire tree. Additionally, a triangular inequality-based rewiring method [124] is applied to the RRT-Connect [86] algorithm, demonstrating faster planning times and shorter path lengths compared to both RRT and RRT-Connect algorithms. Similarly, the Post Triangular Rewiring method [125] further improves the optimality of paths generated by the RRT algorithm by leveraging the triangular inequality principle.

To improve path smoothness and reduce length, path short-cutting and smoothing techniques are commonly used. Path short-cutting optimizes the route by removing redundant intermediate points after checking the feasibility between nodes along the path. Advanced hybrid methods, such as BITKOMO [126] and Regionally Accelerated BIT* (RABIT*) [127], combine sampling-based planners with trajectory optimization frameworks to achieve a balance between global exploration and local optimization. BITKOMO [126] integrates Batch Informed Trees (BIT*) [57,58] with the K-Order Markov Optimization (KOMO) [128] framework, ensuring asymptotic optimality while benefiting from KOMO's fast convergence. Similarly, RABIT* [127] combines Batch Informed Trees (BIT*) [57,58] with Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [129], using local optimization to refine paths, accelerate convergence, and improve performance in challenging regions, such as narrow passages, all while maintaining guarantees of global optimality. Trajectory optimization methods like KOMO [128], CHOMP [129], STOMP [130], and TrajOpt [131] exploit gradient and second-order information to converge efficiently to a local optimal solution. Path smoothing is essential for enhancing the quality of planned paths by reducing sharp turns and discontinuities, ensuring smoothness and feasibility. Among smoothing techniques, Bezier curve smoothing, a parametric method defined by control points, is widely applied. Neto et al. [132] develop a method based on seventh-order Bezier curve improved RRT, designed for autonomous vehicles with holonomic constraints. This approach efficiently generates smooth paths that satisfy kinematic constraints, minimize tree nodes, and enhance obstacle avoidance. Li et al. [133] propose a path planning algorithm for AUVs that integrates improved RRT with Bezier optimization, balancing search efficiency, path smoothness, and feasibility, with its effectiveness validated through simulations. Similarly, Durakli et al. [134] introduce a novel method leveraging Bezier curves to optimize paths from traditional algorithms, using node pruning and curve smoothing to significantly improve path smoothness and optimality in known environments. In addition to Bezier curves, cubic splines are another widely used smoothing method. A cubic spline is a piecewise cubic polynomial that ensures smooth interpolation

by maintaining continuity in the function as well as its first and second derivatives across intervals. Sun et al. [135] propose an online motion planning method that enables safe and smooth autonomous navigation of Mecanum-wheeled robots in industrial environments by combining SC-RRT path planning with cubic spline-based trajectory planning. Ma et al. [136] introduce a bidirectional search PRM global path planning algorithm that enhances planning efficiency and smoothness by improving connection rules and integrating cubic spline interpolation. Simulations and experiments validate its superiority over traditional PRM in terms of both path quality and computational efficiency.

3.9. Hybrid search approaches

In sampling-based algorithms, hybrid methods combine multiple path planning strategies to enhance the algorithm's performance and adaptability in complex environments.

Wei and Liu [137] combine RRT with a variable-length genetic algorithm, enhancing smooth path generation by minimizing path length and curvature, maintaining diversity, and preventing premature convergence. Mashayekhi et al. [138] propose Hybrid RRT, Hybrid RRT utilizes a dual-tree search, enabling it to find solutions more quickly than unidirectional searches. Afterward, it merges the start tree and goal tree from the dual-tree search into a single tree to perform informed sampling, optimizing the current solution. Additionally, Al-Ansarry et al. [139] introduce an enhanced approach known as Hybrid RRT-A*, designed to address the limitations of the original RRT, particularly its slow convergence and high cost. By integrating the heuristic function of the A* algorithm with RRT, the method reduces tree expansion and directs the search towards the goal more efficiently, using fewer nodes and less time. Respectively, Kiani et al. [140] integrate RRT with Grey Wolf Optimization (GWO), demonstrating the ability to efficiently find near-optimal paths in three-dimensional path planning. Pohan et al. [141] propose a hybridization of RRT* and Ant Colony System (RRT-ACS) to generate optimal parking paths efficiently, outperforming other algorithms in common scenarios. Recently, Cao et al. [142] integrate the advantages of RRT-Connect [86] for global path planning, artificial potential field (APF) [143] for local path planning, and cubic B-spline [144] for curve smoothing to optimize the path of unmanned aerial vehicles (UAVs).

4. Application and limitation

4.1. Application 1: Dynamic environments

Dynamic path planning involves determining a path from a starting position to a goal position in an environment that evolves over time. This may include changes caused by moving obstacles, such as individual pedestrians or crowds, alterations in the environment's structure, or other dynamic elements requiring real-time path updates [145]. Various sampling-based algorithms have been developed to efficiently replan paths in such scenarios. One common approach is to use bidirectional search [146,147], which facilitates quicker updates. Another strategy focuses on continuously refining and adjusting the search process during execution [148–151]. Furthermore, biased sampling has proven to be an effective method for improving planning efficiency [152–155]. Additionally, incorporating human awareness and crowd dynamics into the sampling-based planning process further enhances the planner's performance [156–159].

4.2. Application 2: Unknown environments and uncertainty localization

Robot motion planning in unknown environments is challenging, as it requires navigating without prior knowledge of the surroundings. In such cases, robots rely on real-time sensor data (e.g., LiDAR, cameras, sonar) to detect obstacles, track changes, and build an on-the-fly map. This process involves continuous exploration, obstacle avoidance, and dynamic path adjustments. Sampling-based algorithms play a crucial role in exploration tasks, enabling efficient and adaptive navigation in uncertain environments.

In unknown environments, one effective method involves using sampling-based algorithms with a biased direction [160,161] to optimize node generation and reduce computational load, allowing for more efficient real-time path planning. Other strategies include rolling planning combined with node screening [162] and model predictive control to further enhance path optimization [163].

To address localization uncertainties, sampling-based algorithms integrated with the Extended Kalman Filter (EKF) [164] and Simultaneous Localization and Mapping (SLAM) [165] have been employed to improve navigation accuracy. Additionally, sampling-based methods that focus on generating robust trajectories [166] and risk-bounded trajectories [167] provide enhanced path-planning capabilities in uncertain environments. The Min-Max RRT* [168] further refines this process by minimizing the maximum state estimate uncertainty along a path, offering an alternative to traditional additive cost representations of uncertainty.

4.3. Application 3: Nonholonomic constraints

Nonholonomic constraints limit a robot's movement in certain directions due to its physical structure, such as wheeled robots that cannot move sideways. These constraints complicate motion planning, requiring algorithms that generate paths respecting the robot's kinematic properties. To handle these challenges, sampling-based planners are adapted using techniques like steering functions [169–171], and goal bias [172,173]. Another approach combines RRT* with the generalized velocity obstacles (GVO) model to reduce trajectory uncertainty [174]. Similarly, the use of a distance function extends RRT* to handle nonholonomic constraints effectively [175]. S-BRRT* [176] further optimizes planning by combining the strengths of bidirectional RRT* with pruning and smoothing strategies. It incorporates bidirectional trees to enhance search efficiency and employs Bezier curves for path smoothing, significantly improving path quality and exploration efficiency in both sparse and dense environments.

4.4. Application 4: Kinematics

Sampling-based motion planning with kinodynamic constraints considers both the robot's kinematics and dynamics, planning in the state space (including position and velocity). This approach requires generating time-parameterized, dynamically feasible trajectories, making the problem more complex. Path continuity, in particular, depends on the robot's motion mode and kinematic constraints: wheeled robots with non-holonomic constraints require continuous paths to remain feasible, while legged or flying robots can handle certain discontinuities by jumping, repositioning, or adjusting altitude [177]. A wide variety of sampling-based algorithms for kinodynamic planning exist. For instance, using a fixed-final-state-free-final-time controller [178], neural network predictions of cost functions to

achieve the cost/metric between two given states considering the nonlinear constraints [179], deep reinforcement learning to learn an obstacle-avoiding policy that maps a robot's sensor observations to actions [180], precomputed motion primitives [55], lazy-steering techniques [181] and using a partial-final-state-free (PFF) optimal controller in kinodynamic RRT* to reduce the dimensionality of the sampling space [182].

4.5. Application 5: Manifold constraints

Motion planning under manifold constraints in robotics involves finding paths or trajectories that not only guide the robot from start to goal but also satisfy specific constraints defining a "manifold". Manifolds are higher-dimensional spaces representing the set of valid configurations that the robot can assume under its constraints, such as non-holonomic constraints, geometric constraints, kinematic limitations, or contact constraints.

Several sampling-based algorithms have been proposed to address these challenges, such as projection methods [183,184,184], continuation techniques [185–188], reparameterization-based [189,190] and offline methods that construct an approximation of the constraint manifold offline [191,192].

4.6. Application 6: Multi-robot

Multi-robot motion planning focuses on coordinating the movements of multiple robots to complete individual or collective tasks while avoiding collisions. This involves not only determining paths for each robot within a shared environment but also managing their interactions to optimize overall efficiency. Such planning is crucial in applications like warehouse automation, search, and rescue missions, autonomous bio-inspired robot coordination [193], robotic swarms, and robot arm systems [194].

Research in sampling-based algorithms has extensively tackled the challenges inherent in multi-robot motion planning, proposing various modifications and enhancements to traditional methods for improved performance. One approach [195] integrates both tightly-coupled and loosely-coupled strategies within its framework, allowing it to adapt to diverse multi-robot task scenarios. Another method [196] utilizes a more efficient sampling strategy to enhance planning efficiency. Additionally, an optimization-based map exploration strategy [197] equips multiple robots to actively explore their environment.

To efficiently plan within composite spaces, sampling-based algorithms offer effective strategies, where constructing individual roadmaps for each robot and implicitly searching the tensor product of these structures is a common approach [198–200]. For resolving inter-robot conflicts, sampling-based methods incorporate techniques like Prioritized Planning (SI-CPP) and Conflict-Based Search (SI-CCBS) [201] to better coordinate the paths of multiple robots.

4.7. Limitation 1: Narrow passage

In robot motion planning, the narrow passage problem refers to the challenge robots face when navigating through tight or confined spaces within complex environments. Research on sampling-based algorithms has extensively explored the narrow passage problem. Various studies have proposed modifications and enhancements to the traditional sampling-based algorithms to improve their performance in these constrained environments. One approach focuses on optimizing the sampling process to enhance the accuracy and efficiency of path planning in narrow spaces [202–204]. Another strategy involves incorporating an improved bridge test along with a novel search method based on

local guidance to facilitate smoother navigation [205]. Additionally, Szkandera et al. [206] propose a method inspired by the exit points for cavities in protein models, incorporating this concept into sampling-based algorithms. Respectively, Wang et al. [207] model the tree selection process as a multi-armed bandit problem and using a reinforcement learning algorithm with an enhanced ϵ_t -greedy strategy to address the narrow passage problem.

4.8. Limitation 2: Slow convergence and large memory required

Sampling-based algorithms face several limitations and challenges, particularly related to low convergence rates and high memory requirements. **Low Convergence Rates:** sampling-based algorithms can struggle with convergence speed, especially in high-dimensional spaces. The random sampling process, while effective for exploration, can lead to slow convergence towards optimal paths. This is a significant drawback when quick and efficient path planning is required. **High Memory Requirements:** sampling-based algorithms often require large amounts of memory to store the extensive tree structures generated during the planning process. This can become problematic in scenarios with limited computational resources or when dealing with very large state spaces.

To improve the convergence speed and memory efficiency of sampling-based algorithms, researchers have introduced various advanced methods. These include smart sampling techniques [54, 208], bidirectional search [86], lazy search strategies [100], and heuristic-based methods [33,50].

4.9. Comprehensive comparison of sampling-based algorithms (1998–2024)

Table 1 provides a comprehensive comparison of various typical sampling-based algorithms proposed between 1998 and 2024, showcasing the evolution of path planning techniques over the years. The table lists key attributes for each algorithm, including probabilistic completeness, bidirectional approach, heuristic search, batch sampling, asymptotic optimality, anytime capability, lazy collision check, greedy strategy, learning-based planning, static path planning, and dynamic path planning. Probabilistic Completeness indicates whether the algorithm will find a solution, if one exists, given sufficient time. Bidirectional Approach describes algorithms that search simultaneously from the start and goal, potentially reducing search time by meeting in the middle. Heuristic Search refers to algorithms that utilize heuristics to guide the search towards the goal more efficiently. Batch Sampling is a technique where multiple samples are generated in batches, improving performance by reducing the number of collision checks. Asymptotic Optimality implies that as the number of samples increases, the solution converges to an optimal path. Anytime Capability refers to algorithms that can provide a feasible solution quickly and improve it as more time or computational resources become available. Lazy Collision Check is a strategy where collision checks are deferred until absolutely necessary, often leading to faster initial solutions. Greedy Strategy indicates that the algorithm uses a direct approach to reach the goal, which may sometimes result in suboptimal paths. Learning-Based Planning refers to the integration of learning techniques to enhance the algorithm's efficiency or adaptability over time. Static Path Planning describes planning algorithms for environments with fixed, unchanging obstacles, while Dynamic Path Planning indicates algorithms capable of adapting to changes in the environment, suitable for applications with moving obstacles or evolving conditions. These attributes highlight each method's technical advancements and specific features, providing insight into how they address different path planning challenges.

5. Experiment

In this survey, we utilize the Planner-Arena benchmark database [221], the Planner Developer Tools (PDT) [222], and Open Robotics Automation Virtual Environment (OpenRAVE) [223] to benchmark proposed motion planner behaviors.

Ten popular algorithms, including various versions of RRT-Connect, RRT*, RRT#, Informed RRT*, LazyPRM*, BIT*, ABIT*, AIT*, EIRM*, and EIT* from the Open Motion Planning Library (OMPL) [224], were tested in both simulated random scenarios (Fig. 7) and manipulation tasks (Fig. 9, Fig. 10). The evaluations are implemented on a desktop with an Intel i5-12600k processor and 16 GB memory, running Ubuntu 20.04. These comparisons were carried out in simulated environments ranging from \mathbb{R}^4 to \mathbb{R}^{16} , and for Manipulation tasks ranging from \mathbb{R}^7 and \mathbb{R}^{14} . The primary objective for the planners was to minimize path length (cost). The RGG constant η was uniformly set to 1.001, and the rewire factor was set to 1.2 for all planners.

For RRT-based algorithms, a 5% goal bias was used, with maximum edge lengths of 0.5, 1.1, 1.25, 2.4 and 3.0 in \mathbb{R}^4 , \mathbb{R}^7 , \mathbb{R}^8 , \mathbb{R}^{14} , \mathbb{R}^{16} . All batch-sorted planners sampled 100 states per batch, and informed planners defined the informed set X_f using the current best costs.

5.1. Simulation experimental tasks

Fig. 8 shows the performance of all algorithms in the wall gap test benchmarks and ten random rectangle experiments conducted in \mathbb{R}^4 , \mathbb{R}^8 , and \mathbb{R}^{16} . EIT* and EIRM* perform efficiently in terms of both success rate and path cost across various scenarios. RRT*, RRT#, and Informed RRT* do not perform as well in terms of both success rate and path cost across various scenarios. RRT* and RRT# Informed RRT* tend to struggle with convergence speed, often requiring more time to find feasible solutions. As shown in Table 2, various path planning algorithms are compared across multiple benchmark scenarios. Among benchmarked planners, EIT* stands out as the state-of-the-art (SOTA) algorithm, achieves the lowest initial solution time of 0.0056 s in the WG - \mathbb{R}^4 scenario. This impressive performance highlights EIT*'s efficiency in quickly finding feasible paths, outperforming other algorithms such as RRT-Connect, RRT*, and Informed RRT*, which exhibit longer initial solution times.

In the RR - \mathbb{R}^4 scenario, BIT* and ABIT* achieve shorter initial times (0.1239 s and 0.0947 s, respectively), compared to the other methods, with BIT* also yielding a lower final solution cost ($c_{final}^{med} = 2.0457$). Similarly, in RR - \mathbb{R}^{16} , ABIT* outperforms other planners in terms of initial time (0.5259 s), although EIT* demonstrates a lower final cost. In higher-dimensional scenarios, EIT* continues to demonstrate its competitive edge, with notable efficiency observed in both WG - \mathbb{R}^8 and WG - \mathbb{R}^{16} . The ability of EIT* to maintain low initial solution times while achieving satisfactory final solution costs reinforces its status as a leading approach in path planning.

Overall, the table underscores the adaptability of sampling-based planners in high-dimensional spaces, achieving low initial solution times and final costs across various scenarios. ABIT* and BIT* excel in structured environments such as RR - \mathbb{R}^4 and RR - \mathbb{R}^{16} , effectively balancing initial time and solution cost. EIT* and EIRM* exhibit similarly low values for median initial solution time, median initial solution cost, and median solution cost over time. In contrast, RRT*, RRT#, and Informed RRT* demonstrate higher median initial solution times, initial solution costs, and solution costs over time.

Table 1
Key characteristics and algorithmic features of sampling-based planners.

Method	Year	Probabilistic completeness	Bidirectional approach	Heuristic search	Batch sampling	Asymptotic optimal	Anytime capability	Lazy collision check	Greedy strategy	Learning-based planning	Static path planning	Dynamic path planning
RRT [19]	1998	✓	-	-	-	-	-	-	-	-	✓	-
RRT-connect [86]	2000	✓	✓	✓	-	-	-	-	-	-	✓	-
Lazy PRM [98]	2000	✓	-	✓	-	-	-	✓	-	-	✓	-
hRRT [112]	2003	✓	-	✓	-	-	-	-	-	-	✓	-
ADD-RRT [209]	2005	✓	-	✓	-	-	-	-	-	-	✓	-
Anytime RRT [210]	2006	✓	-	✓	-	-	✓	-	-	-	✓	✓
T-RRT [211]	2008	✓	-	✓	-	-	-	-	-	-	✓	-
TSRRT [212]	2010	✓	✓	✓	-	-	-	✓	-	-	✓	-
RRT* [49]	2011	✓	-	✓	-	✓	✓	-	-	-	✓	-
RRT*-SMART [54]	2013	✓	-	✓	-	✓	✓	-	-	-	✓	-
RRT# [120]	2013	✓	-	✓	-	✓	✓	-	-	-	✓	✓
Informed RRT* [33]	2014	✓	-	✓	-	✓	✓	-	-	-	✓	-
FMT* [103]	2015	✓	-	✓	✓	✓	-	✓	-	-	✓	-
BIT* [57]	2015	✓	-	✓	✓	✓	✓	-	-	-	✓	-
Lazy-PRM* [100]	2015	✓	-	✓	-	✓	✓	✓	-	-	✓	✓
Lazy-RRG* [100]	2015	✓	-	✓	-	✓	✓	✓	-	-	✓	✓
RT-RRT* [123]	2015	✓	-	✓	-	✓	✓	-	-	-	✓	✓
BI ² RRT* [91]	2016	✓	✓	✓	-	✓	✓	-	✓	-	✓	-
RRT ^x [122]	2016	✓	-	✓	-	✓	✓	-	-	-	✓	✓
FCRRT [213]	2016	✓	-	✓	-	✓	✓	✓	-	-	✓	✓
HARRT* [92]	2016	✓	✓	✓	-	✓	✓	-	-	-	✓	✓
P-RRT* [104]	2016	✓	-	✓	-	✓	✓	-	-	-	✓	✓
ALCC [101]	2018	✓	-	✓	-	✓	✓	✓	-	-	✓	✓
NoD-RRT [179]	2018	✓	-	✓	-	✓	✓	-	-	✓	✓	-
PB-RRT* [90]	2018	✓	✓	✓	-	✓	✓	-	-	-	✓	-
PIB-RRT* [90]	2018	✓	✓	✓	-	✓	✓	-	-	-	✓	-
Multi-Tree	2018	✓	-	✓	-	✓	-	-	-	-	✓	✓
T-RRT* [214]												
Lazy Steering												
RRT* [181]	2019	✓	-	✓	-	✓	✓	✓	-	✓	✓	-
RL-RRT [180]	2019	✓	-	✓	-	-	-	-	-	✓	✓	-
Quick-RRT* [105]	2019	✓	-	✓	-	✓	✓	-	-	-	✓	-
PQ-RRT* [106]	2020	✓	-	✓	-	✓	✓	-	-	-	✓	-
AIT* [34]	2020	✓	✓	✓	✓	✓	✓	✓	-	-	✓	-
ABIT* [115]	2020	✓	-	✓	✓	✓	✓	-	-	-	✓	-
dRRT* [200]	2020	✓	-	✓	-	✓	✓	-	-	-	✓	✓
Neural RRT* [52]	2020	✓	-	✓	-	✓	✓	-	-	✓	✓	-
GBIT* [116]	2021	✓	-	✓	✓	✓	✓	-	✓	-	✓	-
MP-RRT# [215]	2021	✓	-	✓	-	-	✓	-	-	-	✓	✓
AM-RRT* [109]	2021	✓	-	✓	-	✓	✓	-	-	-	✓	✓
MOD-RRT* [150]	2021	✓	-	✓	-	✓	✓	-	-	-	✓	✓
F-RRT* [216]	2021	✓	-	✓	-	-	✓	-	-	-	✓	✓
EIT* [117]	2022	✓	✓	✓	✓	✓	✓	✓	-	-	✓	-
EIRM* [118]	2022	✓	✓	✓	✓	✓	✓	✓	-	-	✓	-
RJ-RRT [204]	2022	✓	-	✓	-	-	-	-	✓	-	✓	-
BiAIT* [119]	2023	✓	✓	✓	✓	✓	✓	✓	-	-	✓	-
EP-RRT* [51]	2023	✓	✓	✓	-	✓	✓	-	✓	-	✓	-
FA-RRT*N [217]	2023	✓	-	✓	-	✓	✓	-	-	-	✓	-
GMM-RRT* [218]	2023	✓	-	✓	-	✓	✓	-	-	-	✓	-
PROTAMP-RRT [219]	2023	✓	-	✓	-	-	-	-	-	-	✓	-
Neural Informed RRT* [60]	2024	✓	-	✓	-	✓	✓	-	-	✓	✓	✓
iDb-RRT [220]	2024	✓	✓	✓	-	-	-	-	-	-	✓	-
SI-RRT* [201]	2024	✓	-	✓	-	✓	✓	-	-	-	✓	✓

5.2. Simulated manipulation tasks

Fig. 11 shows the performance of all algorithms in the Single-Arm Manipulator Problem and in Dual-Arm Manipulator Problem. RRT-Connect and ABIT* are faster than other algorithms but may result in higher costs, making them suitable for applications

requiring quick solutions. On the other hand, Lazy PRM*, BIT* and AIT* achieve lower costs but require more computation time, making them ideal for scenarios where path optimality is crucial. RRT*, RRT#, and Informed RRT* do not perform as well in terms of both success rate and path cost. As illustrated in Table 3, we compare various algorithms for manipulation tasks, focusing on

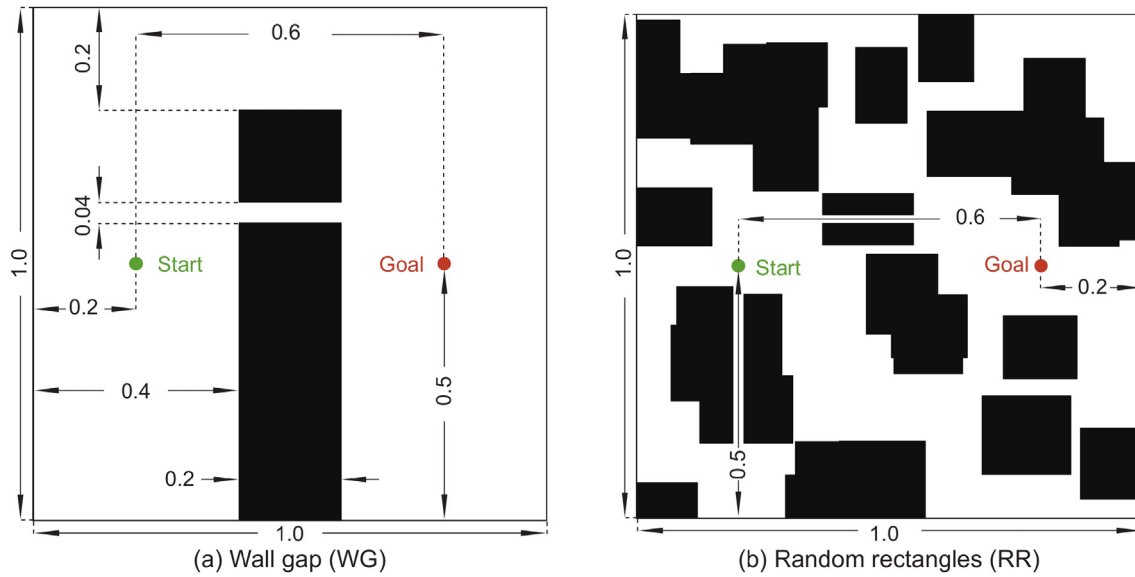


Fig. 7. (a) A two-dimensional illustration of the wall gap experiment. The start and goal states are represented by a green dot and red dot, respectively. Each state space dimension was bounded to the interval $[0, 1]$. (b) A two-dimensional illustration of the Random Rectangles experiment. The start and goal states are represented by a green dot and red dot, respectively. Each state space dimension was bounded to the interval $[0, 1]$.

Table 2

Benchmarks evaluation comparison for simulated random scenarios. t_{init}^{med} : median initial solution time (in seconds); c_{init}^{med} : median initial solution cost; c_{final}^{med} : median solution cost over time (in seconds).

Scenarios	RRT-Connect			RRT*			RRT#			Informed RRT*			LazyPRM*		
	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}
WG - \mathbb{R}^4	0.0189	2.3523	2.3523	0.0450	1.9527	1.8084	0.0469	1.9886	1.8418	0.0422	1.9418	1.8294	0.0095	1.9608	0.9560
WG - \mathbb{R}^8	0.0260	3.0528	3.0528	0.0891	2.9789	2.6418	0.0934	3.2095	2.7300	0.0907	3.1726	2.6795	0.0125	2.6882	1.6260
WG - \mathbb{R}^{16}	0.0542	4.2013	4.2013	1.1494	5.5114	5.4170	1.3328	6.3678	6.0390	1.4954	5.3444	5.2526	0.0202	3.8372	2.8594
RR - \mathbb{R}^4	0.0997	2.5585	2.5585	0.2462	3.266	3.143	0.2480	3.077	3.356	0.2568	3.281	3.281	0.0646	2.0323	1.7571
RR - \mathbb{R}^8	0.5539	5.4617	5.4617	2.0761	7.2886	7.2435	2.1085	7.0050	7.0042	2.2842	7.6778	7.2683	0.3788	5.5239	3.2652
RR - \mathbb{R}^{16}	0.1926	5.3251	5.3251	0.5433	7.8501	7.7856	0.5349	7.9862	7.0149	0.4763	7.9458	7.9198	0.2759	5.2091	3.8871
Scenarios	BIT*			ABIT*			AIT*			EIRM*			EIT*		
	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}
WG - \mathbb{R}^4	0.0263	1.6643	0.9223	0.0084	1.7950	0.9151	0.0132	1.7087	0.7539	0.0058	1.7849	0.7812	0.0056	1.8059	0.7235
WG - \mathbb{R}^8	0.0355	2.4434	1.8654	0.0111	2.5784	1.7255	0.0211	2.5161	1.7029	0.0068	2.5528	1.5862	0.0058	2.5802	1.3561
WG - \mathbb{R}^{16}	0.0533	4.0784	2.8080	0.0174	4.2886	2.8300	0.0369	4.2280	3.1773	0.0104	4.6266	3.0372	0.0102	4.2056	2.9879
RR - \mathbb{R}^4	0.1239	2.2457	2.0457	0.0947	2.4809	2.3671	0.0749	1.9368	1.7195	0.0412	1.9173	1.6124	0.0406	1.9417	1.5854
RR - \mathbb{R}^8	1.6933	5.8140	5.0191	1.2512	6.6619	5.6050	0.9761	5.2000	4.0319	0.1215	4.7398	2.6180	0.1290	4.8876	2.5272
RR - \mathbb{R}^{16}	2.0198	6.1065	5.5245	0.5259	8.2124	5.8792	0.4301	6.5640	4.8204	0.1477	5.7654	3.7562	0.1561	5.7635	3.7288

Table 3

Benchmarks evaluation comparison for Manipulation tasks. SA: single-arm; DA: dual-arm; t_{init}^{med} : median initial solution time (in seconds); c_{init}^{med} : median initial solution cost; c_{final}^{med} : median solution cost over time (in seconds).

Task	RRT-Connect			RRT*			RRT#			Informed RRT*			LazyPRM*		
	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}
SA - \mathbb{R}^7	0.2260	18.0909	18.0909	0.810	20.9410	20.9410	0.8561	20.7685	20.7685	0.818	24.4390	24.4390	0.6554	24.0031	13.2513
DA - \mathbb{R}^{14}	0.2077	16.7844	16.7844	1.440	16.8691	16.8691	1.502	24.7865	24.7865	1.514	24.7505	24.7505	1.2077	23.6651	11.9351
Task	BIT*			ABIT*			AIT*			EIRM*			EIT*		
	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}	t_{init}^{med}	c_{init}^{med}	c_{final}^{med}
SA - \mathbb{R}^7	0.5469	20.2001	11.9929	0.3790	23.9119	12.3742	0.6006	22.2275	13.3074	0.9284	26.3252	14.8356	0.8493	24.6474	16.3089
DA - \mathbb{R}^{14}	0.4973	19.3292	10.2405	0.3562	23.0785	10.6342	0.5466	22.8266	11.7872	0.6015	23.8139	12.5322	0.6177	24.3534	12.4140

both single-arm (SA) and dual-arm (DA) configurations in high-dimensional spaces. The performance metrics include median initial solution time (t_{init}^{med}), median initial solution cost (c_{init}^{med}), and median solution cost over time (c_{final}^{med}).

The single-arm experiment primarily aims to test the performance of motion planning algorithms in obstacle avoidance

scenarios, where a single robotic arm navigates through a complex environment to assess the algorithm's efficiency and accuracy in avoiding obstacles. In contrast, the dual-arm experiment evaluates the algorithms in high-dimensional spaces, as the coordinated movement of two arms introduces increased complexity and dimensionality, providing a thorough assessment of the algorithm's adaptability and effectiveness in high-dimensional,

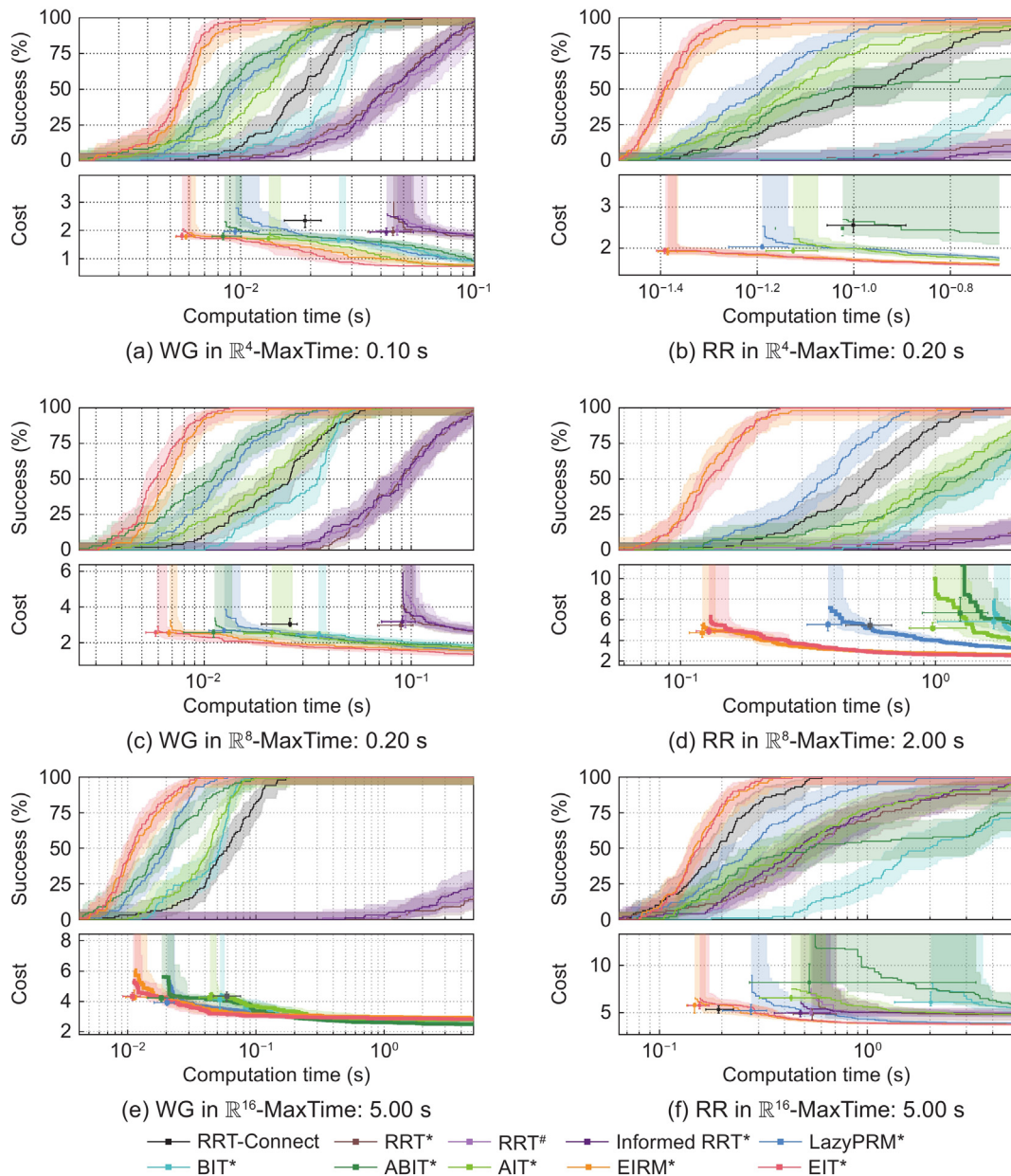


Fig. 8. Detailed experimental results from various motion planners in the Open Motion Planning Library are presented above. Figures (a), (c) and (e) depict test benchmark wall gap outcomes in \mathbb{R}^4 , \mathbb{R}^8 and \mathbb{R}^{16} , respectively. Panel (b) showcases ten random rectangle experiments in \mathbb{R}^4 , while panels (d) and (f) demonstrate in \mathbb{R}^8 and \mathbb{R}^{16} . In the cost plots, subfigures illustrate both the solution cost and time, while lines indicate the progression of cost for a planner that is almost certainly asymptotically optimal (unsuccessful runs are assigned an infinite cost). The error bars depict nonparametric 99% confidence intervals for both solution cost and time.

multi-constraint environments. In the single-arm experiment, the maximum planning time is set to 5 s, while in the dual-arm experiment, it is extended to 10 s. These time limits reflect the algorithm's performance in real-world operations, ensuring that solutions are found within a practical timeframe. The longer time limit for the dual-arm experiment is due to the increased complexity of coordinating two arms, which requires additional computational resources compared to the single-arm task.

In the single-arm scenario (SA – \mathbb{R}^7), RRT-Connect demonstrates a relatively low initial solution time of 0.2260 s, but its final solution cost remains high at 18.0909. Conversely, EIT* offers a more competitive median initial solution time of 0.8493 s while achieving a notably lower median solution cost of 16.3089, indicating its efficiency in path optimization.

The dual-arm configuration (DA – \mathbb{R}^{14}) presents similar trends. EIT* achieves an initial solution time of 0.6177 s, which is comparable to other algorithms, while also has the minimal final solution cost at 12.4140. This positions EIT* as a strong contender in both initial speed and long-term cost efficiency.

Among the other algorithms, RRT*, RRT#, Informed RRT* exhibit higher initial solution times, indicating less efficiency in quickly generating paths. LazyPRM* shows slightly better final solution costs but suffers from higher initial costs and times compared to EIT*.

In the random rectangles and wall gap scenarios, RRT-Connect performs relatively average. Particularly in higher dimensions (e.g., R16), while it can find an initial path relatively quickly, the path cost tends to be higher, and it struggles to find optimized solutions. Additionally, as the dimension increases, the

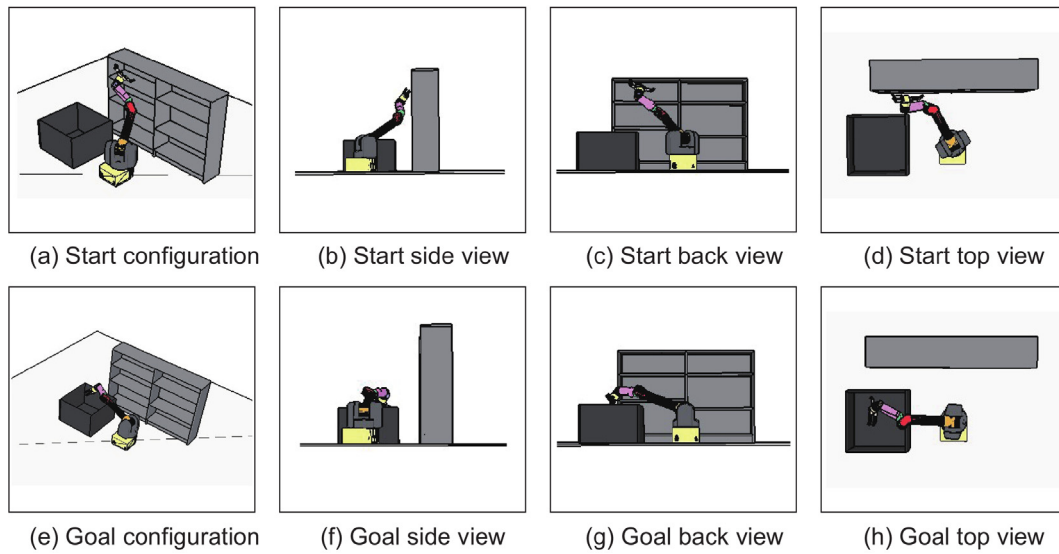


Fig. 9. Illustrations of the single-arm manipulator problem. The top row shows the start configuration of the arm in position to pick up an object on the top shelf (a–d). The bottom row shows the goal configuration of the arms in position to place an object in the box (e–h).

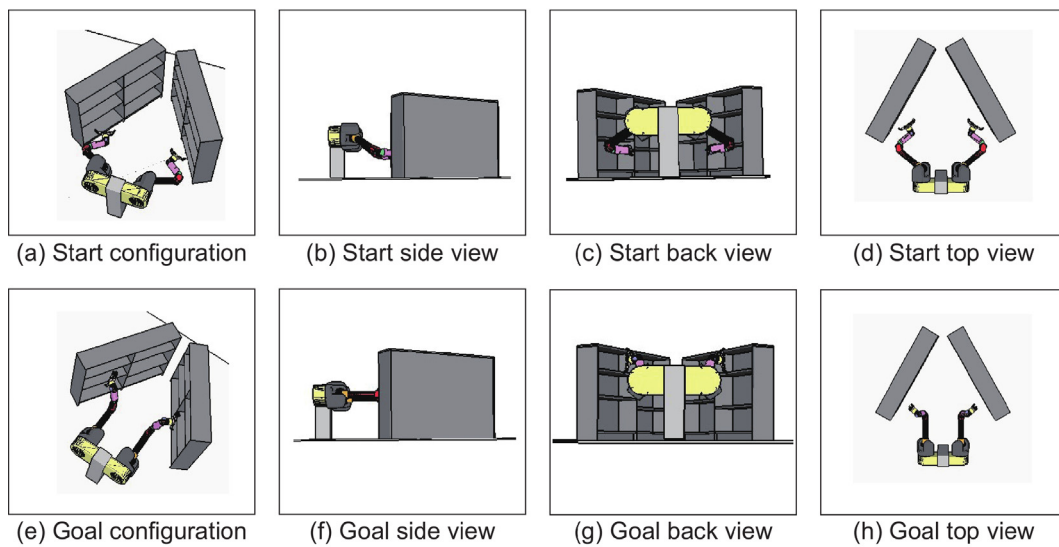


Fig. 10. Illustrations of the dual-arm manipulator problem. The top row shows the start configuration of the arms in position to pick up an object on the bottom shelf (a–d). The bottom row shows the goal configuration of the arms in position to place an object on the top shelf (e–h).

success rate of RRT-Connect decreases, indicating that in high-dimensional complex environments, it is prone to generating redundant paths and is more affected by constraints. However, in the second manipulation experiment (single-arm and dual-arm robots), RRT-Connect performs exceptionally well. In these experiments, RRT-Connect leverages its rapid expansion capabilities to generate paths efficiently, providing effective solutions for manipulation tasks. In single-arm and dual-arm manipulation tasks, where the task space dimensionality is relatively lower and task goals are well-defined, RRT-Connect's fast connection strategy is well-suited, resulting in favorable performance in terms of both path generation time and path cost, achieving high efficiency in these operational tasks.

Overall, these experiments demonstrate the performance of ten popular algorithms across various scenarios, solidifying their status as leading choices in high-dimensional manipulation path

planning. The performance in both single-arm and dual-arm tasks highlights their feasibility and efficiency.

6. Conclusion

Sampling-based motion planning algorithms are highly effective for exploring continuously-valued spaces, which are commonly encountered in robotics. These algorithms rely on generating samples to approximate and explore the search space. Many sampling-based

algorithms are probabilistically complete. But these algorithms do not provide any guarantee on the quality of its solution. In recent years, researchers have focused on addressing this issue. In this article, we have reviewed the progress made. We divide the traditional sampling-based algorithms framework into seven parts, **Sampling**(·), **Nearest**(·), **Steer**(·), **CollisionFree**(·), **Extend**(·)

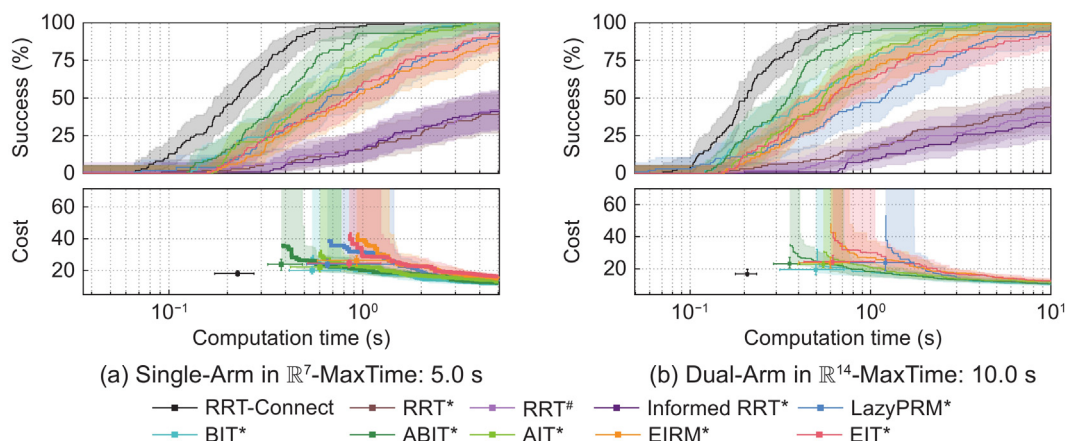


Fig. 11. Detailed experimental results of the planners' performances on the single-arm and dual-arm manipulator are presented above. Figure (a) shows the result of the single-arm manipulator, Figure (b) shows the result of the dual-arm manipulator. In the cost plots, subfigures illustrate both the solution cost and time, while lines indicate the progression of cost for a planner that is almost certainly asymptotically optimal (unsuccessful runs are assigned an infinite cost). The error bars depict nonparametric 99% confidence intervals for both solution cost and time.

including *Heuristic*(\cdot), *Rewire*(\cdot). We conduct an extensive literature review on these seven aspects and provide a summary and analysis of the current research status. Furthermore, this paper tests the performance of ten popular planners in both simulated random scenarios and Manipulation tasks with different dimensions. The results suggest that planning algorithms are capable of solving a wide range of problems, including those with narrow passages and constraints. However, no single planner consistently outperforms others across all problem types. Apart from the comparative evaluations, this review provides a comprehensive overview about applications and limitations, such as dynamic environments, Narrow passage and kinodynamic constraints. This helps researchers and practitioners to make better understanding sampling-based planners in different fields.

In summary, by systematically reviewing the state of the art and addressing the remaining challenges, this survey serves as a valuable resource for researchers and practitioners in the field of robotics. It offers a clear understanding of the current landscape of motion planning techniques and provides insights into the future directions of this rapidly evolving area of study.

CRedit authorship contribution statement

Liding Zhang: Writing – original draft. **Kuanqi Cai:** Writing – original draft. **Zewei Sun:** Writing – original draft. **Zhenshan Bing:** Project administration. **Chaoqun Wang:** Methodology. **Luis Figueredo:** Conceptualization. **Sami Haddadin:** Resources, Funding acquisition. **Alois Knoll:** Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK (Phase 1: Infrastructure and the research and development program under grant no. DIK0249).

References

- [1] M.R. Pedersen, L. Nalpantidis, R.S. Andersen, C. Schou, S. Bøgh, V. Krüger, O. Madsen, Robot skills for manufacturing: From concept to industrial deployment, *Robot. Comput.-Integr. Manuf.* 37 (2016) 282–291.
- [2] E. Matheson, R. Minto, E.G. Zampieri, M. Faccio, G. Rosati, Human-robot collaboration in manufacturing applications: A review, *Robotics* 8 (4) (2019) 100.
- [3] J. Arents, M. Greitans, Smart industrial robot control trends, challenges and opportunities within manufacturing, *Appl. Sci.* 12 (2) (2022) 937.
- [4] W. Echelmeyer, A. Kirchheim, E. Wellbrock, Robotics-logistics: Challenges for automation of logistic processes, in: 2008 IEEE International Conference on Automation and Logistics, IEEE, 2008, pp. 2099–2103.
- [5] R. Lin, H. Huang, M. Li, An automated guided logistics robot for pallet transportation, *Assem. Autom.* 41 (1) (2021) 45–54.
- [6] R. Bernardo, J.M. Sousa, P.J. Gonçalves, Survey on robotic systems for internal logistics, *J. Manuf. Syst.* 65 (2022) 339–350.
- [7] A.R. Lanfranco, A.E. Castellanos, J.P. Desai, W.C. Meyers, Robotic surgery: a current perspective, *Ann. Surg.* 239 (1) (2004) 14–21.
- [8] M. Diana, J. Marescaux, Robotic surgery, *J. Br. Surg.* 102 (2) (2015) e15–e28.
- [9] A. Sozzi, M. Bonfè, S. Farsoni, G.D. Rossi, R. Muradore, Dynamic motion planning for autonomous assistive surgical robots, *Electronics* 8 (9) (2019) 957.
- [10] B.H. Wilcox, Robotic vehicles for planetary exploration, *Appl. Intell.* 2 (1992) 181–193.
- [11] J. Oberländer, S. Klemm, G. Heppner, A. Roennau, R. Dillmann, A multi-resolution 3-D environment model for autonomous planetary exploration, in: 2014 IEEE International Conference on Automation Science and Engineering, CASE, IEEE, 2014, pp. 229–235.
- [12] K. Albee, A.C. Hernandez, O. Jia-Richards, A.T. Espinoza, Real-time motion planning in unknown environments for legged robotic planetary exploration, in: 2020 IEEE Aerospace Conference, IEEE, 2020, pp. 1–9.
- [13] K. Chen, Z. Bing, Y. Wu, F. Wu, L. Zhang, S. Haddadin, A. Knoll, Real-time contact state estimation in shape control of deformable linear objects under small environmental constraints, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 13833–13839.
- [14] E. Dijkstra, A note on two problems in connexion with graphs, *Numer. Math.* 1 (1959) 269–271.
- [15] F. Duchoñ, A. Babinec, M. Kajan, P. Beño, M. Florek, T. Fico, L. Jurišica, Path planning with modified a star algorithm for a mobile robot, *Procedia Eng.* 96 (2014) 59–69.
- [16] A. Stentz, Optimal and efficient path planning for partially-known environments, in: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, IEEE, 1994, pp. 3310–3317.
- [17] J. Tu, S.X. Yang, Genetic algorithm based path planning for a mobile robot, in: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), vol. 1, IEEE, 2003, pp. 1221–1226.
- [18] Q. Zhu, Y. Yan, Z. Xing, Robot path planning based on artificial potential field approach with simulated annealing, in: Sixth International Conference on Intelligent Systems Design and Applications, vol. 2, IEEE, 2006, pp. 622–627.
- [19] S. LaValle, Rapidly-exploring random trees: A new tool for path planning, *Res. Rep.* 9811 (1998).

- [20] L.E. Kavraki, P. Švestka, J.-C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Trans. Robot. Autom.* 12 (4) (1996) 566–580.
- [21] H.-P. Huang, S.-Y. Chung, Dynamic visibility graph for path planning, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, IEEE, 2004, pp. 2813–2818.
- [22] M. Foskey, M. Garber, M.C. Lin, D. Manocha, A Voronoi-based hybrid motion planner, in: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180), vol. 1, IEEE, 2001, pp. 55–60.
- [23] W. Cheah, H.H. Khalili, S. Watson, P. Green, B. Lennox, Grid-based motion planning using advanced motions for hexapod robots, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 3573–3578.
- [24] S.X. Yang, C. Luo, A neural network approach to complete coverage path planning, *IEEE Trans. Syst. Man Cybern. B* 34 (1) (2004) 718–724.
- [25] Y. Zhang, D.-w. Gong, J.-h. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neurocomputing* 103 (2013) 172–185.
- [26] P. Vadakkepat, K.C. Tan, W. Ming-Liang, Evolutionary artificial potential fields and their application in real time robot path planning, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512), vol. 1, IEEE, 2000, pp. 256–263.
- [27] O. Brock, L.E. Kavraki, Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces, in: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), vol. 2, IEEE, 2001, pp. 1469–1474.
- [28] S. Pandya, S. Hutchinson, A case-based approach to robot motion planning, in: [Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 1992, pp. 492–497.
- [29] T. Lozano-Perez, *Spatial Planning: A Configuration Space Approach*, Springer, 1990.
- [30] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, *IEEE Robot. Autom. Mag.* 4 (1) (1997) 23–33.
- [31] A. Zelinsky, R.A. Jarvis, J. Byrne, S. Yuta, et al., Planning paths of complete coverage of an unstructured environment by a mobile robot, in: Proceedings of International Conference on Advanced Robotics, vol. 13, Citeseer, 1993, pp. 533–538.
- [32] M.P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, P. Melin, Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Appl. Soft Comput.* 9 (3) (2009) 1102–1110.
- [33] J.D. Gammell, T.D. Barfoot, S.S. Srinivasa, Batch informed trees (BIT*): Informed asymptotically optimal anytime search, *Int. J. Robot. Res.* 39 (5) (2020) 543–567, <http://dx.doi.org/10.1177/0278364919890396>.
- [34] M.P. Strub, J.D. Gammell, Adaptively Informed Trees (AIT*): Fast asymptotically optimal path planning through adaptive heuristics, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 3191–3198.
- [35] B. Lindqvist, A. Patel, K. Löfgren, G. Nikolakopoulos, A tree-based next-best-trajectory method for 3-D UAV exploration, *IEEE Trans. Robot.* 40 (2024) 3496–3513.
- [36] R. Border, J.D. Gammell, The Surface Edge Explorer (SEE): A measurement-direct approach to next best view planning, *Int. J. Robot. Res. (IJRR)* (2024).
- [37] D. Hsu, R. Kindel, J.-C. Latombe, S. Rock, Randomized kinodynamic motion planning with moving obstacles, *Int. J. Robot. Res.* 21 (3) (2002) 233–255.
- [38] J.-C. Latombe, Motion planning: A journey of robots, molecules, digital actors, and other artifacts, *Int. J. Robot. Res.* 18 (11) (1999) 1119–1128.
- [39] S.R. Lindemann, S.M. LaValle, Current issues in sampling-based motion planning, in: *Robotics Research. The Eleventh International Symposium: with 303 Figures*, Springer, 2005, pp. 36–54.
- [40] K.I. Tsianos, I.A. Sucan, L.E. Kavraki, Sampling-based robot motion planning: Towards realistic applications, *Comp. Sci. Rev.* 1 (1) (2007) 2–11.
- [41] M. Elbanhawi, M. Simic, Sampling-based robot motion planning: A review, *Ieee Access* 2 (2014) 56–77.
- [42] I. Noreen, A. Khan, Z. Habib, Optimal path planning using RRT* based approaches: a survey and future directions, *Int. J. Adv. Comput. Sci. Appl.* 7 (11) (2016).
- [43] Z. Kingston, M. Moll, L.E. Kavraki, Sampling-based methods for motion planning with constraints, *Annu. Rev. Control., Robot., Auton. Syst.* 1 (1) (2018) 159–185.
- [44] K. Cai, C. Wang, J. Cheng, C.W. De Silva, M.Q.-H. Meng, Mobile robot path planning in dynamic environments: A survey, 2020, arXiv preprint arXiv:2006.14195.
- [45] J.R. Sanchez-Ibanez, C.J. Pérez-del Pulgar, A. García-Cerezo, Path planning for autonomous mobile robots: A review, *Sensors* 21 (23) (2021) 7898.
- [46] J.D. Gammell, M.P. Strub, Asymptotically optimal sampling-based motion planning methods, *Annu. Rev. Control., Robot., Auton. Syst.* 4 (1) (2021) 295–318.
- [47] A. Orthey, C. Chamzas, L.E. Kavraki, Sampling-based motion planning: A comparative review, *Annu. Rev. Control., Robot., Auton. Syst.* 7 (2023).
- [48] T. Xu, Recent advances in rapidly-exploring random tree: A review, *Heliyon* (2024).
- [49] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, *Int. J. Robot. Res.* 30 (7) (2011) 846–894.
- [50] J.D. Gammell, T.D. Barfoot, S.S. Srinivasa, Informed sampling for asymptotically optimal path planning, *IEEE Trans. Robot.* 34 (4) (2018) 966–984.
- [51] J. Ding, Y. Zhou, X. Huang, K. Song, S. Lu, L. Wang, An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling, *J. Comput. Sci.* 67 (2023) 101937.
- [52] J. Wang, W. Chi, C. Li, C. Wang, M.Q.-H. Meng, Neural RRT*: Learning-based optimal path planning, *IEEE Trans. Autom. Sci. Eng.* 17 (4) (2020) 1748–1758.
- [53] J. Wang, T. Li, B. Li, M.Q.-H. Meng, GMR-RRT*: Sampling-based path planning using Gaussian mixture regression, *IEEE Trans. Intell. Veh.* 7 (3) (2022) 789–800.
- [54] J. Nasir, F. Islam, U. Malik, et al., RRT*-SMART: A rapid convergence implementation of RRT*, *Int. J. Adv. Robot. Syst.* 10 (7) (2013).
- [55] B. Sakkak, L. Bascetta, G. Ferretti, M. Prandini, Sampling-based optimal kinodynamic planning with motion primitives, *Auton. Robots* 43 (7) (2019) 1715–1732.
- [56] L. Zhang, Z. Bing, K. Chen, L. Chen, K. Cai, Y. Zhang, F. Wu, P. Krumbholz, Z. Yuan, S. Haddadin, A. Knoll, Flexible Informed Trees (FIT*): Adaptive batch-size approach in informed sampling-based path planning, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2024, pp. 3146–3152.
- [57] J.D. Gammell, S.S. Srinivasa, T.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, 2015, pp. 3067–3074.
- [58] J.D. Gammell, T.D. Barfoot, S.S. Srinivasa, Batch Informed Trees (BIT*): Informed asymptotically optimal anytime search, *Int. J. Robot. Res.* 39 (5) (2020) 543–567.
- [59] F. Yu, Y. Chen, Cyl-IRRT*: Homotopy optimal 3D path planning for AUVs by biasing the sampling into a cylindrical informed subset, *IEEE Trans. Ind. Electron.* 70 (4) (2022) 3985–3994.
- [60] Z. Huang, H. Chen, J. Pohovey, K. Driggs-Campbell, Neural informed RRT*: Learning-based path planning with point cloud state representations under admissible ellipsoidal constraints, 2024, arXiv:2309.14595, [Online] Available: <https://arxiv.org/abs/2309.14595>.
- [61] B. Ihter, J. Harrison, M. Pavone, Learning sampling distributions for robot motion planning, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 7087–7094.
- [62] D. Molina, K. Kumar, S. Srivastava, Learn and link: Learning critical regions for efficient planning, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 10605–10611.
- [63] C. Xiong, H. Zhou, D. Lu, Z. Zeng, L. Lian, C. Yu, Rapidly-exploring adaptive sampling tree*: a sample-based path-planning algorithm for unmanned marine vehicles information gathering in variable ocean environments, *Sensors* 20 (9) (2020) 2515.
- [64] K. Cai, C. Wang, S. Song, H. Chen, M.Q.-H. Meng, Risk-aware path planning under uncertainty in dynamic environments, *J. Intell. Robot. Syst.* 101 (2021) 1–15.
- [65] K. Cai, W. Chen, C. Wang, H. Zhang, M.Q.-H. Meng, Curiosity-based robot navigation under uncertainty in crowded environments, *IEEE Robot. Autom. Lett.* 8 (2) (2022) 800–807.
- [66] H.-T. Tak, C.-G. Park, S.-C. Lee, Improvement of RRT*-smart algorithm for optimal path planning and application of the algorithm in 2 & 3-dimension environment, *J. Korean Soc. Aviat. Aeronaut.* 27 (2) (2019) 1–8.
- [67] H. Suwoyo, A. Adriansyah, J. Andika, A. Ubaidillah, M.F. Zakaria, An integrated RRT* SMART-A* algorithm for solving the global path planning problem in a static environment, *IJUM Eng. J.* 24 (1) (2023) 269–284.
- [68] A. Boeuf, J. Cortés, R. Alami, T. Siméon, Enhancing sampling-based kinodynamic motion planning for quadrotors, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2015, pp. 2447–2452.
- [69] C.K. Verginis, D.V. Dimarogonas, L.E. Kavraki, Kdf: Kinodynamic motion planning via geometric sampling-based algorithms and funnel control, *IEEE Trans. Robot.* 39 (2) (2022) 978–997.
- [70] S.M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.
- [71] S.M. LaValle, J.J. Kuffner Jr., Randomized kinodynamic planning, *Int. J. Robot. Res.* 20 (5) (2001) 378–400.
- [72] A. Atramentov, S.M. LaValle, Efficient nearest neighbor searching for motion planning, in: Proceedings 2002 IEEE International Conference on Robotics and Automation, vol. 1, IEEE, 2002, pp. 632–637.
- [73] J. Pan, S. Chitta, D. Manocha, Faster sample-based motion planning using instance-based learning, in: *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*, Springer, 2013, pp. 381–396.

- [74] L. Zhang, Z. Bing, Y. Zhang, K. Cai, L. Chen, F. Wu, S. Haddadin, A. Knoll, Elliptical K-nearest neighbors: Path optimization via Coulomb's law and invalid vertices in C-space obstacles, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 12032–12039.
- [75] M. Kleinbort, O. Salzman, D. Halperin, Collision detection or nearest-neighbor search? On the computational bottleneck in sampling-based motion planning, in: Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, Springer, 2020, pp. 624–639.
- [76] M. Kleinbort, E. Granados, K. Solovey, R. Bonalli, K.E. Bekris, D. Halperin, Refined analysis of asymptotically-optimal kinodynamic planning in the state-cost space, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 6344–6350.
- [77] L. Janson, E. Schmerling, A. Clark, M. Pavone, Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions, *Int. J. Robot. Res.* 34 (7) (2015) 883–921.
- [78] L. Janson, B. Ihter, M. Pavone, Deterministic sampling-based motion planning: Optimality, complexity, and performance, *Int. J. Robot. Res.* 37 (1) (2018) 46–61.
- [79] M. Tsao, K. Solovey, M. Pavone, Sample complexity of probabilistic roadmaps via ϵ -nets, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 2196–2202.
- [80] C. Wang, M.Q.-H. Meng, Variant step size RRT: An efficient path planner for UAV in complex environments, in: 2016 IEEE International Conference on Real-Time Computing and Robotics, RCAR, IEEE, 2016, pp. 555–560.
- [81] C. Wang, L. Meng, S. She, I.M. Mitchell, T. Li, F. Tung, W. Wan, M.Q.-H. Meng, C.W. de Silva, Autonomous mobile robot navigation in uneven and unstructured indoor environments, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 109–116.
- [82] Y. Zhang, R. Wang, C. Song, J. Xu, An improved dynamic step size RRT algorithm in complex environments, in: 2021 33rd Chinese Control and Decision Conference, CCDC, IEEE, 2021, pp. 3835–3840.
- [83] J. Yang, J. Tian, T. Chao, Variable step size strategy for RRT* algorithm, in: International Conference on Signal and Information Processing, Networking and Computers, Springer, 2023, pp. 12–19.
- [84] H. Shen, W.-F. Xie, J. Tang, T. Zhou, Adaptive manipulability-based path planning strategy for industrial robot manipulators, *IEEE/ASME Trans. Mechatronics* 28 (3) (2023) 1742–1753.
- [85] Q. Li, H. Zhao, M. Zhang, Z. Sun, A path planning algorithm for mobile robots based on DGABI-RRT, in: Intelligent Robotics and Applications: 14th International Conference, ICIRA 2021, Yantai, China, October 22–25, 2021, Proceedings, Part IV 14, Springer, 2021, pp. 554–564.
- [86] J. Kuffner, S. LaValle, RRT-connect: An efficient approach to single-query path planning, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 2, 2000, pp. 995–1001.
- [87] B. Akgun, M. Stilman, Sampling heuristics for optimal motion planning in high dimensions, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 2640–2645.
- [88] M. Jordan, A. Perez, Optimal Bidirectional Rapidly-Exploring Random Trees, MIT-CSAIL-TR-2013-021, Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA, 2013.
- [89] A.H. Qureshi, Y. Ayaz, Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments, *Robot. Auton. Syst.* 68 (2015) 1–11.
- [90] Z. Tahir, A.H. Qureshi, Y. Ayaz, R. Nawaz, Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments, *Robot. Auton. Syst.* 108 (2018) 13–27.
- [91] F. Burget, M. Bennewitz, W. Burgard, BI 2 RRT*: An efficient sampling-based path planning framework for task-constrained mobile manipulation, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2016, pp. 3714–3721.
- [92] D. Yi, M.A. Goodrich, K.D. Seppi, Homotopy-aware RRT*: Toward human-robot topological path-planning, in: Proceedings of the 2016 11th ACM/IEEE International Conference on Human-Robot Interaction, HRI, IEEE, 2016.
- [93] Z. Lin, Y. Li, J. Xiang, G. Ling, F. Suo, Bidirectional homotopy-guided RRT for path planning, in: 2020 16th International Conference on Control, Automation, Robotics and Vision, ICARCV, IEEE, 2020, pp. 333–338.
- [94] J. Wang, W. Chi, C. Li, M.Q.-H. Meng, Efficient robot motion planning using bidirectional-unidirectional RRT extend function, *IEEE Trans. Autom. Sci. Eng.* 19 (3) (2021) 1859–1868.
- [95] H. Liu, X. Zhang, J. Wen, R. Wang, X. Chen, Goal-biased bidirectional RRT based on curve-smoothing, *IFAC-PapersOnLine* 52 (24) (2019) 255–260.
- [96] P. Xin, X. Wang, X. Liu, Y. Wang, Z. Zhai, X. Ma, Improved bidirectional RRT* algorithm for robot path planning, *Sensors* 23 (2) (2023) 1041.
- [97] S.M. LaValle, J.J. Kuffner, Rapidly-exploring random trees: Progress and prospects: Steven M. Lavalle, Iowa State University, A James J. Kuffner, Jr., University of Tokyo, Tokyo, Japan, in: Algorithmic and computational robotics, AK Peters/CRC Press, 2001, pp. 303–307.
- [98] R. Bohlin, L. Kavraki, Path planning using lazy PRM, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1, 2000, pp. 521–528.
- [99] C.L. Nielsen, L.E. Kavraki, A two level fuzzy PRM for manipulation planning, in: Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113), vol. 3, IEEE, 2000, pp. 1716–1721.
- [100] K. Hauser, Lazy collision checking in asymptotically-optimal motion planning, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2015, pp. 2951–2957.
- [101] D. Kim, Y. Kwon, S.-e. Yoon, Adaptive lazy collision checking for optimal sampling-based motion planning, in: 2018 15th International Conference on Ubiquitous Robots, UR, IEEE, 2018, pp. 320–327.
- [102] J. Chase Kew, B. Ihter, M. Bandari, T.-W.E. Lee, A. Faust, Neural collision clearance estimator for batched motion planning, in: International Workshop on the Algorithmic Foundations of Robotics, Springer, 2020, pp. 73–89.
- [103] L. Janson, E. Schmerling, A. Clark, M. Pavone, Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions, 2015, arXiv:1306.3532, [Online] Available: <https://arxiv.org/abs/1306.3532>.
- [104] A.H. Qureshi, Y. Ayaz, Potential functions based sampling heuristic for optimal path planning, *Auton. Robots* 40 (2016) 1079–1093.
- [105] I.-B. Jeong, S.-J. Lee, J.-H. Kim, Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate, *Expert Syst. Appl.* 123 (2019) 82–90.
- [106] Y. Li, W. Wei, Y. Gao, D. Wang, Z. Fan, PQ-RRT*: An improved path planning algorithm for mobile robots, *Expert Syst. Appl.* 152 (2020) 113425.
- [107] B. Liao, F. Wan, Y. Hua, R. Ma, S. Zhu, X. Qing, F-RRT*: An improved path planning algorithm with improved initial solution and convergence rate, *Expert Syst. Appl.* 184 (2021) 115457.
- [108] Q. Li, J. Wang, H. Li, B. Wang, C. Feng, Fast-RRT*: An improved motion planner for mobile robot in two-dimensional space, *IEEJ Trans. Electr. Electron. Eng.* 17 (2) (2022) 200–208.
- [109] D. Armstrong, A. Jonasson, AM-RRT*: Informed sampling-based planning with assisting metric, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 10093–10099.
- [110] H. Wanga, X. Liu, S. Song, B. Li, X. Lu, J. Nie, X. Zhao, Improved RRT path planning algorithm based on growth evaluation, in: International Conference on Intelligent Equipment and Special Robots, vol. 12127, ICIESR 2021, SPIE, 2021, pp. 521–527.
- [111] M. Rickert, O. Brock, A. Knoll, Balancing exploration and exploitation in motion planning, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 2812–2817.
- [112] C. Urmson, R. Simmons, Approaches for heuristically biasing RRT growth, in: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), vol. 2, IEEE, 2003, pp. 1178–1183.
- [113] M. Penrose, *Random Geometric Graphs*, vol. 5, OUP Oxford, 2003.
- [114] S. Koenig, M. Likhachev, D. Furcy, Lifelong planning A*, *Artificial Intelligence* 155 (1–2) (2004) 93–146.
- [115] M.P. Strub, J.D. Gammell, Advanced BIT*(ABIT*): Sampling-based planning with advanced graph-search techniques, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 130–136.
- [116] L. Chen, L. Yu, S. Libin, Z. Jiwen, Greedy BIT*(GBIT*): greedy search policy for sampling-based optimal planning with a faster initial solution and convergence, in: 2021 International Conference on Computer, Control and Robotics, ICCCR, IEEE, 2021, pp. 30–36.
- [117] M.P. Strub, J.D. Gammell, Adaptively Informed Trees (AIT*) and Effort Informed Trees (EIT*): Asymmetric bidirectional sampling-based path planning, *Int. J. Robot. Res.* 41 (4) (2022) 390–417.
- [118] V.N. Hartmann, M.P. Strub, M. Toussaint, J.D. Gammell, Effort Informed Roadmaps (EIRM*): Efficient asymptotically optimal multiquery planning by actively reusing validation effort, in: The International Symposium of Robotics Research, Springer, 2022, pp. 555–571.
- [119] C. Li, H. Ma, P. Xu, J. Wang, M.Q.-H. Meng, BiAIT*: Symmetrical bidirectional optimal path planning with adaptive heuristic, *IEEE Trans. Autom. Sci. Eng.* (2023) 1–13.
- [120] O. Arslan, P. Tsiotras, Use of relaxation methods in sampling-based algorithms for optimal motion planning, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 2421–2428.
- [121] O. Arslan, P. Tsiotras, Dynamic programming guided exploration for sampling-based motion planning algorithms, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, Seattle, WA, USA, 2015, pp. 4819–4826.
- [122] M. Otte, E. Frazzoli, RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning, *Int. J. Robot. Res.* 35 (7) (2016) 797–822.

- [123] K. Naderi, J. Rajamäki, P. Hämmäläinen, RT-RRT* a real-time path planning algorithm based on RRT, in: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, 2015, pp. 113–118.
- [124] J.-G. Kang, D.-W. Lim, Y.-S. Choi, W.-J. Jang, J.-W. Jung, Improved RRT-connect algorithm based on triangular inequality for robot path planning, *Sensors* 21 (2) (2021) 333.
- [125] J.-G. Kang, J.-W. Jung, Post triangular rewiring method for shorter RRT robot path planning, 2021, arXiv preprint arXiv:2107.05344.
- [126] J. Kamat, J. Ortiz-Haro, M. Toussaint, F.T. Pokorny, A. Orthey, Bitkomo: Combining sampling and optimization for fast convergence in optimal motion planning, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2022, pp. 4492–4497.
- [127] S. Choudhury, J.D. Gammell, T.D. Barfoot, S.S. Srinivasa, S. Scherer, Regionally Accelerated Batch Informed Trees (RABIT*): A framework to integrate local information into optimal path planning, in: 2016 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2016, pp. 4207–4214.
- [128] M. Toussaint, Newton methods for k-order markov constrained motion problems, 2014, arXiv preprint arXiv:1407.0414.
- [129] N. Ratliff, M. Zucker, J.A. Bagnell, S. Srinivasa, CHOMP: Gradient optimization techniques for efficient motion planning, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 489–494.
- [130] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, S. Schaal, STOMP: Stochastic trajectory optimization for motion planning, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 4569–4574.
- [131] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, P. Abbeel, Motion planning with sequential convex optimization and convex collision checking, *Int. J. Robot. Res.* 33 (9) (2014) 1251–1270.
- [132] A.A. Neto, D.G. Macharet, M.F. Campos, Feasible RRT-based path planning using seventh order Bézier curves, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 1445–1450.
- [133] J. Li, C. Yang, AUV path planning based on improved RRT and Bezier curve optimization, in: 2020 IEEE International Conference on Mechatronics and Automation, ICMA, IEEE, 2020, pp. 1359–1364.
- [134] Z. Durakli, V. Nabiyev, A new approach based on Bezier curves to solve path planning problems for mobile robots, *J. Comput. Sci.* 58 (2022) 101540.
- [135] Y. Sun, C. Zhang, P. Sun, C. Liu, Safe and smooth motion planning for Mecanum-Wheeled robot using improved RRT and cubic spline, *Arab. J. Sci. Eng.* 45 (4) (2020) 3075–3090.
- [136] X. Ma, R. Gong, Y. Tan, H. Mei, C. Li, Path planning of mobile robot based on improved PRM based on cubic spline, *Wirel. Commun. Mob. Comput.* 2022 (1) (2022) 1632698.
- [137] C.-H. Wei, J.-S. Liu, Hybridizing RRT and variable-length genetic algorithm for smooth path generation, in: 2011 IEEE International Conference on Robotics and Biomimetics, IEEE, 2011, pp. 626–632.
- [138] R. Mashayekhi, M.Y.I. Idris, M.H. Anisi, I. Ahmady, Hybrid RRT: A semi-dual-tree RRT-based motion planner, *IEEE Access* 8 (2020) 18658–18668.
- [139] S. Al-Ansary, S. Al-Darraj, Hybrid RRT-A*: An improved path planning method for an autonomous mobile robots, *Iraqi J. Electr. Electron. Eng.* 17 (1) (2021).
- [140] F. Kiani, A. Seyyedabbasi, R. Aliyev, M.U. Gulle, H. Basyildiz, M.A. Shah, Adapted-RRT: novel hybrid method to solve three-dimensional path planning problem using sampling and metaheuristic-based algorithms, *Neural Comput. Appl.* 33 (22) (2021) 15569–15599.
- [141] M.A.R. Pohan, J. Utama, Efficient sampling-based for mobile robot path planning in a dynamic environment based on the rapidly-exploring random tree and a rule-template sets, *Int. J. Eng.* 36 (4) (2023) 797–806.
- [142] L. Cao, L. Wang, Y. Liu, S. Yan, 3D trajectory planning based on the rapidly-exploring random tree-connect and artificial potential fields method for unmanned aerial vehicles, *Int. J. Adv. Robot. Syst.* 19 (5) (2022) 17298806221118867.
- [143] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robot. Res.* 5 (1) (1986) 90–98.
- [144] Z. Wang, K. Wang, S. An, Cubic B-spline interpolation and realization, in: Information Computing and Applications: Second International Conference, ICICA 2011, Qinhuangdao, China, October 28–31, 2011. Proceedings, Part 1 2, Springer, 2011, pp. 82–89.
- [145] Z. Bing, D. Lerch, K. Huang, A. Knoll, Meta-reinforcement learning in non-stationary and dynamic environments, *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (3) (2023) 3476–3491.
- [146] X. Zhang, T. Zhu, L. Du, Y. Hu, H. Liu, Local path planning of autonomous vehicle based on an improved heuristic bi-RRT algorithm in dynamic obstacle avoidance environment, *Sensors* 22 (20) (2022) 7968.
- [147] Q. Zou, X. Du, Y. Liu, H. Chen, Y. Wang, J. Yu, Dynamic path planning and motion control of microbotic swarms for mobile target tracking, *IEEE Trans. Autom. Sci. Eng.* 20 (4) (2023) 2454–2468.
- [148] R. Seif, M.A. Oskoei, Mobile robot path planning by RRT* in dynamic environments, *Int. J. Intell. Syst. Appl.* 7 (5) (2015) 24.
- [149] D. Connell, H.M. La, Dynamic path planning and replanning for mobile robots using RRT*, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2017, pp. 1429–1434.
- [150] J. Qi, H. Yang, H. Sun, MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment, *IEEE Trans. Ind. Electron.* 68 (8) (2021) 7244–7251.
- [151] P. Zhao, Y. Chang, W. Wu, H. Luo, Z. Zhou, Y. Qiao, Y. Li, C. Zhao, Z. Huang, B. Liu, et al., Dynamic RRT: fast feasible path planning in randomly distributed obstacle environments, *J. Intell. Robot. Syst.* 107 (4) (2023) 48.
- [152] O. Adiyatov, H.A. Varol, A novel RRT*-based algorithm for motion planning in dynamic environments, in: 2017 IEEE International Conference on Mechatronics and Automation, ICMA, IEEE, 2017, pp. 1416–1421.
- [153] K. Wei, B. Ren, A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm, *Sensors* 18 (2) (2018) 571.
- [154] K. Cai, C. Wang, C. Li, S. Song, M.Q.-H. Meng, Adaptive sampling for human-aware path planning in dynamic environments, in: 2019 IEEE International Conference on Robotics and Biomimetics, ROBIO, IEEE, 2019, pp. 1987–1994.
- [155] C. Yuan, G. Liu, W. Zhang, X. Pan, An efficient RRT cache method in dynamic environments for path planning, *Robot. Auton. Syst.* 131 (2020) 103595.
- [156] W. Chi, M.Q.-H. Meng, Risk-RRT*: A robot motion planning algorithm for the human robot coexisting environment, in: 2017 18th International Conference on Advanced Robotics, ICAR, IEEE, 2017, pp. 583–588.
- [157] W. Chi, C. Wang, J. Wang, M.Q.-H. Meng, Risk-DTRRT-based optimal motion planning algorithm for mobile robots, *IEEE Trans. Autom. Sci. Eng.* 16 (3) (2018) 1271–1288.
- [158] K. Cai, W. Chen, D. Dugas, R. Siegart, J.J. Chung, Sampling-based path planning in highly dynamic and crowded pedestrian flow, *IEEE Trans. Intell. Transp. Syst.* (2023).
- [159] K. Cai, W. Chen, C. Wang, S. Song, M.Q.-H. Meng, Human-aware path planning with improved virtual doppler method in highly dynamic environments, *IEEE Trans. Autom. Sci. Eng.* 20 (2) (2022) 1304–1321.
- [160] Y. Tian, L. Yan, G.-Y. Park, S. Yang, Y.-S. Kim, S.-R. Lee, C.-Y. Lee, Application of RRT-based local path planning algorithm in unknown environment, in: 2007 International Symposium on Computational Intelligence in Robotics and Automation, 2007, pp. 456–460.
- [161] L. Chang-an, C. Jin-gang, L. Guo-dong, L. Chun-yang, Mobile robot path planning based on an improved rapidly-exploring random tree in unknown environment, in: 2008 IEEE International Conference on Automation and Logistics, IEEE, 2008, pp. 2375–2379.
- [162] J. Li, C. Li, T. Chen, Y. Zhang, Improved RRT algorithm for AUV target search in unknown 3D environment, *J. Mar. Sci. Eng.* 10 (6) (2022) 826.
- [163] B. Lindqvist, A. Patel, K. Löfgren, G. Nikolakopoulos, A tree-based next-best-trajectory method for 3D UAV exploration, *IEEE Trans. Robot.* (2024).
- [164] R. Pepy, A. Lambert, Safe path planning in an uncertain-configuration space using RRT, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2006, pp. 5376–5381.
- [165] Y. Huang, K. Gupta, RRT-SLAM for motion planning with motion and map uncertainty for robot exploration, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2008, pp. 1077–1082.
- [166] B.D. Luders, J.P. How, An optimizing sampling-based motion planner with guaranteed robustness to bounded uncertainty, in: 2014 American Control Conference, IEEE, 2014, pp. 771–777.
- [167] T. Summers, Distributionally robust sampling-based motion planning under uncertainty, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 6518–6523.
- [168] B. Englot, T. Shan, S.D. Bopardikar, A. Speranzon, Sampling-based min-max uncertainty path planning, in: 2016 IEEE 55th Conference on Decision and Control, CDC, IEEE, 2016, pp. 6863–6870.
- [169] H. Banzhaf, L. Palmieri, D. Nienhüser, T. Schamm, S. Knoop, J.M. Zöllner, Hybrid curvature steer: A novel extend function for sampling-based nonholonomic motion planning in tight environments, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2017, pp. 1–8.
- [170] J. Peng, Y. Chen, Y. Duan, Y. Zhang, J. Ji, Y. Zhang, Towards an online RRT-based path planning algorithm for ackermann-steering vehicles, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 7407–7413.
- [171] R. Reyes, I. Becerra, R. Murrieta-Cid, S. Hutchinson, Visual-RRT: Integrating IBVS as a steering method in an RRT planner, *Robot. Auton. Syst.* 169 (2023) 104525.
- [172] Y. Gan, B. Zhang, C. Ke, X. Zhu, W. He, T. Ihara, Research on robot motion planning based on RRT algorithm with nonholonomic constraints, *Neural Process. Lett.* 53 (2021) 3011–3029.
- [173] Y. Dong, E. Camci, E. Kayacan, Faster RRT-based nonholonomic path planning in 2D building environments using skeleton-constrained path biasing, *J. Intell. Robot. Syst.* 89 (2018) 387–401.

- [174] Y. Chen, M. Liu, L. Wang, RRT* combined with gvo for real-time non-holonomic robot navigation in dynamic environment, in: 2018 IEEE International Conference on Real-Time Computing and Robotics, RCAR, IEEE, 2018, pp. 479–484.
- [175] J.J. Park, B. Kuipers, Feedback motion planning via non-holonomic RRT* for mobile robots, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2015, pp. 4035–4040.
- [176] Y. Zhang, D. Gong, S-BRRT*: A spline-based bidirectional RRT with strategies under nonholonomic constraint, in: 2021 33rd Chinese Control and Decision Conference, CCDC, IEEE, 2021, pp. 1753–1758.
- [177] K. Yang, S. Sukkarieh, Real-time continuous curvature path planning of UAVs in cluttered environments, in: 2008 5th International Symposium on Mechatronics and Its Applications, IEEE, 2008, pp. 1–6.
- [178] D.J. Webb, J.v.d. Berg, Kinodynamic RRT*: Optimal motion planning for systems with linear differential constraints, 2012, arXiv preprint arXiv:1205.5088.
- [179] Y. Li, R. Cui, Z. Li, D. Xu, Neural network approximation based near-optimal motion planning with kinodynamic constraints using RRT, IEEE Trans. Ind. Electron. 65 (11) (2018) 8718–8729.
- [180] H.-T.L. Chiang, J. Hsu, M. Fiser, L. Tapia, A. Faust, RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies, IEEE Robot. Autom. Lett. 4 (4) (2019) 4298–4305.
- [181] M. Yavari, K. Gupta, M. Mehrandezh, Lazy steering RRT*: An optimal constrained kinodynamic neural network based planner with no in-exploration steering, in: 2019 19th International Conference on Advanced Robotics, ICAR, IEEE, 2019, pp. 400–407.
- [182] D. Zheng, P. Tsiotras, Accelerating kinodynamic RRT* through dimensionality reduction, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 3674–3680.
- [183] D. Berenson, S.S. Srinivasa, D. Ferguson, J.J. Kuffner, Manipulation planning on constraint manifolds, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 625–632.
- [184] C. Suh, T.T. Um, B. Kim, H. Noh, M. Kim, F.C. Park, Tangent space RRT: A randomized planning algorithm on constraint manifolds, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 4968–4973.
- [185] L. Jaillet, J.M. Porta, Path planning under kinematic constraints by rapidly exploring manifolds, IEEE Trans. Robot. 29 (1) (2012) 105–117.
- [186] L. Jaillet, J.M. Porta, Efficient asymptotically-optimal path planning on manifolds, Robot. Auton. Syst. 61 (8) (2013) 797–807.
- [187] L. Jaillet, J.M. Porta, Path planning with loop closure constraints using an atlas-based RRT, in: Robotics Research: The 15th International Symposium ISRR, Springer, 2017, pp. 345–362.
- [188] B. Kim, T.T. Um, C. Suh, F.C. Park, Tangent bundle RRT: A randomized algorithm for constrained motion planning, Robotica 34 (1) (2016) 202–225.
- [189] L. Han, L. Rudolph, J. Blumenthal, I. Valodzin, Convexly stratified deformation spaces and efficient path planning for planar closed chains with revolute joints, Int. J. Robot. Res. 27 (11–12) (2008) 1189–1212.
- [190] T.A. McMahon, Sampling Based Motion Planning with Reachable Volumes (Ph.D. thesis), 2016.
- [191] I.A. Şucan, S. Chitta, Motion planning with constraints using configuration space approximations, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 1904–1910.
- [192] F. Burget, A. Hornung, M. Bennewitz, Whole-body motion planning for manipulation of articulated objects, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 1656–1662.
- [193] Z. Bing, A. Rohregger, F. Walter, Y. Huang, P. Lucas, F.O. Morin, K. Huang, A. Knoll, Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot, Science Robotics 8 (85) (2023).
- [194] K. Cai, R. Laha, Y. Gong, L. Chen, L. Zhang, L.F. Figueredo, S. Haddadin, Demonstration to adaptation: A user-guided framework for sequential and real-time planning, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2024, pp. 9871–9878.
- [195] B. Cain, M. Kalaitzakis, N. Vitzilaios, MK-RRT*: Multi-robot kinodynamic RRT trajectory planning, in: 2021 International Conference on Unmanned Aircraft Systems, ICUAS, IEEE, 2021, pp. 868–876.
- [196] J. Hvězda, M. Kulich, L. Pfeučil, Improved discrete RRT for coordinated multi-robot planning, 2019, arXiv preprint arXiv:1901.07363.
- [197] L. Zhang, Z. Lin, J. Wang, B. He, Rapidly-exploring random trees multi-robot map exploration under optimization framework, Robot. Auton. Syst. 131 (2020) 103565.
- [198] A. Adler, M. De Berg, D. Halperin, K. Solovey, Efficient multi-robot motion planning for unlabeled discs in simple polygons, in: Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics, Springer, 2015, pp. 1–17.
- [199] K. Solovey, O. Salzman, D. Halperin, Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning, Int. J. Robot. Res. 35 (5) (2016) 501–513.
- [200] R. Shome, K. Solovey, A. Dobson, D. Halperin, K.E. Bekris, drtt*: Scalable and informed asymptotically-optimal multi-robot motion planning, Auton. Robots 44 (3) (2020) 443–467.
- [201] J. Sim, J. Kim, C. Nam, Safe interval RRT* for scalable multi-robot path planning in continuous space, 2024, arXiv preprint arXiv:2404.01752.
- [202] L. Zhou, J. Ding, X. Fan, An adaptive RRT algorithm based on narrow passage recognition for assembly path planning, in: 2023 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM, IEEE, 2023, pp. 0203–0208.
- [203] A. Belaid, B. Mendil, A. Djenadi, Narrow passage RRT*: a new variant of RRT, Int. J. Comput. Vis. Robot. 12 (1) (2022) 85–100.
- [204] Q. Chai, Y. Wang, RJ-RRT: improved RRT for path planning in narrow passages, Appl. Sci. 12 (23) (2022) 12033.
- [205] X. Shu, F. Ni, Z. Zhou, Y. Liu, H. Liu, T. Zou, Locally guided multiple Bi-RRT* for fast path planning in narrow passages, in: 2019 IEEE International Conference on Robotics and Biomimetics, ROBIO, IEEE, 2019, pp. 2085–2091.
- [206] J. Szkandera, I. Kolingerová, M. Maňák, Narrow passage problem solution for motion planning, in: Computational Science–ICCS 2020: 20th International Conference, Amsterdam, the Netherlands, June 3–5, 2020, Proceedings, Part I 20, Springer, 2020, pp. 459–470.
- [207] W. Wang, L. Zuo, X. Xu, A learning-based multi-RRT approach for robot path planning in narrow passages, J. Intell. Robot. Syst. 90 (2018) 81–100.
- [208] F. Islam, J. Nasir, U. Malik, Y. Ayaz, O. Hasan, RRT*-smart: Rapid convergence implementation of RRT* towards optimal solution, in: 2012 IEEE International Conference on Mechatronics and Automation, IEEE, 2012, pp. 1651–1656.
- [209] L. Jaillet, A. Yershova, S. La Valle, T. Simeon, Adaptive tuning of the sampling domain for dynamic-domain RRTs, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 2851–2856.
- [210] D. Ferguson, A. Stentz, Anytime RRTs, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2006, pp. 5369–5375.
- [211] L. Jaillet, J. Cortes, T. Simeon, Transition-based RRT for path planning in continuous cost spaces, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 2145–2150.
- [212] T.T. Um, B. Kim, C. Suh, F.C. Park, Tangent space RRT with lazy projection: An efficient planning algorithm for constrained motions, in: Advances in Robot Kinematics: Motion in Man and Machine: Motion in Man and Machine, Springer, 2010, pp. 251–260.
- [213] R. Kang, H. Liu, Z. Wang, Fast convergence RRT for asymptotically-optimal motion planning, in: 2016 IEEE International Conference on Robotics and Biomimetics, ROBIO, IEEE, 2016, pp. 2111–2116.
- [214] C. Wong, E. Yang, X.-T. Yan, D. Gu, Optimal path planning based on a multi-tree T-RRT* approach for robotic task planning in continuous cost spaces, in: 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, IEEE, 2018, pp. 242–247.
- [215] S. Primatesta, A. Osman, A. Rizzo, MP-RRT*: A model predictive sampling-based motion planning algorithm for unmanned aircraft systems, J. Intell. Robot. Syst. 103 (2021) 59.
- [216] X. Liao, Z. Zhu, H. Tang, W. Zhang, F-RRT: Fast RRT algorithm for path planning in dynamic environments, Int. J. Adv. Robot. Syst. 18 (1) (2021) 1–12.
- [217] O. Khattab, A. Yasser, M.A. Jaradat, L. Romdhane, Intelligent adaptive RRT* path planning algorithm for mobile robots, in: 2023 Advances in Science and Engineering Technology International Conferences, ASET, 2023, pp. 01–06.
- [218] Z. Lv, H. Zhang, W. Liu, GMM-RRT for kiwifruit picking path planning, J. Agric. Robot. (2023) 1–15.
- [219] A. Saccuti, R. Monica, J. Aleotti, PROTAMP-RRT: A probabilistic integrated task and motion planner based on RRT, IEEE Robot. Autom. Lett. 8 (12) (2023) 8398–8405.
- [220] J. Ortiz-Haro, W. Hönig, V.N. Hartmann, M. Toussaint, L. Righetti, iDb-RRT: Sampling-based kinodynamic motion planning with motion primitives and trajectory optimization, 2024, arXiv:2403.10745, [Online] Available: <https://arxiv.org/abs/2403.10745>.
- [221] M. Moll, I.A. Sucan, L.E. Kavraki, Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization, IEEE Robot. Autom. Mag. 22 (3) (2015) 96–102.
- [222] J.D. Gammell, M.P. Strub, V.N. Hartmann, Planner Developer Tools (PDT): Reproducible experiments and statistical analysis for developing and testing motion planners, in: Proceedings of the Workshop on Evaluating Motion Planning Performance (EMPP), IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2022.
- [223] R. Diankov, Automated Construction of Robotic Manipulation Programs (Ph.D. thesis), Carnegie Mellon University, 2010.
- [224] I.A. Sucan, M. Moll, L.E. Kavraki, The open motion planning library, IEEE Robot. Autom. Mag. 19 (4) (2012) 72–82.