Article

The International Journal of Robotics Research
2018, Vol. 37(1) 46–61

DOI: 10.1177/0278364917714338
journals.sagepub.com/home/ijr

Deterministic sampling-based motion planning: Optimality, complexity, and performance

Lucas Janson[1], Brian Ichter[2] and Marco Pavone[2]

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. Specifically, we first show that PRM, for a certain selection of tuning parameters and deterministic low-dispersion sampling sequences, is deterministically *asymptotically optimal, in other words, 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 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 contrast to the $O(n \log n)$ complexity results for existing asymptotically optimal probabilistic planners. Fourth, we discuss extending our theoretical results and insights to other batch-processing algorithms such as FMT*, to non-uniform sampling strategies, to k-nearest-neighbor implementations, and to differentially constrained problems. Importantly, our main theoretical tool is the ℓ_2-dispersion, an interesting consequence of which is that* all *our theoretical results also hold for low-ℓ_2-dispersion* random *sampling (which i.i.d. sampling does not satisfy). In other words, 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. Finally, through numerical experiments, we show that planning with deterministic (or random) low-dispersion sampling generally provides superior performance in terms of path cost and success rate.*

Keywords

Motion planning, optimal path planning, sampling-based algorithms, motion control, ℓ_2-dispersion

1. Introduction

Probabilistic sampling-based algorithms represent a particularly successful approach to robotic motion planning problems (Lavalle, 2006; Thrun et al., 2005). 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" (Lavalle, 2006). Examples, roughly in chronological order, include the probabilistic roadmap (PRM) algorithm (Kavraki et al., 1996), expansive space trees (Hsu et al., 1999; Phillips et al., 2004), Lazy-PRM (Bohlin and Kavraki, 2000), the rapidly exploring random tree

[1] Department of Statistics, Stanford University, CA, USA
[2]Department of Aeronautics and Astronautics, Stanford University, CA, USA

Corresponding author:
Marco Pavone, Stanford University, Department of Aeronautics & Astronautics William F. Durand Building, Rm. 261 496 Lomita Mall Stanford, CA 94305-4035.
Email: pavone@stanford.edu

(RRT) algorithm (LaValle and Kuffner, 2001), sampling-based roadmap of trees (Plaku et al., 2005), rapidly exploring roadmap (Alterovitz et al., 2011), PRM* and RRT* (Karaman and Frazzoli, 2011), RRT# (Arslan and Tsiotras, 2013), and the fast marching tree algorithm (FMT*) (Janson et al., 2015). 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 (Barraquand et al., 2000). Recently, it has been proven that RRT*, PRM*, RRT#, and FMT* are asymptotically optimal, that is, the cost of the returned solution converges probabilistically to the optimum (Arslan and Tsiotras, 2013; Janson et al., 2015; Karaman and Frazzoli, 2011).

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, in other words, 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 (Branicky et al., 2001; Hsu et al., 2006; LaValle et al., 2004) 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 in an extensive set of numerical experiments (Branicky et al., 2001; LaValle et al., 2004) (see also Section 5 of Hsu et al., 2006, for a similar conclusion). Prompted by these results, a number of deterministic low-dispersion incremental sequences have been specifically tailored to motion planning problems (Lindemann et al., 2005; Yershova and LaValle, 2004; Yershova et al., 2010).

The results in Branicky et al. (2001), LaValle et al. (2004), and Hsu et al. (2006) are restricted to convergence to feasible, as opposed to *optimal*, paths. Several questions are still open. Are there advantages to 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 probabilistic planners in Janson et al. (2015) and Dobson et al. (2015)? 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, we show that similar results hold for many of the existing batch (i.e. not anytime) algorithms, including Lazy-PRM and FMT* .

1.1. 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 in d dimensions 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 analogous result for the case of i.i.d. random sampling holds probabilistically or in probability (Janson et al., 2015; Karaman and Frazzoli, 2011) (as opposed to deterministically) and requires a connection radius $\Omega\left((\log(n)/n)^{1/d}\right)$, that is, a bigger connection radius.

Convergence rate. We show that, in the absence of obstacles, the factor of sub-optimality 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 analogous result for the case of i.i.d. sampling only holds in probability and is much more involved (and less interpretable) (Janson et al., 2015). 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 $O(n\, \log(n))$ (Karaman and Frazzoli, 2011).

Extensions. We extend the contributions in all three of the preceding categories to much broader settings. Specifically, we find that many of the results that hold for PRM run on a low-dispersion lattice hold either exactly or approximately for k-nearest-neighbor algorithms, for other batch-processing algorithms such as FMT*, for non-lattice low-dispersion sampling such as the Halton sequence, for non-uniform sampling, and for kinodynamic planning.

Experimental performance. Finally, we compare performance (in terms of path cost and success rates) of deterministic low-dispersion sampling versus i.i.d. sampling on a variety of test cases ranging from two to eight dimensions and including geometric, kinematic chain, and kinodynamic planning problems. 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 $O((\log n)^{1/d}\, n^{-1/d})$ (Deheuvels, 1983; Niederreiter, 1992). Interestingly, the same $O(n^{-1/d})$ dispersion can be achieved with non-i.i.d. *random* sequences, for example randomly rotated and offset lattices. As we will show, these sequences enjoy the same *deterministic* performance guarantees as 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), for example 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 Branicky et al. (2001) and LaValle et al. (2004), 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, for example D* (Stentz, 1995) and related kinodynamic variants (Pivtoraiko et al., 2009).

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 extend the results from Section 4 to other batch-processing algorithms, non-uniform sampling, and kinodynamic motion planning. In Section 6 we present results from numerical experiments supporting our statements. Finally, in Section 7, 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, non-empty set S of points contained in a d-dimensional compact Euclidean subspace $\mathcal{X}$ with positive Lebesgue measure, its ℓ_2-dispersion $D(S)$ is defined as*

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

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 (Niederreiter, 1992)

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

which allows us to bound $D(S)$ when $D_\infty(S)$ is easier to compute. In particular, an important result due to Deheuvels (1983) 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 (LaValle et al., 2004); in this paper we will simply refer to both as "sequences".) For instance, the Sukharev sequence (Sukharev, 1971), 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 (Lavalle, 2006). In this paper, we will focus on the Sukharev (Sukharev, 1971) and Halton sequences (Halton, 1960), except in two dimensions when we will consider the triangular lattice as well,

though there are many other deterministic sequences with ℓ_2-dispersion of order $O(n^{-1/d})$; see Yershova and LaValle (2004), Lindemann et al. (2005), and Yershova et al. (2010) for other examples.

3. Problem statement

The problem formulation follows that in Janson et al. (2015) very closely. Let $\mathcal{X} = [0, 1]^d$ be the configuration space, where $d \in \mathbb{N}$. Let $\mathcal{X}_{\text{obs}}$ be a closed set representing the obstacles, and let $\mathcal{X}_{\text{free}} = \text{cl}(\mathcal{X} \setminus \mathcal{X}_{\text{obs}})$ be the obstacle-free space, where $\text{cl}(\cdot)$ denotes the closure of a set. The initial condition is $x_{\text{init}} \in \mathcal{X}_{\text{free}}$, and the goal region is $\mathcal{X}_{\text{goal}} \subset \mathcal{X}_{\text{free}}$. A specific path planning problem is characterized by a triplet $(\mathcal{X}_{\text{free}}, x_{\text{init}}, \mathcal{X}_{\text{goal}})$. A function $\sigma : [0, 1] \to \mathbb{R}^d$ is a *path* if it is continuous and has bounded variation. If $\sigma(\tau) \in \mathcal{X}_{\text{free}}$ 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 $\sigma(1) \in \text{cl}(\mathcal{X}_{\text{goal}})$, then σ is said to be a *feasible path* for the planning problem $(\mathcal{X}_{\text{free}}, x_{\text{init}}, \mathcal{X}_{\text{goal}})$.

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

> **Optimal path planning problem.** Given a path planning problem $(\mathcal{X}_{\text{free}}, x_{\text{init}}, \mathcal{X}_{\text{goal}})$ with an arc length cost 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 (Karaman and Frazzoli, 2011). For a given $\delta > 0$, define the δ-interior of $\mathcal{X}_{\text{free}}$ as the set of all configurations that are at least a distance δ from $\mathcal{X}_{\text{obs}}$. Then a path is said to have strong δ-clearance if it lies entirely inside the δ-interior of $\mathcal{X}_{\text{free}}$. Further, a path planning problem with optimal path cost c^* is called δ-robustly feasible if there exists a strictly positive sequence $\delta_n \to 0$, 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, $\sigma_n(1) \in \partial\mathcal{X}_{\text{goal}}$, and $\sigma_n(\tau) \notin \mathcal{X}_{\text{goal}}$ 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 $\mathcal{X}_{\text{free}}$. 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 linear interpolation between 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 $\mathcal{X}_{\text{goal}}$ in the graph (V, E) (if one exists, otherwise it reports failure), where a path's length is defined as the sum of the costs (Euclidean distances) of its edges. Deliberately, we do not specify the definition of `SampleFree` and have left r_n in line 3 of Algorithm 1 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 Karaman and Frazzoli (2011).

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 $(\mathcal{X}_{\text{free}}, x_{\text{init}}, \mathcal{X}_{\text{goal}})$ be a δ-robustly feasible path planning problem in d dimensions, with $\delta > 0$ and $\mathcal{X}_{\text{goal}}$ 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

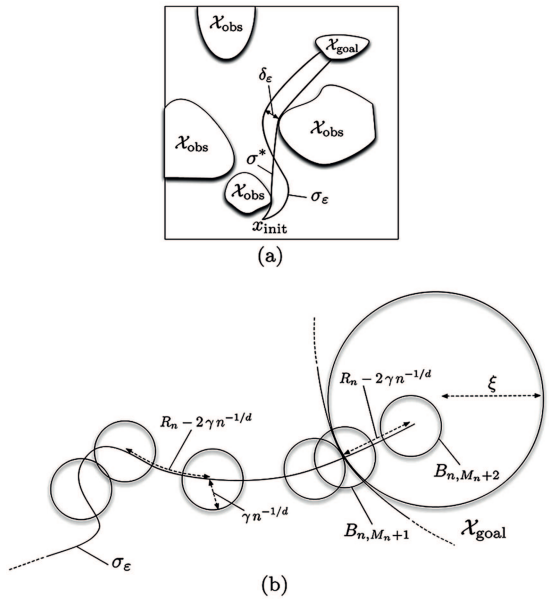

Fig. 1. (a) Illustration in two dimensions of σ_ε as the shortest strongly δ_ε-robust feasible path, as compared to the optimal path σ^*, as used in the proof of Theorem 1. (b) Illustration in two dimensions of the construction of $B_1, \ldots, B_{M_n+2}$ in the proof of Theorem 1.

has strong δ_ε-clearance for some $\delta_\varepsilon > 0$; see Figure 1(a). Let $\{R_n\}_{n=1}^{\infty}$ be a sequence such that $R_n \leq r_n$, $n^{1/d}R_n \to \infty$, and $R_n \to 0$, guaranteeing that there exists a $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,

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

For any $n \geq n_0$, construct the sequence of closed balls $\{B_{n,i}\}_{i=1}^{M_n+2}$ 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 $\mathcal{X}_{\text{goal}}$. 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 $\mathcal{X}_{\text{goal}}$ by a distance $R_n - 2\gamma n^{-1/d}$ in the direction perpendicular to $\partial\mathcal{X}_{\text{goal}}$. Note that by equation (3), $B_{n,M_n+2} \subset \mathcal{X}_{\text{goal}}$. See Figure 1(b) for an illustration.

Since the dispersion is upper-bounded by the radii of all the $B_{n,i}$ balls, each $B_{n,i}$ 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 $R_n \leq \delta_\varepsilon - \gamma n^{-1/d}$ 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 $(x_0, x_1), \ldots, (x_{M_n+1}, x_{M_n+2})$:

$$\begin{aligned} c_n &\leq \sum_{i=0}^{M_n+1} \|x_{i+1} - x_i\| \leq (M_n + 2)\,R_n \\ &\leq \frac{c(\sigma_\varepsilon)}{R_n - 2\gamma n^{-1/d}} R_n + 2R_n \\ &\leq c(\sigma_\varepsilon) + \frac{2\gamma n^{-1/d}}{R_n - 2\gamma n^{-1/d}} c(\sigma_\varepsilon) \\ &\quad + 2R_n = c(\sigma_\varepsilon) + \frac{1}{\frac{R_n}{2\gamma n^{-1/d}} - 1} c(\sigma_\varepsilon) + 2R_n \\ &\leq \left(1 + \frac{\varepsilon}{3}\right) c^* + \frac{1}{\frac{3}{\varepsilon} + 1}\left(1 + \frac{\varepsilon}{3}\right) c^* + \frac{\varepsilon}{3} c^* = (1 + \varepsilon)\,c^* \end{aligned}$$

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 fact 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 $2\,D(V)$. Such a result can be quite useful in practice, as solutions going through narrow corridors could be undesirable anyway (see LaValle et al., 2004, 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, for example Janson et al. (2015), 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) with which the probability of returning a greater-than-ε suboptimal solution goes to zero. In contrast, we compute the rate (in n) with which 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 $\mathcal{X}_{\text{free}}$, assuming that both the initial point and the final point have already been sampled. Define*

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

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, using a connection radius r_n. Then for all n such that $r_n > 2D_n$ and $r_n + D_n < \delta$,

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

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 centered at x_{i-1}, and that ball is collision-free because of the assumption that $r_n + D_n < \delta$), and x_f is connected to x_{M_n} as well. Thus c_n is upper-bounded by the path tracing through $x_0, x_1, \ldots, x_{M_n}, x_f$:

$$\begin{aligned} c_n &\le \|x_1 - x_0\| + \sum_{i=2}^{M_n} \|x_i - x_{i-1}\| + \|x_f - x_{M_n}\| \\ &\le r_n - D_n + \sum_{i=2}^{M_n} r_n + \|x_f - x_{M_n}\| \\ &\le (M_n r_n - D_n) + \left(\left(\frac{c^{(\delta)}}{r_n - 2D_n} - \left\lfloor \frac{c^{(\delta)}}{r_n - 2D_n} \right\rfloor \right) \right. \\ &\quad \left. (r_n - 2D_n) + D_n \right) \\ &= c^{(\delta)} + 2D_n M_n \le \left(1 + \frac{2D_n}{r_n - 2D_n} \right) c^{(\delta)} \end{aligned}$$

where the second and third inequalities follow by considering the greatest 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). For instance, consider a deterministic low-dispersion sequence with dispersion upper-bounded by $\gamma\, n^{-1/d}$, and assume one has time to plan on $\bar{n}$ samples. Choose $r_{\bar{n}} = 2\psi\gamma\, \bar{n}^{-1/d}$ for some $\psi > 1$ and choose $\delta_0 > r_{\bar{n}}$ (one can readily verify that such a choice of $r_{\bar{n}}$ satisfies the assumptions of Theorem 2). Then one can state the deterministic guarantee "For* all *planning problems where there exists a feasible δ_0-clear path (see Figure 2), the cost of the path returned is within a factor $1/(\psi - 1)$ of the cost of the* best *δ_0-clear path." For given values of ψ and δ_0, one can therefore "certify" the performance of the planner for a desired target performance. An interesting avenue for future research is to use information about the curvature of the obstacles to quantify the difference between $c^{(\delta)}$ and the true optimal cost c^*.*

Remark 3 (Alternative cost functions)**.** *Although we have used arc length as the cost function in Theorems 1 and 2, the proofs only rely on a few of its properties: the triangle inequality, additivity, and its natural relationship with Euclidean distance. This allows our results to be easily extended to a number of other cost functions. In particular, let $c(\sigma)$ denote the cost of the path σ, let $\sigma(x, y)$ denote the minimum-cost (with respect to c) path between x and y, and let $\sigma_1\sigma_2$ denote the concatenation of two paths when σ_1 ends at the point at which σ_2 begins. Then suppose the cost function c satisfies $c(\sigma(x, z)) \le c(\sigma(x, y)) + c(\sigma(y, z))$, $c(\sigma_1\sigma_2) = c(\sigma_1) + c(\sigma_2)$, and $\beta_1\|x - y\| \le c(\sigma(x, y)) \le \beta_2\|x - y\|$ for some $0 < \beta_1 \le \beta_2 < \infty$. Then if gPRM measures distances and connects points using $c(\sigma(\cdot, \cdot))$ and $\sigma(\cdot, \cdot)$, respectively,*[2] *Theorem 1 holds unchanged and the statement of Theorem 2 becomes: for all n such that $r_n > 2\beta_2 D_n$ and $r_n + \beta_2 D_n < \beta_1\delta$,*

$$c_n \le \left(1 + \frac{2\beta_2 D_n}{r_n - 2\beta_2 D_n} \right) c^{(\delta)}$$

when c_n, c^*, and $c^{(\delta)}$ are redefined as path costs as opposed to arc lengths. The altered proofs simply require replacing all arc lengths/Euclidean distances (including ball radii) by $c(\sigma(\cdot, \cdot))$, multiplying all $\gamma\, n^{-1/d}$ and D_n by β_2, and being careful with constants in dealing with δ-clearance and ξ-regularity.

A common cost function satisfying these conditions is the line integral evaluated over a function on the configuration space which is bounded above and away from zero.

Both the asymptotic optimality and convergence rate results can be extended to other batch planners such as Lazy-PRM or FMT*, as discussed in Section 5.1.

Lastly, we show that using a low-ℓ_2-dispersion lattice sample set an asymptotically optimal version of gPRM can be run that has lower-order computational complexity than any existing asymptotically optimal 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*

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

Furthermore, as long as $n r_n^d \nrightarrow 0$, the space complexity is also $O(n^2 r_n^d)$.

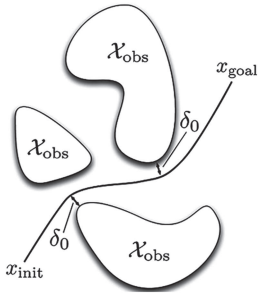

Fig. 2. An example of a planning problem with a feasible δ_0-clear path. For a given clearance parameter δ_0 and suboptimality factor ψ, one can readily determine the number of samples (and, hence, computation time) that is *deterministically* guaranteed to return a path whose cost is within $1/(\psi - 1)$ of the cost of the *best* δ_0-clear path, for *all* planning problems where a feasible δ_0-clear path exists.

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 need 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 Orlin et al. (2010) 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).

The space complexity is proportional to the number of edges plus the number of vertices, which are $O(n^2 r_n^d)$ and $O(n)$, respectively. By assumption that $nr_n \nrightarrow 0$, $O(n^2 r_n^d)$ will be the (possibly co-) dominant term. □

Since Theorem 1 allows $r_n \in \omega(n^{-1/d})$ while maintaining asymptotic optimality, Theorem 3 implies that cubic-lattice sampling allows for an asymptotically optimal algorithm with computational and space complexity $\omega(n)$. All other asymptotically optimal algorithms in the literature have computational and space complexity of 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 discuss the limitations implied by this in the next section.

5. Extensions

In this section we discuss several extensions to the results presented in Section 4. In particular, we discuss extensions to alternative implementations and other types of batch-processing algorithms (Sections 5.1 and 5.2), to non-uniform sampling sequences (Section 5.3), and to kinodynamic motion planning (Section 5.4).

5.1. Convergence results for other batch-processing algorithms

The theoretical convergence results of the previous section (Theorems 1 and 2) extend to alternative implementations of gPRM and other batch-processing algorithms. We first discuss a variant of gPRM which makes connections based on k nearest neighbors instead of a fixed connection radius. This variant, referred to as k-nearest gPRM, has the advantage of being more adaptive to different obstacle spaces than its connection-radius counterpart (we refer the reader to Janson et al., 2015, Section 5.3, for a more detailed discussion about the relative benefits of a k-nearest-neighbors implementation). Assuming that k-nearest gPRM is implemented so that the number of neighbors k_n (a function of n, as r_n) is taken for each sample to be roughly connected to its r_n-neighborhood if there are no obstacles, then deterministic asymptotic optimality and rate convergence guarantees can be readily derived. Specifically, assuming $k_n = (1+\epsilon)\, n\zeta_d r_n^d$, where $\epsilon > 0$ and ζ_d is the volume of the unit ball in d dimensions, a graph constructed in k-nearest gPRM with k_n neighbors will asymptotically contain all the edges (and more) of the graph constructed in gPRM with radius r_n, and thus will return a path at least as low-cost.

A second example is Lazy-PRM (Bohlin and Kavraki, 2000), where the extension of the theoretical convergence results is straightforward, as the path returned by Lazy-PRM is identical to that returned by gPRM (using the same radius).

A third example is FMT* (Janson et al., 2015). The only difference in the proof for FMT* from that for gPRM is that c_n cannot naïvely be upper-bounded by the length of a path tracing through a sequence of points in the gPRM graph, as FMT* may not use some edges. However, as shown in the proof of Lemma 4.2 (pp. 912–913) in Janson et al. (2015,

Appendix A), FMT*'s c_n *can* be bounded by the length of a path tracing through a sequence of points in the gPRM graph *if those points are contained in a suitable sequence of balls*. In particular, the sequence of balls needs to be sufficiently far from obstacles and adjacent balls need to be sufficiently close together. It is easy to check that the sequences $\{B_i\}$ used in the proofs of the previous section satisfy these conditions.

5.2. Complexity results for different sampling schemes and other batch-processing algorithms

The complexity result in Theorem 3, strictly speaking, only applies to gPRM as defined in Algorithm 1 run on n samples arranged in a d-dimensional cubic lattice. Using other low-dispersion but non-lattice sampling schemes, such as the Halton/Hammersley sequence, would not give $O(nr_n^d)$ distinct edge lengths, which precludes the use of the improved implementation of Dijkstra's algorithm (Orlin et al., 2010). A k-nearest-neighbor implementation (where k_n is selected as specified in Section 5.1 in order to ensure convergence), even on a lattice, would also no longer in general have $O(nr_n^d)$ distinct edge lengths. Lastly, other batch-processing algorithms, such as FMT*, do not separate graph construction and shortest-path computation, again precluding, at least naïvely, the use of the improved implementation of Dijkstra's algorithm.

In all the above cases, all the same computational advantages as those stated in the proof of Theorem 3 would hold, *except* the advantage from the sped-up shortest-path algorithm in step (3). 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. In other words, the constant hidden in the big-O notation for the shortest-path algorithm is drastically smaller than that in the other steps. Thus, a practical approximation of the runtime could ignore the shortest-path component, in which case the result of Theorem 3 can be extended to all of the aforementioned examples.

5.3. Non-uniform sampling

A popular method for incorporating prior knowledge about a problem is to draw samples in a way that is not uniform throughout the configuration space. Increased sampling density in areas that are especially hard to traverse gives a motion planning algorithm help in that area. Thus if these areas can be identified a priori or even adaptively (see, for example, Boor et al., 1999), observed convergence can be substantially sped up. Sampling non-uniformly in the i.i.d. setting is often fairly straightforward, for instance by simply adding sample-rejection rules to the uniform strategy (Hsu et al., 2006). In fact, every non-uniform sampling strategy that we are aware of can be formulated as a transformation of uniformly sampled points, and we expect that applying the same transformation to a low-dispersion set of points will provide an improvement similar to that in the uniformly sampled case. The intuitive reasoning is the same: non-uniform sampling seeks to place samples in regions of space that are chosen better than uniformly, but low-dispersion samples characterize a volume better locally, so that the non-uniformly chosen regions will be better covered. Section 6.6 shows simulations in support of this conjecture.

Furthermore, as long as there remains a "baseline density" of samples everywhere in the space, similar guarantees for the uniform case can still be made. Specifically, think of the baseline density of samples as a low-dispersion set of samples whose size is a constant fraction α of n. Then those samples alone have low dispersion, not only with respect to their size αn, but also with respect to the full sample size n, since the two only differ by a constant multiple. Since adding back the rest of the samples can only lower the dispersion further, such a non-uniform set of samples must still have ℓ_2-dispersion that scales as $n^{-1/d}$, and thus the theoretical results of this paper apply.

There is a rich literature in different methods for non-uniform sampling, but providing theoretical results quantifying their improvement over uniform sampling is still a topic of current research. Once such results are established, an interesting future direction will be to understand the benefits of the non-uniform analog of low-dispersion sampling. Finally, although our rigorous theoretical results for non-uniform sampling are only worst-case in the baseline sampling density, it is worth noting that such results often form the core of proofs of theoretical results for non-uniform sampling. For instance, the asymptotic optimality of RRT*, whose sampling is non-uniform through a steering parameter, is proved under the assumption of no steering (and thus, uniform sampling). As another example, the asymptotic optimality of the batch informed trees (BIT*) (Gammell et al., 2015) algorithm is proved by showing that BIT*'s solution is no worse than that of RRT* with uniform sampling.

5.4. Deterministic kinodynamic planning

Another important extension is to motion planning with differential constraints. In particular, we consider here the extension to systems with linear affine dynamics of the form $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{c}$, where A, B, and $\mathbf{c}$ are constants. The extension of the ℓ_2-dispersion-based analysis of this paper to that case poses some challenges. The key roadblock is that the ℓ_2-dispersion is no longer a particularly accurate measure of how suitable a set of points is for tracking an optimal differentially constrained path. Essentially, Euclidean balls must be replaced by "perturbation" balls (Schmerling et al., 2015a), which are high-dimensional ellipses. To be clear, by "high-dimensional ellipse" we mean a volume defined by

$$\{x : x^{\mathrm{T}} Q x < r\} \tag{6}$$

for some positive-definite matrix Q and scalar r. Although such ellipses may be inner-bounded by a Euclidean ball, this (poor) approximation adds the exponential factor of the *controllability index* of the pair (A, B) (Schmerling et al., 2015a) to the analysis. (Assuming the pair (A, B) is controllable, the controllability indices ν_i give a fundamental notion of how difficult a linear system is to control in the various directions; see Kailath, 1980, p. 431, or Chen, 1995, p. 150, for a detailed treatment. The number of controllability indices is equal to the number of control inputs, that is, to the number of columns of B. The maximum, that is, $\nu = \max \nu_i$, is referred to as the controllability index of the pair (A, B).) The following theorem (whose proof is largely based on the analysis framework devised in Schmerling et al., 2015a) summarizes the optimality result. Here gDPRM is just Algorithm 1 except that `Near` uses the cost in Schmerling et al. (2015a, equation (2)) instead of arc length (The D stands for "Differential").

Theorem 4 (Asymptotic optimality with deterministic sampling for systems with linear affine dynamics)**.** *Under the assumptions of Schmerling et al. (2015a, Theorem VI.1), gDPRM with deterministic low-dispersion sampling is asymptotically optimal for*

$$r_n = C_1 n^{-1/(\nu d)} \tag{7}$$

for some constant C_1, while gDPRM with i.i.d. uniform sampling is asymptotically optimal for

$$r_n = C_2 \left(\frac{\log(n)}{n} \right)^{1/\tilde{D}} \tag{8}$$

for some constant C_2, where $\tilde{D} = (d + \sum \nu_i^2)/2$, the ν_i are the controllability indices of the pair (A, B), and $\nu = \max \nu_i$.

Proof sketch. For the sake of brevity, we only describe here the changes needed to the theory in Schmerling et al. (2015a), as the results have much in common. The proof of (7) is nearly identical to that of Schmerling et al. (2015a, Theorem VI.1) except with a low-dispersion analog of Schmerling et al. (2015a, Theorem IV.6) which uses the same r_n rate as in (7). To see this analogous result, note that Schmerling et al. (2015a, Lemma IV.5) holds deterministically under low-dispersion sampling with the log term removed from the condition on the volume $\mu[T_k]$. Then the proof of the analog of Schmerling et al. (2015a, Theorem IV.6) only requires construction of the sets S_k (no T_k), again without the log term. From the deterministic version of Schmerling et al. (2015a, Lemma IV.5), we have that for the scaling of r_n as in (7), *every* set S_k will contain a sample, and equation (7) follows.

Finally, equation (8) is a direct corollary of Schmerling et al. (2015a, Theorem VI.1). □

If $\nu = 1$ (i.e. all directions are "equally difficult" to control), deterministic sampling and our analysis show all the same benefits as in the case of the path planning (non-kinodynamic) problem by getting rid of the $\log(n)$ term required by i.i.d. sampling without changing the exponent (as in this case $\nu d = d$ and $\tilde{D} = d$). Note that a special case is where $\nu = 1$ is represented by the single-integrator model $\dot{\mathbf{x}}(t) = \mathbf{u}(t)$, which effectively reduces the kinodynamic planning problem to the path planning problem stated in Section 3—in this sense, equation 7 recovers Theorem 1 when $\dot{\mathbf{x}}(t) = \mathbf{u}(t)$. However, in general, the exponent for the case of deterministic low-dispersion sampling (i.e. the exponent in equation (7)) may be worse. For instance, for the double-integrator model in three dimensions, namely $\ddot{\mathbf{x}}(t) = \mathbf{u}(t)$ and $d = 6$, the three controllability indices are $\nu_1 = \nu_2 = \nu_3 = 2$. As a consequence, one obtains $\nu d = 12$ and $\tilde{D} = 9$, and the radius in equation (7) (i.e. for the deterministic case) is *larger* than the radius for equation (8) (i.e. for the case with i.i.d. uniform sampling).

This is *not* to say that deterministic sampling is necessarily inappropriate or not advantageous for differentially constrained problems, but just that the analysis used here is inadequate (most critically, we crudely inner-bound ellipses via Euclidean balls). Our analysis does, however, suggest possible ways forward. One could consider a measure of dispersion which applies more specifically to ellipses, and possibly tailor a deterministic sequence to be low-dispersion in this sense. To our knowledge, no assessment of sample sequences in terms of this type of dispersion has been performed previously, and this represents a theoretically and practically important direction for future research (together with studying tailored notions of sampling sequences and dispersion for other classes of dynamical systems, e.g. driftless systems).

We will further study kinodynamic motion planning via deterministic sampling sequences through numerical experiments in Section 6.

6. Numerical experiments

In this section we numerically investigate the benefits of planning with deterministic low-dispersion sampling instead of i.i.d. sampling. Section 6.1 overviews the simulation environment used for this investigation. Section 6.2 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 6.3 and 6.4. Finally, we briefly discuss non-i.i.d., random, low-dispersion sampling schemes in Section 6.5 and show simulation results for deterministic low-dispersion sampling applied to non-uniform sampling in Section 6.6.

6.1. Simulation environment

Simulations were written in C++, MATLAB, and Julia,[3] and run using a Unix operating system with a 2.3 GHz processor and 8 GB of RAM (the reason we used MATLAB

and Julia for some of the experiments is simply for ease of implementation; since, as discussed below, the focus is on results in terms of sample count as opposed to runtimes, the choice of programming language is immaterial to the validity of our conclusions). The C++ simulations were run through the Open Motion Planning Library (OMPL) (Şucan et al., 2012). The planning problems simulated in OMPL were rigid body problems, the simulations in MATLAB involved point robots and kinematic chains, and those in Julia incorporated kinodynamic constraints. 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 the chain and kinodynamic simulations), we use as connection radius $r_n = \gamma_{\text{PRM}} (\log(n)/n)^{1/d}$, where $\gamma_{\text{PRM}} = 2.2 \left(1 + 1/d\right)^{1/d} \left(1/\zeta_d\right)^{1/d}$ and ζ_d is again the volume of the unit ball in d-dimensional Euclidean space. This choice ensures asymptotic optimality both for i.i.d. sample sequences (Karaman and Frazzoli, 2011; Janson et al., 2015) 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, which have the advantage that they do not depend on the specific hardware or software we use. (Note that drawing samples represents a trivial fraction of the total algorithm runtime. Furthermore, as mentioned in the introduction, deterministic sampling even allows for possible speedups in computation.) The code for these results can be found at `https://github.com/stanfordasl`.

6.2. Sampling sequences

We consider two deterministic low-dispersion sequences, namely the Halton sequence (Halton, 1960) 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 deterministic low-dispersion sequence of points (Halton, 1960; LaValle et al., 2004). Lattices in this work were implemented as a triangular lattice in two dimensions and a Sukharev grid in higher dimensions (Sukharev, 1971).

Along with the benefits described throughout the paper, lattices also present some challenges (LaValle et al., 2004). First, for basic cubic lattices with the same number of lattice points k per side, the total number of samples n is constrained 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 (Lindemann et al., 2005; Yershova and LaValle, 2004; Yershova et al., 2010), 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 one increases the allowed resolution of n by a factor of d, as it allows all numbers of the form $n = (k-1)^m (k)^{d-m}$.

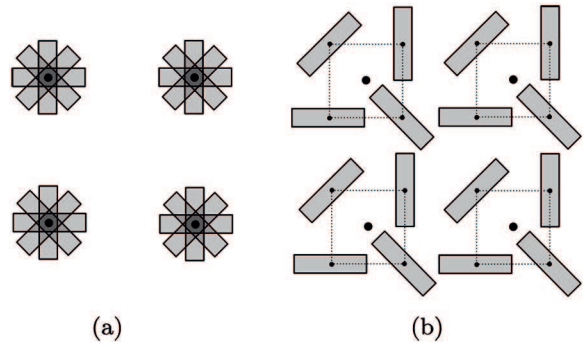

Fig. 3. (a) Rectangular rigid body lattice samples with one rotational and two spatial degrees of freedom. (b) A "spread" lattice formed by spreading the different rigid body orientations spatially around the original point results in a better spatial coverage of the configuration space.

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 $10\pi^\circ$ as an arbitrary angle far from zero (in practice, problems often show a "preferential" direction, for example vertical, so there may be an a priori natural choice of rotation angle).

Finally, we include examples in SE(2) and SE(3) (the configuration spaces for rigid bodies in 2 and 3 dimensions, respectively) because they are standard, even though the cost function is no longer arc length in the configuration space, but closer to arc length in the translational dimensions. As this is technically outside the theory in this paper, we must consider an appropriate analog of deterministic low-dispersion sampling. In the naïve Sukharev lattice, each translational lattice point represents many rotational orientations. For instance, in SE(3), this means there are only k^3 translational points when $n = k^6$, providing a poor extension of the theory in this paper to the rigid body planning problem. A better approximate extension is to "spread" the points, which entails de-collapsing all the rotational orientations at each translational lattice point by spreading them deterministically in the translational dimensions around the original point. An example of this process with one rotational and two spatial dimensions is shown in Figure 3. A similar phenomenon occurs with kinodynamic sampling, in which case it is beneficial to spread velocity samples. For the double-integrator model, as shown in Figure 4, this process entails offsetting velocity samples from their original positions by an amount proportional to their velocities. To then reduce unfavorable structure between neighboring lattice points and increase variation in connection types, alternating samples were rotated 180° for the final implementation in Figure 4(c).

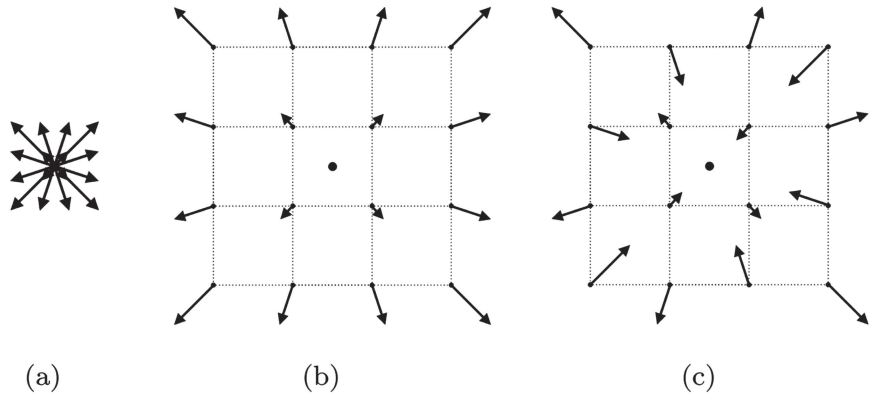

Fig. 4. (a) A lattice for the double-integrator model with two spatial and two velocity degrees of freedom. (b) A "spread" lattice formed by offsetting the velocity samples from their original positions (by an amount proportional to their velocities) to obtain better spatial coverage. (c) The same lattice with alternating samples rotated 180° to increase variation in types of connections and reduce unfavorable structure (this implementation was used for our results).

6.3. 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 two dimensions and three dimensions were generated with a fixed configuration that allowed for several homotopy classes of solutions. A two-dimensional maze with rectangular obstacles was also created (Figure 5(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 two-dimensional maze in Figure 5 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.

To illustrate planning with rectangular obstacles in higher-dimensional space, we constructed a recursive maze obstacle environment. Each instance of the maze consists of two copies of the previous dimension's maze, separated by an obstacle with an opening through the new dimension, as detailed in Janson et al. (2015). Figure 6(a) shows the maze in two dimensions and Figure 6(b) shows the maze in three dimensions with the two copies of the two-dimensional maze in black and the opening in red. Halton and lattice sampling conferred similar benefits in the recursive mazes in two, three, four, five, six, and eight dimensions as they did in other simulations (see Table 1).

Along with the rectangular obstacles, hyperspherical obstacles within a Euclidean unit hypercube were generated to compare performances on a smooth obstacle set with no possibility of axis alignment between samples and obstacles. The setups for two dimensions and three dimensions (Figure 6(c) and (d)) were fixed, while in four dimensions obstacles were randomly generated to match a specified spatial coverage. Again, Halton and lattice sampling consistently outperformed random sampling, as shown in Table 1.

As an additional high-dimensional example, an eight-dimensional kinematic chain planning problem with rotational joints, eight links, and a fixed base was created (Figure 6(e)). The solution required the chain to be extracted from one opening and inserted into the other, as inspired by Lindemann et al. (2005). The chain cost function was set as the sum of the absolute values of the angle differences over all links and the connection radius was thus scaled by $\sqrt{d}\pi$. With this high dimension and new cost function, the Halton and lattice still perform as well as or better than i.i.d. sampling (see 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 6(f). 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 6(g). As already mentioned, the rigid body planning problems are not strictly covered by the theory in this paper, and thus the SE(2) and SE(3) lattices use the spreading method described in Section 6.2. The results, summarized in Table 1, show that Halton and lattice sampling generally outperform i.i.d. random sampling.

Lastly, planning problems with kinodynamic constraints were simulated within the Julia environment using the gDPRM algorithm defined in Section 5.4, with reference to Schmerling et al. (2015a). The connection radius was computed using equation (8) in Theorem 4. First, a double-integrator model with two spatial dimensions and two velocity dimensions was posed to simulate a system with drift (Figure 6(h)). To obtain better spatial coverage, the lattice was spread as described in Section 6.2. Second, results were simulated for the Reeds–Shepp car system (Reeds

Table 1. Summary of results; recall that entries represent quantities relative to i.i.d. sampling, hence the percentage signs. Each entry is divided by the results of i.i.d. sampling (averaged over 50 runs). For Halton sampling and lattice sampling, the number of samples at which 90% success is achieved and the costs at a medium number of samples (near 700) and a high number of samples are shown (highest samples simulated, always 3000 or greater). Note that nearly all table entries are below 100%, meaning the Halton and lattice sampling outperformed i.i.d. sampling.

		Halton vs i.i.d.			Lattice vs i.i.d.		
Dim.	Obstacles	90% Success	Medium	High	90% Success	Medium	High
2	Rectangular	38%	118%	80%	15%	56%	80%
3	Rectangular	36%	88%	94%	19%	80%	87%
2	Rectangular Maze	13%	98%	99%	13%	100%	99%
2	Sphere	16%	93%	99%	7%	93%	99%
3	Sphere	36%	97%	100%	8%	97%	99%
4	Sphere	100%	97%	97%	100%	97%	100%
2	Recursive Maze	33%	100%	100%	18%	100%	100%
3	Recursive Maze	22%	95%	99%	22%	96%	98%
4	Recursive Maze	56%	95%	98%	56%	100%	100%
5	Recursive Maze	45%	97%	96%	60%	95%	96%
6	Recursive Maze	56%	95%	97%	75%	94%	96%
8	Recursive Maze	56%	98%	99%	75%	99%	99%
8	Chain	67%	112%	91%	7%	76%	87%
3	SE(2)	81%	96%	100%	81%	101%	101%
6	SE(3)	32%	96%	93%	42%	94%	95%
4	Double-Integrator	53%	90%	93%	30%	92%	96%
3	Reeds–Shepp Car	44%	97%	99%	44%	98%	100%

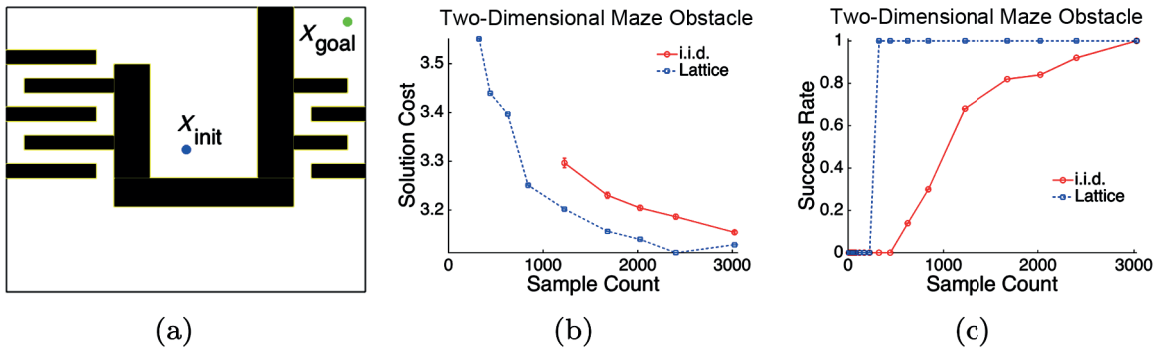

Fig. 5. (a) The planning setup for a point robot with rectangular obstacles in a two-dimensional maze. (b) and (c) The results for solution cost and success rate versus sample count (averaged over 50 runs). For clarity we only report data for i.i.d. sampling and lattice sequences; results including Halton sampling are reported in Table 1. Only points with greater than 50% success are shown in (b).

and Shepp, 1990), a regular driftless control-affine system with one rotational and two translational degrees of freedom (Figure 6(h)). The Reeds–Shepp car is constrained to move with a unit speed (forwards or backwards) and turn with a fixed radius. Although these dynamics are different from those discussed in Section 5.4, we can still define gDPRM with respect to Schmerling et al. (2015b) and evaluate the benefits of deterministic low-dispersion sampling. Again, spreading was used in the spatial dimension. With the addition of kinodynamic constraints, the Halton and lattice sampling continue to outperform i.i.d. random sampling, as summarized in Table 1.

6.4. Summary of results

Table 1 shows a summary of the results from the simulations detailed in Section 6.3. 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 as a ratio of the same quantity for i.i.d. sampling. Additionally, the solution costs at medium and high sampling counts are shown, again as a ratio of the cost using the same number of i.i.d. samples. For all cases the lattice sampling finds a solution with fewer or an equal number of samples and of lower or essentially equal cost

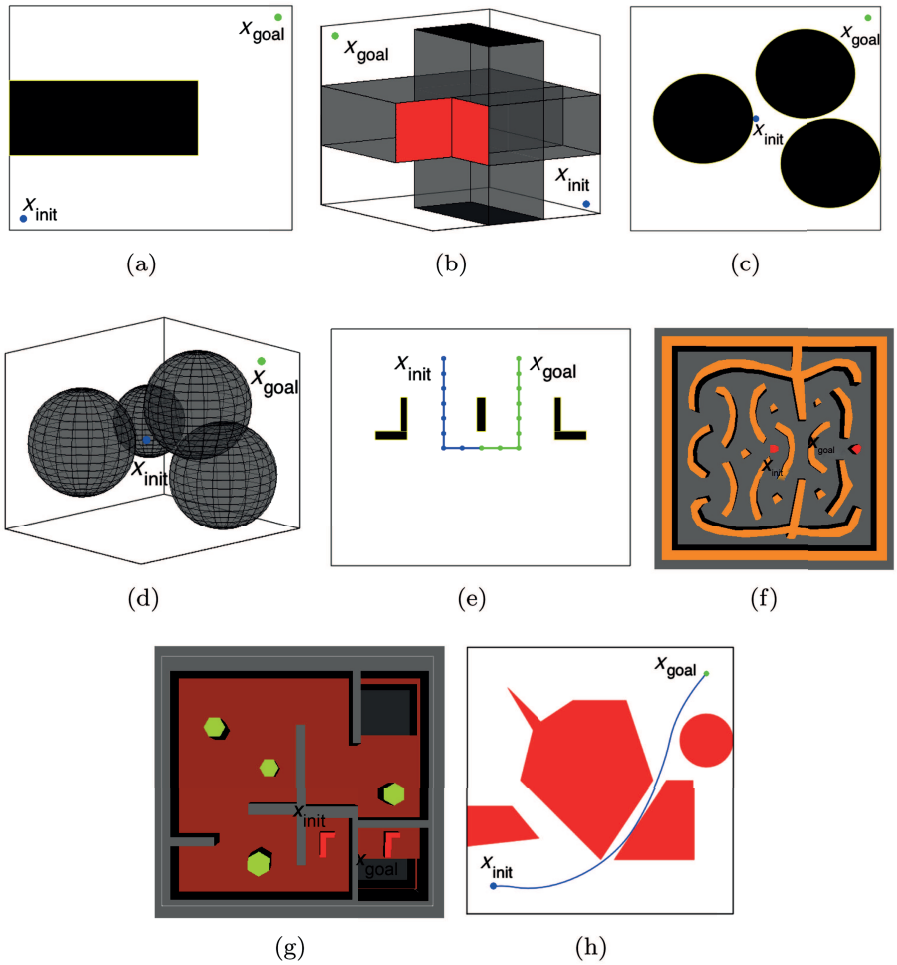

Fig. 6. Images of the recursive maze planning problem in two dimensions (a) and three dimensions (b) and the spherical obstacle sets in two dimensions (c) and three dimensions (d). Also shown are an eight-dimensional kinematic chain planning problem in (e) and the OMPL rigid body planning problems for SE(2) and SE(3) in (f) and (g) respectively. Lastly, (h) shows the setup for the double-integrator and Reeds–Shepp car. Note that the problems in (h) have kinodynamic constraints, where we are not only considering straight line connections, but also ones with curvature. A summary of results can be found in Table 1.

to 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 deterministic low-dispersion sequences particularly outperform random sampling in terms of number of samples required for a 90% success rate.

6.5. Non-deterministic sampling sequences

The above simulations showed deterministic lattice sampling, with a fixed rotation around each axis, and the deterministic Halton sequence outperformed 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 7 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, for example eliminating axis alignment issues while still enjoying deterministic guarantees (see Section 4). Hsu et al. (2006) reported on similar advantages to randomization for finding a feasible solution. Their further study represents an interesting direction for future research.

6.6. Non-uniform sampling

Following the discussion in Section 5.3, we consider the extension of deterministic low-dispersion sampling to two planning algorithms with non-uniform sampling. First, using OMPL's default RRT* (Karaman and Frazzoli, 2011) implementation, the SE(3) problem in Figure 6(g) was simulated with samples drawn from i.i.d. and Halton sequences.

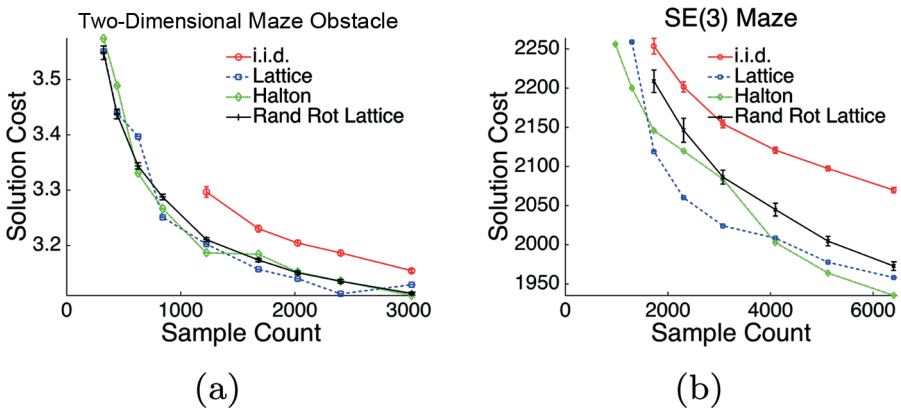

Fig. 7. Results for deterministic and non-deterministic low-dispersion sampling. "Rand Rot Lattice" refers to a randomly rotated lattice.

Table 2. Results for non-uniform sampling algorithms. Each entry shows the results of Halton sampling divided by the results of i.i.d. sampling (averaged over 50 runs). For RRT*, the number of samples at which 90% success is achieved and the costs at a medium and high number of samples are shown, while BIT* compares time rather than sample count[4]. Again note that all table entries are at or below 100%, meaning Halton sampling outperformed i.i.d. sampling.

			Halton vs i.i.d.		
Algorithm	Dim.	Obstacles	90% Success	Medium	High
RRT*	6	SE(3)	32%	95%	99%
BIT*	2	Rectangular	24%	100%	100%
BIT*	3	Sphere	51%	99%	100%
BIT*	8	Chain	45%	100%	95%
BIT*	3	SE(2)	62%	97%	99%
BIT*	6	SE(3)	30%	97%	98%

Second, a suite of tests was performed with the BIT* algorithm (Gammell et al., 2015), including one from each geometric obstacle type in Table 1 except the recursive maze.[5] Paralleling the aforementioned implementation approach, BIT* was implemented in MATLAB (for the rectangular, spherical, and chain problems) and OMPL (for the SE(2) and SE(3) problems) with samples drawn from i.i.d. and Halton sequences and only relevant samples kept. The non-uniformity of RRT* samples arises from the use of a steering function, which limits individual connection distance when growing the tree, thus only allowing for nearby connections, while BIT*'s non-uniformity is a consequence of adaptively resampling only relevant samples at each batch. The results, shown in Table 2, show similar benefits to those in Table 1 for uniform sampling, demonstrating that non-uniform sampling strategies derive the same improvement in solution quality as uniform sampling does, when i.i.d. sampling is replaced by low-dispersion sampling in the pipeline that generates the non-uniform samples.

7. 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 performance 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 deepen our study of deterministic kinodynamic motion planning, in particular in terms of tailored notions of sampling sequences and dispersion and more general dynamical models. Second, it is of interest to extend the results herein to other classes of sampling-based motion planning algorithms (beyond the ones studied in this paper), especially the large class of *anytime* algorithms (e.g. RRT/RRT*). This leads directly to 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 (Lindemann et al., 2005; Yershova and LaValle, 2004; Yershova et al., 2010). It may also be of interest to study low-dispersion sampling strategies that incorporate prior knowledge of the problem by sampling non-uniformly, as discussed in Section 5.3. Fourth, 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. Fifth, 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 further investigate the issue of certification for sampling-based planners, for example in the context of trajectory planning for drones or self-driving cars.

Acknowledgements

The authors would like to thank Javier V Gómez for his help with the implementation of the algorithms in OMPL. This work was originally presented at the 17th International Symposium on Robotics Research, ISRR 2015. This extended and revised version includes a novel section on extensions of this work to k-nearest-neighbor algorithms, non-PRM algorithms, non-lattice sampling, non-uniform sampling, and kinodynamic motion planning. It also includes a significantly extended simulation section.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by NASA under the Space Technology Research Grants Program (grant NNX12AQ43G). Lucas Janson was partially supported by the NIH National Institutes of Health (training grant T32GM096982). Brian Ichter was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

Notes

1. For $f, g : \mathbb{N} \to \mathbb{R}$, we say $f \in O(g)$ if there exist $n_0 \in \mathbb{N}$ 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 exist $n_0 \in \mathbb{N}$ 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 $\lim_{n\to\infty} f(n)/g(n) = \infty$.
2. In particular, `Near`(V, v, r) returns $\{u \in V : c(\sigma(u, v)) < r\}$, `CollisionFree`$(u, v)$ checks $\sigma(u, v)$ for obstacle intersection, and `ShortestPath` defines the length of a path $x_0, \ldots, x_M$ by $\sum_{i=1}^{M} c(\sigma(x_{i-1}, x_i))$.
3. http://arxiv.org/abs/1209.5145
4. Because the default implementation of BIT* uses time instead of sample count as its tuning parameter, we compare BIT* across different sample sequences using the same time instead of sample count. However, since the only difference between sample sequences is how they are sampled, sample count and time are in an essentially one-to-one relationship.
5. BIT*'s non-uniform sampling is mainly advantageous when the optimal solution is shorter than the diameter of the configuration space, which is not the case for the recursive maze environments.

References

Alterovitz R, Patil S and Derbakova A (2011) Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning. In: *Proceedings of the IEEE conference on robotics and automation*, Shanghai, China, 9–13 May 2011, pp. 3706–3712. Piscataway: IEEE.

Arslan O and Tsiotras P (2013) Use of relaxation methods in sampling-based algorithms for optimal motion planning. In: *Proceedings of the IEEE conference on robotics and automation*, Karlsruhe, Germany, 6–10 May 2013, pp. 2421–2428. Piscataway: IEEE.

Barraquand J, Kavraki L, Motwani R, et al. (2000) A random sampling scheme for path planning. *The International Journal of Robotics Research* 16(6): 759–774.

Bohlin R and Kavraki L (2000) Path planning using lazy PRM. In: *Proceedings of the IEEE conference on robotics and automation*, San Francisco, CA, USA, 24–28 April 2000, pp. 521–528. Piscataway: IEEE.

Boor V, Overmars MH and van der Stappen AF (1999) The Gaussian sampling strategy for probabilistic roadmap planners. In: *Proceedings of the IEEE conference on robotics and automation*, Detroit, MI, 10–15 May 1999, pp. 1018–1023. Piscataway: IEEE.

Branicky MS, LaValle SM, Olson K, et al. (2001) Quasi-randomized path planning. In: *Proceedings of the IEEE conference on robotics and automation*, Seoul, South Korea, 21–26 May 2001, pp. 1481–1487. Piscataway: IEEE.

Chen CT (1995) *Linear System Theory and Design*. Oxford: Oxford University Press.

Deheuvels P (1983) Strong bounds for multidimensional spacings. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 64(4): 411–424.

Dobson A, Moustakides G and Bekris KE (2015) Geometric probability results for bounding path quality in sampling-based roadmaps after finite computation. In: *Proceedings of the IEEE conference on robotics and automation*. Seattle, WA, USA, 26–30 May 2015, pp. 4180–4186. Piscataway: IEEE.

Gammell JD, Srinivasa SS and Barfoot TD (2015) Batch Informed Trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: *Proceedings of the IEEE conference on robotics and automation*. Seattle, WA, USA, 26–30 May 2015, pp. 3067–3074. Piscataway: IEEE.

Halton JH (1960) On the efficiency of certain quasirandom sequences of points in evaluating multidimensional integrals. *Numerische Mathematik* 2(1): 84–90.

Hsu D, Latombe JC and Kurniawati H (2006) On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research* 25(7): 627–643.

Hsu D, Latombe JC and Motwani R (1999) Path planning in expansive configuration spaces. *International Journal of Computational Geometry & Applications* 9(4): 495–512.

Janson L, Schmerling E, Clark A, et al. (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.

Kailath T (1980) *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall.

Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30(7): 846–894.

Kavraki LE, Švestka P, Latombe JC, et al. (1996) Probabilistic roadmaps for path planning in high-dimensional spaces. *IEEE Transactions on Robotics and Automation* 12(4): 566–580.

Lavalle S (2006) *Planning Algorithms*. Cambridge: Cambridge University Press.

LaValle SM and Kuffner JJ (2001) Randomized kinodynamic planning. *The International Journal of Robotics Research* 20(5): 378–400.

LaValle SM, Branicky MS and Lindemann SR (2004) On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research* 23(7–8): 673–692.

Lindemann SR, Yershova A and LaValle SM (2005) Incremental grid sampling strategies in robotics. In: *Workshop on algorithmic foundations of robotics*, Utrecht/Zeist, The Netherlands, 11–13 July 2004, pp. 313–328. Berlin: Springer-Verlag.

Niederreiter H (1992) *Random Number Generation and Quasi-Monte Carlo Methods* (*CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63). Philadelphia, PA: Society for Industrial & Applied Mathematics.

Orlin JB, Madduri K, Subramani K, et al. (2010) A faster algorithm for the single source shortest path problem with few distinct positive lengths. *Journal of Discrete Algorithms* 8(2): 189–198.

Phillips JM, Bedrossian N and Kavraki LE (2004) Guided expansive spaces trees: A search strategy for motion- and cost-constrained state spaces. In: *Proceedings of the IEEE conference on robotics and automation*, New Orleans, LA, 26 April–1 May 2004, pp. 3968–3973. Piscataway: IEEE.

Pivtoraiko M, Knepper RA and Kelly A (2009) Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics* 26(3): 308–333.

Plaku E, Bekris KE, Chen BY, et al. (2005) Sampling-based roadmap of trees for parallel motion planning. *IEEE Transactions on Robotics* 21(4): 597–608.

Reeds JA and Shepp LA (1990) Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics* 145(2): 367–393.

Schmerling E, Janson L and Pavone M (2015a) Optimal sampling-based motion planning under differential constraints: The drift case with linear affine dynamics. In: *Proceedings of the IEEE conference on decision and control*. Osaka, Japan, 15–18 December 2015 pp. 2574–2581. Piscataway: IEEE.

Schmerling E, Janson L and Pavone M (2015b) Optimal sampling-based motion planning under differential constraints: The driftless case. In: *Proceedings of the IEEE conference on robotics and automation*, Seattle, WA, 26–30 May 2015, pp. 2368–2375. Piscataway: IEEE.

Stentz A (1995) Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics & Automation* 10(3): 89–100.

Şucan IA, Moll M and Kavraki LE (2012) The Open Motion Planning Library. *IEEE Robotics and Automation Magazine* 19(4): 72–82.

Sukharev AG (1971) Optimal strategies of the search for an extremum. *USSR Computational Mathematics and Mathematical Physics* 11(4): 119–137.

Thrun S, Burgard W and Fox D (2005) *Probabilistic Robotics*. Cambridge, MA: MIT Press.

Yershova A and LaValle SM (2004) Deterministic sampling methods for spheres and SO(3). In: *Proceedings of the IEEE conference on robotics and automation*, New Orleans, LA, USA, 26 April–1 May 2004, pp. 3974–3980. Piscataway: IEEE.

Yershova A, Jain S, Lavalle SM, et al. (2010) Generating uniform incremental grids on SO(3) using the Hopf fibration. *The International Journal of Robotics Research* 29(7): 801–812.