



Adaptive hybrid local–global sampling for fast informed sampling-based optimal path planning

Marco Faroni¹ · Nicola Pedrocchi² · Manuel Beschi³

Received: 14 August 2022 / Accepted: 17 March 2024
© The Author(s) 2024

Abstract

This paper improves the performance of RRT*-like sampling-based path planners by combining admissible informed sampling and local sampling (i.e., sampling the neighborhood of the current solution). An adaptive strategy regulates the trade-off between exploration (admissible informed sampling) and exploitation (local sampling) based on online rewards from previous samples. The paper demonstrates that the algorithm is asymptotically optimal and has a better convergence rate than state-of-the-art path planners (e.g., Informed-RRT*) in several simulated and real-world scenarios. An open-source, ROS-compatible implementation of the algorithm is publicly available.

Keywords Motion planning · Path planning · Sampling-based algorithms · Informed sampling · Online learning for motion planning · Informed-RRT*

1 Introduction

Path planning is a fundamental problem in robotics and with a heavy impact on a broad variety of applications. For example, the recent developments in humanoid robotics require fast planning tools to handle high-dimensional systems. Similarly, industrial and service robotics often deal with dynamic environments where the robot must plan the motion on the fly. An example is a robot arm that picks objects from a conveyor belt or cooperates with humans to assemble a piece of furniture. A common thread of these applications is the high dimensionality of the search space and the limited available computing time to find a solution.

Path-planning problems are solved mainly through graph-based or sampling-based approaches. Graph-based methods (Hart et al., 1968; Likhachev et al., 2008) are used mainly for navigation problems, while sampling-based methods are the most widespread in robotic manipulation because they are more efficient with high-dimensional systems. Sampling-based methods explore the search space by randomly sampling the robot configuration space to find a sequence of feasible nodes from start to goal. Different strategies for sampling and connecting nodes have given birth to different algorithms, such as RRT (LaValle, 1998), EST (Hsu et al., 1997), and PRM (Kavraki et al., 1996).

Sampling-based methods are successful in robotics because they do not require discretizing the search space, do not explicitly require the construction of the obstacle space, and generalize well to different robots' structures and specifications. These advantages come at the cost of weaker completeness and optimality guarantees. In particular, they can provide *asymptotic optimality*; that is, the probability of converging to the optimal solution approaches one as the number of samples goes to infinity (Karaman & Frazzoli, 2011). The convergence rate of such algorithms is relatively slow, and actual implementations usually stop the search way before they reach the optimum. A meaningful improvement to optimal planners came with the introduction of informed sampling (Gammell et al., 2018). Informed sampling-based planners shrink the sampling space every time the solution

✉ Marco Faroni
marco.faroni@polimi.it

Nicola Pedrocchi
nicola.pedrocchi@stiima.cnr.it

Manuel Beschi
manuel.beschi@unibs.it

¹ Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan 20133, Italy

² STIIMA-CNR - Institute of Intelligent Industrial Technologies and Systems, National Research Council of Italy, Via Alfonso Corti 12, Milan 20133, Italy

³ Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Via Branze 38, Brescia 25123, Italy

cost decreases, making the convergence to the optimal solution faster. These planners show a slow convergence rate when the cost heuristic is poorly informative. In the case of path length minimization, the Euclidean distance can be chosen as a heuristic of the cost between two points. However, with many obstacles, there is a large difference between the Euclidean distance and the actual minimum path between the two points. In these cases, the convergence speed resembles that of uninformed planners (e.g., RRT* (Karaman & Frazzoli, 2011)). This paper tackles this issue by proposing a mixed strategy that alternates sampling the informed set and the neighborhood of the current solution. The rationale is that the cost of the solution improves by sampling its neighborhood (i.e., local sampling), with a consequent quick reduction of the measure of the informed set.

Alternating admissible and locally informed sampling is an example of the classic exploration-versus-exploitation dilemma, which is hardly solvable with a fixed ratio between the usage of the two sampling strategies. To overcome this issue, we propose an adaptive technique to dynamically balance the choice of one sampling strategy over the other. The result is that the search algorithm prefers exploitation (i.e., local sampling) only as long as it is useful and switches to exploration (i.e., admissible informed sampling) to avoid stagnation.

The paper's contribution is twofold. First, it defines a mixed sampling strategy that combines global and local informed sampling for asymptotically optimal sampling-based path planners. Local informed sampling oversamples the neighborhood of the current solution to quickly reach a local optimum, while global informed sampling guarantees asymptotic optimality. Second, it proposes an asymptotically optimal algorithm that uses the mixed sampling strategy and dynamically adjusts the trade-off between global and local sampling, showing that this outperforms state-of-the-art planners, such as Informed-RRT*, on different classes of problems.

An open-source ROS-compatible version of the planner is publicly available https://github.com/JRL-CARI-CNR-UNIBS/cari_motion_planning.

The paper is organized as follows. Section 2 introduces the reader to optimal planning and informed sampling. Section 3 discusses previous works on the acceleration of informed sampling-based planners. Section 4 discusses the motivation of this work through some illustrative examples. Section 5 describes the proposed method. Section 6 compares it with other methods. Section 7 concludes and discusses future works.

2 Informed sampling-based optimal path planning

This section introduces the concepts of path planning, informed sets, and informed sampling used throughout the paper.

The path planning problem is formulated in the configuration space, $X \subseteq \mathbb{R}^n$, which denotes all possible configurations x of the system (for robot manipulators, x is usually a vector of joint angles). Let X_{obs} be the space of all those configurations in collision with an obstacle, and $X_{\text{free}} = \text{cl}(X \setminus X_{\text{obs}})$ the obstacle-free configuration space, where $\text{cl}(\cdot)$ denotes the closure of the set.

Definition 1 (*optimal path planning*) (adapted from Gammell et al. 2018) Given a starting point x_{start} and a set of desired goal points $X_{\text{goal}} \subset X$, *optimal path planning* is the problem of finding a curve $\sigma^* : [0, 1] \rightarrow X_{\text{free}}$ such that:

$$\sigma^* = \underset{\sigma \in \Sigma}{\operatorname{argmin}} \{c(\sigma) \mid \sigma(0) = x_{\text{start}}, \sigma(1) \in X_{\text{goal}}\} \quad (1)$$

where $c : \Sigma \rightarrow \mathbb{R}_{\geq 0}$ is a Lipschitz continuous cost function associating a cost $c(\sigma)$ to a curve $\sigma \in \Sigma$, Σ is the set of solution paths, and $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers.

Remark 1 Cost function c is often the length of the path so that the optimal motion plan is the shortest collision-free path from x_{start} to X_{goal} .

If an algorithm can find a solution to the optimal path planning problem, then it is said to be an *optimal path planner*. Sampling-based path planners, such as RRT* (Karaman & Frazzoli, 2011), can only ensure the probabilistic convergence to the optimal solution. This weaker form of optimality is referred to as (*almost-sure*) *asymptotic optimality*. The convergence rate of an asymptotically optimal planner is related to the probability of sampling points that can improve the current solution. This set is referred to as the *omniscient set* (Gammell et al., 2018). RRT* and similar algorithms, as proposed in Karaman and Frazzoli (2011), are very inefficient at sampling the omniscient set (the probability that RRT* samples a point that belongs to the omniscient set decreases factorially in the state dimension (Gammell et al., 2018)). To increase the probability of sampling the omniscient set, Gammell et al. (2018) coined the concept of informed sampling; that is, sampling an approximation of the omniscient set (the *informed set*) so that the probability of finding a point that improves the current solution is higher. If the informed set is a superset of the omniscient set, it is referred to as an *admissible informed set*.

Definition 2 (*admissible informed set*) (adapted from Gammell et al. 2018) An informed set $X_{\hat{f}}$ is a heuristic estimate

of the omniscient set X_f . If $X_{\hat{f}} \supseteq X_f$, the informed set is said to be *admissible*.

In minimum-length path planning, it is always possible to construct an admissible informed set by considering that the shortest path through a sample $x \in X$ is lower bounded by the sum of Euclidean distances from x_{start} to x and from x to $x_{\text{goal}} \in X_{\text{goal}}$. As a consequence, all possibly improving points lie in the so-called \mathcal{L}_2 -informed set, $X_{\hat{f}}$, given by:

$$X_{\hat{f}} = \{x \in X_{\text{free}} \mid \|x - x_{\text{start}}\|_2 + \|x_{\text{goal}} - x\|_2 < c_k\} \quad (2)$$

where c_k is the cost of the best solution at iteration k . Notice that such an informed set is equivalent to the intersection of the free space X_{free} and an n -dimensional hyper-ellipsoid symmetric about its transverse axis with focal points at x_{start} and x_{goal} , transverse diameter equal to c_k , and conjugate diameters equal to $\sqrt{c_k^2 - c_{\text{min}}^2}$, where

$$c_{\text{min}} = \|x_{\text{goal}} - x_{\text{start}}\|_2. \quad (3)$$

The volume of the hyper-ellipsoid decreases progressively as the solution cost c_k decreases, improving the convergence rate of the algorithm.

3 Related works

Informed sampling stems from the simple but effective idea of sampling only points with a higher probability of improving the solution. This is not a new idea, in principle, as several works use heuristics to bias sampling (Urmson & Simmons, 2003; Rodriguez et al., 2008; Salzman & Halperin, 2013; Shan et al., 2014; Ge et al., 2016; Santana Correia et al., 2018; Yu et al., 2019; Lai et al., 2020; Faroni & Berenson, 2023).

The main issue with sampling bias is that, depending on the geometry of the problem, the heuristic may discard points of X_f . (Gammell et al., 2018). This can be deleterious for the convergence speed, and it may even compromise the optimality of the algorithm. Compared to these works, admissible informed sampling never excludes any points possibly belonging to the omniscient set; thus, it retains asymptotic optimality regardless of the geometry of the problem. Nonetheless, convergence speed may be slow when the admissible heuristic is not informative.

Few works attempted to speed up the convergence rate by combining informed planning and local techniques. In (Kim & Song, 2015, 2018), Kim and Song propose to run a deterministic path short-cutter every time the algorithm improves the solution. The short-cutting procedure acts as follows: i) it considers three consecutive nodes on the path at a time; ii) it discretizes the two corresponding edges; iii) it tries to

connect the extreme nodes to the sampled segment until it finds a collision; iv) it moves the central node to the intersection of the two segments found in the previous step. Such an approach has two main drawbacks. First, the computational time owed to the short-cutting is significant as it requires an iterative edge evaluation (i.e., collision checking) every time it tries to refine a triple of nodes. Second, this approach is suitable only for minimum-path problems, as it relies on the triangular inequality applied to each triple of nodes. Hauer and Tsiotras (2017) proposes to refine the current solution by moving the nodes of the tree based on gradient descent. Yet, every time a node is moved, the refinement process requires an intensive edge evaluation.

The idea of combining global and local optimization was also explored by Choudhury et al. (2016), who propose a hybrid use of BIT* (Gammell et al., 2015), a lazy heuristic-driven informed planner and CHOMP (Ratliff et al., 2009), a gradient-based local planner. Roughly speaking, the local planner is used to solve a two-point problem between a pair of nodes. One main drawback is that the local planner is called every time an edge is evaluated, which may be computationally counter-effective. Other variants of BIT* were proposed in Strub and Gammell (2020), and Strub and Gammell (2020), but focusing on how to improve the heuristics by experience. Faroni and Berenson (2023) uses online learning (clustering of previous edges and Multi-Armed Bandits) to oversample promising regions.

Finally, Joshi and Tsiotras (2020) and Mandalika et al. (2021) propose approaches to focus the search on subsets of the informed set. Mandalika et al. (2021) decomposes a planning problem into two sub-problems and applies informed sampling to them. The union of the informed sets of the sub-problems is strictly contained in the informed set of the initial problem; thus, the search focuses on a smaller region. However, the performance of such an approximation strongly depends on the problem geometry, and the authors do not discuss how to retain asymptotic optimality. Joshi and Tsiotras (2020) estimates the cost-to-come of the tree leaves to bias the search towards a subset of the informed set, called the relevant region. In this case, the trade-off between exploration and exploitation is fixed. Thus, the performance depends on the problem geometry, and it may be even worse than admissible informed sampling.

Our approach is similar to the works mentioned above as it alternates informed sampling and local refinement of the path. Compared to Kim and Song (2015, 2018) and Hauer and Tsiotras (2017), our method refines the path by sampling the neighborhood of the current solution, and this allows for gradient-free refinement, also with generic cost functions. Moreover, Kim and Song (2015, 2018) and Hauer and Tsiotras (2017) tend to favor exploitation (i.e., path refinement) rather than exploration, wasting time optimizing suboptimal solutions (see numerical results in Sect. 6). Similarly,

Choudhury et al. (2016) and Joshi and Tsiotras (2020) use a fixed balance between exploration and exploitation; thus, the performance may vary a lot across different problems. Our method adjusts the trade-off between exploration and exploitation according to the cost progression, adapting to different problems. In this sense, we could use our adaptive scheme in Joshi and Tsiotras (2020) to dynamically balance the trade-off between exploration and exploitation and in Mandalika et al. (2021) to retain asymptotic optimality.

4 Motivation for an adaptive mixed sampling strategy

To understand the motivation behind this work, consider the minimum-path problem in Fig. 1a. Because of the presence of a large obstacle between x_{start} and x_{goal} , the \mathcal{L}_2 -informed set is large and poorly informative. Sampling the neighborhood of the current solution would be much more efficient than considering the whole \mathcal{L}_2 -informed set, as the path would quickly converge to the global optimum. This situation is expected when the current and the optimal solutions are homotopic. We will refer to this sampling strategy as *local sampling*.

On the other hand, when the current solution is locally optimal, any efforts on the local optimization would be useless. For example, in Fig. 1b, the optimal solution passes through the narrow passage between the two obstacles; thus, sampling the neighborhood of the current solution would lead to a local optimum (yellow in Fig. 1c). As the solution approaches the local optimum, the probability of improving the solution via local sampling is equal to zero.

Notice that a fast convergence to a local minimum quickly reduces the volume of the informed set. However, it is crucial to understand when the local sampling is beneficial without losing the asymptotic global optimality.

In this paper, we combine admissible informed sampling and local sampling in a *mixed sampling strategy*. On the one hand, sampling the admissible informed set guarantees that all points from the omniscient set are taken into account. On the other hand, local sampling has a twofold role. First, if the local and the global optima correspond, it quickly converges to the solution, as in Fig. 1c. Second, it reduces the size of the admissible informed set. Indeed, the Lebesgue measure of the \mathcal{L}_2 -informed set is directly related to the best cost to date c_k as follows:

$$\lambda(X_{\hat{f}}) = c_k (c_k^2 - c_{\min}^2)^{\frac{n-1}{2}} \frac{\zeta_n}{2^n} \quad (4)$$

where ζ_n is the Lebesgue measure of the unit ball (dependent only on n) (Gammell et al., 2018). Hence, improving the current solution (even in the neighborhood of a local minimum)

enhances the convergence speed to the globally optimal solution.

5 Proposed approach

This section describes the proposed mixed sampling strategy. First, it defines the local informed set. Second, it designs an algorithm to dynamically change the local sampling probability based on the cost evolution. Finally, it proves asymptotic optimality.

5.1 Mixed-strategy sampling

Consider an n -dimensional path planning problem solved by a sampling-based planner. Let $\sigma_k \in X_{\text{free}}$ be the current solution at iteration k and $c_k = c(\sigma_k)$. An RRT*-like planner is asymptotically optimal if the algorithm that connects nodes satisfies conditions on the minimum rewire radius and the sampler draws nodes from a superset of the omniscient set. If we drop the second condition, such a relaxed planner would converge to a local optimum. To formulate this idea more formally, we introduce the notion of *local informed set*. Then, we combine it with admissible informed sampling to obtain the adaptive *mixed-strategy sampler* used in the proposed planner.

Definition 3 (*local informed set*) The *local informed set* of the current solution σ_k is the intersection of the admissible informed set and the set of points with distance smaller than R from σ_k :

$$X_{\hat{f},l} = \left\{ x \in X_{\hat{f}} \mid \min_{y \in \sigma_k} (\|x - y\|_2) < R \right\}. \quad (5)$$

Lemma 1 (local optimality of local sampling) Consider an asymptotically optimal path planner and let the sampling algorithm draw samples only from the local informed set. The planner converges to a local minimum with a probability equal to one.

Proof : If the current solution is not a local optimum, the intersection of the omniscient set and any neighborhoods of σ_k is not empty (c is Lipschitz):

$$X_f \cap X_{\hat{f},l} \neq \emptyset \quad \text{with } X_{\hat{f},l} \text{ neighborhood of } \sigma_k$$

It follows that local sampling improves the solution with a probability greater than zero whenever the solution is not (locally) optimal. \square

We define hereafter a mixed-strategy sampler to combine admissible and locally informed sampling soundly.

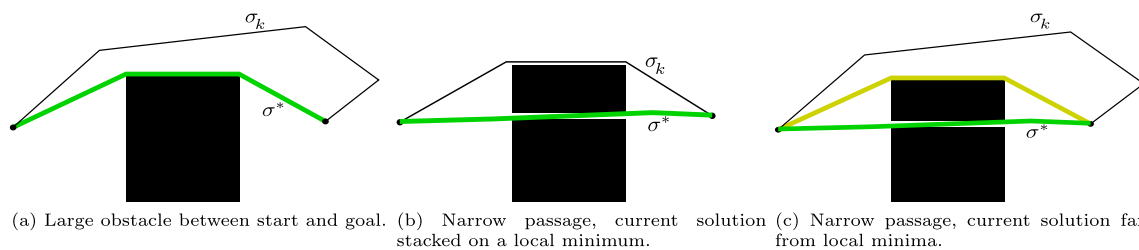


Fig. 1 Examples of planning situations where local sampling is useful (a), deleterious (b), or unable to find the global optimum but useful to reduce the measure of the informed set (c)

Definition 4 (*mixed-strategy sampler*) A *local sampler* and a *global sampler* are algorithms that draw samples from $X_{\hat{f},l}$ and $X_{\hat{f}}$, respectively. A *mixed-strategy sampler* draws samples by using a local sampler with probability ϕ and a global sampler with probability $1 - \phi$.

Remark 2 A mixed-strategy sampler is admissible if $\phi < 1$.

Lemma 2 (optimality of admissible mixed-strategy samplers) A *sampling-based path planner that is asymptotically optimal under uniform sampling distribution is asymptotically optimal also under admissible mixed-strategy sampling.*

Proof : A mixed-strategy sampler samples X_{free} with non-uniform probability density d . If such a sampler is admissible, d can be seen as a mixture of probability densities such that:

$$d = (1 - \phi) d_1 + \phi d_2$$

where d_1 is a strictly positive uniform probability density over X_{free} and

$$d_2 = \frac{d \lambda(X_{\text{free}}) - 1 + \phi}{\phi \lambda(X_{\text{free}})}.$$

Based on this consideration, the asymptotic optimality of the path planner traces back to the proof of asymptotic optimality of Janson et al. (2015) with non-uniform sampling. In particular, the planner is still asymptotically optimal by adjusting the rewire radius of a factor $(1 - \phi)^{-1/n}$, as proved in Appendix D of Janson et al. (2015). \square

At each iteration, the mixed-strategy sampler should select an appropriate value of ϕ based on the likelihood of improving the current solution. This is important to exploit the advantages of both admissible and local informed sampling (respectively, global asymptotic optimality and fast convergence to local optima) and mitigate the flaws (slow convergence speed and stagnation into local minima). We denote the guess that σ_k is not a local optimum at iteration k by $p_k \in [0, 1]$. If $c_k < c_{k-1}$, we increase p_{k+1} proportionally to the relative improvement of the cost such that:

$$p_{k+1} = \nu p_k + (1 - \nu) \frac{c_{k-1} - c_k}{c_{k-1} - u} \tag{6}$$

Algorithm 1: Mixed-strategy informed planner

```

Input:  $p_k, \sigma_{k-1}, c_{k-1}, \nu, R_0$ 
Output:  $p_{k+1}, \sigma_k, c_k$ 
1  $j = \text{rand.uniform}([0, 1]);$ 
2 if  $j < p_k$  then
3    $R = R_0(c_k - u);$ 
4    $x = \text{localSampling}(\sigma_{k-1}, R);$ 
5 else
6    $x = \text{informedSampling}(\sigma_{k-1}, c_k);$ 
7  $(\sigma_k, c_k) = \text{connectAndRewire}(x);$ 
8 if  $c_k < c_{k-1}$  then
9    $p_{k+1} = \nu p_k + (1 - \nu) \frac{c_{k-1} - c_k}{c_{k-1} - u}$ 
10 else
11    $p_{k+1} = \nu p_k$ 
    
```

where $\nu \in [0, 1)$ is a forgetting factor that smooths the evolution of p and u is an admissible estimate of the best cost c^* . Note that the cost c_k is non-increasing (namely, $c_k \leq c_{k-1}$), therefore p_k is a strictly positive number (assuming $p_0 > 0$). Moreover, $p_{k+1} \leq 1$ because $c_k \leq u$ and $p_0 \leq 1$.

It follows that a selector that uses $\phi = p_k$ is admissible.

5.2 Proposed algorithm

The proposed planner is the variant of Informed-RRT* in Algorithm 1. It uses the guess p_k as the probability to sample the local informed set (lines 1–6). Sample x is used to extend the tree (line 7); and p_{k+1} is updated according to (6) (lines 8 and 9).

Procedures `informedSampling` and `localSampling` sample the admissible informed set (2) and the local informed set (5), respectively. The former follows the implementation of Gammell et al. (2018), and the latter uses Algorithm 2.

Algorithm 2 randomly samples a ball of radius R centered at a random point along the current solution path. First, it uniformly samples the n -dimensional unit ball and assigns the value to b (lines 2–4). Then, it picks a random point, $\sigma(s)$, on path σ . Therefore, the final candidate sample is obtained by scaling b , from the unit ball to the ball of radius R and centered in $\sigma(s)$ (lines 5–6). Finally, it uses rejection to

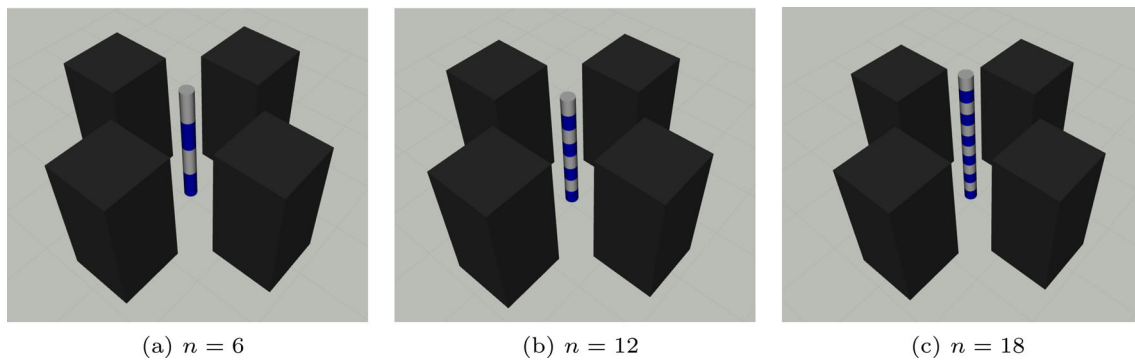


Fig. 2 Robot manipulator benchmark. n is the robot's number of degrees of freedom

Algorithm 2: localSampling procedure

Input: curve σ , radius R of the local informed set

Output: sample $x \in X_{f,l}$

```

1 repeat
2    $w = \text{rand.normal}(0, 1);$ 
3    $r = \text{rand.uniform}([0, 1]);$ 
4    $b = \left( \sqrt[n]{r} \frac{w}{\|w\|_2} \right);$ 
5    $s = \text{rand.uniform}([0, 1]);$ 
6    $x = \sigma(s) + Rb;$ 
7 until  $x \in X_{f,l};$ 

```

ensure that $x \in X_{f,l}$. Note that the rejection of the candidate is unlikely if R is small.

Algorithm 2 does not sample the local informed set uniformly. The points closer to the path have a higher probability of being sampled than points near the boundary of the tube. Moreover, Algorithm 2 over-samples regions “inside” the corners of the path. Non-uniform local sampling does not affect the asymptotic optimality of the planner (Lemma 2). Moreover, in minimum-length problems, over-sampling regions inside the corners may be beneficial in reducing the path length.

5.3 Algorithm tuning and convergence performance

Algorithm 1 has two parameters more than Informed-RRT*: the radius R_0 and the forgetting factor ν . Appendix Appendix A provides an illustrative example showing the effect of the parameters on the convergence. Summarizing the results, $R_0 \in [0.01, 0.02]$ and $\nu \approx 0.999$ consistently provide the best results across problems of different dimensionality and geometry.

6 Experiments

We test our Mixed-strategy Informed planner (MI-RRT*) with robot manipulators (6, 12, 18 degrees of freedom), nav-

igation of mobile manipulators, and a real manufacturing case study. We demonstrate that MI-RRT* consistently outperforms the baselines.

6.1 Robot manipulators

We consider three robotic cells (Fig. 2). Each cell has four rectangular obstacles and a serial manipulator (6, 12, and 18 degrees of freedom, respectively). The cell descriptions and usage examples are available at https://github.com/JRL-CARI-CNR-UNIBS/high_dof_snake_robots.

We compare our planner (MI-RRT*) with Informed-RRT* (Gammell et al., 2018), which uses a pure admissible informed sampling method, and wrapping-based Informed-RRT* (Wrap-RRT*) (Kim & Song, 2018), which applies a shortcutting procedure whenever it improves the solution. The additional parameters of MI-RRT* are tuned according to Sect. 5.3, namely $R = 0.02(c_k - u)$ and $\nu = 0.999$.

First of all, we show an example of a query to illustrate the behavior of the algorithms. Figure 3a shows the cost trend for a random planning query with $n = 6$, repeated 30 times for each planner. MI-RRT* provides a faster convergence rate and a smaller variance. Moreover, the median cost of the proposed algorithm is closer to the 10%-percentile than the other strategies, highlighting the capability of MI-RRT* to converge sooner to the global minimum. The same behavior is clear also for $n = 12$, as shown in Fig. 3b. In this case, Informed-RRT* suffers more from the curse of dimensionality, while Wrapping-based RRT* gets stuck in a local minimum for several iterations.

For an exhaustive comparison, we set up a benchmark as follows. Thirty queries are generated randomly (queries for which a direct connection between start and goal exists are discarded). The queries are solved with different planning times, between 0.5 and 5 s. Bounding the maximum planning time instead of the maximum number of iterations has been preferred because the algorithms perform different operations during the iterations. Moreover, planning time is more meaningful in practical applications.

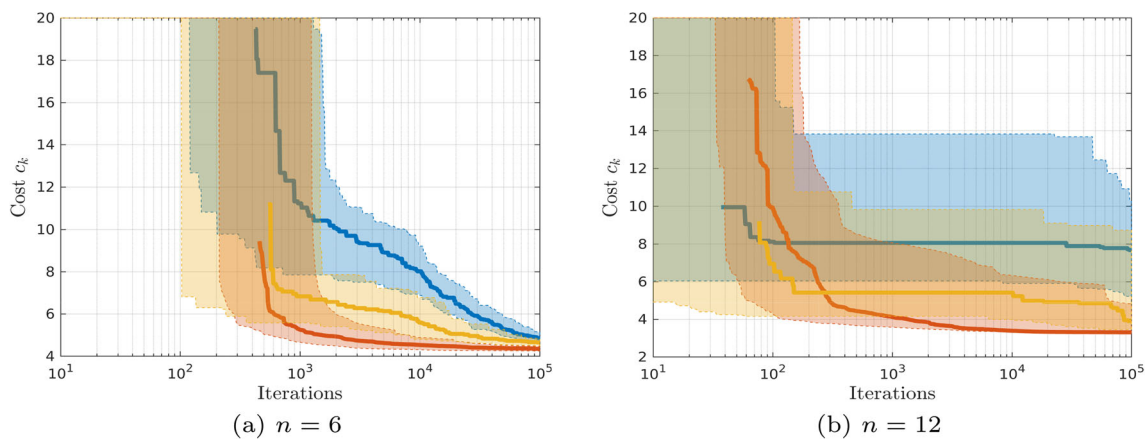


Fig. 3 Cost trends over iterations for the 6-dof and 12-dof manipulators for one query. Blue: Informed-RRT*; Orange: Wrap-RRT*; Red: Mixed-strategy RRT*. Solid lines denote the median; shaded regions denote the region between the 10% and 90% percentile

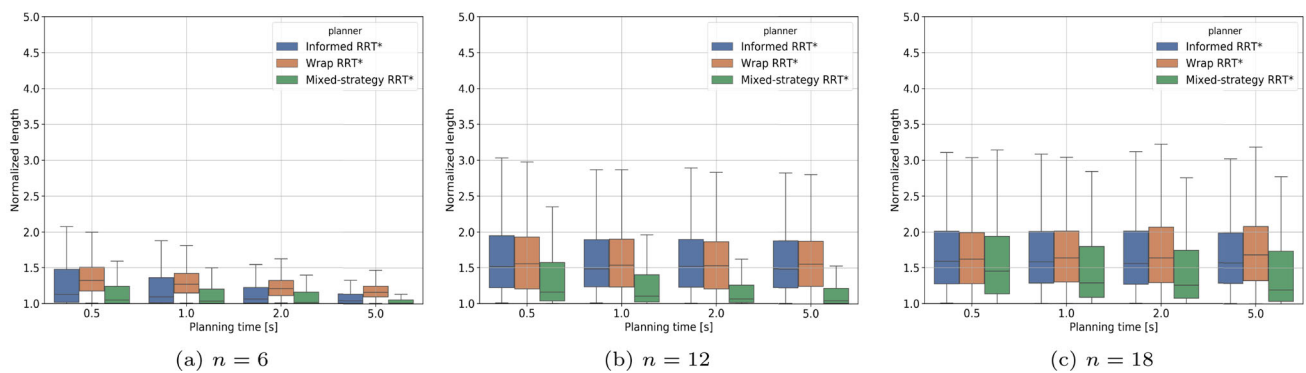


Fig. 4 Robot manipulators (6-, 12-, 18-dof): Cost trends over planning time for randomized queries

Each planner solves each query 30 times for maximum planning times equal to 0.5, 1.0, 2.0, 5.0 seconds. The final cost of each query is normalized by an estimate of the minimum cost, obtained by solving the query with a maximum planning time equal to 60 s.

The box-plots of Fig. 4 show that MI-RRT* has a faster convergence rate as well as a smaller variance compared to both Informed-RRT* and Wrap-RRT*. Therefore, the proposed approach finds better and more repeatable paths given the same amount of time. This result is emphasized for larger values of n , as shown in Fig. 4b, c.

We did not observe significant differences between Wrap-RRT* and Informed-RRT*, probably because the improvement of the convergence rate is counterbalanced by the computational overload owed to the wrapping procedures, as mentioned in Sect. 3.

6.2 Mobile manipulators

We consider navigation scenarios with mobile manipulators. Each robot consists of a 6-degree-of-freedom manipulator mounted on an omnidirectional mobile platform (two lin-

ear and one rotational degree of freedom). In the first case, the robot has to move from one side to the other of a wall with a narrow opening (see Fig. 5a). The problem has at least three homotopy classes: two circumnavigate the wall, and one passes through the narrow passage, requiring the re-configuration of the robot to fit the passage. We compare our MI-RRT* with Informed-RRT* (Gammell et al., 2018) and Wrap-RRT* (Kim & Song, 2018) over 30 repetitions. Results are in Fig. 5b: MI-RRT* has the best convergence rate, followed by Wrap-RRT*, and Informed-RRT*.

We also consider a second scenario with two mobile manipulators (for a total of 18 degrees of freedom) required to move from one side to the other of a wall with two openings. Results are in Fig. 5c: similarly to the single-robot case, MI-RRT* has the best convergence rate, followed by Wrap-RRT*, and Informed-RRT*, despite the greater number of iterations required by all methods to solve the problem.

6.3 Real-world case study

We validated our algorithm in a manufacturing mock-up cell designed within the EU-funded project *Sharework*. The

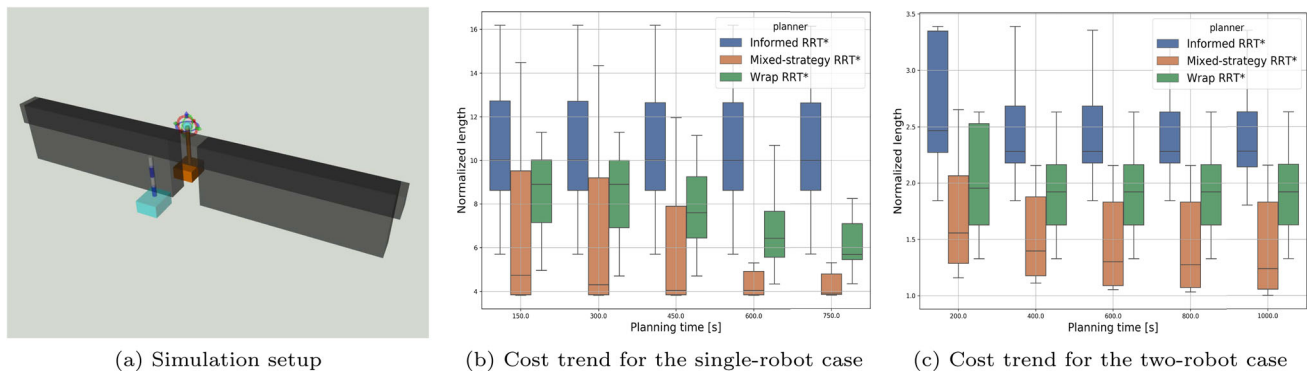


Fig. 5 Mobile manipulators: cost trends over planning time with one and two robots

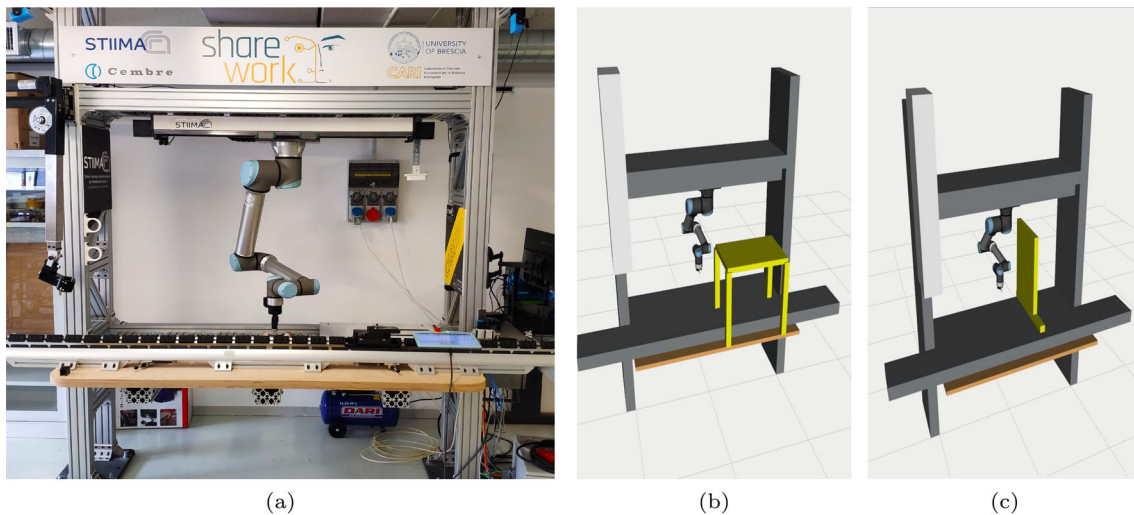


Fig. 6 Experimental setup. A Universal Robots UR10e mounted upside down works on the panel in front of it. **a** Actual setup; **b** first experiment: table-shaped obstacle upon placing position; **c** second experiment: barrier with a narrow passage between picking and placing positions

cell consists of a 6-degree-of-freedom collaborative robot, Universal Robots UR10e, mounted upside down and working on a work table in front of it (Fig. 6). The proposed motion planner is implemented in C++ within ROS/MoveIt! (Coleman et al., 2014). An open-source version of the code is available at https://github.com/JRL-CARI-CNR-UNIBS/cari_motion_planning. ROS/MoveIt! runs on an external computer from which it sends the planned trajectory to the robot controller.

The robot is tasked with a sequence of fifty pick-and-place operations. We consider two experiments. In the first one, a table-shaped obstacle is placed upon the placing goal (Fig. 6b). In the second one, a barrier separates the picking and placing goals (Fig. 6c). These scenarios simulate realistic machine-tending operations, in which the robot needs to access a confined space. From a planning perspective, they introduce narrow passages, complicating the planning problem. For example, in the barrier experiment, the shortest path passes through the narrow space below the barrier, close to the table surface.

Figure 7 compares the performance of MI-RRT*, Informed-RRT*, and Wrap-RRT* with different planning times, for the table and the barrier experiments. Similar to Sects. 6.1 and 6.2, MI-RRT* has a faster convergence speed in both experiments. In the table experiment, MI-RRT* reduces the planning time up to -34% and -13% compared to Informed-RRT* and Wrap-RRT*. In the barrier experiment, MI-RRT* reduces the planning time up to -37% and -18% compared to Informed-RRT* and Wrap-RRT*. Note that, contrary to the simulations, Wrap-RRT* showed a significant improvement compared to Informed-RRT*. This suggests that the advantages of Wrap-RRT* are problem-dependent.

Overall, MI-RRT* finds better solutions with the same maximum planning time. As a matter of example, Fig. 7c shows the continuous trend of the normalized costs for the barrier experiment. The key result is that MI-RRT* approaches the best cost faster in the initial phase. For example, after 10 s, MI-RRT* reaches $1.4c^*$, while Informed-RRT* and Wrap-RRT* reach $1.6c^*$ and $1.9c^*$, respectively;

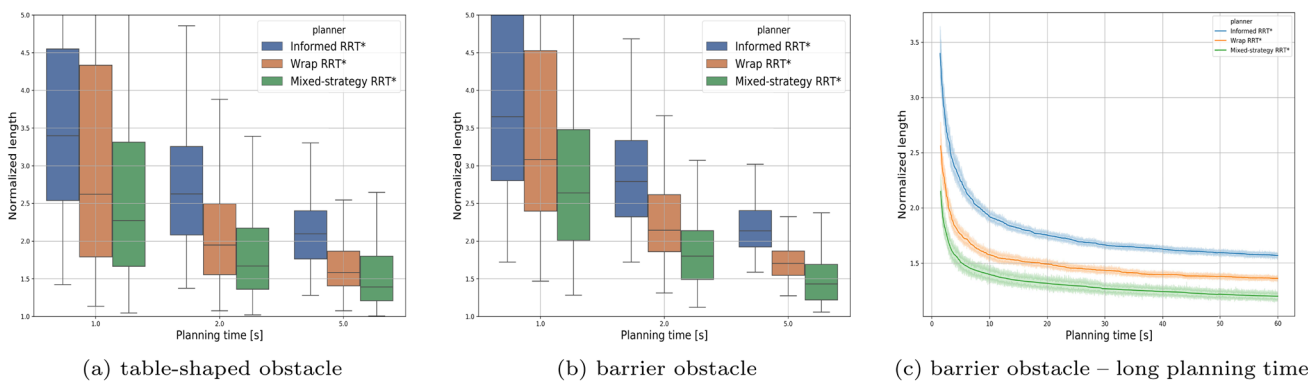


Fig. 7 Experimental results: cost trends over planning time

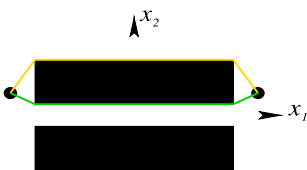


Fig. 8 Narrow-passage example of Sect. A.1 for $n = 2$. The planning problem has one global optimum (green line) and one local optimum (yellow line). For the sake of readability, axis scales are not equal

after 60s, MI-RRT* reaches $1.2c^*$, while Informed-RRT* and Wrap-RRT* reach $1.35c^*$ and $1.55c^*$, respectively.

7 Conclusions

Comparisons with state-of-the-art methods highlight the effectiveness of the proposed method in improving the convergence speed, especially in high-dimensional problems. The method is implemented in a manufacturing-oriented case study, where the robot is tasked with a sequence of pick-and-place operations. Results show that the proposed planner converges quicker to the optimal solution, allowing for shorter planning latencies in online applications.

An open-source implementation of the algorithm is available at https://github.com/JRL-CARI-CNR-UNIBS/cari_motion_planning. The algorithm is implemented in C++ and is fully compatible with ROS/MoveIt! (Coleman et al., 2014). Examples of usage and benchmarking are also available at https://github.com/JRL-CARI-CNR-UNIBS/high_dof_snake_robots.

Appendix A Effect of the tuning parameters

We analyze the effect of the parameters R_0 and ν used in Algorithm 1. To do so, we use an illustrative example consisting of a narrow-passage problem with one local minimum c_{local} and one global minimum c_{global} . Different cardinalities

of the configuration spaces are tested. We run 200 queries for each parameter set; each time, the algorithm runs for 10^6 iterations with an early stop condition if the cost c_k satisfies the condition $c_k < 1.01c_{global}$. Although this analysis is limited to an illustrative example, the results can serve as tuning guidelines for parameters R_0 and ν , as demonstrated in Sect. 6.

A.1 Narrow-passage example

We consider the configuration space

$$X = \{x \in X \subseteq \mathbb{R}^n \mid -5 \leq x_i \leq 5\} \tag{A1}$$

and an hollow hyper-spherindrical obstacle:

$$X_{obs} = \left\{ x \in X \mid |x_1| \leq \frac{l_c}{2}, r_{c1}^2 \leq \sum_{i=2}^n x_i^2 \leq r_{c2}^2 \right\} \tag{A2}$$

where $l_c = 1$ is length of the hyper-spherinder, $r_{c2} = 1$ is the external radius, and r_{c1} is the cavity radius. The cavity radius is such that the ratio between the volume of the cavity and that of the external cylinder is equal to 0.5 for all values of n . The starting and goal points are set equal to

$$x_{start} = [-0.6l_c, a, 0, \dots, 0], x_{goal} = [0.6l_c, a, 0, \dots, 0]$$

with

$$a = \frac{r_{c2} + 3r_{c1}}{4}$$

The problem has a local and a global minimum:

$$c_{local} = l_c + 2\sqrt{(0.1l_c)^2 + (r_{c1} - a)^2}$$

$$c_{global} = l_c + 2\sqrt{(0.1l_c)^2 + (r_{c2} - a)^2}$$

An example of the planning problem for $n = 2$ is in Fig. 8.

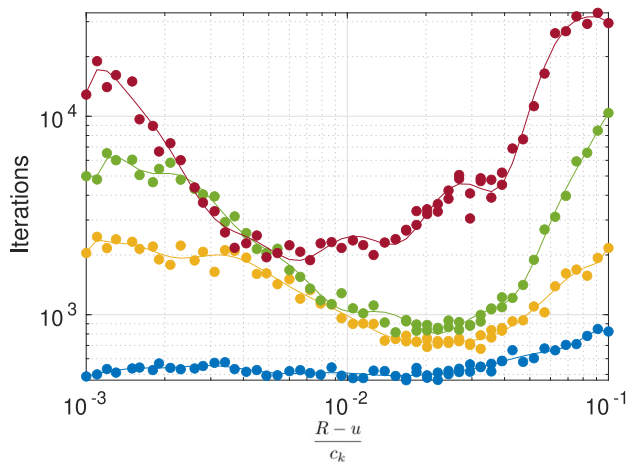
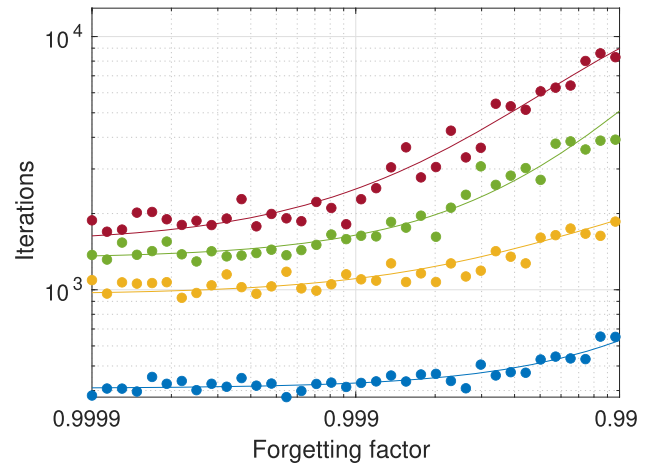
(a) Effect of radius R_0 (b) Effect of forgetting factor ν

Fig. 9 Effect of the parameters R_0 and ν . Iterations needed to reach $c_k = 1.01c_{\text{global}}$ (90%-percentile from 200 tests) for different values of R_0 and ν . Blue line: $n = 2$; orange line: $n = 3$; green line: $n = 4$, red line: $n = 7$

A.2 Effect of R_0

R_0 should be adequately small compared to the current cost. We run tests for $R_0 \in [10^{-3}, 10^{-1}]$ and $\nu = 0.999$. For each test, we count the number of iterations needed to reach $c_k \leq 1.01c_{\text{global}}$. The 90%-percentile, computed over 200 tests, is used as the performance index. Figure 9a shows the performance obtained for different values of $R_0 = \frac{R}{c_k - u}$ and n . Values around 0.02 provide the best results, while the local optimization is less effective with higher values. Smaller values of R_0 provide minimal improvements to the cost function.

A.3 Effect of forgetting factor ν

The forgetting factor allows smoothing the switching between the two sampling strategies by averaging out the cost changes over multiple iterations. Figure 9b shows the relation between the forgetting factor ν and the number of iterations required to reach $c_k = 1.01c_{\text{global}}$ (90%-percentile), the tube radius R_0 was set equal to 0.02 according to Sect. A.2. If $\nu > 0.999$, results do not vary significantly; however, selecting values of ν too close to 1 could lead the solver to get stuck in local minima for many iterations. $\nu = 0.999$ is a reasonable value for most cases in our experience.

Author Contributions M.F. and M.B. devised the methodology and wrote the main manuscript text. All authors contributed to implementing the software and conceiving the experiments. All authors read and reviewed the manuscript.

Funding This study was partially carried out within the MICS (Made in Italy-Circular and Sustainable) Extended Partnership and received

funding from Next-Generation EU (Italian PNRR-M4 C2, Invest 1.3-D.D. 1551.11-10-2022, PE00000004). CUPMICS D43C22003120001.

Declarations

Ethical Statement The authors declare that the following is fulfilled: (1) This material is the authors' original work, which has not been previously published elsewhere; (2) The paper is not currently being considered for publication elsewhere; (3) The paper reflects the authors' research and analysis truthfully and completely; (4) The paper properly credits the meaningful contributions of co-authors and co-researchers; (5) All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Choudhury, S., Gammell, J.D., Barfoot, T.D., Srinivasa, S.S., & Scherer, S. (2016). Regionally accelerated batch informed trees (RABIT*): A framework to integrate local information into optimal path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 4207–4214).
- Coleman, D., Şucan, I. A., Chitta, S., & Correll, N. (2014). Reducing the barrier to entry of complex robotic software: A MoveIt! case study. *Journal of Software Engineering for Robotics*, 5(1), 3–16.

- Faroni, M., & Berenson, D. (2023). Motion planning as online learning: A multi-armed bandit approach to kinodynamic sampling-based planning. *IEEE Robotics and Automation Letters*, 8(10), 6651–6658.
- Gammell, J. D., Srinivasa, S. S., & Barfoot, T. D. (2015). Batch Informed Trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 3067–3074).
- Gammell, J. D., Barfoot, T. D., & Srinivasa, S. S. (2018). Informed sampling for asymptotically optimal path planning. *IEEE Transactions on Robotics*, 34(4), 966–984.
- Ge, J., Sun, F., & Liu, C. (2016). RRT-GD: An efficient rapidly-exploring random tree approach with goal directionality for redundant manipulator path planning. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics* (pp. 1983–1988).
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107.
- Hauer, F., & Tsiotras, P. (2017). Deformable rapidly-exploring random trees. In *Robotics: Science and Systems*.
- Hsu, D., Latombe, J.-C., & Motwani, R. (1997). Path planning in expansive configuration spaces. In *Proceedings of International Conference on Robotics and Automation* (vol. 3, pp. 2719–2726).
- Janson, L., Schmerling, E., Clark, A., & Pavone, M. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International journal of robotics research*, 34(7), 883–921.
- Joshi, S. S. & Tsiotras, P. (2020). Relevant region exploration on general cost-maps for sampling-based motion planning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 6689–6695).
- Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894.
- Kavraki, L. E., Svestka, P., Latombe, J., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580.
- Kim, M.-C., & Song, J.-B. (2015). Informed RRT* towards optimality by reducing size of hyperellipsoid. In *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics* (pp. 244–248).
- Kim, M.-C., & Song, J.-B. (2018). Informed RRT* with improved converging rate by adopting wrapping procedure. *Intelligent Service Robotics*, 11, 53–60.
- Lai, T., Morere, P., Ramos, F., & Francis, G. (2020). Bayesian local sampling-based planning. *IEEE Robotics and Automation Letters*.
- LaValle, S. (1998). Rapidly-exploring random trees: a new tool for path planning. *The annual research report*.
- Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., & Thrun, S. (2008). Anytime search in dynamic graphs. *Artificial Intelligence*, 172(14), 1613–1643.
- Mandalika, A., Scalise, R., Hou, B., Choudhury, S., & Srinivasa, S. S. (2021). Guided incremental local densification for accelerated sampling-based motion planning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- MI-RRT*: a ROS-MoveIt! plugin. https://github.com/JRL-CARI-CNR-UNIBS/cari_motion_planning.
- Ratliff, N., Zucker, M., Bagnell, J. A., & Srinivasa, S. (2009). Chomp: Gradient optimization techniques for efficient motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 489–494).
- Robotic cell description for benchmarking of motion planners in MoveIt!. https://github.com/JRL-CARI-CNR-UNIBS/high_dof_snake_robots.
- Rodriguez, S., Thomas, S., Pearce, R., & Amato, N. M. (2008). Resampl: A region-sensitive adaptive motion planner. In *Algorithmic Foundation of Robotics VII* (pp. 285–300).
- Salzman, O., & Halperin, D. (2013). Asymptotically near-optimal RRT for fast, high-quality, motion planning. *IEEE Transactions on Robotics*, 32(3), 473–483.
- Santana Correia, A. D., Freire, E. O., Kamarry, S., Carvalho, É. Á. N., & Molina, L. (2018). The polarized RRT-edge approach. In *Proceedings of the Latin American Robotics Symposium* (pp. 277–282).
- Shan, Y. X., Li, B. J., Jian-Zhou, & Yue-Zhang, (2014). An approach to speed up RRT. In *Proceedings of the IEEE Intelligent Vehicles Symposium* (pp. 594–598).
- Strub, M. P., & Gammell, J. D. (2020). Adaptively informed trees (AIT*): Fast asymptotically optimal path planning through adaptive heuristics. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 3191–3198).
- Strub, M. P., & Gammell, J. D. (2020). Advanced BIT* (ABIT*)-sampling-based planning with advanced graph-search techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 130–136).
- Urmson, C., & Simmons, R. (2003). Approaches for heuristically biasing RRT growth. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (pp. 1178–1183).
- Yu, H., Lu, W., Liu, D., Han, Y., & Wu, Q. (2019). Speeding up Gaussian belief space planning for underwater robots through a covariance upper bound. *IEEE Access*, 7, 121961–121974.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.