On the Probabilistic Foundations of Probabilistic Roadmap Planning

David Hsu[*]   Jean-Claude Latombe[†,1]   Hanna Kurniawati[*]

[*]*Department of Computer Science*
National University of Singapore
Singapore, 117543, Singapore

[†]*Department of Computer Science*
Stanford University
Stanford, CA 94305, USA

Abstract

Why is probabilistic roadmap (PRM) planning probabilistic? How does the probability measure used for sampling a robot's configuration space affect the performance of a PRM planner? These questions have received little attention so far. This paper tries to fill this gap, with the intent to identify promising directions to improve future planners. It introduces the probabilistic foundations of PRM planning and examines previous work in this context. It shows that the success of PRM planning depends mainly and critically on favorable "visibility" properties of a robot's configuration space. A promising direction for speeding up PRM planners is to infer partial knowledge on such properties from both workspace geometry and information gathered during roadmap construction, and to use this knowledge to adapt the probability measure for sampling. This paper also shows that the choice of the sampling source— pseudo-random or deterministic—has small impact on a PRM planner's performance, compared to that of the sampling measure. These conclusions are supported by both theoretical and empirical results.

1   Introduction

Probabilistic roadmap (PRM) planners [10, 27] solve seemingly difficult motion planning problems, such as the one shown in Figure 1, where the robot's configuration space $\mathcal{C}$ is six-dimensional and the environment is described by tens of thousands of triangles. While an algebraic planner would be overwhelmed by the prohibitive cost of computing an exact representation of the free space $\mathcal{F}$, defined as the collision-free subset of $\mathcal{C}$, a PRM planner builds only an extremely simplified representation of $\mathcal{F}$, called a *probabilistic roadmap*. A roadmap is a graph whose nodes are configurations sampled from $\mathcal{F}$ according to a suitable probability measure and whose edges are simple collision-free paths, *e.g.*, straight-line segments, between sampled configurations. PRM planners work surprisingly well in practice. Why?

Previous work has partially addressed this question by identifying and formalizing free space properties that guarantee good performance for a PRM planner using the uniform sampling measure (*e.g.*, [21, 25, 26, 32, 48]) . Several systematic experimental studies have also compared various PRM planners, in terms of their sampling and connection strategies (*e.g.*, [12, 14, 39]). However, the underlying question "Why are PRM planners probabilistic?" has received little attention so far, and consequently the importance of probabilistic sampling measures for PRM planning remains poorly understood. Since no inherent randomness or uncertainty exists in the classic formulation of motion planning problems like the one depicted in Figure 1, one may wonder why probabilistic sampling helps to solve them.

---

[1]Part of this work was completed while the author was at the National University of Singapore, supported by the Kwan Im Thong Hood Cho Temple Professorship.
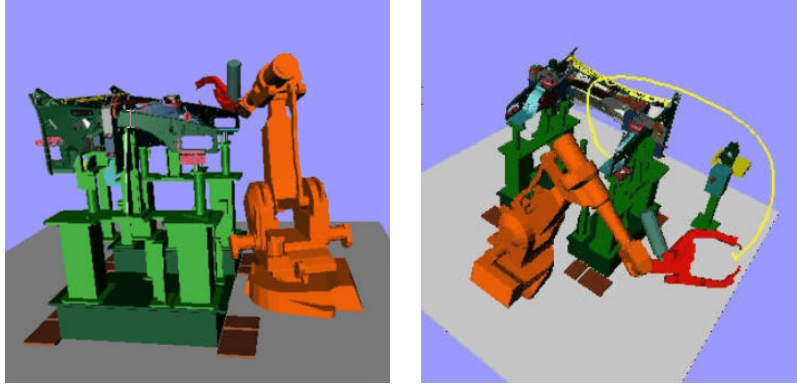
Figure 1. A practical motion planning problem.

In this paper, we attempt to fill this gap, with the intent to identify promising directions to improve future PRM planners. We introduce the probabilistic foundations of PRM planning (Section 2). We examine previous work in this context and argue that the empirical success of PRM planning tells us as much about the nature of motion planning problems encountered in practice as about PRM planning itself (Section 3). We emphasize the key distinction between the sampling *measure*, a notion firmly rooted in probability theory, and the sampling *source*, and show that the source has small impact on a planner's performance, compared to the measure (Sections 4 and 5). Finally, we briefly discuss the role of connection strategies in PRM planning (Section 6). Details of the software implementation used in our experiments are described in Appendix A.

The main questions addressed in this paper are summarized below:

- **Why is PRM planning probabilistic?** A foundational choice in PRM planning is to avoid the prohibitive cost of computing an exact representation of $\mathcal{F}$. So, a PRM planner never knows the exact shape of $\mathcal{F}$, in particular, its connectivity. It works very much like a robot exploring an *unknown* environment to build a map. At any moment during planning, many hypotheses on the shape of $\mathcal{F}$ are consistent with the information gathered so far. The probability measure for sampling $\mathcal{F}$ derives from this uncertainty. Hence, PRM planning trades the cost of computing $\mathcal{F}$ exactly against the cost of dealing with uncertainty. This choice is beneficial only if probabilistic sampling leads to a roadmap that is much smaller in size than that of an exact representation of $\mathcal{F}$ and still represents $\mathcal{F}$ well enough to answer motion planning queries correctly.

- **What does the empirical success of PRM planning imply?** One can think of the nodes of a roadmap as a network of guards watching over $\mathcal{F}$. To guarantee that a PRM planner finds a solution quickly whenever one exists, $\mathcal{F}$ should satisfy favorable "visibility" properties. A key contribution of PRM planning has been to reveal that many free spaces encountered in practice satisfy such properties, despite their high algebraic complexity. In retrospective, this can be explained by the fact that poor visibility properties in $\mathcal{F}$ are caused by narrow passages, which are unstable under small perturbations of the geometry of robots or obstacles. Narrow passages are therefore unlikely to occur by accident. Furthermore, visibility properties can be defined in terms of volume ratios over certain subsets of $\mathcal{F}$ and do not directly depend on $\dim(\mathcal{C})$, the dimensionality of $\mathcal{C}$. This explains why PRM planning scales up reasonably well when $\dim(\mathcal{C})$ increases.

- **How important is the sampling measure?** In every PRM planner, a probability measure prescribes how sampled configurations are distributed over $\mathcal{F}$. Since visibility properties are generally not uniformly favorable over $\mathcal{F}$, this measure plays a central role in determining the efficiency of PRM plan-

ning by allocating a higher density of samples to regions expected to have poor visibility properties, using the hypotheses on the shape of $\mathcal{F}$. For computational efficiency, most existing PRM planners use only heuristic estimates of visibility properties to locate such regions, by using workspace geometry or by exploiting information gathered during roadmap construction, but experiments show that the resulting non-uniform sampling measures dramatically improve the planners' performance.

- **How important is the sampling source?** To sample a configuration in $\mathcal{C}$, a PRM planner needs a source S of uniformly distributed, random or deterministic numbers. For example, the planner may use S to sample a point uniformly from the unit hypercube $[0, 1]^{\dim(\mathcal{C})}$ and then maps the point into $\mathcal{C}$ according to a given probabilistic sampling measure. The source S has limited effect on the efficiency of PRM planning. When $\dim(\mathcal{C})$ is small, low-discrepancy or low-dispersion deterministic sources achieve some speedup over random sources [33]. The speedup is, however, very modest compared to that achieved by good sampling measures. It also fades away quickly, as $\dim(\mathcal{C})$ increases.

This paper does not introduce any new PRM planner or sampling strategy. Instead, its main contribution, based on a synthesis of previous work, is to articulate a coherent framework centered on the probabilistic foundations of PRM planning. It re-examines several ideas, considered separately before, in this framework and also establishes a new theoretical result (Theorem 2 in Section 3.2) that lends further support to the importance of visibility properties for PRM planning. Overall, the paper brings new understanding of what makes PRM planning effective, which in turn may help us to design better planners in the future.

2   Why is PRM planning probabilistic?

For many robots, computing an exact representation of the free space $\mathcal{F}$ takes prohibitive time, but fast, exact algorithms exist to test whether a given configuration or path is collision-free, and some of them can handle large, complex geometric models efficiently (see [36] for a survey). PRM planners use two *probes* based on such algorithms to access geometric information on the configuration space $\mathcal{C}$:

- For any $q \in \mathcal{C}, \texttt{FreeConf}(q)$ is true if and only if $q \in \mathcal{F}$.

- For any pair $q, q' \in \mathcal{C}, \texttt{FreePath}(q, q')$ is true if and only if q and q' can be connected with a straight-line path lying entirely in $\mathcal{F}$.

The choice of using only these two probes is foundational. Since a PRM planner does not compute the exact shape of $\mathcal{F}$, it never gains this information. At any moment during planning, many hypotheses on the shape of $\mathcal{F}$ are consistent with the information gathered so far by the probes, and each hypothesis has some probability of being correct. The probabilistic nature of PRM planners comes from the fact that this uncertainty is modeled implicitly by a probability measure over the set of hypotheses.

In the rest of this paper, we use the following reference PRM planner, $\texttt{BasicPRM}$, which takes as inputs two query configurations q_1 and q_2 in $\mathcal{F}$ as well as an integer N. To connect q_1 and q_2, $\texttt{BasicPRM}$ builds a roadmap R by sampling up to N configurations from $\mathcal{F}$. The nodes of R consist of the sampled configurations as well as the two query configurations. The edges of R represent collision-free straight-line paths between the nodes.

Procedure 1 `BasicPRM`(q_1, q_2, N)

1: **if** `FreePath`(q_1, q_2) is true **then return** the path between q_1 and q_2.
2: Initialize the roadmap R with two nodes, q_1 and q_2.
3: **repeat**
4:     Sample a configuration q from $\mathcal{C}$ uniformly at random.
5:     **if** `FreeConf`(q) is true **then** add q as a new node of R.
6:     **for** every node v of R such that $v \neq q$ **do**
7:         **if** `FreePath`(q, v) is true **then** add (q, v) as a new edge of R.
8: **until** q_1 and q_2 are in the same connected component of R or R contains $N + 2$ nodes.
9: **if** q_1 and q_2 are in the same connected component of R **then**
10:     **return** a path between q_1 and q_2.
11: **else**
12:     **return** `NoPath`.

Most PRM planners use more sophisticated sampling strategies than the uniform random one in Line 4 of `BasicPRM`. A sampling strategy can be formally specified as a pair (π, S), where

- π is a probability measure that prescribes how sampled configurations are distributed over $\mathcal{C}$, and

- S is a source of uniformly distributed, random or deterministic numbers.

The simplest measure π is the uniform one, which makes all the configurations to be sampled with equal probability. More sophisticated measures are non-uniform over $\mathcal{C}$ and sometimes change during the roadmap construction. We will show in Sections 4 and 5 that designing good sampling measures is one of the most promising ways to speed up PRM planning, while sampling sources have only limited impact on the overall performance.

In Lines 6 and 7, the planner checks for connection between every pair of nodes in R, resulting in $O(N^2)$ invocations of `FreePath`. To improve computational efficiency, most PRM planners use connection strategies that only attempt to connect selected pairs of nodes, reducing the number of invocations of `FreePath` to roughly $O(N)$. The implementation of `BasicPRM` used in our experiments also incorporates such strategies, as described in Appendix A.2.

`BasicPRM` has two possible outcomes. If it returns a path, the answer is always correct. However, the `NoPath` answer may or may not be correct, because, after N nodes have been sampled, R still may not capture the connectivity of $\mathcal{F}$ well enough to connect the query configurations.

Let us now return to the question "Why is PRM planning probabilistic?". Suppose that while constructing a roadmap, the planner could maintain a representation (H, η), where H is the set of all hypotheses over the shape of $\mathcal{F}$ and η is a probability measure that assigns to each hypothesis in H the probability of it being correct. Then, in each iteration of the planner, the optimal sampling measure $\hat{\pi}$ would be the one that minimizes the expected number of remaining iterations until the roadmap connects q_1 and q_2, whenever these two configurations lie in the same connected component of $\mathcal{F}$. In principle, $\hat{\pi}$ could be inferred from (H, η). In practice, maintaining (H, η) explicitly would be too expensive. So, existing PRM planners use heuristics to approximate the optimal sampling measure (see Section 4). Many of them make hypotheses on the shape of $\mathcal{F}$, *e.g.*, assuming that certain regions of $\mathcal{F}$ are likely to contain narrow passages and thus increasing the sampling density there.

3 What does the empirical success of PRM planning imply?

`BasicPRM` has some probability γ of returning an incorrect `NoPath` answer, but the effectiveness of PRM planning demonstrated on many practical problems indicates that γ is usually small. Experiments show
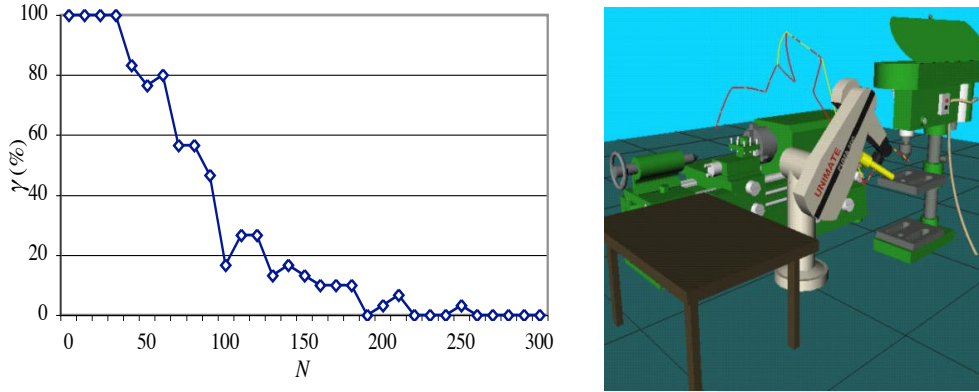
Figure 2. The experimental convergence rate of `BasicPRM`. The graph plots the percentage of unsuccessful outcomes out of 100 independent runs for the same query in the environment shown on the right, as the number of roadmap nodes increases.
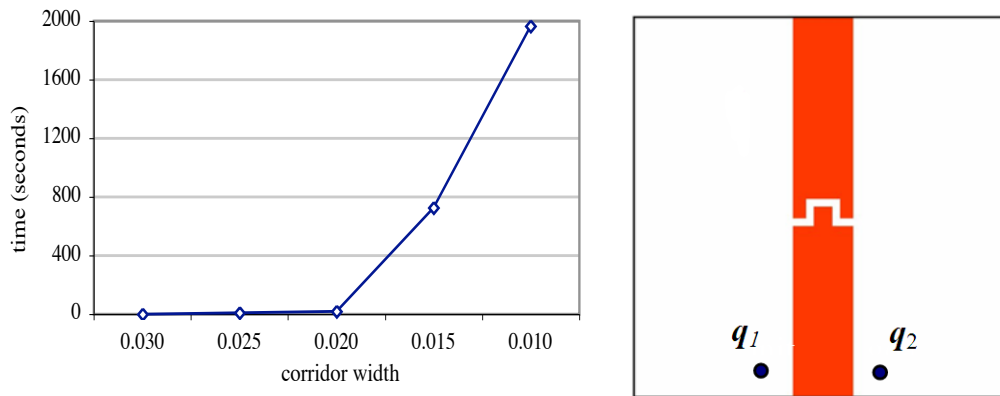

Figure 3. A difficult example for PRM planning. $\mathcal{F}$ consists of two rectangular chambers connected by a narrow corridor and is bounded by a square box of size 1 by 1. The plot shows the average running time for `BasicPRM` to connect the two query configurations q_1 and q_2, as the corridor width decreases.

that even in complex geometric environments, γ often converges to 0 quickly, as the number of roadmap nodes increases. See Figure 2 for an example. However, one can also easily construct apparently simple environments, like the one in Figure 3, where PRM planners perform poorly. Together, these two examples suggest that many environments encountered in practice satisfy favorable properties that PRM planners exploit well. What are these properties?

In this section, we argue that $\mathcal{F}$ must satisfy a visibility property called *expansiveness*. We first review a result from [21, 26] showing that if $\mathcal{F}$ is expansive, then `BasicPRM` answers planning queries correctly with high probability (Section 3.1). We then establish a new result, which can be seen as a weak converse of this earlier result: if $\mathcal{F}$ is poorly expansive, then there exist queries for which we cannot expect `BasicPRM` to work well (Section 3.2). In light of these two theoretical results, PRM's empirical success suggests that most free spaces encountered in practice have favorable expansiveness properties. In Section 3.3, we discuss why this is the case.
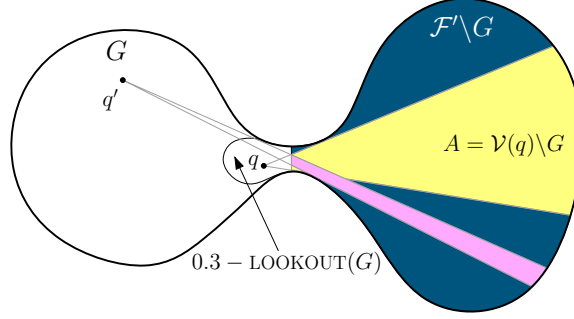
Figure 4. The β-LOOKOUT. $\mathcal{F}'$ consists of two large open regions connected by a narrow passage. G is the subset of $\mathcal{F}'$ to the left of the vertical line segment drawn roughly half way in the passage. For $\beta = 0.3$, $q \in$ 0.3-LOOKOUT(G), because $\mu(A) \geq 0.3 \times \mu(\mathcal{F}'\backslash G)$, while $q' \notin$ 0.3-LOOKOUT(G).

3.1   Performance of `BasicPRM` in expansive free spaces

Let us start with a few definitions that will lead to the concept of an expansive free space. We say that two configurations q and q' in $\mathcal{F}$ *see* each other if `FreePath`(q, q') is true. The visibility set of a configuration $q \in \mathcal{F}$ is the set $\mathcal{V}(q) = \{q' \in \mathcal{F} \mid$ `FreePath`(q, q') is true$\}$. The visibility set of a set M of configurations in $\mathcal{F}$ is then $\mathcal{V}(M) = \bigcup_{q \in M} \mathcal{V}(q)$.

Let us now define the notion of an ε-good free space [26]. Intuitively, it is a space in which every configuration q has a relatively large visibility set, so that it is easy to sample a set of configurations that, collectively, see most of $\mathcal{F}$.

Definition 1 Let ε be a constant in $\in (0, 1]$. A point $q \in \mathcal{F}$ is ε-*good* if it sees at least an ε-fraction of $\mathcal{F}$, *i.e.*, if $\mu(\mathcal{V}(q)) \geq \varepsilon\,\mu(\mathcal{F})$, where $\mu(S)$ denotes the volume of a subset $S \subseteq \mathcal{C}$. The free space $\mathcal{F}$ is ε-*good* if every point $q \in \mathcal{F}$ is ε-good.

However, ε-goodness is too weak to imply anything on the *connectivity* of the constructed roadmap. A stronger property is needed to "connect" a visibility set to its complement in $\mathcal{F}$. This leads to the notion of the lookout of a subset G of $\mathcal{F}$ [21].

Definition 2 Let $\mathcal{F}'$ be a connected component of $\mathcal{F}$ and G be any subset of $\mathcal{F}'$. Let β be a constant in $(0, 1]$. The β-LOOKOUT of G is the set of all points in G such that each point sees at least a β-fraction of the complement of G: β-LOOKOUT$(G) = \{q \in G \mid \mu(\mathcal{V}(q)\backslash G) \geq \beta\,\mu(\mathcal{F}'\backslash G)\}$.

See Figure 4 for an illustration of this definition.

Suppose that the volume of β-LOOKOUT(G) is $\alpha\,\mu(G)$. If either α or β is small, then it would be difficult to sample a point in G and another in $\mathcal{F}'\backslash G$ so that the two points see each other and thus build a roadmap connecting G and $\mathcal{F}'\backslash G$. An example is the free space of Figure 3 when the corridor is very narrow. Hence, the concept of expansiveness defined below [21].

Definition 3 Let ε, α, and β be constants in $(0, 1]$. A connected component $\mathcal{F}'$ of $\mathcal{F}$ is $(\varepsilon, \alpha, \beta)$-*expansive* if (*i*) every point $q \in \mathcal{F}'$ is ε-good and (*ii*) for any set M of points in $\mathcal{F}'$, $\mu(\beta$-LOOKOUT$(\mathcal{V}(M))) \geq \alpha\,\mu(\mathcal{V}(M))$. The free space $\mathcal{F}$ is $(\varepsilon, \alpha, \beta)$-*expansive*, if its connected components are all $(\varepsilon, \alpha, \beta)$-expansive.

The following theorem, which is based directly on the results established in [21, 26], characterizes the performance of `BasicPRM` in an expansive free space.

Theorem 1 *Let $\mathcal{F}$ be $(\varepsilon, \alpha, \beta)$-expansive, and p_1 and p_2 be two configurations in the same connected component of $\mathcal{F}$.* `BasicPRM`(p_1, p_2, N) *returns a path connecting p_1 and p_2 with probability converging to 1 at an exponential rate, as N increases.*

The precise bound given in [21] shows that the failure probability of `BasicPRM` is upper-bounded by $\left(\frac{c_1}{\varepsilon\alpha}\right)\exp(c_2\varepsilon\alpha(-N + \frac{c_3}{\beta}))$, for some fixed positive constants c_1, c_2, and c_3. This theoretical bound is consistent with the convergence plot in Figure 2 and similar plots in earlier work (*e.g.*, [45]). The exponential convergence rate of this probability bound is crucial. Mere asymptotic convergence, often regarded as the definition of probabilistic completeness, is insufficient to ensure the practical effectiveness of PRM planners.

When ε, α, and β are reasonably large, the expansiveness of $\mathcal{F}$ guarantees that the visibility set $\mathcal{V}(M)$ of any set M of configurations in a connected component $\mathcal{F}'$ of $\mathcal{F}$ has a large lookout. So, it is easy to construct a small roadmap that captures the connectivity of $\mathcal{F}$ well. The values of ε, α, and β measure the extent to which $\mathcal{F}$ is expansive. The larger these values are, the smaller N needs to be for `BasicPRM` to answer queries correctly, resulting in faster computation time.

3.2 Performance of `BasicPRM` in poorly expansive free spaces

Let us now consider free spaces that are poorly expansive. The following theorem states that when α and β are too small, we cannot expect `BasicPRM` to give a correct answer for all query configurations, even if it constructs a very large roadmap.

Theorem 2 *For any $\varepsilon > 0$, any arbitrarily large $N > 0$, and any $\gamma \in (0, 1]$, there exist α_0 and β_0 such that if $\mathcal{F}$ is not $(\varepsilon, \alpha, \beta)$-expansive for $\alpha \geq \alpha_0$ and $\beta \geq \beta_0$, then there exists a pair of query configurations p_1 and p_2 in the same connected component of $\mathcal{F}$ such that* `BasicPRM`(p_1, p_2, N) *fails to return a path between q_1 and q_2 with probability greater than γ.*

Proof. If $\mathcal{F}$ is not $(\varepsilon, \alpha, \beta)$-expansive for $\alpha \geq \alpha_0$ and $\beta \geq \beta_0$, then, by definition, there exists a connected component $\mathcal{F}'$ of $\mathcal{F}$ and a subset $G \subseteq \mathcal{F}'$ with a small lookout such that

$$\mu(\beta_0\text{-LOOKOUT}(G)) < \alpha_0\,\mu(G). \tag{1}$$

Let us pick configurations q_1 and q_2 uniformly at random from G and from its complement $\mathcal{F}'\backslash G$, respectively. We want to derive a lower bound on the probability that `BasicPRM`(q_1, q_2, N) generates a roadmap that does not contain a path between q_1 and q_2.

Let A denote the event that there is a path in R between q_1 and q_2. Since $q_1 \in G$ and $q_2 \in \mathcal{F}'\backslash G$, this path must contain two consecutive nodes q and q' of R such that $q \in G$, $q' \in \mathcal{F}'\backslash G$, and $q' \in \mathcal{V}(q)$. We bound the probability $P(q' \in \mathcal{V}(q))$ by conditioning on the location of q within G. Let L denote the lookout of G and L' denote its complement: $L = \beta_0\text{-LOOKOUT}(G)$ and $L' = G\backslash L$. We have

$$\begin{aligned}
P(q' \in \mathcal{V}(q)) &= P(q' \in \mathcal{V}(q) \mid q \in L)P(q \in L) + P(q' \in \mathcal{V}(q) \mid q \in L')P(q \in L') \\
&< 1 \cdot P(q \in L) + P(q' \in \mathcal{V}(q) \mid q \in L') \cdot 1 \tag{2}
\end{aligned}$$

First, we show

$$P(q \in L) < \alpha_0. \tag{3}$$

There are two cases. If $q = q_1$, then $P(q \in L) = \mu(L)/\mu(G)$, as q_1 is chosen uniformly at random from G. It then follows from (1) that $P(q \in L) < \alpha_0$. Instead, if $q \neq q_1$, then q must be a node corresponding to a configuration sampled uniformly at random from $\mathcal{F}$. We then have $P(q \in L) = \mu(L)/\mu(\mathcal{F}) < \alpha_0\mu(G)/\mu(\mathcal{F})$ and, again, $P(q \in L) < \alpha_0$, because $G \subseteq \mathcal{F}$ and $\mu(G)/\mu(\mathcal{F}) \leq 1$.

Next, we show

$$P(q' \in \mathcal{V}(q) \mid q \in L') < \beta_0. \tag{4}$$

7

For every $q \in L'$, we have
$$\mu(\mathcal{V}(q) \backslash G) < \beta_0 \, \mu(\mathcal{F}' \backslash G), \tag{5}$$
by the definition of a lookout set. Again, there are two cases. If $q' = q_2$, then since q_2 is chosen uniformly at random from $\mathcal{F}' \backslash G$, $\mathrm{P}(q' \in \mathcal{V}(q) \mid q \in L') = \mu(\mathcal{V}(q) \backslash G)/\mu(\mathcal{F}' \backslash G)$. Together with (5), this implies $\mathrm{P}(q' \in \mathcal{V}(q) \mid q \in L') < \beta_0$. Instead, if $q \neq q_2$, then $\mathrm{P}(q' \in \mathcal{V}(q) \mid q \in L') = \mu(\mathcal{V}(q) \backslash G)/\mu(\mathcal{F}) < \beta_0 \, \mu(\mathcal{F}' \backslash G)/\mu(\mathcal{F}) < \beta_0$.

Substituting (3) and (4) into (2), we get $\mathrm{P}(q' \in \mathcal{V}(q)) < \alpha_0 + \beta_0 \leq 2c$, where $c = \max\{\alpha_0, \beta_0\}$. Since the roadmap R has $N + 2$ nodes, we have

$$\mathrm{P}(A) \leq \binom{N+2}{2} \mathrm{P}(q' \in \mathcal{V}(q)) \leq 3N^2 \cdot 2c.$$

If we choose $c \leq (1-\gamma)/6N^2$, then $\mathrm{P}(A) \leq 1-\gamma$. In other words, if we choose α_0 and β_0 to be both smaller than $c \leq (1 - \gamma)/6N^2$, then for a randomly chosen pair of configurations q_1 and q_2, `BasicPRM`(q_1, q_2, N) fails to produce a path between q_1 and q_2 with probability greater than γ. Since q_1 and q_2, are chosen uniformly at random from G and $\mathcal{F}' \backslash G$, respectively, this implies that there exists at least one particular pair of configurations $p_1 \in G$ and $p_2 \in \mathcal{F}' \backslash G$ such that `BasicPRM`(p_1, p_2, N) fails to produce a path between p_1 and p_2 with probability greater than γ. $\quad\square$

Theorem 2 tells us that in an expansive space with small α and β values, there exist *some* path-connected query configurations for which `BasicPRM` will often fail. This result does not contradict Theorem 1. Even in a poorly expansive space, the probability of `BasicPRM` giving correct answers *eventually* converges to 1 at an exponential rate for sufficiently large N.

Theorem 2 also does not imply that `BasicPRM` performs badly all the time when α and β are small. A poorly expansive space may still contain a subset that is expansive with large α and β values; if query configurations are chosen within this subset, then `BasicPRM` will give correct answers with high probability. Here it is important to remark that the values of ε, α, and β are determined by the worst configurations and lookouts in $\mathcal{F}$. Thus, they do not reflect the fact that visibility properties are usually not uniformly favorable or unfavorable across $\mathcal{F}$. This is precisely what non-uniform sampling measures try to exploit (see Section 4).

In practice, a surprisingly small number of roadmap nodes is often sufficient to answer queries correctly. So, a key contribution of PRM planning is to reveal this *a priori* unsuspected property of motion planning problems: despite their high algebraic complexity, many free spaces encountered in practice or their subsets relevant to the queries verify favorable visibility properties, such as expansiveness. By chance, more than by design, PRM planners exploit this property well. The reason why PRM planners often scale up well when $\dim(\mathcal{C})$ increases is that visibility properties can be defined in terms of volume ratios over subsets of $\mathcal{F}$ and thus do not directly depend on $\dim(\mathcal{C})$.

3.3   Why are many free spaces expansive in practice?

In retrospective, it is not really surprising that many free spaces encountered in practice are expansive with large ε, α, and β. Experimental results indicate that only very narrow passages slow down PRM planners significantly [44]. To form such narrow passages, distinct portions of the free space boundary must almost coincide. For example, in a two-dimensional free space, two curve segments on the free space boundary must be almost parallel and very close to each other. Hence, a narrow passage is not a stable geometric feature: small random perturbations of workspace geometry are likely to either eliminate the passage or make it wider. In high-dimensional configuration spaces, passages that are narrow in several dimensions are even more unstable, as more near coincidences must occur simultaneously.
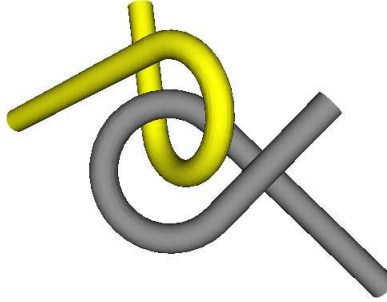
Figure 5. The alpha-puzzle consists of two identical rigid tubes: one is treated as a moving robot, the other as a fixed obstacle. The two tubes have been intentionally shaped and dimensioned to create a non-obvious narrow passage between the configurations where the two tubes are intertwined and those where they are separated.

A more thorough analysis based on the above intuition is developed in [8], using a technique known as *smoothed analysis* [47]. Smoothed analysis measures the worst-case performance of an algorithm over all possible inputs under random perturbations. This is a realistic model for many real-world scenarios, as their inputs are inherently noisy and are not *intentionally* chosen to create worst-case scenarios. It is shown in [8] that PRM planning with uniform sampling has polynomial smoothed running time in configuration spaces of constant dimensions. Specifically, given a configuration space of fixed dimensions and with polyhedral obstacles bounded by n simplices whose vertices are perturbed according to a normal distribution of variance σ^2, a set of randomly sampled points with size polynomial in n and $1/\sigma$ results in a roadmap that covers the free space well and captures its connectivity with high probability.

This smoothed analysis does not suggest perturbing the geometry of $\mathcal{F}$ at random before planning. Instead, it indicates that poorly expansive free spaces are unlikely to occur by accident. It is not easy to intentionally design the geometry of a robot and its workspace so that the resulting free space contains non-obvious narrow passages. A rare example of an intentionally designed and reasonably complicated narrow passage in a six-dimensional configuration space is the alpha puzzle (Figure 5). In the motion planning literature, most examples of narrow passages in $\mathcal{F}$ derive directly from intentional narrow passages in the workspace, *e.g.*, when two mechanical parts must be assembled together by inserting one into the other, or a robot must pass between two closely spaced obstacles. Thus, several sampling strategies try to infer effective non-uniform sampling measures from workspace geometry.

4   How important is the sampling measure?

In the previous section, we have analyzed the performance of `BasicPRM` when the uniform sampling measure is used. However, most PRM planners employ non-uniform measures that dramatically improve performance. To illustrate, Figure 6 compares the average running times of three versions of `BasicPRM` with distinct sampling strategies: the uniform strategy (whose sampling measure π is uniform over $\mathcal{C}$), the Gaussian strategy of [4], and the two-phase connectivity expansion strategy of [27]. The last two strategies, which employ non-uniform sampling measures, perform much better than the uniform one. How can such improvements be explained? What information can a PRM planner use to bias the sampling measure to its advantage?

In this section, we address these questions. We first discuss the rationale for non-uniform sampling measures (Section 4.1). We then review previous work and describe how non-uniform sampling measures are usually constructed (Section 4.2).
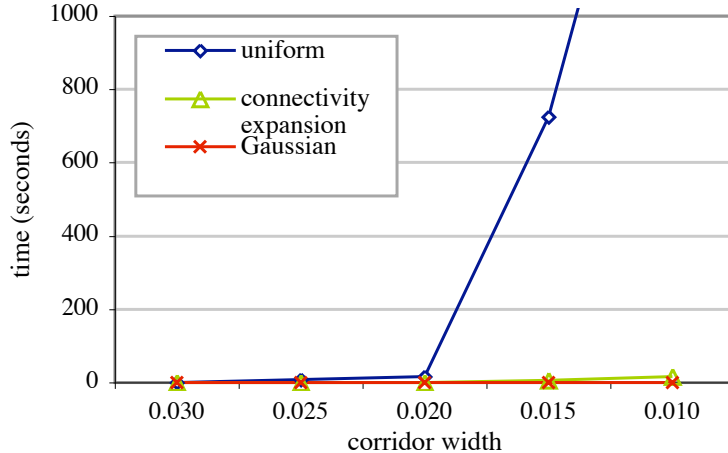
Figure 6. Comparison of three strategies with different sampling measures. The plot shows the average running time over 30 runs on the problem in Figure 3, as the corridor width decreases.
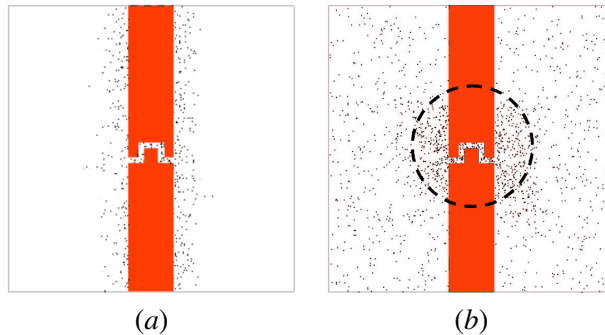

Figure 7. Sampled configurations generated by (*a*) the Gaussian strategy and (*b*) the two-phase connectivity expansion strategy. Both strategies sample the region containing the narrow passage more densely than most of the rest of $\mathcal{F}$.

4.1 Rationale for non-uniform sampling

Consider again the distribution (H, η) of hypotheses on the shape of $\mathcal{F}$ (Section 2). If nothing is assumed about $\mathcal{F}$, then the uniform measure is the best that a PRM planner can use. The fact that a sampled configuration q lies in $\mathcal{F}$ or not provides no information whatsoever on whether another configuration q' lie in $\mathcal{F}$, even if q' lies very close to q. All remaining hypotheses in H that are consistent with the information obtained on q are still equally likely, and there is no reason to sample one region of $\mathcal{C}$ more densely than another. Without prior assumptions, there is also little that we can say about the expected performance of PRM planning. If we persist in using a PRM planner, it must be that $\mathcal{F}$ is assumed to satisfy certain favorable properties that allow a small roadmap to represent $\mathcal{F}$ well enough to answer motion planning queries correctly. Note here the analogy with inductive learning, more specifically, PAC learning [28], where one can expect to learn a concept from examples only if the concept is assumed to have a simple representation. Similarly, *we can expect a PRM planner to work well—i.e., to "learn" the connectivity of $\mathcal{F}$ by sampling— only if $\mathcal{F}$ is assumed to satisfy favorable visibility properties, such as expansiveness, which allows $\mathcal{F}$ to be adequately represented by a small roadmap.*

Now, if $\mathcal{F}$ is indeed expansive, can non-uniform sampling measures work better than the uniform one? As mentioned earlier, visibility properties are usually not uniformly favorable across $\mathcal{F}$. The values of ε, α,

and β that characterize the expansiveness of $\mathcal{F}$ are determined by the worst-case configurations and lookouts in $\mathcal{F}$. To illustrate, consider the example in Figure 3. Points in the two large rectangular chambers are ε-good for $\varepsilon \approx 0.5$, but points inside the narrow corridor are ε-good for a much smaller value of ε. Furthermore, each chamber has a small lookout located near the entrance of the corridor. These considerations, along with the results of Section 3, suggest sampling more densely the region of $\mathcal{C}$ containing the narrow passage. However, a PRM planner is not given *a priori* information on the location of the passage, or even on the fact that one exists. Nevertheless, as Figure 7 shows, both the Gaussian strategy and the two-phase connectivity expansion strategy succeed in sampling the region containing the corridor more densely than most of the rest of $\mathcal{F}$. This ability allows them to achieve huge speedup over uniform sampling. As Figure 6 indicates, when the corridor width increases, visibility properties become more uniformly favorable, and the benefit of non-uniform sampling then decreases.

4.2 Constructing non-uniform sampling measures

Ideally, one would like to maintain a representation (H, η) of the possible shapes of $\mathcal{F}$ and, in each sampling operation, infer from (H, η) the optimal sampling measure that minimizes the expected number of remaining steps before connecting the two query configurations. However, this would have prohibitive computational cost. So, most planners only make local hypotheses that identify regions of $\mathcal{F}$ expected to have poor visibility and use them to construct non-uniform sampling measures. Some planners derive such local hypotheses from workspace geometry *prior* to roadmap construction. Others use information gained *during* roadmap construction. Some planners do not make any hypotheses explicitly, but instead continuously adapt the sampling measure using unsupervised learning techniques. In general, non-uniform sampling strategies spend more time generating a node of a roadmap than the uniform strategy, but the expectation is that the nodes are better distributed, so that a much smaller roadmap is sufficient to answer queries correctly, resulting in faster computation time.

In the following, we review existing sampling strategies and classify them according to what information is used for constructing sampling measures and how such information is used.

Workspace-based strategies. We have argued in Section 3.3 that narrow passages in $\mathcal{F}$ are often caused by narrow passages in the workspace. So, a number of sampling strategies infer hypotheses on the locations of poor-visibility regions in $\mathcal{F}$ from workspace geometry. For example, the watershed labeling algorithm identifies small corridors connecting large open regions in a cell decomposition of the workspace [49]. Workspace importance sampling tetrahedralizes the workspace to locate narrow passages and other regions with small local feature size [31]. Several strategies extract the medial axis of the workspace layout [11, 15, 18, 50]; the clearance along the medial axis measures the amount of open space around it.

Once narrow passages in the workspace have been identified, they must be mapped to corresponding regions in the configuration space for sampling. There are various ways to do this. For example, if the robot is a rigid object, one may sample the positional parameters of the configuration more densely around selected workspace passages, and sample the rotational parameters uniformly at random. For an articulated arm, one may sample a configuration of the arm's end-effector in the same way and then use the arm's inverse kinematics to get one or several configurations of the entire arm [31].

Instead of exploiting the geometry of obstacles in the workspace, the strategy proposed in [35] uses the kinematic singularities of a robot arm to infer regions of $\mathcal{F}$ with poor visibility. At a singular configuration q_{s}, the arm's end-effector loses some degrees of freedom and has reduced maneuverability. Thus, the region of $\mathcal{F}$ around q_{s} tends to have a flattened shape, resulting in poor visibility. Experiments show that sampling more densely around singular configurations improves the performance of PRM planning.

Filtering strategies. Filtering strategies over-sample $\mathcal{C}$, but quickly reject many unpromising samples by using local geometric patterns suggesting good or poor visibility. They rely on the fact that the probe `FreePath`, which checks the connection between two configurations, has much higher computational cost than `FreeConf`. Although filtering increases the number of calls to `FreeConf`, it eventually yields a smaller set of better placed roadmap nodes and thus reduces the number of calls to `FreePath`. It often leads to significant savings in computational time.

The Gaussian strategy [4] is one such strategy. In each sampling operation, it picks independently a pair of configurations. The first one, q, is sampled uniformly at random from $\mathcal{C}$, and the next one q' is sampled according to a Gaussian measure centered at q. If exactly one of the two sampled configurations lies in $\mathcal{F}$, this configuration is retained as a roadmap node. Otherwise, both configurations are discarded. This strategy tries to locate the boundary of $\mathcal{F}$ and sample more densely there (see Figure 7a). The rationale is that configurations with poor visibility often lie close to the boundary of $\mathcal{F}$.

The bridge test strategy [19] tries to capture a different kind of geometric pattern. If both q and q' are in collision, then it checks whether the midpoint q_{m} between q and q' is collision-free. If so, it retains q_{m} as a new roadmap nodes. The local geometric pattern captured by the three configurations q, q', and q_{m} suggests strongly that q_{m} lies in a narrow passage [19].

Visibility-based PRM [46] uses a slightly different filtering strategy. It estimates local visibility by testing the connections among sampled configurations and decimate the roadmap in regions where visibility is expected to be good, in order to avoid wasting effort there.

Adaptive strategies Adaptive strategies use information gained during roadmap construction to generate and adapt sampling measures. The two-phase connectivity expansion strategy [27] is one example. It builds an initial roadmap by sampling $\mathcal{C}$ uniformly at random. While doing so, for each node q in the roadmap, it counts the numbers of successful and unsuccessful connections to the nodes nearby and evaluates a criterion that amounts to a local estimate of the size of the visibility set of q. Next, it samples more configurations using a measure restricted to the neighborhoods of the nodes estimated to have small visibility sets, so that the final distribution of sampled configurations is denser around such nodes. In Figure 7b, this strategy produces a denser distribution of samples in the circled region around the corridor. The multi-phase sampling strategy described in [40] recursively subdivides $\mathcal{C}$ into regions, classifies them, and assigns a sampling measure to each region based on the classification. The classifier is trained in a pre-processing phase on a variety of configuration spaces.

Tree expansion strategies [1, 21, 34, 45] are other examples of adaptive sampling strategies. They hypothesize the location of the boundary of the portion of $\mathcal{F}$ represented by the current roadmap. In each sampling step, they try to expand this boundary by sampling new configurations around a node of the roadmap believed to be close to the boundary. So, the probability measure for sampling a new configuration is conditioned on the existing roadmap nodes, and thus automatically adapts over time. Tree-expansion strategies do not intentionally try to sample more densely in regions with poor visibility, but their success relies critically on the expansiveness of $\mathcal{F}$ and on their ability to effectively "link" together sequences of visibility sets [21].

Sampling strategies using unsupervised on-line learning are closest in their form to the strategy for constructing optimal sampling measures. The strategy proposed in [6] creates and updates an approximate model of $\mathcal{F}$ in the form of a collection of Gaussian functions and uses this model to sample configurations so that the expected value of a utility function is maximized. Some other adaptive strategies do not make explicit hypotheses on $\mathcal{F}$. For example, in [22], the sampling measure is constructed as a linearly weighted combination of component measures with complementary strengths. To adapt the sampling measure, the weights are adjusted after each sampling operation to favor the component measures that give the most promising results.

Deformation strategies They try to deform $\mathcal{F}$ into a more expansive space $\mathcal{F}_\mathrm{d}$, in which PRM planning becomes easier. For example, the deformation strategies proposed in [9, 20, 44] dilate $\mathcal{F}$, so that the dilated free space $\mathcal{F}_\mathrm{d}$ fully contains $\mathcal{F}$. As dilatation usually widens narrow passages more than the rest of $\mathcal{F}$ in relative terms, $\mathcal{F}_\mathrm{d}$ is expected to be more expansive than $\mathcal{F}$.

Deformation strategies require using modified probes `FreeConf` and `FreePath`. In [9, 44] a shrunken version of the geometry of robots and/or obstacles is pre-computed. The probes then use the shrunken geometric models to test whether a configuration or a path lies in $\mathcal{F}_\mathrm{d}$. The roadmap is built in $\mathcal{F}_\mathbf{d}$. Once a path joining the query configurations has been found in $\mathcal{F}_\mathbf{d}$, the planner samples additional configurations around it to generate a path that lies entirely in $\mathcal{F}$. Dilation may create false passages in $\mathcal{F}_\mathbf{d}$, *i.e.*, passages that are not present in $\mathcal{F}$. However, false passages occur rarely in practice for the reasons discussed in Section 3.3. Furthermore, a small dilatation is usually sufficient to achieve large speedup in computation time; this further reduces the risk of creating false passages.

5 How important is the sampling source?

To sample a configuration, a PRM planner needs both a probability measure π and a source S of random or deterministic numbers. The "sampling measure" and the "sampling source" are distinct concepts. The distinction is, however, blurred in the literature. With the use of deterministic sources to PRM planning [33], this distinction becomes important. Typically, the planner uses S to sample a point from a unit hypercube of suitable dimensionality and then maps the point into $\mathcal{C}$ according to π. As true randomness is difficult to obtain on a computer, the source most commonly used in existing PRM planners is the pseudo-random source S_ran, which generates a sequence of numbers that closely approximate the statistical properties of true random numbers.[2] In particular, a pseudo-random sequence is slightly irregular to simulate the effect that each number is chosen independently. In the proof of Theorem 1, this independence guarantees that samples spread evenly over $\mathcal{F}$. However, deterministic sources can achieve the same goal, sometimes even better, by minimizing discrepancy or dispersion [33]. The grid is a familiar deterministic source. In this section, we compare pseudo-random and deterministic sources. We also compare the impact of the sampling source to that of the sampling measure on the overall efficiency of PRM planning.

Our experiments use a pseudo-random source S_ran and two deterministic sources—the Halton sequence S_hal [38] and the incremental discrepancy-optimal sequence S_opt [37], both of which have been reported in [33] to often outperform S_ran. We pair each source with two probability measures, the uniform measure π_U and the measure π_G used in the Gaussian strategy. So, we have in total six sampling strategies $\{\pi_\mathrm{U}, \pi_\mathrm{G}\} \times \{S_\mathrm{ran}, S_\mathrm{hal}, S_\mathrm{opt}\}$, each of which is embedded in a distinct version of `BasicPRM` . See Appendix A.1 for implementation details, in particular the delicate combination of π_G and a deterministic source.

The sampling measure versus the sampling source. Figure 8 compares the six strategies on the example shown in Figure 3. Each entry of the table in Figure 8*a* was generated by dividing the running time of the uniform random strategy $(\pi_\mathrm{U}, S_\mathrm{ran})$ by the running time of the strategy corresponding to the table entry. So, the table reports the speedups over $(\pi_\mathrm{U}, S_\mathrm{ran})$. To generate the table, we averaged the running times for $(\pi_\mathrm{U}, S_\mathrm{ran})$ and $(\pi_\mathrm{G}, S_\mathrm{ran})$ over 30 independent runs. The second column (π_U) of the table shows that S_hal and S_opt achieve some speedup over S_ran, but a comparison of the second and the third columns shows that far greater speedup is achieved by switching to the better sampling measure π_G. Furthermore, the advantage of S_hal and S_opt over S_ran observed with π_U diminishes when we switch to π_G.

These results are reinforced in Figure 8*b*, which plots the running times of the six strategies, as the corridor width decreases. The three indistinguishable curves bundled together at the bottom of the plot all

---

[2]If we fix the seed of a pseudo-random source, the numbers generated by a pseudo-random seed are in fact deterministic. To have multiple independent runs of a PRM planner, we must use a different seed for each run.

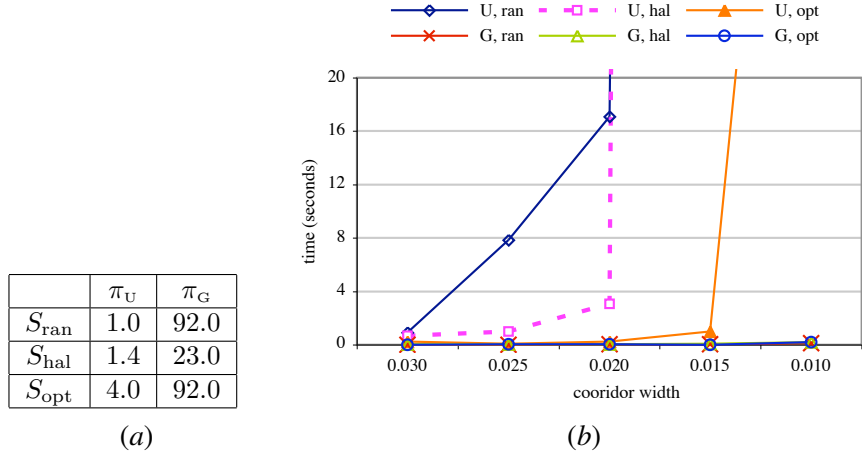|	π_U	π_G
S_{ran}	1.0	92.0
S_{hal}	1.4	23.0
S_{opt}	4.0	92.0

(a)   (b)

Figure 8. Running time comparison of six sampling strategies on the problem of Figure 3, when (a) the corridor width is set to 0.03, (b) the width decreases.
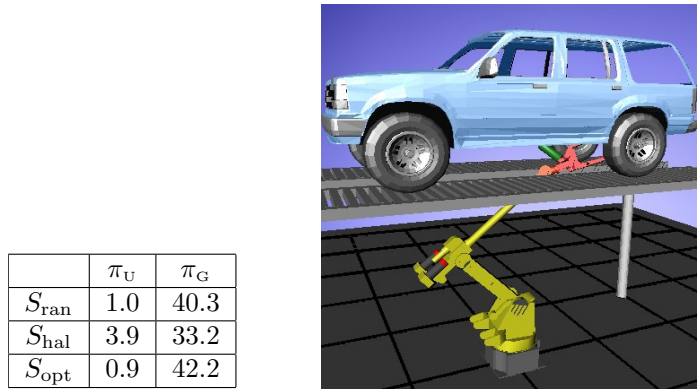

	π_U	π_G
S_{ran}	1.0	40.3
S_{hal}	3.9	33.2
S_{opt}	0.9	42.2

Figure 9. Comparison of six sampling strategies on a more realistic problem.

correspond to strategies using π_G, demonstrating that the sampling measure plays the critical role in determining the overall efficiency of the planner. Similar results have been obtained on more realistic problems, *e.g.*, the one in Figure 9, in which a six-degrees-of-freedom robot manipulator needs to access the bottom of a car through the narrow slot between the lift supports.

Dependence on dimensionality. The main rationale for using deterministic sources is that they reduce the discrepancy or dispersion of the samples. However, the computational cost of achieving a fixed discrepancy or dispersion grows exponentially with $\dim(\mathcal{C})$ [38, 42]. The samples generated by a deterministic source are distributed evenly and regularly over $[0, 1]^{\dim(\mathcal{C})}$. So they roughly correspond to a grid with $N^{1/\dim(\mathcal{C})}$ discretized intervals per axis, where N is the number of samples. In typical PRM planning problems, N is relatively small, while $\dim(C)$ could be large (greater than 6). This leads to large discrepancy and dispersion, even when a deterministic source is used.

Hence, the advantage that deterministic sources can possibly achieve over pseudo-random sources necessarily fades away as $\dim(\mathcal{C})$ increases. To illustrate, Figures 10 and 11 compare the performance of the six sampling strategies in configuration spaces of varying dimensionality. In Figure 10, as $\dim(\mathcal{C})$ increases, the volume of the narrow passage gets smaller, relative to the total volume of $\mathcal{F}$, and so it becomes more difficult to sample in it. The plot shows that (π_U, S_{opt}), which optimizes discrepancy, has the fastest increase in com-
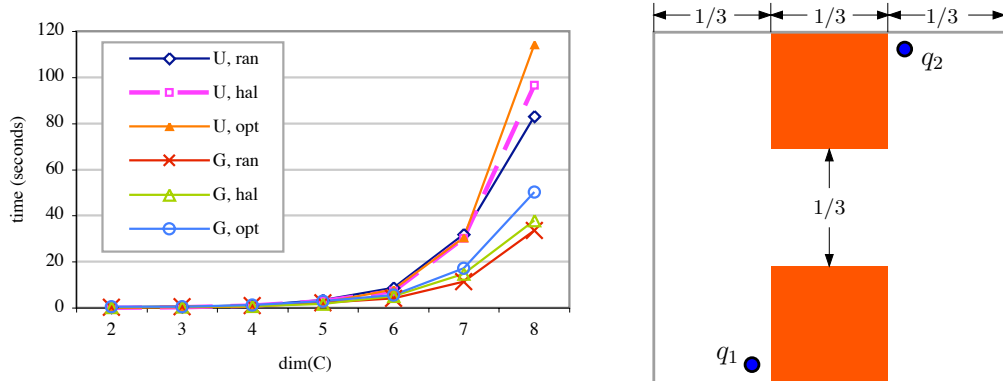
Figure 10. Running time comparison of six sampling strategies, as $\dim(\mathcal{C})$ increases. The plot shows the average running time for answering the query in the environment shown on the right. The free space $\mathcal{F}$ contains two large free regions connected with a multi-dimensional narrow passage that has length $1/3$ in each dimension.
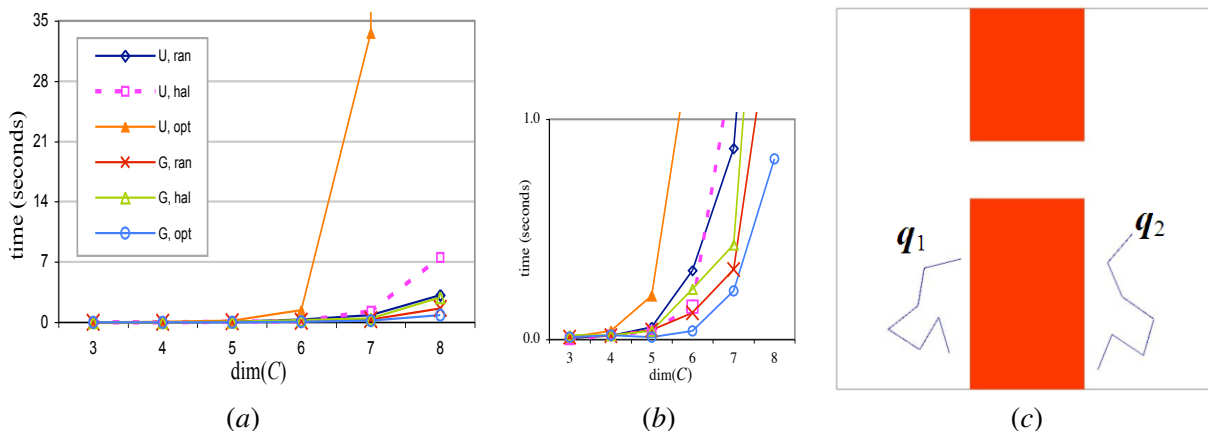

(a) $\qquad\qquad$ (b) $\qquad\qquad$ (c)

Figure 11. Running time comparison of six sampling strategies on a planar articulated arm with a moving base. (a) The plot shows the running times for each sampling strategy as $\dim(\mathcal{C})$ increases. (b) A zoom of the lower portions of the curves in (a). (c) The test environment.

putational cost, as $\dim(\mathcal{C})$ grows. The increase is slightly slower for $(\pi_{\mathrm{U}}, S_{\mathrm{ran}})$ and $(\pi_{\mathrm{U}}, S_{\mathrm{hal}})$. Figure 11 gives another example, where the robot is a planar articulated arm with a moving base. The dimensionality of $\mathcal{C}$ is increased by adding up to six links to the robot. The results here are similar to those of the previous example. It is interesting to observe that in this example, $(\pi_{\mathrm{U}}, S_{\mathrm{hal}})$ performs slightly better than $(\pi_{\mathrm{U}}, S_{\mathrm{ran}})$ when $\dim(\mathcal{C}) \leq 6$, but worsens afterwards (Figure 11b). In both examples, the three strategies using π_{G} have more moderate increases in computational cost. As $\dim(\mathcal{C})$ grows, visibility properties become less uniform over $\mathcal{C}$, and the advantage of π_{G} over π_{U} grows.

Robustness. For a given problem, the running times of a planner with a pseudo-random source vary from one run to another, while the running times of a planner with a deterministic source stay constant. This has led to the thinking that deterministic sampling has more consistent performance. This conclusion, however, is somewhat misleading, because consistency is better assessed across small variations of the same problem, a more realistic scenario in practice.

This led us to run the three strategies $\{\pi_{\mathrm{U}}\} \times \{S_{\mathrm{ran}}, S_{\mathrm{hal}}, S_{\mathrm{opt}}\}$ on 100 perturbed versions of the problem shown in Figure 11, when the robot contains four links (so, $\dim(\mathcal{C}) = 6$). Each perturbation is a small ran-
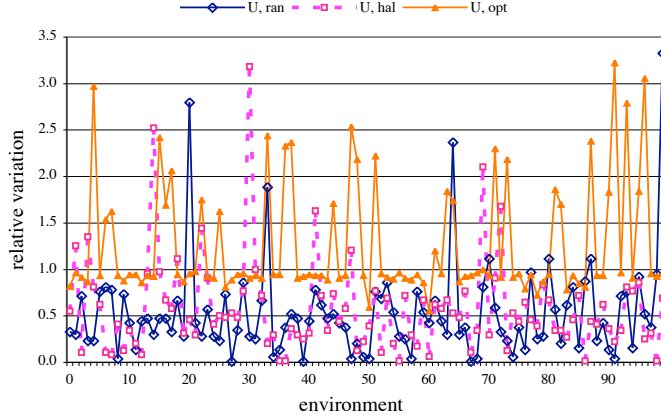
(a)

	mean	std	relative std
	μ	σ	σ/μ
$(\pi_{\mathrm{U}}, S_{\mathrm{ran}})$	0.208	0.154	0.741
$(\pi_{\mathrm{U}}, S_{\mathrm{hal}})$	0.213	0.164	0.768
$(\pi_{\mathrm{U}}, S_{\mathrm{opt}})$	1.055	1.469	1.393

(b)

Figure 12. Running time variations of three strategies that differ in sampling sources. (*a*) The horizontal axis indexes the 100 perturbed environments. The vertical axis shows the relative variation, defined as $|t_i - \bar{t}|/\bar{t}$, where t_i is the running time of a particular sampling strategy on the perturbed environment i, and $\bar{t} = \sum_{i=1}^{100} t_i/100$ is the average running time. (*b*) Summary statistics.

dom translation of the origin of $\mathcal{C}$. So, the shape of $\mathcal{F}$ remains unchanged. For each perturbed environment, we ran *each* of the three sampling strategies *exactly once*. As usual, in each run of $(\pi_{\mathrm{U}}, S_{\mathrm{ran}})$, the seed of the random sequence is set to a different, uncorrelated value. The results are shown in Figure 12. They indicate that the running times of the two deterministic sampling strategies vary at least as much as those of the pseudo-random sampling strategy.

6 Connection strategies

While the quality of a roadmap depends mostly on how its nodes are sampled, the dominant cost of roadmap construction is incurred in connecting these nodes. There are two reasons for this: the probe `FreePath` is much more expensive to evaluate than `FreeConf`, and there are potentially $O(N^2)$ pairs of nodes to be checked for connection. For computational efficiency, most PRM planners employ another important strategy, the connection strategy, that (*i*) selects a class of paths for connecting the nodes and (*ii*) chooses which pairs of nodes should be tested for connection. Systematic experiments have been carried out to compare various connection strategies [12]. Below, we focus on the relationship between connection strategies and visibility properties in $\mathcal{F}$. This relationship has also been studied in [13, 14].

Like many other PRM planners, `BasicPRM` connects roadmap nodes with straight-line paths, a choice encoded in the probe `FreePath`. However, paths with other fixed shapes could be used as well, by simply redefining `FreePath` appropriately. They lead to visibility sets with different shapes[3] and thus affect the expansiveness of $\mathcal{F}$. Experiments show that no particular path shapes are inherently better than others [2].

---

[3]The theoretical results in Section 3 hold without any modification under this new definition of visibility sets.

On the other hand, on problems with narrow passages, improvements have been obtained with a probe `FreePath` that connects nodes by performing heuristic search in restricted regions of the configuration space [14, 24]. Restricted local search is still more expensive than checking a path of a given shape for collision, especially in spaces of high dimensionality, but it allows a much broader class of paths, which are no longer restricted to pre-defined shapes, to be used for connecting the nodes. So, it may significantly increase the expansiveness of $\mathcal{F}$ by making lookout sets much larger. The tradeoff between the cost of local search and the resulting increase in expansiveness has not received enough attention so far.

To reduce the number of calls to `FreePath` and speed up planning, most PRM planners only try to connect each node to other close-by nodes, which can be found efficiently through hashing [10, 23, 51]. In principle, this heuristics may reduces the visibility sets of roadmap nodes, but empirical results show that it works well in practice. The chance that two nodes see each other decreases sharply when the distance between them increases. Even if two distant nodes see each other and the PRM planner makes no attempt to connect them, it is very likely that they can be still connected in the roadmap through intermediate nodes. So, we lose little by not trying to connect nodes far apart.

7 Conclusion

A foundational choice made in PRM planning is to avoid computing the exact shape of free space $\mathcal{F}$ in order to gain computational efficiency. As a result, a PRM planner never knows the exact shape of $\mathcal{F}$. At any time during planning, many hypotheses on the shape of $\mathcal{F}$ are consistent with the information that the planner has obtained so far. The probability measure used for sampling $\mathcal{F}$ derives from this uncertainty.

The success of PRM planning depends mainly and critically on favorable visibility properties in $\mathcal{F}$, such as expansiveness. These properties do not depend directly on the dimensionality of $\mathcal{F}$. By exploiting the fact that visibility is not uniformly favorable across $\mathcal{F}$, non-uniform sampling measures dramatically improve the efficiency of PRM planning. The construction of effective non-uniform sampling measures underlies the probabilistic foundations of PRM planning. In contrast, the choice of the sampling source has only small impact on the overall efficiency of a PRM planner.

To speed up PRM planning, one promising direction is to design better sampling strategies (and perhaps connection strategies as well) by exploiting the partial knowledge acquired during roadmap construction and using this knowledge to adjust the sampling measure on-line to make it more effective. One idea is to design probes that provide more information than `FreeConf` and `FreePath`, *e.g.*, probes that return local algebraic models of $\mathcal{F}$. The improved probes may allow the planner to better identify regions with poor visibility, such as the entrance of a narrow passage. This idea is related to the hybrid planner introduced in [17]. Since $\mathcal{F}$ often satisfies favorable visibility properties, one could exploit this fact to generate simplified algebraic models that avoid the details not needed for PRM planning. Another idea, in an seemingly opposite direction, is to acquire information on $\mathcal{F}$ in low-cost increments by creating probes that break the process of testing whether a configuration or a path lies in $\mathcal{F}$ into a series of small steps. Using such probes, a PRM planner can construct roadmaps whose nodes and edges are not fully tested, but instead carry weights estimating their probability of lying in $\mathcal{F}$. At each step, the planner may choose to add new nodes or edges to the roadmap, or refine the weights of its existing nodes and edges by performing additional probing steps on them [3, 45]. This approach allows the planner to obtain additional information on $\mathcal{F}$ at low incremental cost, and use this information to adapt the sampling measure and focus on the most interesting parts of $\mathcal{F}$. Although no existing PRM planner is able to compute an optimal sampling measure, considerable progress has been made in recent years by developing increasingly more sophisticated techniques to extract information on the visibility properties of $\mathcal{F}$ and adapt the sampling measure accordingly. More progress in this direction can be expected in the future, as some promising approaches, such as learning-based adaptive strategies [6, 22] and deformation strategies [9, 20, 44], have barely been touched on so far.

Although the "narrow passage" issue has often been seen as a main bottleneck in PRM planning, a perhaps more important issue in the future is the relatively high variance of the running time of PRM planners (see, *e.g.*, Figure 12), especially in poorly expansive free spaces. The high variance, combined with our inability to estimate the values of ε, α, and β in advance, often requires setting a large time limit (or, equivalently, a large number N of roadmap nodes) for the planner to terminate without a path. This may not be acceptable in some applications where many planning queries have to be processed sequentially [16]. Experiments indicate that some non-uniform sampling strategies also reduce the variance on the running times as well, but more research is needed to address this important issue.

The main drive for improving PRM planning in the future will come from problems far more challenging than those considered today, problems with configuration spaces of hundreds or thousands of dimensions or with new types of motion constraints. One such problem is motion planning for self-reconfigurable robots consisting of many identical or similar modules (*e.g.*, [7, 30, 43]). Another problem, protein folding, takes us beyond robotics to study the motion of bio-molecules [5]. A protein, which typically consists of thousands of atoms, can be seen as a hyper-redundant articulated linkage moving in a coordinated fashion to achieve a compact folded state while avoiding self-collision, *i.e.*, collision among its constituent atoms. The design of new motion planning algorithms should be aimed at problems of such scale.

References

[1] M. Akinc, K.E. Bekris, B. Y. Chen, A.M. Ladd, E. Plaku, and L.E. Kavraki. Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps. In M. Erdmann et al., editors, *Algorithmic Foundations of Robotics VI*, pages 80–89. Springer-Verlag, 2004.

[2] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In P.K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 155–168. A. K. Peters, Wellesley, MA, 1998.

[3] R. Bohlin and L.E. Kavraki. Path planning using lazy PRM. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 521–528, 2000.

[4] V. Boor, M.H. Overmars, and F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1018–1023, 1999.

[5] C.-I. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing, 1999.

[6] B. Burns and O. Brock. Toward optimal configuration space sampling. In *Robotics: Science and Systems*, 2005.

[7] A. Casal and M. Yim. Self-reconfiguration planning for a class of modular robots. In *Proc. SPIE Symp. on Intelligent Systems & Advanced Manufacturing*, volume 3839, pages 246–255, 1999.

[8] S. Chaudhuri and V. Koltun. Smoothed analysis of probabilistic roadmaps. Unpublished manuscript, 2006.

[9] H.-L. Cheng, D. Hsu, J.-C. Latombe, and G. Sánchez-Ante. Multi-level free-space dilation for sampling narrow passages in PRM planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2006. To appear.

[10] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun. *Principles of Robot Motion : Theory, Algorithms, and Implementations*, chapter 7. The MIT Press, 2005.

[11] M. Foskey, M. Garber, M.C. Lin, and D. Manocha. A Voronoi-based hybrid motion planner. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 55–60, 2001.

[12] R. Geraerts and M.H. Overmars. A comparative study of probabilistic roadmap planners. In J.D. Boissonnat et al., editors, *Algorithmic Foundations of Robotics V*, pages 43–59. Springer, 2002.

[13] R. Geraerts and M.H. Overmars. Creating small roadmaps for solving motion planning problems. In *IEEE Int. Conf. on Methods & Models in Automation & Robotics*, pages 531–536, 2005.

[14] R. Geraerts and M.H. Overmars. Reachability analysis of sampling based planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 406–412, 2005.

[15] L. Guibas, C. Holleman, and L.E. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis based sampling approach. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 254–260, 1999.

[16] K. Hauser, T. Bretl, and J.C. Latombe. Learning-assisted multi-step planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2005.

[17] S. Hirsch and D. Halperin. Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane. In J.D. Boissonnat et al., editors, *Algorithmic Foundations of Robotics V*, pages 239–256. Springer, 2002.

[18] K.E. Hoff III, T. Culver, J. Keyser, M. Lin, and D. Manocha. Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 2931–2937, 2000.

[19] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 4420–4426, 2003.

[20] D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P.K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 141–154. A. K. Peters, Wellesley, MA, 1998.

[21] D. Hsu, J.C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 2719–2726, 1997.

[22] D. Hsu, G. Sánchez-Ante, and Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 3885–3891, 2005.

[23] P. Indyk. Nearest neighbors in high-dimensional spaces. In 2nd, editor, *Handbook of Discrete and Computational Geometry*, chapter 39, pages 877–892. CRC Press, 2004.

[24] P. Isto. Constructing probabilistic roadmaps with powerful local planning and path optimization. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 2323–2328, 2002.

[25] L.E. Kavraki, M.N. Kolountzakis, and J.C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Trans. on Robotics & Automation*, 14(1):166–171, 1998.

[26] L.E. Kavraki, J.C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot path planning. In *Proc. ACM Symp. on Theory of Computing*, pages 353–362, 1995.

[27] L.E. Kavraki, P. Švestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics & Automation*, 12(4):566–580, 1996.

[28] M.J. Kearns and U.V. Vazirani. *An Introduction to Computational learning Theory*. The MIT Press, Cambridge, MA, 1994.

[29] D.E. Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional, 3rd edition, 1997.

[30] K. Kotay, D. Rus, M. Vona, and C. McGray. The self-reconfiguring robotic molecule: Design and control algorithms. In P. K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 375–386. A. K. Peters, Wellesley, MA, 1998.

[31] H. Kurniawati and D. Hsu. Workspace importance sampling for probabilistic roadmap planning. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1618–1623, 2004.

[32] A.M. Ladd and L.E. Kavraki. Theoretic analysis of probabilistic path planning. *IEEE Trans. on Robotics & Automation*, 20(2):229–242, 2004.

[33] S.M. LaValle, M.S. Branicky, and S.R. Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *Int. J. Robotics Research*, 23(7/8):673–692, 2004.

[34] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 473–479, 1999.

[35] P. Leven and S. Hutchinson. Using manipulability to bias sampling during the construction of probabilistic roadmaps. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 2134–2140, 2002.

[36] M. Lin and D. Manocha. Collision and proximity queries. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 35. CRC Press, 2004.

[37] S.R. Lindemann and S.M. LaValle. Incremental low-discrepancy lattice methods for motion planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 2920–2927, 2003.

[38] J. Matoušek. *Geometric Discrepancy*. Springer-Verlag, 1999.

[39] M.A. Morales A., R. Pearce, and N.M. Amato. Metrics for analyzing the evolution of c-space models. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2006.

[40] M.A. Morales A., L. Tapia, R. Pearce, S. Rodriguez, and N.M. Amato. A machine learning approach for feature-sensitive motion planning. In M. Erdmann et al., editors, *Algorithmic Foundations of Robotics VI*, pages 361–276. Springer-Verlag, 2004.

[41] B. Moro. The full Monte. *Risk*, 8(2):53–57, 1995.

[42] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial & Applied Mathematics, 1992.

[43] A. Pamecha, I. Ebert-Uphoff, and G. S. Chirikjian. Useful metrics for modular robot motion planning. *IEEE Trans. on Robotics & Automation*, 13(4):531–545, 1997.

[44] M. Saha, J.C. Latombe, Y.C. Chang, and F. Prinz. Finding narrow passages with probabilistic roadmaps: The small-step retraction method. *Autonomous Robots*, 19(3):301–319, 2005.

[45] G. Sánchez-Ante and J.C. Latombe. On delaying collision checking in PRM planning—application to multi-robot coordination. *Int. J. Robotics Research*, 21(1):5–26, 2002.

[46] T. Siméon, J.P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *J. Advanced Robotics*, 14(6):477–494, 2000.

[47] D.A. Spielman and S. Teng. Smoothed analysis: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[48] P. Švestka. On probabilistic completeness and expected complexity for probabilistic path planning. Technical Report UU-CS-1996-08, Utrecht University, Department of Information & Computing Sciences, Utrecht, the Netherlands, 1996.

[49] J.P. van den Berg and M.H. Overmars. Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners. *Int. J. Robotics Research*, 24(12):1055–1071, 2005.

[50] Y. Yang and O. Brock. Adapting the sampling distribution in PRM planners based on an approximated medial axis. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2004.

[51] A. Yershova and S.M. Lavalle. Efficient nearest neighbor searching for motion planning. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 632–637, 2002.

A   Implementation details

This appendix describes implementation details of `BasicPRM` with various sampling strategies and the connection strategy which are used in our experiments.

A.1 The sampling strategies

We used several sampling measures in our experiments. The measure for the two-phase connectivity expansion strategy was implemented according to its description in [27]. As the original algorithm is intended for a multi-query PRM setting, we made a small change in our experiments. Our implementation runs the first phase for 100 iterations, the second phase for 200 iterations, and then repeat. The ratio of iterations between the two phases is the same as that recommended in [27].

To implement the sampling measure for the Gaussian strategy, we pick a pair of configurations in each iteration. The first configuration q is sampled according to the uniform measure. The second configuration is chosen according to a radially symmetric Gaussian with its center at q and a small standard deviation σ. To choose the value of σ, we performed a few preliminary runs for each test environment and used the values that generated the best results.

We also used three different sampling sources. The pseudo-random source S_{ran} was implemented with an algorithm due to Knuth [29]. The algorithms for the two deterministic sources are described in [38] and [37], respectively.

The implementation of $(\pi_{\mathrm{G}}, S_{\mathrm{hal}})$ and $(\pi_{\mathrm{G}}, S_{\mathrm{opt}})$ requires more careful explanation. This particular combination of a non-uniform sampling measure and a deterministic sampling source has not appeared in the literature before. Our implementation takes a straightforward approach by replacing the pseudo-random source of the original Gaussian strategy with a deterministic source. To generate a sample according to the Gaussian measure, but with a deterministic source, it uses Moro's inversion [41], instead of the more commonly used Box-Mueller method, in order to better preserve low discrepancy. Furthermore, our implementation uses two copies of the deterministic source, one for each configuration in the pair of configurations sampled, in order to maintain independence between them.

A.2 The connection strategy

When adding a new node q to the roadmap, our implementation of `BasicPRM` checks the connection between q and an existing roadmap node v, only if (i) q and v are in different connected components, (ii) the distance between q and v is smaller than a threshold D, and (iii) v is a K-nearest neighbor of q for some fixed constant K. Based on earlier experimental results (*e.g.*, [12]) and our own experiences, we chose D to be 0.25 and K to be 30.