Deterministic Sampling-Based Motion Planning: Optimality, Complexity, and Performance

Lucas Janson, Brian Ichter, and Marco Pavone

Abstract Probabilistic sampling-based algorithms, such as the probabilistic roadmap (PRM) and the rapidly-exploring random tree (RRT) algorithms, represent one of the most successful approaches to robotic motion planning, due to their strong theoretical properties (in terms of probabilistic completeness or even asymptotic optimality) and remarkable practical performance. Such algorithms are probabilistic in that they compute a path by connecting independently and identically distributed (i.i.d.) random points in the configuration space. Their randomization aspect, however, makes several tasks challenging, including certification for safety-critical applications and use of offline computation to improve real-time execution. Hence, an important open question is whether similar (or better) theoretical guarantees and practical performance could be obtained by considering deterministic, as opposed to random sampling sequences. The objective of this paper is to provide a rigorous answer to this question. The focus is on the PRM algorithm—our results, however, generalize to other batch-processing algorithms such as FMT*. Specifically, we first show that PRM, for a certain selection of tuning parameters and deterministic low-dispersion sampling sequences, is *deterministically* asymptotically optimal, i.e., it returns a path whose cost converges deterministically to the optimal one as the number of points goes to infinity. Second, we characterize the convergence rate, and we find that the factor of sub-optimality can be very explicitly upper-bounded

Lucas Janson
Department of Statistics, Stanford University, Stanford, CA 94305, e-mail: ljanson@stanford.edu

Brian Ichter
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, e-mail: ichter@stanford.edu

Marco Pavone
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, e-mail: pavone@stanford.edu

This work was supported by NASA under the Space Technology Research Grants Program, Grant NNX12AQ43G. Lucas Janson was partially supported by NIH training grant T32GM096982. Brian Ichter was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

in terms of the ℓ_2-dispersion of the sampling sequence and the connection radius of PRM. Third, we show that an asymptotically optimal version of PRM exists with computational and space complexity arbitrarily close to $O(n)$ (the theoretical lower bound), where n is the number of points in the sequence. This is in stark contrast to the $O(n \log n)$ complexity results for existing asymptotically-optimal probabilistic planners. Finally, through numerical experiments, we show that planning with deterministic low-dispersion sampling generally provides superior performance in terms of path cost and success rate.

1 Introduction

Probabilistic sampling-based algorithms represent a particularly successful approach to robotic motion planning problems [26, 15]. The key idea behind probabilistic sampling-based algorithms is to avoid the explicit construction of the configuration space (which can be prohibitive in complex planning problems) and instead conduct a search that probabilistically probes the configuration space with independently and identically distributed (i.i.d.) random samples. This probing is enabled by a collision detection module, which the motion planning algorithm considers as a "black box" [15]. Examples, roughly in chronological order, include the probabilistic roadmap algorithm (PRM) [14], expansive space trees (EST) [10, 21], Lazy-PRM [4], the rapidly exploring random trees algorithm (RRT) [16], sampling-based roadmap of trees (SRT) [23], rapidly-exploring roadmap [1], PRM* and RRT* [13], RRT$^\#$ [2], and the fast marching tree algorithm (FMT*) [12]. A central result is that these algorithms provide *probabilistic completeness* guarantees in the sense that the probability that the planner fails to return a solution, if one exists, decays to zero as the number of samples approaches infinity [3]. Recently, it has been proven that RRT*, PRM*, RRT$^\#$, and FMT* are asymptotically optimal, i.e., the cost of the returned solution converges almost surely to the optimum [13, 2, 12].

It is natural to wonder whether the theoretical guarantees and practical performance of sampling-based algorithms would hold if these algorithms were to be de-randomized, i.e., run on a *deterministic*, as opposed to random sampling sequence. This is an important question, as de-randomized planners would significantly simplify the certification process (as needed for safety-critical applications), enable the use of offline computation (particularly important for planning under differential constraints or in high dimensional spaces—exactly the regime for which sampling-based planners are designed), and, in the case of lattice sequences, drastically simplify a number of operations (e.g., locating nearby samples). This question has received relatively little attention in the literature. Specifically, previous research [5, 17, 11] has focused on the performance of de-randomized versions of sampling-based planners in terms of convergence to feasible paths. A number of deterministic variants of the PRM algorithm were shown to be resolution complete (i.e., provably converging to a feasible solution as $n \to \infty$) and, perhaps surprisingly, offer superior performance on an extensive set of numerical experiments [5, 17]. Prompted by these results, a number of deterministic low-dispersion, incremental sequences have been designed specifically tailored to motion planning problems [27, 18, 28].

The results in [5, 17, 11] are restricted to convergence to feasible, as opposed to *optimal* paths. Several questions are still open. Are there advantages of i.i.d. sampling in terms of convergence to an optimal path? Can convergence rate guarantees for the case of deterministic sampling be provided, similar to what is done for prob-

abilistic planners in [12, 8]? For a given number of samples, are there advantages in terms of computational and space complexity? The objective of this paper is to rigorously address these questions. Our focus is on the PRM algorithm. However, our results extend to most of the existing batch (i.e., not anytime) algorithms, including Lazy-PRM and FMT*.

Statement of Contributions: The contributions of this paper are as follows.

Deterministic asymptotic optimality of sampling-based planning: We show that the PRM algorithm is asymptotically optimal when run on *deterministic* sampling sequences whose ℓ_2-dispersion is upper-bounded by $\gamma n^{-1/d}$, for some $\gamma \in \mathbb{R}_{>0}$ (we refer to such sequences as deterministic low-dispersion sequences), and with a connection radius $r_n \in \omega(n^{-1/d})$[1]. In other words, the cost of the solution computed over n samples converges deterministically to the optimum as $n \to \infty$. As a comparison, the analogue result for the case of i.i.d. random sampling holds almost surely [13, 12] (as opposed to deterministically) and requires a connection radius $\Omega\left((\log(n)/n)^{1/d}\right)$, i.e., bigger.

Convergence rate: We show that, in the absence of obstacles, the factor of suboptimality of PRM is upper-bounded by $2D_n/(r_n - 2D_n)$, where D_n is the ℓ_2-dispersion of the sampling sequence. A slightly more sophisticated result holds for the obstacle-cluttered case. As a comparison, the analogue result for the case of i.i.d. sampling only holds in probability and is much more involved (and less interpretable) [12]. Our results could be instrumental to the certification of sampling-based planners.

Computational and space complexity: We prove that PRM, when run on a low-dispersion lattice, has computational and space complexity $O(n^2 r_n^d)$. As asymptotic optimality can be obtained using $r_n \in \omega(n^{-1/d})$, there exists an asymptotically optimal version of PRM with computational and space complexity $\omega(n)$, where $O(n)$ represents the theoretical lower bound (as, at the very least, n operations need to be carried out to load samples into memory). As a comparison, the analogous complexity results for the case of i.i.d. sampling are of order $O(n \log(n))$ [13]. We also extend our complexity analysis to non-lattice deterministic sampling sequences.

Experimental performance: Finally, we compare performance (in terms of path cost) of deterministic low-dispersion sampling versus i.i.d. sampling on a variety of test cases ranging from two to six dimensions and from simple sets of rectangular obstacles to rather complicated mazes. In all our examples, for a given number of samples, deterministic low-dispersion sampling performs no worse and sometimes substantially better than i.i.d. sampling (this is not even accounting for the potential significant speed-ups in runtime, e.g., due to fast nearest-neighbor indexing).

The key insight behind our theoretical results (e.g., smaller required connection radius, better complexity, etc.) is the factor difference in dispersion between deterministic low-dispersion sequences versus i.i.d. sequences, namely $O(n^{-1/d})$ versus $O((\log n)^{1/d} n^{-1/d})$ [7, 19]. Interestingly, the same $O(n^{-1/d})$ dispersion can be

[1] For $f, g : \mathbb{N} \to \mathbb{R}$, we say $f \in O(g)$ if there exists $n_0 \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$ such that $|f(n)| \leq k|g(n)|$ for all $n \geq n_0$. We say $f \in \Omega(g)$ if there exists $n_0 \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$ such that $|f(n)| \geq k|g(n)|$ for all $n \geq n_0$. Finally, we say $f \in \omega(g)$ if $\lim_{n\to\infty} f(n)/g(n) = \infty$.

achieved with non-i.i.d. *random* sequences, e.g., randomly rotated and offset lattices. As we will show, these sequences enjoy the same *deterministic* performance guarantees of deterministic low-dispersion sequences and retain many of the benefits of deterministic sampling (e.g., fast nearest-neighbor indexing). Additionally, their "controlled" randomness may allow them to address some potential issues with deterministic sequences (in particular lattices), e.g., avoiding axis-alignment issues in which entire rows of samples may become infeasible due to alignment along an obstacle boundary. In this perspective, achieving deterministic guarantees is really a matter of i.i.d. sampling versus non-i.i.d., low-dispersion sampling (with deterministic sampling as a prominent case), as opposed to random versus deterministic. Collectively, our results, complementing and corroborating those in [5, 17], strongly suggest that both the study and application of sampling-based algorithms should adopt non-i.i.d. low-dispersion sampling. From a different viewpoint, our results provide a theoretical bridge between sampling-based algorithms with i.i.d. sampling and non-sampling-based algorithms on regular grids (e.g., D* [24] and related kinodynamic variants [22]).

Organization: This paper is structured as follows. In Section 2 we provide a review of known concepts from low-dispersion sampling, with a focus on ℓ_2-dispersion. In Section 3 we formally define the optimal path planning problem. In Section 4 we present our three main theoretical results for planning with low-dispersion sequences: asymptotic optimality, convergence rate, and computational and space complexity. In Section 5 we present results from numerical experiments supporting our statements. Finally, in Section 6, we draw some conclusions and discuss directions for future work.

2 Background

A key characteristic of any set of points on a finite domain is its ℓ_2-dispersion. This concept will be particularly useful in elucidating the advantages of deterministic sampling over i.i.d. sampling. As such, in this section we review some relevant properties and results on the ℓ_2-dispersion.

Definition 1 (ℓ_2-dispersion). For a finite, nonempty set S of points contained in a d-dimensional compact Euclidean subspace $\mathscr{X}$ with positive Lebesgue measure, its ℓ_2-dispersion $D(S)$ is defined as

$$D(S) := \sup_{x\in\mathscr{X}} \min_{s\in S} \|s-x\|_2 = \sup\{r>0 : \exists x \in \mathscr{X} \text{ with } B(x,r)\cap S = \emptyset\}, \quad (1)$$

where $B(x,r)$ is the *open* ball of radius r centered at x.

Intuitively, the ℓ_2-dispersion quantifies how well a space is covered by a set of points S in terms of the largest open Euclidean ball that touches none of the points. The quantity $D(S)$ is important in the analysis of path optimality as an optimal path may pass through an empty ball of radius $D(S)$. Hence, $D(S)$ bounds how closely any path tracing through points in S can possibly approximate that optimal path.

The ℓ_2-dispersion of a set of deterministic or random points is often hard to compute, but luckily it can be bounded by the more-analytically-tractable ℓ_∞-dispersion. The ℓ_∞-dispersion is defined by simply replacing the ℓ_2-norm in equation (1) by the ℓ_∞-norm, or max-norm. The ℓ_∞-dispersion of a set S, which we will denote by $D_\infty(S)$, is related to the ℓ_2-dispersion in d dimensions by [19],

$$D_\infty(S) \leq D(S) \leq \sqrt{d} D_\infty(S),$$

which allows us to bound $D(S)$ when $D_\infty(S)$ is easier to compute. In particular, an important result due to [7] is that the ℓ_∞-dispersion of n independent uniformly sampled points on $[0,1]^d$ is $O((\log(n)/n)^{1/d})$ with probability 1. Corollary to this is that the ℓ_2-dispersion is also $O((\log(n)/n)^{1/d})$ with probability 1.

Remarkably, there are deterministic sequences with ℓ_2-dispersions of order $O(n^{-1/d})$, an improvement by a factor $\log(n)^{1/d}$. (Strictly speaking, one should distinguish point sets, where the number of points is specified in advance, from sequences [17]—in this paper we will simply refer to both as "sequences.") For instance, the Sukharev sequence [25], whereby $[0,1]^d$ is gridded into $n = k^d$ hypercubes and their centers are taken as the sampled points, can easily be shown to have ℓ_2-dispersion of $(\sqrt{d}/2)n^{-1/d}$ for $n = k^d$ points. As we will see in Section 4, the use of sample sequences with lower ℓ_2-dispersions confers on PRM a number of beneficial properties, thus justifying the use of certain deterministic sequences instead of i.i.d. ones. In the remainder of the paper, we will refer to sequences with ℓ_2-dispersion of order $O(n^{-1/d})$ as *low-dispersion* sequences. A natural question to ask is whether we can use a sequence that *minimizes* the ℓ_2-dispersion. Unfortunately, such an optimal sequence is only known for $d = 2$, in which case it is represented by the centers of the equilateral triangle tiling [15]. In this paper, we will focus on the Sukharev [25] and Halton sequences [9], except in two dimensions when we will consider the triangular lattice as well, but there are many other deterministic sequences with ℓ_2-dispersion of order $O(n^{-1/d})$; see [27, 18, 28] for other examples.

3 Problem Statement

The problem formulation follows very closely the problem formulation in [12]. Let $\mathscr{X} = [0,1]^d$ be the configuration space, where $d \in \mathbb{N}$. Let $\mathscr{X}_{\text{obs}}$ be a closed set representing the obstacles, and let $\mathscr{X}_{\text{free}} = \text{cl}(\mathscr{X} \setminus \mathscr{X}_{\text{obs}})$ be the obstacle-free space, where $\text{cl}(\cdot)$ denotes the closure of a set. The initial condition is $x_{\text{init}} \in \mathscr{X}_{\text{free}}$, and the goal region is $\mathscr{X}_{\text{goal}} \subset \mathscr{X}_{\text{free}}$. A specific path planning problem is characterized by a triplet $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$. A function $\sigma : [0,1] \to \mathbb{R}^d$ is a *path* if it is continuous and has bounded variation. If $\sigma(\tau) \in \mathscr{X}_{\text{free}}$ for all $\tau \in [0,1]$, σ is said to be *collision-free*. Finally, if σ is collision-free, $\sigma(0) = x_{\text{init}}$, and $\sigma(1) \in cl(\mathscr{X}_{\text{goal}})$, then σ is said to be a *feasible path* for the planning problem $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$.

The goal region $\mathscr{X}_{\text{goal}}$ is said to be *regular* if there exists $\xi > 0$ such that $\forall y \in \partial\mathscr{X}_{\text{goal}}$, there exists $z \in \mathscr{X}_{\text{goal}}$ with $B(z;\xi) \subseteq \mathscr{X}_{\text{goal}}$ and $y \in \partial B(z;\xi)$ (the notation $\partial\mathscr{X}$ denotes the boundary of set $\mathscr{X}$). Intuitively, a regular goal region is a smooth set with a boundary that has bounded curvature. Furthermore, we will say $\mathscr{X}_{\text{goal}}$ is ξ-regular if $\mathscr{X}_{\text{goal}}$ is regular for the parameter ξ. Denote the set of all paths by Σ. A cost function for the planning problem $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$ is a function $c : \Sigma \to \mathbb{R}_{\geq 0}$; in this paper we will focus on the *arc length* function. The optimal path planning problem is then defined as follows:

> **Optimal path planning problem**: Given a path planning problem $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$ with a regular goal region and the arc length function $c : \Sigma \to \mathbb{R}_{\geq 0}$, find a feasible path σ^* such that $c(\sigma^*) = \min\{c(\sigma) : \sigma \text{ is feasible}\}$. If no such path exists, report failure.

A path planning problem can be arbitrarily difficult if the solution traces through a narrow corridor, which motivates the standard notion of path clearance [13]. For a

given $\delta > 0$, define the δ-interior of $\mathscr{X}_{\text{free}}$ as the set of all configurations that are at least a distance δ from $\mathscr{X}_{\text{obs}}$. Then a path is said to have strong δ-clearance if it lies entirely inside the δ-interior of $\mathscr{X}_{\text{free}}$. Further, a path planning problem with optimal path cost c^* is called δ-robustly feasible if there exists a strictly positive sequence $\delta_n \to 0$, and a sequence $\{\sigma_n\}_{i=1}^n$ of feasible paths such that $\lim_{n\to\infty} c(\sigma_n) = c^*$ and for all $n \in \mathbb{N}$, σ_n has strong δ_n-clearance, $\sigma_n(1) \in \partial\mathscr{X}_{\text{goal}}$, and $\sigma_n(\tau) \notin \mathscr{X}_{\text{goal}}$ for all $\tau \in (0,1)$.

Lastly, in this paper we will be considering a generic form of the PRM algorithm. That is, denote by gPRM (for generic PRM) the algorithm given by Algorithm 1. The function `SampleFree`(n) is a function that returns a set of $n \in \mathbb{N}$ points in $\mathscr{X}_{\text{free}}$. Given a set of samples V, a sample $v \in V$, and a positive number r, `Near`(V,v,r) is a function that returns the set of samples $\{u \in V : \|u-v\|_2 < r\}$. Given two samples $u,v \in V$, `CollisionFree`(u,v) denotes the boolean function which is true if and only if the line joining u and v does not intersect an obstacle. Given a graph (V,E), where the node set V contains x_{init} and E is the edge set, `ShortestPath`(x_{init},V,E) is a function returning a shortest path from x_{init} to $\mathscr{X}_{\text{goal}}$ in the graph (V,E) (if one exists, otherwise it reports failure). Deliberately, we do not specify the definition of `SampleFree` and have left r_n unspecified, thus allowing for any sequence of points—deterministic or random—to be used, with any connection radius. These "tuning" choices will be studied in Section 4. We want to clarify that we are in no way proposing a new algorithm, but just defining an umbrella term for the PRM class of algorithms which includes, for instance, sPRM and PRM* as defined in [13].

Algorithm 1 gPRM Algorithm

```
1  V ← {x_init} ∪ SampleFree(n); E ← ∅
2  for all v ∈ V do
3     X_near ← Near(V\{v}, v, r_n)
4     for x ∈ X_near do
5        if CollisionFree(v,x) then
6           E ← E ∪ {(v,x)} ∪ {(x,v)}
7        end if
8     end for
9  end for
10 return ShortestPath(x_init, V, E)
```

4 Theoretical Results

In this section we present our main theoretical results. We begin by proving that gPRM on low-dispersion sequences is asymptotically optimal, in the deterministic sense, for connection radius $r_n \in \omega(n^{-1/d})$. Previous work has required r_n to be at least $\Omega((\log(n)/n)^{1/d})$ for asymptotic optimality.

Theorem 1 (Asymptotic optimality with deterministic sampling). *Let $(\mathscr{X}_{free}, x_{init}, \mathscr{X}_{goal})$ be a δ-robustly feasible path planning problem in d dimensions, with $\delta > 0$ and $\mathscr{X}_{goal}$ ξ-regular. Let c^* denote the arc length of an optimal path σ^*, and let c_n denote the arc length of the path returned by gPRM (or ∞ if gPRM returns failure) with n vertices whose ℓ_2-dispersion is $D(V)$ using a radius r_n. Then if $D(V) \leq \gamma n^{-1/d}$ for some $\gamma \in \mathbb{R}$ and*

$$n^{1/d} r_n \to \infty, \tag{2}$$

then $\lim_{n\to\infty} c_n = c^*$.

Proof. Fix $\varepsilon > 0$. By the δ-robust feasibility of the problem, there exists a σ_ε such that $c(\sigma_\varepsilon) \leq (1+\varepsilon/3)c^*$ and σ_ε has strong δ_ε-clearance for some $\delta_\varepsilon > 0$, see Figure 1(a). Let R_n be a sequence such that $R_n \leq r_n$, $n^{1/d}R_n \to \infty$, and $R_n \to 0$, guaran-

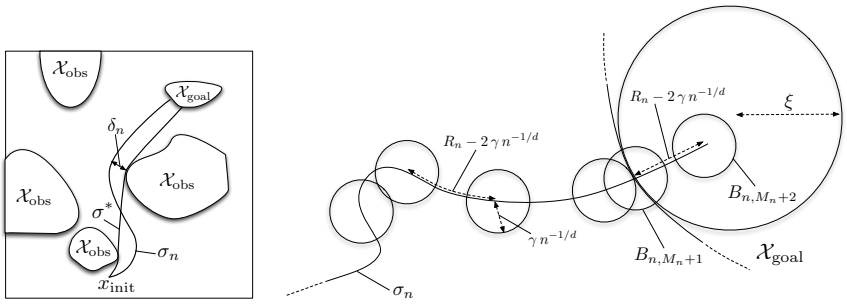


Fig. 1 Figure 1(a): Illustration in 2D of σ_n as the shortest strongly δ_n-robust feasible path, as compared to the optimal path σ^*, as used in the proof of Theorem 1. Figure 1(b): Illustration in 2D of the construction of $B_1, \ldots, B_{M_n+2}$ in the proof of Theorem 1.

teeing that there exists a $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,

$$(4+6/\varepsilon)\gamma n^{-1/d} \leq R_n \leq \min\{\delta_\varepsilon, \xi, c^*\varepsilon/6\}. \tag{3}$$

For any $n \geq n_0$, construct the closed balls $B_{n,m}$ such that $B_{n,i}$ has radius $\gamma n^{-1/d}$ and has center given by tracing a distance $(R_n - 2\gamma n^{-1/d})i$ from x_0 along σ_ε (this distance is positive by Equation (3)) until $(R_n - 2\gamma n^{-1/d})i > c(\sigma_\varepsilon)$. This will generate $M_n = \lfloor c(\sigma_\varepsilon)/(R_n - 2\gamma n^{-1/d})\rfloor$ balls. Define B_{n,M_n+1} to also have radius $\gamma n^{-1/d}$ but center given by the point where σ_ε meets $\mathscr{X}_{\text{goal}}$. Finally, define B_{n,M_n+2} to have radius $\gamma n^{-1/d}$ and center defined by extending the center of B_{n,M_n+1} into $\mathscr{X}_{\text{goal}}$ by a distance $R_n - 2\gamma n^{-1/d}$ in the direction perpendicular to $\partial\mathscr{X}_{\text{goal}}$. Note that by Equation (3), $B_{n,M_n+2} \subset \mathscr{X}_{\text{goal}}$. See Figure 1(b) for an illustration.

Since the dispersion matches the radii of all the $B_{n,m}$, each $B_{n,m}$ has at least one sampled point within it. Label these points $x_1, \ldots, x_{M_n+2}$, with the subscripts matching their respective balls of containment. For notational convenience, define $x_0 := x_{\text{init}}$. Note that by construction of the balls, for $i \in \{0, \ldots, M_n+1\}$, each pair of consecutively indexed points (x_i, x_{i+1}) is separated by no more than $R_n \leq r_n$. Furthermore, since $R_n \leq \delta_\varepsilon$ by Equation (3) above, there cannot be an obstacle between any such pair, and thus each pair constitutes an edge in the gPRM graph. Thus, we can upper-bound the cost c_n of the gPRM solution by the sum of the lengths of the edges $(x_0, x_1), \ldots, (x_{M_n+1}, x_{M_n+2})$:

$$c_n \leq \sum_{i=0}^{M_n+1} \|x_{i+1}-x_i\| \leq (M_n+2)R_n \leq \frac{c(\sigma_\varepsilon)}{R_n-2\gamma n^{-1/d}}R_n + 2R_n$$
$$\leq c(\sigma_\varepsilon) + \frac{2\gamma n^{-1/d}}{R_n-2\gamma n^{-1/d}}c(\sigma_\varepsilon) + 2R_n = c(\sigma_\varepsilon) + \frac{1}{\frac{R_n}{2\gamma n^{-1/d}}-1}c(\sigma_\varepsilon) + 2R_n$$
$$\leq (1+\frac{\varepsilon}{3})c^* + \frac{1}{\frac{3}{\varepsilon}+1}(1+\frac{\varepsilon}{3})c^* + \frac{\varepsilon}{3}c^* = (1+\varepsilon)c^*.$$

The second inequality follows from the fact that the distance between x_i and x_{i+1} is upper-bounded by the distance between the centers of $B_{n,i}$ and $B_{n,i+1}$ (which is at most $R_n - 2\gamma n^{-1/d}$) plus the sum of their radii (which is $2\gamma n^{-1/d}$). The last inequality follows from the facts that $c(\sigma_\varepsilon) \leq (1+\varepsilon/3)c^*$ and Equation (3). □

Note that if gPRM using $r_n > 2D(V)$ reports failure, then there are two possibilities: (i) a solution does not exist, or (ii) all solution paths go through corridors whose widths are smaller than $2D(V)$. Such a result can be quite useful in practice, as solutions going through narrow corridors could be undesirable anyways (see [17, Section 5] for the same conclusion).

Next, we relate the solution cost returned by gPRM to the best cost of a path with strong δ-clearance in terms of the ℓ_2-dispersion of the samples used. This is a generalization of previous convergence rates, e.g. [12], which only apply to obstacle-free spaces. Previous work also defined convergence rate as, for a fixed level of suboptimality ε, the rate (in n) that the probability of returning a greater-than-ε-suboptimal solution goes to zero. In contrast, we compute the rate (in n) that solution suboptimality approaches zero. Lastly, previous work focused on asymptotic rates in big-O notation, while here we provide exact upper-bounds for finite samples.

Theorem 2 (Convergence rate in terms of dispersion). *Consider the simplified problem of finding the shortest feasible path between two points x_0 and x_f in $\mathcal{X}_{free}$, assuming that both the initial point and final point have already been sampled. Define*

$$\delta_{max} = \sup\{\delta > 0 : \exists \text{ a feasible } \sigma \in \Sigma \text{ with strong } \delta\text{-clearance}\}$$

and assume $\delta_{max} > 0$. For all $\delta < \delta_{max}$, let $c^{(\delta)}$ be the cost of the shortest path with strong δ-clearance. Let c_n be the length of the path returned by running gPRM on n points whose ℓ_2-dispersion is D_n and using a connection radius r_n. Then for all n such that $r_n > 2D_n$ and $r_n < \delta$,

$$c_n \leq \left(1 + \frac{2D_n}{r_n - 2D_n}\right) c^{(\delta)}. \tag{4}$$

Proof. Let σ_δ be a feasible path of length $c^{(\delta)}$ with strong δ-clearance. Construct the balls $B_1, \ldots, B_{M_n}$ with centers along σ_δ as in the proof of Theorem 1 (note we are not constructing B_{M_n+1} or B_{M_n+2}), except with radii D_n and centers separated by a segment of σ_δ of arc-length $r_n - 2D_n$. Note that $M_n = \lfloor c^{(\delta)}/(r_n - 2D_n) \rfloor$. Then by definition each B_i contains at least one point x_i. Furthermore, each x_i is connected to x_{i-1} in the gPRM graph (because x_i is contained in the ball of radius $r_n - D_n$ centered at x_{i-1}, and that ball is collision-free), and x_f is connected to x_{M_n} as well. Thus c_n is upper-bounded by the path tracing through $x_0, x_1, \ldots, x_{M_n}, x_f$:

$$c_n \leq \|x_1 - x_0\| + \sum_{i=2}^{M_n} \|x_i - x_{i-1}\| + \|x_f - x_{M_n}\| \leq r_n - D_n + \sum_{i=2}^{M_n} r_n + \|x_f - x_{M_n}\|$$
$$\leq (M_n r_n - D_n) + \left(\left(\frac{c^{(\delta)}}{r_n - 2D_n} - \left\lfloor \frac{c^{(\delta)}}{r_n - 2D_n} \right\rfloor \right) (r_n - 2D_n) + D_n \right)$$
$$= c^{(\delta)} + 2D_n M_n \leq \left(1 + \frac{2D_n}{r_n - 2D_n} \right) c^{(\delta)},$$

where the second and third inequalities follow by considering the farthest possible distance between neighboring points, given their inclusion in their respective balls. □

Remark 1 (Convergence rate in obstacle-free environments). Note that when there are no obstacles, $\delta_{\max} = \infty$ and $c^{(\delta)} = \|x_f - x_0\|$ for all $\delta > 0$. Therefore, an immediate corollary of Theorem 2 is that the convergence rate in free space of gPRM to the *optimal* solution is upper-bounded by $2D_n/(r_n - 2D_n)$ for $r_n > 2D_n$.

Remark 2 (Practical use of convergence rate). Theorem 2 provides a convergence rate result to a shortest path with strong δ-clearance. This result is useful for two main reasons. First, in practice, the objective of path planning is often to find a high-quality path with some "buffer distance" from the obstacles, which is precisely captured by the notion of δ-clearance. Second, the convergence rate in Equation (4) could be used to certify the performance of gPRM (and related batch planners) by placing some assumptions on the obstacle set (e.g., minimum separation distance among obstacles and/or curvature of their boundaries)—this is an interesting avenue for future research.

Both the asymptotic optimality and convergence rate results can be extended to other batch planners such as Lazy-PRM or FMT*. This is however omitted due to space limitations.

Lastly, we show that using a low-ℓ_2-dispersion lattice sample set, an asymptotically-optimal (AO) version of gPRM can be run that has lower-order computational complexity than any existing AO algorithm, namely $\omega(n)$ instead of $O(n\log(n))$.

Theorem 3 (Computational complexity with deterministic sampling). *gPRM run on n samples arranged in a d-dimensional cubic lattice with connection radius r_n satisfying Equation 2 has computational complexity and space complexity*

$$O(n^2 r_n^d). \tag{5}$$

Proof. The algorithm gPRM has three steps: (1) For each sampled point x, it needs to compute which other sampled points are within a distance r_n of x. (2) For each pair of sampled points within r_n of one another, their connecting edge needs to be checked for collision, and if collision-free, its edge-length needs to be computed. (3) The shortest path through the graph produced in steps (1) and (2) from the initial point to the goal region needs to be computed.

The lattice structure makes it trivially easy to bound a point's r_n-neighborhood by a bounding hypercube with side-length $2r_n$, ensuring only $O(nr_n^d)$ nearby points need to be checked for each of the n samples, so this step takes $O(n^2 r_n^d)$ time.

In step (2), one collision-check and at most one cost computation needs to be performed for each pair of points found in step (1) to be within r_n of one another. The number of such pairs can be bounded above by the number of sampled points times the size of each one's neighborhood, leading to a bound of the form $O(n \cdot nr_n^d)$. Thus step (2) takes $O(n^2 r_n^d)$ time.

After steps (1) and (2), a weighted (weights correspond to edge lengths) graph has been constructed on n vertices with a number of edges asymptotically upper-bounded by $n^2 r_n^d$. One more property of this graph, because it is on the cubic lattice, is that the number of distinct edge lengths is asymptotically upper-bounded by nr_n^d. An implementation of Dijkstra's algorithm for the single source shortest path problem is presented in [20] with running time linear in both the number of edges and the number of vertices times the number of distinct edge lengths. Since both are $O(n^2 r_n^d)$, that is the time complexity of step (3). □

Since Theorem 1 allows $r_n \in \omega(n^{-1/d})$ while maintaining AO, Theorem 3 implies that cubic-lattice sampling allows for an AO algorithm with computational and space complexity $\omega(n)$. All other AO algorithms in the literature have computational and space complexity at least $O(n\log(n))$. While the use of an $r_n \in \omega(n^{-1/d})$ makes the graph construction phase (steps (1) and (2)) $\omega(n)$, step (3) would in general take longer, as shortest-path algorithms on a general graph with n vertices requires $\Omega(n\log(n))$. Thus the lattice structure must be leveraged to improve the complexity of step (3). We note, however, that in practice the shortest-path algorithm is typically a trivial fraction of path planning runtime, as it requires none of the much-more-complex edge-cost and collision-free computations. If we ignore this component of the runtime, the result of Theorem 3 can be extended to other PRM-like algorithms such as FMT*, non-lattice sampling schemes such as the Halton/Hammersley sequence, and a k-nearest-neighbor implementation on a lattice (where k_n is taken so that each point would be connected to its r_n-neighborhood if there were no obstacles).

5 Numerical Experiments

In this section we numerically investigate the benefits of planning with low-dispersion sampling instead of i.i.d. sampling. Section 5.1 overviews the simulation environment used for this investigation. Section 5.2 details the deterministic low-dispersion sequences used, namely, lattices and the Halton sequence. Several simulations are then introduced and results compared to i.i.d. sampling in Sections 5.3 and 5.4. We then briefly discuss non-i.i.d., random, low-dispersion sampling schemes in Section 5.5. Finally, in Section 5.6 we discuss the results of different connection radius scalings, specifically $(\log(\log(n))/n)^{1/d}$, to numerically validate the theoretical results in Section 4.

5.1 Simulation Environment

Simulations were written in C++ and MATLAB, and run using a Unix operating system with a 2.3 GHz processor and 8 GB of RAM. The C++ simulations were run through the Open Motion Planning Library (OMPL) [6]. The planning problems simulated in OMPL were rigid body problems whereas the simulations in MATLAB used point robots. For each planning problem, the entire implementation of

gPRM was held fixed (including the sequence of r_n) except for the sampling strategy. Specifically, for all simulations (except those in Section 5.6), we use as connection radius $r_n = \gamma_{\mathrm{PRM}} (\log(n)/n)^{1/d}$, where $\gamma_{\mathrm{PRM}} = 2.2 \left(1 + 1/d\right)^{1/d} \left(1/\zeta_d\right)^{1/d}$ and ζ_d is the volume of the unit ball in the d-dimensional Euclidean space. This choice ensures asymptotic optimality both for i.i.d. sample sequences [13, 12] and deterministic low-dispersion sequences (Theorem 1). Because this is an exact "apples-to-apples" comparison, we do not present runtime results, but only results in terms of sample count (note that drawing samples represents a trivial fraction of the total algorithm runtime). As mentioned in the introduction, deterministic sampling even allows for possible speed-ups in computation (e.g., fast nearest-neighbor indexing).

5.2 Sampling Sequences

We consider two deterministic low-dispersion sequences, namely the Halton sequence [9] and lattices. Halton sampling is based on a generalization of the van der Corput sequence and uses prime numbers as its base to form a low-dispersion sequence of points [9, 17]. Lattices in this work were implemented as a triangular lattice in two dimensions and a Sukharev grid in higher dimensions [25].

Along with the benefits of lattices described throughout the paper, they also present some challenges [17]. First, for basic cubic lattices with the same number of lattice points k per side, the total number of samples n is fixed to k^d, which limits the potential number of samples. For example, in 10 dimensions, the first four available sample counts are 1, 1024, 59,049, and 1,048,576. There are some strategies to allow for incremental sampling [27, 18, 28], but in this paper we overcome this difficulty by simply "weighting" dimensions. Explicitly, we allow each side to have a different number of lattice points. Independently incrementing each side's number of points by 1 increases the allowed resolution of n by a factor of d, as it allows all numbers of the form $n = (k-1)^m (k)^{d-m}$.

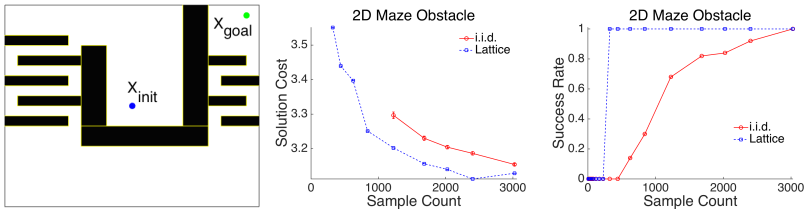
Second, lattices are sensitive to axis-alignment effects, whereby an axis-aligned obstacle can invalidate an entire axis-aligned row of samples. A simple solution to this problem is to rotate the entire lattice by a fixed amount (note this changes nothing about the ℓ_2-dispersion or nearest-neighbor sets). We chose to rotate each dimension by 10π degrees as an arbitrary angle far from zero (in practice, problems often show a "preferential" direction, e.g., vertical, so there may be an *a priori* natural choice of rotation angle).

Finally, in a Sukharev lattice in SE(2) or SE(3), each translational lattice point represents many rotational orientations. In SE(3), this means there are only k^3 translational points when $n = k^6$. In many rigid body planning problems, the translational dimension is more cluttered and the dominating term in the cost, so the effective reduction in translational samples severely hampers planning. A solution is to perform "spreading," which entails de-collapsing all the rotational orientations at each translational lattice point by spreading them deterministically in the translational dimensions.

5.3 Simulation Test Cases

Within the MATLAB environment, results were simulated for a point robot within a Euclidean unit hypercube with a variety of geometric obstacles over a range of dimensions. First, rectangular obstacles in 2D and 3D were generated with a fixed configuration that allowed for several homotopy classes of solutions. A 2D maze

with rectangular obstacles was also created (Figure 2(a)). These sets of rectangular obstacles are of particular interest as they represent a possible "worst-case" for lattice-based sampling because of the potential for axis alignment between samples and obstacles. The results, shown for the 2D maze in Figure 2 and for all experiments in Table 1, show that Halton and lattice sampling outperform i.i.d. sampling in both success rate and solution cost.



(a) (b) (c)

Fig. 2 Figure 2(a): The planning setup for a point robot with rectangular obstacles in a 2D maze. Figures 2(b) and 2(c): The results for solution cost and success rate versus sample count (averaged over 50 runs). For clarity we only report data for i.i.d. sampling and lattice sequences, results including Halton sampling are reported in Table 1. Only points with greater than 50% success are shown in Figure 2(b).

To illustrate rectangular planning scenarios in higher dimensional space, we constructed a recursive maze obstacle environment. Each instance of the maze consists of two copies of the maze in the previous dimension separated by a divider and connected through the last dimension. See Figures 3(a) and 3(b) for the first two instances of the maze in two and three dimensions. This recursion produces a configuration space with only one homotopy class of solution, any element of which is necessarily long and features sections that are spatially close, but far away in terms of their distance along the solution path. Halton and lattice sampling confered similar benefits in the recursive mazes in 2D, 3D, 4D, 5D, and 6D as they did in other simulations (see Table 1).

In addition to the rectangular obstacles, hyperspherical obstacles within a Euclidean unit hypercube were generated to compare performance on a smooth obstacle set with no possibility of axis alignment between samples and obstacles. The setups for 2D and 3D (Figures 3(c) and 3(d)) were fixed, while in 4D, obstacles were randomly generated to match a specified spatial coverage. Again, Halton and lattice sampling consistently outperformed random sampling, as shown in Table 1.

Within the OMPL environment, rigid body planning problems from the OMPL test banks were posed for SE(2) and SE(3) configuration spaces. In the SE(2) case, one rotational and two translational degrees of freedom are available, resulting in a three dimensional space, shown in Figure 3(e). The SE(3) problem consists of an "L-shaped" robot moving with three translational and three rotational degrees of freedom, resulting in six total problem dimensions, shown in Figure 3(f). The SE(2) and SE(3) lattices use the spreading method described in Section 5.2. The results, summarized in Table 1, show that Halton and lattice sampling generally outperform i.i.d. random sampling.

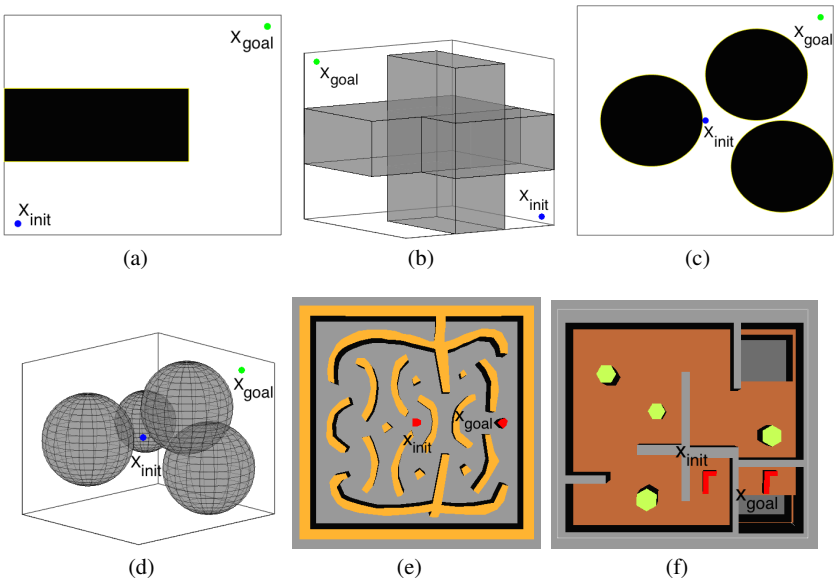


Fig. 3 Images of the recursive maze planning problem in 2D (3(a)) and 3D (3(b)) and the spherical obstacle sets in 2D (3(c)) and 3D (3(d)). Also shown are the OMPL rigid body planning problems for SE(2) and SE(3) in 3(e) and 3(f) respectively. A summary of results can be found in Table 1.

5.4 Summary of Results

Table 1 shows a summary of the results from simulations detailed in Section 5.3. Results are shown normalized by the i.i.d. sampling results. In each case the sample count at which a success rate greater than 90% is achieved and sustained is reported. Additionally, the solution costs at a medium and high sampling count are shown. For all cases the lattice sampling finds a solution with fewer or an equal number of samples and of lower or equal cost than that found by i.i.d. sampling. The Halton sampling also always finds a solution at lower sample counts than i.i.d. sampling, and almost always finds solutions of lower cost as well. The low-dispersion sequences particularly outperform random sampling in terms of number of samples required for a 90% success rate.

5.5 Nondeterministic Sampling Sequences

The above simulations showed deterministic lattice sampling, with a fixed rotation around each axis, and the deterministic Halton sequence outperform uniform i.i.d. sampling. Both deterministic sequences have low ℓ_2-dispersions of $O(n^{-1/d})$, but sequences with the same order ℓ_2-dispersion need not be deterministic. Figure 4 shows results for a randomly rotated and randomly offset version of the lattice (again, the ℓ_2-dispersion and neighborhoods are all still deterministically the same). The same cases in Table 1 were run for the randomly rotated lattice and the results showed it performed as well as or better than random sampling (over 50 runs). In general, low-dispersion random sequences might provide some advantages, e.g., eliminating axis alignment issues while still enjoying deterministic guarantees (see

		Halton			Lattice		
Dim	Obstacles	90%	Medium	High	90%	Medium	High
2	Rectangular	0.38	1.18	0.80	0.15	0.56	0.80
3	Rectangular	0.36	0.88	0.94	0.19	0.80	0.87
2	Rect Maze	0.13	0.98	0.99	0.13	1.00	0.99
2	Sphere	0.16	0.93	0.99	0.07	0.93	0.99
3	Sphere	0.36	0.97	1.00	0.08	0.97	0.99
4	Sphere	1.00	0.97	0.97	1.00	0.97	1.00
2	Recursive Maze	0.33	1.00	1.00	0.18	1.00	1.00
3	Recursive Maze	0.22	0.95	0.99	0.22	0.96	0.98
4	Recursive Maze	0.56	0.95	0.98	0.56	1.00	1.00
5	Recursive Maze	0.45	0.97	0.96	0.60	0.95	0.96
6	Recursive Maze	0.56	0.95	0.97	0.75	0.94	0.96
3	SE(2)	0.81	0.96	1.00	0.81	1.01	1.01
6	SE(3)	0.32	0.96	0.93	0.42	0.94	0.95

Table 1 Summary of results. Each column is normalized by the results of random sampling (averaged over 50 runs). For Halton sampling and lattice sampling, the number of samples at which 90% success is achieved and the cost at a medium number of samples (near 700) and a high number of samples are shown (highest samples simulated, always 3000 or greater).

Section 4). Their further study represents an interesting direction for future research.

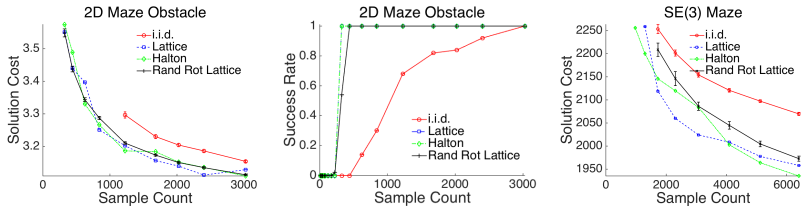


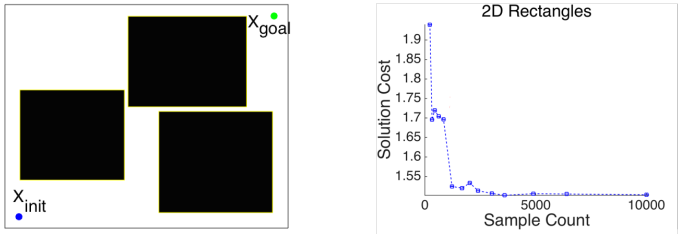
Fig. 4 Results for deterministic and nondeterministic low-dispersion sampling. "Rand Rot Lattice" refers to a randomly rotated lattice.

5.6 Radius Scaling of $(\log(\log(n))/n)^{1/d}$

To validate the results in Section 4 (specifically, Theorem 1), we investigate the effects of scaling the radius as $\gamma_{\text{PRM}} \log(\log(n))/n)^{1/d}$. The results of these simulations are shown in Figure 5 for 2D rectangular obstacles. In the case of $\gamma_{\text{PRM}} (\log(\log(n))/n)^{1/d}$, the lattice planning appears to still converge to the optimal cost, as predicted by Theorem 1.

6 Conclusions

This paper has shown that using low-dispersion sampling strategies (in particular, deterministic) can provide substantial benefits for solving the optimal path planning problem with sampling-based algorithms, in terms of deterministic perfor-



(a) (b)

Fig. 5 Figure 5(a): planning setup for study of radius scaling. Figure 5(b): planning results for lattice sampling with radius scaling of $\gamma_{\text{PRM}}\,(\log(\log(n))/n)^{1/d}$.

mance guarantees, reduced computational complexity per given number of samples, and superior practical performance.

This paper opens several directions for future research. First, we plan to extend our results to the case of kinodynamic motion planning. Second, it is of interest to extend the results herein to other classes of sampling-based motion planning algorithms, especially the large class of *anytime* algorithms (e.g., RRT/RRT*). This leads directly into a third key direction, which is to study alternative low-dispersion sampling strategies beyond the few considered here, particularly *incremental* sequences for use in anytime algorithms. There is already some work in this area, although thus far it has focused on the use of such sequences for the feasibility problem [27, 18, 28]. It may also be of interest to study low-dispersion sampling strategies that incorporate prior knowledge of the problem by sampling non-uniformly. Third, we plan to investigate the topological relationship between the optimal path cost and that of the best strong-δ-clear path, in order to frame the convergence rate in terms of the true optimal cost. Fourth, from a practical standpoint, it is of interest to adapt existing algorithms or design new ones that explicitly leverage the structure of low-dispersion sequences (e.g., fast nearest neighbor indexing or precomputed data structures). This would be especially beneficial in the domain of kinodynamic motion planning. Finally, leveraging our convergence rate results, we plan to investigate the issue of certification for sampling-based planners, e.g., in the context of trajectory planning for drones or self-driving cars.

References

[1] R. Alterovitz, S. Patil, and A. Derbakova. Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning. In *Proc. IEEE Conf. on Robotics and Automation*, pages 3706–3712, 2011.

[2] O. Arslan and P. Tsiotras. Use of relaxation methods in sampling-based algorithms for optimal motion planning. In *Proc. IEEE Conf. on Robotics and Automation*, pages 2421–2428, May 2013.

[3] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, Tsai-Y. Li, and P. Raghavan. A random sampling scheme for path planning. In *International Journal of Robotics Research*, pages 249–264. Springer, 2000.

[4] R. Bohlin and L. Kavraki. Path planning using lazy PRM. In *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, CA, April 2000.

[5] M. S. Branicky, S. M. LaValle, K. Olson, and L. Yang. Quasi-randomized path planning. In *Proc. IEEE Conf. on Robotics and Automation*, volume 2, pages 1481–1487, 2001.

[6] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics and Automation Magazine*, 19(4):72–82, December 2012.
[7] P. Deheuvels. Strong bounds for multidimensional spacings. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 64(4):411–424, 1983.
[8] A. Dobson, G. Moustakides, and Bekris. K. E. Geometric probability results for bounding path quality in sampling-based roadmaps after finite computation. In *Proc. IEEE Conf. on Robotics and Automation*, 2015. To Appear.
[9] J. H. Halton. On the efficiency of certain quasirandom sequences of points in evaluating multidimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.
[10] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *International Journal of Computational Geometry & Applications*, 9(4):495–512, 1999.
[11] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *International Journal of Robotics Research*, 25(7):627–643, 2006.
[12] L. Janson, E. Schmerling, A. Clark, and M. Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *International Journal of Robotics Research*, 2015. In Press.
[13] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.
[14] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996. doi: 10.1109/70.508439.
[15] S. Lavalle. *Planning Algorithms*. Cambridge University Press, 2006.
[16] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, 2001. doi: 10.1177/02783640122067453.
[17] S. M. LaValle, M. S. Branicky, and S. R. Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *International Journal of Robotics Research*, 23(7-8): 673–692, 2004.
[18] S. R. Lindemann, A. Yershova, and S. M. LaValle. Incremental grid sampling strategies in robotics. In *Workshop on Algorithmic Foundations of Robotics*, pages 313–328. 2005.
[19] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Number 63 in CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial & Applied Mathematics, 1992.
[20] J. B. Orlin, K. Madduri, K. Subramani, and M. Williamson. A faster algorithm for the single source shortest path problem with few distinct positive lengths. *Journal of Discrete Algorithms*, 8(2):189 – 198, 2010.
[21] J. M. Phillips, N. Bedrossian, and L. E. Kavraki. Guided expansive spaces trees: A search strategy for motion- and cost-constrained state spaces. In *Proc. IEEE Conf. on Robotics and Automation*, volume 4, pages 3968–3973, 2004.
[22] M. Pivtoraiko, R. A. Knepper, and A. Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.
[23] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and L. E. Kavraki. Sampling-based roadmap of trees for parallel motion planning. *IEEE Transactions on Robotics*, 21(4):597–608, 2005.
[24] A. Stentz. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics & Automation*, 10(3):89–100, 1995.
[25] A. G. Sukharev. Optimal strategies of the search for an extremum. *USSR Computational Mathematics and Mathematical Physics*, 11(4):119–137, 1971.
[26] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
[27] A. Yershova and S. M. LaValle. Deterministic sampling methods for spheres and SO(3). In *Proc. IEEE Conf. on Robotics and Automation*, volume 4, pages 3974–3980, 2004.
[28] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell. Generating uniform incremental grids on SO(3) using the Hopf fibration. *International Journal of Robotics Research*, 2009.