

1 On the Relationship Between Classical Grid Search and Probabilistic Roadmaps

Steven M. LaValle¹ and Michael S. Branicky²

¹ Dept. of Computer Science, University of Illinois, Urbana, IL 61801 USA

² Dept. of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106 USA

Abstract. *We present, implement, and analyze a spectrum of closely-related planners, designed to gain insight into the relationship between classical grid search and probabilistic roadmaps (PRMs). Building on quasi-Monte Carlo sampling literature, we have developed deterministic variants of the PRM that use low-discrepancy and low-dispersion samples, including lattices. Classical grid search is extended using subsampling for collision detection and also the optimal-dispersion Sukharev grid, which can be considered as a kind of lattice-based roadmap to complete the spectrum. Our experimental results show that the deterministic variants of the PRM offer performance advantages in comparison to the original PRM and the recent Lazy PRM. This even includes searching using a grid with subsampled collision checking. Our theoretical analysis shows that all of our deterministic PRM variants are resolution complete and achieve the best possible asymptotic convergence rate, which is shown superior to that obtained by random sampling. Thus, in surprising contrast to recent trends, there is both experimental and theoretical evidence that some forms of grid search are superior to the original PRM.*

1.1 Introduction

The main motivation of this paper is to provide insight into fundamental questions that arise in the development of sampling-based motion planning algorithms. What factors lead to good computational performance in practice? Is randomization really important to breaking the curse of dimensionality? While the number of samples required for a grid is known to increase exponentially in dimension, is it true that the probabilistic roadmap (PRM), which was “primarily developed for robots with many dofs” [19], overcomes this difficulty through random sampling? By building on existing sampling literature, developing theoretical analysis, and performing experimental studies, we conclude that the *original* PRM does not offer clear advantages over some forms of grid-based search. Furthermore, we even provide theoretical and experimental evidence that suggests the superiority of grids and other deterministic sampling schemes. One theoretical result, which is expressed in terms of a measure from the sampling literature (dispersion), implies that neither the PRM, grids, nor other sampling schemes can avoid the need for an exponential number of samples in dimension.

This paper builds on our previous work [9], in which well-known deterministic sampling techniques, such as Halton/Hammersley points and low-discrepancy lattices, were shown to offer advantages as replacements to pseudo-random sampling in the PRM context. Improved performance using Halton points was also observed recently in the context of the Visibility PRM [33]. An interesting deterministic alternative to PRMs was also proposed in [5].

Given the tremendous variety of possible path planning examples, and the fact that algorithm performance often depends greatly on parameter settings, it is a daunting task to compare path planners. Our approach to this problem is to define a spectrum of planners that has classical grid search and the original PRM at its ends. Neighboring planners in this spectrum vary only slightly, which helps us understand and assess how one small variation affects each algorithm. By chaining these variations together and drawing conclusions along the way, we provide a clearer understanding of how the planners at opposite ends are related.

There are important qualifications to the assertions made in this paper:

- We believe that importance sampling is important. Many recent probabilistic roadmap methods [1,2,8,14,23,30,32,36] have demonstrated improved performance by concentrating samples in a nonuniform way, such as along C-space boundaries [2,8], or the medial axis [14,30,36]. Our work can be viewed as complementary to importance sampling. In many of these approaches it may be possible to obtain performance improvements by replacing random sampling with deterministic schemes; however, given the variety of methods and heuristics often involved, this task is beyond the scope of the present paper.
- We are not recommending any of the planners presented here as the fastest available. Our intent is to gain insight into planning issues, as opposed to delivering the best planner.
- We believe that randomization is useful in many contexts. Its value, however, depends greatly on the paradigm within which it is used.

1.2 Grids and PRMs

This section introduces a few influential planners that will be discussed in detail in this paper, rather than providing a complete survey of techniques.

The path planning problem Let \mathcal{C} denote the configuration space (or *C-space*) of a d -DOF robot in a 2D or 3D world that contains static obstacles. For convenience, assume that the configuration space is parameterized so that $\mathcal{C} = [0, 1]^d \subset \mathbb{R}^d$, and some boundary points are identified to respect topology. Let \mathcal{C}_{free} denote the set of all collision-free configurations. A path planning *query* is a pair of configurations, $\langle q_{init}, q_{goal} \rangle$. The path planning problem is to find a continuous path, $\tau : [0, 1] \rightarrow \mathcal{C}_{free}$ such that $\tau(0) = q_{init}$ and $\tau(1) = q_{goal}$. It is assumed to be computationally prohibitive to construct an explicit representation of \mathcal{C}_{free} ; however, a *collision detection* algorithm

is available, which can quickly indicate whether a given configuration lies in \mathcal{C}_{free} (distance information could also be computed).

Classical grid-based search Grid-based search is considered by many to be the most straightforward form of path planning. Each component of \mathcal{C} is quantized, and a d -dimensional bitmap representation can be precomputed by iterating a collision detector over all quantized configuration values. A neighborhood structure must be defined, such as the set of $2d$ neighbors for each interior element of the bitmap (“up”, “down”, “left”, “right” in the case of $d = 2$). A given query is quantized, and the bitmap can then be searched using a classical AI or graph search algorithm, such as dynamic programming, A^* , best-first, or bidirectional search, to connect q_{init} to q_{goal} . In fact, the bitmap could also be searched using recent path planning methods that are based on incremental search, such as randomized potential fields [4], Ariadne’s clew [27,26], RRTs [22], and the planner in [16,31]. It is well-known that only *resolution completeness* can be obtained, and that for a fixed resolution, the number of samples (bitmap size) increases exponentially in d .

The original PRM The Probabilistic Roadmap (PRM) was introduced in [19] as a way to overcome the well-known curse of dimensionality that exists in grid search. The primary philosophy behind the PRM was to perform substantial preprocessing so that *multiple queries* for the same environment could be handled efficiently. This is analogous to the bitmap precomputation in classical grid based search. First, a *roadmap* encoded as an undirected graph, G , is constructed in a *preprocessing phase*. In a *query phase*, G is used to solve a particular path planning question for a given q_{init} and q_{goal} . Each vertex in G represents an element of \mathcal{C}_{free} , and each edge represents a collision-free path between two configurations.

The algorithm outlined in Figure 1.1 constructs a PRM with N vertices. In Step 3, a random configuration in \mathcal{C}_{free} is found by repeatedly picking a random configuration until one is determined by a collision detection algorithm to be in \mathcal{C}_{free} .¹ The NBHD function in Line 5 is a range query in which all vertices within a specified distance of q are returned, sorted by distance from q . Other variations are possible, of course. The variant considered here is the original PRM algorithm described in [18], page 25. The CONNECT function in Line 7 uses a fast local planner to attempt a connection between q and v . Usually, a “straight line” path in \mathcal{C}_{free} is evaluated between q and v by stepping along incrementally with a collision detection algorithm. A heuristic, node-enhancement phase, described in [19], is not considered here.

Once the PRM has been constructed, the query phase attempts to solve planning problems. Essentially, q_{init} and q_{goal} are treated as new nodes in the PRM, and connections are attempted. Then, standard graph search is performed to connect q_{init} to q_{goal} . If the method fails, then either more vertices are needed in the PRM, or there is no solution. This is analogous to the problem of insufficient resolution in classical grid search.

¹ In practice, of course, pseudo-random samples are generated.

```

BUILD.PRM
1 G.init();
2 for  $i = 1$  to  $N$ 
3    $q \leftarrow \text{RAND\_FREE\_CONF}(q)$ ;
4   G.add_vertex( $q$ );
5   for each  $v \in \text{NBHD}(q, G)$ 
6     if CONNECT( $q, v$ ) then
7       G.add_edge( $q, v$ );

```

Fig. 1.1. The preprocessing phase: build a PRM.

The Lazy PRM A recent PRM variant called the Lazy PRM has been proposed for the problem of answering single planning queries efficiently, as opposed to building an extensive roadmap prior to consideration of a planning query [6]. The resulting planner is sometimes very efficient in comparison to the original PRM. This represents a shift from the *multiple query* philosophy of the original PRM [19], and returns to the *single query* philosophy which was used in some earlier planners [4,13,27].

The key idea in the Lazy PRM is to build the roadmap initially without the use of a collision detector. The difference with respect to the algorithm in Figure 1.1 is that the condition in Line 6 is dropped, and Line 7 is executed every time. This allows the PRM to be constructed quickly; however, more burden is placed on searching in the query phase. Once an initial-goal query is given, the planner performs A^* search on the roadmap to find a solution. If any of the solution edges are in collision, they are removed from the roadmap, and the A^* search is repeated. Eventually, all edges may have to be checked for collision, but often the problem is solved well before this happens. Alternatively, it might be preferable to run the search only once on the initial graph, while validating edges during the search (instead of waiting for a solution and then validating it) [9]. If no solution is found, then more nodes may need to be added to the roadmap. The advantage of the Lazy PRM is that the collision checking is only performed as needed. Thus, all edges do not have to be collision checked as in the case of the original PRM. In classical grid search, this philosophy implies that the bitmap is not precomputed; collision checking is performed only as needed during the search.

1.3 A Spectrum of Sampling Techniques

A first step toward constructing a spectrum of planners from PRMs to classical grid search is to characterize a spectrum of sampling techniques to cover \mathcal{C} , from pseudo-random sampling to grids. A brief introduction to sampling is given here. See [9,25,28] for more details. The key philosophy is that sampling is viewed as an optimization problem in which a finite point set or infinite sequence is designed that minimizes some performance criterion relevant to a particular application.

Motivated by applications in numerical integration and optimization, the most common criteria are discrepancy and dispersion. Let $X = [0, 1]^d \subset \mathbb{R}^d$ define a space over which to generate samples. Consider evaluating the uniformity of a set, P , of N d -dimensional sample points $\{p_0, \dots, p_{N-1}\}$. The *discrepancy* is defined as

$$D(P, \mathcal{R}) = \sup_{R \in \mathcal{R}} |\mu(R) - |P \cap R|/N| \quad (1.1)$$

in which $|\cdot|$ of a set denotes its cardinality, μ denotes the Lebesgue measure, and \mathcal{R} is a *range space*, which will be taken by default in this paper to be the set of all axis-aligned rectangular subsets of X . The *dispersion* is defined as

$$\delta(P, \rho) = \sup_{x \in X} \min_{p \in P} \rho(x, p), \quad (1.2)$$

in which ρ denotes any metric; unless otherwise stated, the default metric in this paper will be ℓ^∞ . Dispersion can also be considered as the “radius” of the largest empty ρ -ball, among all balls whose centers lie in X . Both discrepancy and dispersion seem very relevant in the PRM context because range queries are repeatedly performed, and these criteria ensure that either the appropriate number of samples (discrepancy) or at least one sample (dispersion) will fall within a range.

A *low-discrepancy* point set or sequence is one that yields the best-possible asymptotic discrepancy, which is $O(N^{-1} \log^d N)$ for infinite sequences and $O(N^{-1} \log^{d-1} N)$ for finite point sets. The simplest low-discrepancy point sets and sequences are Hammersley and Halton points, respectively, which were applied to motion planning in [9]. Other low-discrepancy techniques exist that produce smaller constants in the asymptotic convergence rate. The best family of methods are the (t,m,s)-nets and (t,s)-sequences [28], and the current best within this family are the Niederreiter-Xing sequences [29].

Regarding ℓ^∞ dispersion, the *Sukharev sampling criterion* [34] states that for any point set P , $\delta(P) \geq \frac{1}{2} \lfloor N^{\frac{1}{d}} \rfloor$. Thus, to keep dispersion fixed, it is impossible to avoid exponentially-many samples in dimension. A *low-dispersion* point set or sequence is one that produces the best possible asymptotic dispersion, which is $O(N^{-1/d})$. For a fixed N , if $N^{\frac{1}{d}}$ is an integer, k , the *Sukharev grid* yields the best possible dispersion, which is precisely $\frac{1}{2} N^{-1/d}$. In this case, the grid is constructed by partitioning $[0, 1]^d$ into N cubes of width $1/k$ so that a tiling of $k \times k \times \dots \times k$ is obtained, and a sample is placed at the center of each cube. Nongrid, low-dispersion infinite sequences exist that have $\frac{1}{\ln 4}$ as the constant in the asymptotic convergence rate [28].

Grids fall under a more-general class of sample sets known as *lattices*, which have a regular neighborhood structure (they are essentially grids that may have nonorthogonal alignments). Most low-discrepancy sequences, such as Halton points, are not lattices. Surprisingly, lattices exist that have very low discrepancy; one was used in the PRM context in [9]. These offer the additional advantage of avoiding nearest-neighbor computations in the PRM.

1.4 A Spectrum of Planners

We take a bidirectional approach to relating grid search to PRMs. Section 1.4.1 extends and adapts grid search, and Section 1.4.2 presents variations of the PRM. Section 1.4.3 discusses the overlap between the two families of planners, which includes a planner based on the Sukharev grid.

1.4.1 Extensions of Grid Search

Let CGS refer to *classical grid search*, as described in Section 1.2; for simplicity, assume that A^* search is used. As a simple first extension, consider making a Lazy CGS planner. The grid implicitly encodes a roadmap because the location of all samples and the neighborhood structure is implied in its definition. A Lazy CGS planner is simply obtained by not precomputing the collision bitmap, and deferring all collision checks until each is required in the search (either the multi-iteration searching approach in [6], or the single-iteration approach in [9] can be used). This straightforward idea (undoubtedly considered previously) avoids performing unnecessary collision checks. Just as in the case of a PRM, the original CGS is best suited for multiple queries, and the Lazy CGS is suitable for single-query problems.

The next extension may be applied to either the CGS or Lazy CGS. Consider allowing two different sampling resolutions: one is the grid resolution, and the other is the sampling rate required for collision checking. We assume that the latter sampling rate is much higher than the grid resolution. If we consider a motion from one grid point, q , to an adjacent grid point, q' , then collision checking must be performed along a sequence of points obtained by linear interpolation (respecting topology) between q and q' . The grid can now be interpreted as a kind of trellis of paths. A similar idea was applied long ago for grid search in [10]. The advantage of this approach is that we might be able to solve a query using a low resolution for the grid, yet still be able to check for collisions at the required level of resolution. Good performance can be obtained if there are no narrow corridors in the configuration space. If q_{init} and/or q_{goal} do not lie on the grid points, then attempts can be made to connect them (again by interpolation) to a set of nearby grid points. We will refer to this extension as *subsampling grid search*. If a bitmap is precomputed, then the approach will be called SGS; otherwise, it is called a Lazy SGS.

One final point of concern is: where should the grid be placed? It seems reasonable to align the grid axes with the coordinate axes, but the translation remains to be chosen. Typically, the “origin” of a grid is at the coordinate origin (i.e, a sample appears at $(0, 0, \dots, 0)$). Let k denote the number of points per axis, and let d denote the dimension. Suppose $k = 3$ and $d = 2$. A classical grid would place one point in the center and the rest along the boundary (assuming no topological identifications). The maximum ℓ^∞ distance possible from a point $x \in [0, 1]^d$ to a grid point is $1/4$ (in a unit cube). Now, consider the Sukharev grid, which provides optimal dispersion. In this

case, the maximum distance is only $1/6$. Using only two points per axis ($k = 2$), the Sukharev grid yields a maximum distance of $1/4$, which is equivalent to that of a classical grid for which $k = 3$. These differences might appear small; however, for high-dimensional problems for which only low-resolution sampling is possible, the difference is dramatic. For example, suppose $d = 10$. In this case, a Sukharev grid with $2^{10} = 1024$ points provides the same quality coverage (in terms of dispersion) as a classical grid with $3^{10} = 59049$ points. Of course, as k becomes large, the improvement diminishes; however, for large d , it is impractical to make k large. Thus, the difference remains significant. Note also that the improvement diminishes in other topological spaces. In fact, on a toroidal manifold, the two grids are equivalent. That said, the Sukharev grid is never worse than the classical grid, and it is usually much better. Careful grid placement, as opposed to naive grids, is a theme also evident in kinodynamic planning [11].

1.4.2 Variations of the PRM

This section briefly reviews some of the PRM variants and experiments from [9], and also presents new experiments that use the Sukharev grid. For any sampling method, we can consider two variations: 1) a version that precomputes the roadmap, G , and is suitable for multiple queries, and 2) a lazy version that performs collision checking during the search, and is suitable for single-query problems.

First, consider constructing a *quasi-random roadmap*, QRM, by using the d -dimensional low-discrepancy sequence (such as Halton) instead of points generated from a pseudo-random number generator. The operation of the QRM should appear identical to that of the PRM; there are no new requirements. If the samples happen to form a lattice, costly neighborhood range queries can be avoided because the neighborhood structure is implicitly defined by the lattice rules. For many problems this can result in substantial performance improvement, particularly for lazy planners (because the time required to precompute the initial graph in the Lazy PRM represents a sizable fraction of the total running time for some problems [6,9]). Let LRM refer to a roadmap in which lattice points are used.

The experiments presented here can be considered as a continuation of those in [9]. Section 3 of [9]² showed improvement factors between 1.02 and 4.08 for a series of “bent corridor” examples (see Figure 1.2 from dimension 2 to 10, using Hammersley points in the place of pseudo-random samples (obtained from a nonlinear congruential generator)). We recently performed hundreds of new experiments based on the Sukharev grid that consider many different radii, corridor thicknesses, dimensions, and random rotations of samples. We generally observed that the Sukharev grid performs better than

² Regrettably, the original figure of bent corridor experiments contains a typographical error: the width values given are actually $1/2$ of the actual width used.

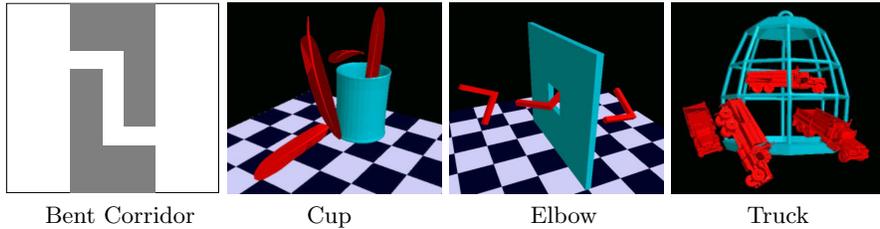


Fig. 1.2. The first frame shows a 2D rendering of the bent corridor series of experiments. The cup, elbow, and truck are 6-DOF problems that were solved using a Sukharev grid with 3, 4, and 5 points per axis, respectively, which is an improvement over PRM performance, and comparable to the low-discrepancy lattices used in [9].

pseudo-random sampling, especially in the case of a narrow corridor and small radius. As the radius increases, the pseudo-random samples are able to overcome much of their high-dispersion flaws; however, Sukharev grids still remain advantageous. The following table

Dim.	Width	Rad	Min	Max	Mean	Sukh-1st	Last-fail
2	0.02	0.5	65	652	461.97	25	400
3	0.05	0.6	131	1861	757.52	125	1000
6	0.15	0.6	195	1444	1237.85	729	64

shows a summary of the experiments for large radius values; with smaller values the results were even more in favor of Sukharev grids. The columns marked Min, Max, and Avg, give the number of nodes used in the PRM. The “Sukh-1st” column gives the smallest Sukharev grid that solved the problem. In some cases, the next-largest Sukharev grid does not solve the problem; thus, the last column gives the largest grid that failed to solve the problem. To avoid problems of axis alignment, we performed numerous experiments in which the Sukharev grid was rotated at random. This had no significant effect on the outcomes. For example, for 5 random rotations of the Sukharev grid, the same number of points, 729 (3 points per axis), was sufficient to solve the 6D problem listed in the table. Several other 6-DOF examples were solved using Sukharev grids, as indicated in the caption of Figure 1.2.

1.4.3 Sukharev Grid: Closing the Gap

Good performance was obtained using the Sukharev grid, but what kind of method is this? It can be considered as an extension of subsampled grid search that uses subsampling and Sukharev samples. The Sukharev grid is also a kind of lattice; therefore, the method can be considered as a special kind of LRM, and the spectrum of planners appears as:

$$CGS - SGS - LRM - QRM - PRM$$

Alternatively, the “Lazy” term could be added to each of these, to make single-query versions. The entire spectrum can be considered as families of *sampling-based motion planning* algorithms.

We now describe some basic relationships between the planners in the spectrum; additional conclusions and insights are derived from the theory in Section 1.5 and the discussion in Section 1.6. Two problems with classical grid search (CGS) were: high sampling density were required, and too many points were concentrated along the boundary. The first problem was fixed by subsampling (SGS), and the second was fixed by using a Sukharev grid. Once these changes are made, grid search appears to be comparable to other lattice-based roadmaps (LRM). Based on our experiments presented here and in [9], the LRMs (including grids) yield performance that is comparable to that of QRMs; however, LRMs have the additional advantage of known neighborhood structure. Finally, the deterministic planners appear superior to the original PRM in performance because they avoids the clumpiness and sparseness that must occur for a set of pseudo-random samples. This will be captured theoretically by Proposition 6 in Section 1.5.

1.5 Theoretical Considerations

We provide some theoretical analysis for all of the deterministic planners presented in Section 1.4, which includes the QRM, LRM, CGS, and SGS, and their Lazy counterparts. Let DRM refer to this collection of planners (for *deterministic roadmaps*), under the condition that they use asymptotically dispersion-optimal samples. This includes Halton, Hammersley, and many other low-discrepancy samples, and also the Sukharev grid.

Deterministic sampling enables the DRM planners to be *resolution complete*, in the sense that if it is possible to solve the query at a given sampling resolution, they will solve it. The resolution can be increased arbitrarily to ensure that any problem can be solved, if a solution exists. This is in contrast to the original PRM and other randomized variants, which are only probabilistically complete [21] (the probability tends to one that a solution will be found as the number of samples grows to infinity).

We exploit dispersion bounds to characterize the set of configuration spaces over which all queries can be correctly answered. This characterization is in terms of a parameter that measures the narrowest corridor width, in a manner similar to that of [3,7,15]. We define a cylindrical tube, and the “width” of \mathcal{C}_{free} is expressed in terms of the largest possible cross section of the tube, over all possible queries. Measuring this parameter may be as difficult as the planning problem; however, the expression of planner performance in terms of parameters that are difficult to measure is common in randomized planning analysis [3,6,15,16,20]. If a solution does not exist, our deterministic planners are able to declare that either the solution path must travel through a narrow passage that has a width smaller than a specified value, or there

is no solution. Such a result might be useful in applications because once the corridor is known to be narrower than a reasonable precision level, the solution might be impractical anyway.

We assume that for any PRM-like method, the radius parameter used to select neighbors in the NBHD function from Line 5 of Figure 1.1 is always sufficiently large. In theory, the radius can be made large enough so that an attempt is made to connect every vertex to every other vertex. In practice, this often becomes inefficient; therefore, a smaller value is typically used.

Let $\gamma = \langle q_{init}, q_{goal} \rangle$ denote a *query*. The set, $\Gamma(\mathcal{C}_{free})$, of all queries in which $q_{init} \in \mathcal{C}_{free}$, $q_{goal} \in \mathcal{C}_{free}$, and $q_{init} \neq q_{goal}$, for a given \mathcal{C}_{free} is called the *query space* of \mathcal{C}_{free} . Let $\Gamma_s(\mathcal{C}_{free}) \subseteq \Gamma(\mathcal{C}_{free})$ denote the set of all queries for which a solution exists.

Let \mathcal{C} represent any d -dimensional configuration space, parameterized to yield $\mathcal{C} = [0, 1]^d \setminus \sim$, in which $\setminus \sim$ denotes appropriate topological identifications along the boundary of the unit cube. Let Ψ represent the subset of the power set of \mathcal{C} corresponding to all open subsets that can be constructed with algebraic constraints, as formulated in [21].

Let a *tube*, \mathcal{B} , represent an uncountable collection of balls of equal radius whose centers are generated by a continuous path, $\tau : [0, 1] \rightarrow \mathcal{C}_{free}$. For each $s \in [0, 1]$ there exists an open ball $B \in \mathcal{B}$ that is centered at $\tau(s)$ and has radius r , which is fixed for all $B \in \mathcal{B}$; let $B(s)$ denote the ball centered at $\tau(s)$. We call $2r$ the *width*, $w(\mathcal{B})$, of the tube.

Let $V(q)$ denote the set of all points visible from a set $q \in \mathcal{C}_{free}$ (i.e., for each $q' \in V(q)$, $\lambda q + (1 - \lambda)q' \in \mathcal{C}_{free}$ for all $\lambda \in [0, 1]$, assuming that topology is respected in the interpolation).

Suppose that a query $\gamma \in \Gamma_s(\mathcal{C}_{free})$ is given. Among all possible tubes, let $\mathcal{B}(\gamma)$ denote the tube with the largest width such that $B(0) \subset V(q_{init})$ and $B(1) \subset V(q_{goal})$. In other words, the entire first ball is visible from q_{init} , and the entire last ball is visible from q_{goal} . Denote this largest-width tube as the $\mathcal{B}(\gamma)$, and call its width the *width*, $w(\gamma)$, of the query. For any query $\gamma \in \Gamma(\mathcal{C}_{free}) \setminus \Gamma_s(\mathcal{C}_{free})$, we say that its width is zero because no tube exists.

Define the *width of \mathcal{C}_{free}* as

$$w(\mathcal{C}_{free}) = \inf_{\gamma \in \Gamma_s(\mathcal{C}_{free})} w(\gamma). \quad (1.3)$$

Let $\Psi(x)$ for $x \in (0, \infty)$ denote the set of all $\mathcal{C}_{free} \in \Psi$ such that $w(\mathcal{C}_{free}) \geq x$. Intuitively, this can be considered as the set of problems for which the width of the narrowest corridor is at least x .

Suppose that the roadmap, G , is constructed for a particular \mathcal{C}_{free} . Then, the algorithm is said to be *complete for \mathcal{C}_{free}* if all queries in $\Gamma(\mathcal{C}_{free})$ are answered correctly in the query phase. A solution path must be reported if one exists; otherwise, failure is reported.

Our first two results establish the resolution completeness and complexity of all of the DRM planners. Therein, we only assume that sampling is accomplished using a set P of *low-dispersion points*, for which $\delta(P, \rho) < b(d)N^{-1/d}$,

in which $b(d)$ is a constant that may depend on the dimension d . For Halton/Hammersley points, b is a function of the largest prime used in the construction. For many low-dispersion sequences, including a multiresolution Sukharev grid, $b(d) \equiv 1$, which is a small constant that is independent of dimension. If N is restricted to values such that $N^{1/d}$ is an integer, then $b(d) \equiv 1/2$. A low-dispersion sequence for which $b(d) \equiv \frac{1}{\ln 4}$ and no restriction is placed on N is given in [28] (surprisingly, it is not uniform).

Propositions 1 through 3 below hold for any metric, provided that both tube width and dispersion are measured using the same metric, ρ . Propositions 4 through 6 hold for norms (which are all related by constants in subsets of \mathbb{R}^d) under the same provision.

Proposition 1 *After N iterations, the DRM planners are complete for all $\mathcal{C}_{free} \in \Psi(4b(d)N^{-1/d})$, in which N is the number of points, d is the dimension of \mathcal{C} , and $b(d)$ is a factor that depends on the sampling method.*

Proof: Suppose first that $\mathcal{C} = [0, 1]^d$ (ignoring any identifications). Assume that $\mathcal{C}_{free} \in \Psi(2b(d)N^{-1/d})$. To show completeness, we establish that for any solvable query, a solution path will be found; let $\gamma \in \Gamma_s(\mathcal{C}_{free})$ be such a query. Because $\mathcal{C}_{free} \in \Psi(2b(d)N^{-1/d})$, there exists a tube, \mathcal{B} , of width at least $2b(d)N^{-1/d}$, such that $B(0) \subset V(q_{init})$ and $B(1) \subset V(q_{goal})$.

Let P denote the set of sample points, which is also the set of vertices in the roadmap, G . Each ball $B \in \mathcal{B}$ must contain at least one $q \in P$. This follows from the fact that N samples were generated, and the asymptotic dispersion bound is $\delta(P, \rho) < b(d)N^{-1/d}$. If any ball of radius $b(d)N^{-1/d}$ is empty, then the dispersion would violate this upper bound.

First, consider connecting q_{init} and q_{goal} to the roadmap. Since $B(0) \subset V(q_{init})$, all configurations found by linear interpolation between q_{init} and any point in $B(0)$ are collision free. Therefore, q_{init} will be connected to a configuration in P (either one contained in $B(0)$, or at least one in the same connected component of G as a configuration of P that lies in $B(0)$). Using a similar argument for $B(1)$, q_{goal} will also be connected to a roadmap vertex.

It finally remains to show that there exists a path in G between the two configurations in P to which q_{init} and q_{goal} are connected. Consider the balls of \mathcal{B} as parameterized using $B(s)$ for $s \in [0, 1]$. We construct a sequence, q_0, \dots, q_{k-1} of k configurations as follows. Let q_0 be any element of $P \cap B(0)$. Let $s_1 \in [0, 1]$ denote the last point at which the ball $B(s)$ contains q_0 , by starting with $B(0)$ and increasing s continuously. Let q_1 be any element of $P \cap B(s_1) \setminus \{q_0\}$. Note that the $B(s_1)$ must contain at least two points in P because q_0 lies on its boundary. Inductively, let q_i be any element of $P \cap B(s_i) \setminus \{q_{i-1}\}$, where s_i is the first point at which $B(s)$ does not contain q_{i-1} . Note that the induction is finite, and let q_{k-1} denote the final configuration in the sequence.

We argue that there must exist a path in G between each pair, q_i, q_{i+1} , of configurations for $i \in \{0, \dots, k-2\}$. The point q_i must lie on the boundary of $B(s_{i+1})$; therefore, $B(s_{i+1})$ contains two points of P . Furthermore, all

points between q_i and q_{i+1} via linear interpolation must be collision free. The algorithm in Figure 1.1 would either have produced an edge between them, or failed to because both were already part of the same connected component of G . Either way, there exists a path in G between q_i and q_{i+1} . By applying this for each configuration in the sequence, there exists a path in G between q_0 and q_{k-1} . Furthermore, q_0 is connected to q_{init} , and q_{k-1} is connected to q_{goal} . Therefore, the query is correctly answered by returning a solution path.

We now turn to the case in which $\mathcal{C} = [0, 1]^d \setminus \sim$, in which \sim denotes boundary identifications needed to appropriately reflect the topology of transformation groups that arise in motion planning: S^1 , P^3 , etc. For the dispersion measurements in $[0, 1]^d$, balls near the boundary have to be contained entirely inside the unit cube. Once identifications are considered, some balls are allowed to overflow as long as their center lies in $[0, 1]^d$. Since $\delta(P, rho) < b(d)N^{-1/d}$ for $[0, 1]^d$, these overflowing empty balls cannot have radius larger than $2b(d)N^{-1/d}$. Thus, the dispersion in the part of the proof for $[0, 1]^d$ is simply scaled by two for the case of $[0, 1]^d \setminus \sim$ by assuming $\mathcal{C}_{free} \in \Psi(4b(d)N^{-1/d})$ in the first step, which establishes the proposition. ■

Proposition 1 can be reworked to bound the width of the query:

Proposition 2 *After N iterations, the DRM planners, for a query γ , either report a solution path or correctly declare that one of the following is true: there is no solution path, or $w(\gamma) < 4b(d)N^{-1/d}$.*

Proof: This follows directly from Proposition 1. Since the DRM is complete for $\Psi(4b(d)N^{-1/d})$, if no solution is found after N iterations, then $w(\mathcal{C}_{free}) < 4b(d)N^{-1/d}$ and $w(\gamma) < 4b(d)N^{-1/d}$. ■

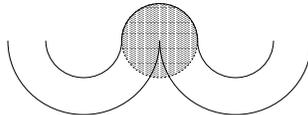


Fig. 1.3. A narrow corridor in \mathcal{C}_{free} , used in the proof of Proposition 3.

The next proposition indicates that if the dispersion is at least δ , then a DRM or PRM planner might miss solutions in corridors of width δ .

Proposition 3 *For any sample set, P , that has dispersion at least δ , no roadmap constructed using the algorithm in Figure 1.1 can be complete for $\Psi \setminus \Psi(\delta)$.*

Proof: We argue that completeness is lost by producing a \mathcal{C}_{free} and query $\gamma \in \Gamma_s(\mathcal{C}_{free})$ that will be answered incorrectly. If the dispersion is δ , then there exists a ball, $B \subset [0, 1]^d$ with radius δ such that $P \cap B = \emptyset$. Consider the corridor shown in Figure 1.3. Assume that no tube greater than width δ can be placed in the corridor. If \mathcal{C}_{free} is chosen so that B is located as

shown in the shaded area, then there will be no path in G that traverses the corridor. For any point on one side of the corridor outside of B , the straight-line path to any point on the other side of the corridor outside of B will intersect $\mathcal{C} \setminus \mathcal{C}_{free}$. Thus, a solution path will not be found. ■

From this the next proposition follows, which establishes that any PRM approach will require an exponential number of samples. It is assumed that the sampling scheme generates samples independently of the obstacle region.

Proposition 4 *Under any sampling scheme (including pseudo-random), a roadmap requires a number of samples exponential in dimension, d , to be complete for $\Psi(\delta)$.*

Proof: This follows immediately from Proposition 3 and the Sukharev sampling criterion [34]. ■

We now consider asymptotic bounds for DRM planners. The next proposition indicates that the DRM planners do the best possible, asymptotically.

Proposition 5 *The number of samples required by each DRM planner to be complete for $\Psi(\delta)$ is asymptotically optimal.*

Proof: By Proposition 3, to be complete for $\Psi(\delta)$, the dispersion must be less than δ . Thus, the goal of a DRM algorithm should be to reduce δ using as few samples as possible. The low-dispersion sequences achieve the best possible asymptotic dispersion. Therefore, the number of samples used in the DRM planners is asymptotically optimal. ■

The following proposition gives some indication that random sampling does not yield the best possible asymptotic convergence in the PRM; in fact, it is significantly worse than using deterministic sampling.

Proposition 6 *For a fixed dimension, d , the PRM with random sampling requires $O((\log N)^{\frac{1}{d}})$ times as many samples (with probability one) as the DRM planners to achieve the same ℓ^∞ dispersion.*

Proof: It was shown by Deheuvels [28] that ℓ^∞ dispersion for random samples is $O((\log N)^{\frac{1}{d}} N^{-\frac{1}{d}})$ with probability one. The asymptotic dispersion in a DRM achieves $O(N^{-\frac{1}{d}})$. The factor difference between the two is $O((\log N)^{\frac{1}{d}})$. ■

Relating the PRM to grid search We now discuss the implications of the previous propositions. Consider Propositions 1 and 2. If we use ℓ^∞ dispersion and select N such that $N^{1/d}$ is an integer, then the Sukharev grid yields $b(d) \equiv 1/2$, which is the best possible performance that can be obtained. An exponential number of samples in d is required, but according to Proposition 4, this is unavoidable. Note that the PRM cannot even provide these deterministic guarantees. The PRM at least has asymptotic analysis that establishes probabilistic completeness and convergence [3,19]. However, Propositions 5 and 6 indicate that the asymptotic rate of convergence obtained by the best deterministic sequences (including the Sukharev grid)

is asymptotically optimal, and the use of random samples is a significant factor worse (with probability one). Proposition 6 quantifies the problems with random sampling. To satisfy statistical tests for uniformity, a pseudo-random number generator must clump too many points in some places, and not enough in others. If the distribution of points is “too uniform,” then it will fail statistical tests. The deterministic samplers, on the other hand, are able to carefully *disperse* the points without this wasteful concern. Thus, in addition to the experimental advantages discussed in Section 1.4, there are strong theoretical indications that deterministic sampling methods, including grids, offer advantages over the original PRM.

1.6 Discussion

We have provided a spectrum of planners that ranges from classical grid search to the probabilistic roadmap. Surprisingly, the original PRM does not appear to be advantageous over deterministic approaches, including grid search, according to our experiments and theoretical analysis.³ Based on our work, we believe one of the main factors for the success of the original PRM was the excellent use of subsampling, which for grid sampling, enables challenging problems to be solved with only a few points per axis. In the original PRM, this idea also contributed greatly to the performance. Although there appears at first to be no exponential dependency on dimension because N is chosen directly, the exponential dependency reveals itself once again if we try to hold dispersion fixed. The notion of fixing the dispersion is general enough to mean keeping the resolution the same for a grid, while also applying to any sampling scheme. Using the Sukharev sampling criterion, $N^{\frac{1}{d}}$ can be considered as the best possible “points per axis” for any sampling scheme, whether or not it is a grid, and also whether it is random or deterministic.

Very high dimensions We note that our experiments focused mostly on six dimensions, with some examples in [9] up to ten dimensions. This includes many problems of interest in robotics, but examples exist in robotics and computational biology in which there are dozens or hundreds of dimensions. Obviously, even a Sukharev grid with two points per axis would be impossible to manage for some problems, while pseudo-random samples would appear to have no trouble. One straightforward way to use deterministic sampling would be to use non-lattice sequence in the place of pseudo-random samples. In very high dimensional problems we expect, however, that the performance differences between using pseudo-random sampling and deterministic sampling would be negligible for most problems. This is due primarily to the fact that spaces of this dimension will be severely undersampled, regardless of the

³ Since the “probabilistic” theme of PRMs does not seem advantageous, it is perhaps better to refer to these, grid search, and other related planners as instances of *sampling-based motion planning*.

sampling scheme used (i.e., the “points per axis” for any sampling scheme would be approximately one).

Search issues Searching is one of the oldest and most fundamental issues in motion planning. Although most PRM-based planners use A^* search, it is important to revisit some of the basic search issues in this context. Given the close connection drawn between classical grid search and the PRM in this paper, it should be clear that classical search issues which apply to grids should also apply to PRMs, QRMs, Sukharev grids, etc. that have a large number of samples. If a PRM has thousands of nodes in a high-dimensional space, then the common local minima problems most likely exist. Since optimality is not important in this context, many motion planning ideas that are based directly on incremental search could be used to search the roadmaps obtained in the family of planners described in this paper. Potential field approaches have improved search by using heuristics to escape local minima [4,13,17]. For the randomized potential field planner in [4], the search is even performed over an implicit grid. Thus, it can be considered as a Lazy CGS, with A^* search replaced by the potential field method. The planner in [10] can also be considered as a Lazy CGS planner. Of course many other planning approaches can be considered. Incremental searching methods, such as Ariadne’s clew [27], bidirectional RRTs [22], or the planners in [16,31], can be considered as alternatives to A^* search in any roadmap approach. As the implicit grid resolution in a Lazy CGS planner becomes higher, it is clear that the sampling itself is not important. The key is how to search in a bounded metric space.

Multiresolution approaches One disadvantage of the Sukharev grid is that for each increment of the points per axis, the total number of points increases dramatically for higher-dimensional problems. In general, for sample sets that require N to be specified, what happens when N needs to be increased? Many planning algorithms involve a repeated alternation between path searching and improving the sampling resolution. Of course, numerous multiresolution approaches have been considered in planning literature. For example, in [12], the sampling can be improved locally. For the Lazy LRM, the number of lattice points could be iteratively doubled, while still remaining low discrepancy [9]. In [5], a trellis is iteratively maintained (similar to SGS), and a hyperplane of samples is added in each iteration before search is performed again. In the original PRM, multiresolution issues appear to be equivalent to the problem of determining how many samples are necessary before a solution can be found. Any number of points can be easily added before search is performed again. A similar behavior can be obtained for grid-based sequences by adding points one at a time and maintaining low discrepancy along the way [24].

Bringing back randomization A recent trend in quasi-Monte Carlo literature is to consider randomized versions of Halton sequences, lattices, and (t,s)-sequences. There exist simple techniques that preserve the low-

discrepancy/low-dispersion properties of deterministic sequences while at the same time each sample is uniform randomly distributed.⁴ An elegant example of this work appears in [35], in which a randomized Halton sequence is obtained by cleverly constructing the first element at random, and computing the appropriate continuations of classical van der Corput sequences for each coordinate. Such sequences can be used in motion planning to obtain *both* probabilistic analysis and the deterministic guarantees from Section 1.5!

Acknowledgments We are grateful for the funding provided in part by NASA grant NAG2-1333 (Branicky), NSF awards 9875304, 0208891 (LaValle), NSF awards 0208891 and 0208919 (Branicky, Frazzoli, and LaValle). Steve Lindemann performed the bent corridor series of experiments. We thank Robert Bohlin, Francesco Bullo, Bruce Donald, Mike Erdmann, Leo Guibas, Blake Hannaford, Lydia Kavraki, Steve Lindemann, Kari Olson, Shai Sachs, Nicola Simeon, and Libo Yang for helpful discussions, comments, and suggestions. We also thank Eric Larsen, Stefan Gottschalk, Ming Lin, and Dinesh Manocha for making the PQP collision detection package available.

References

1. N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, pages 155–168, 1998.
2. N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 113–120, 1996.
3. J. Barraquand, L. Kavraki, J.-C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling scheme for robot path planning. In G. Giralt and G. Hirzinger, editors, *Proc. of the 7th International Symposium on Robotics Research*, pages 249–264. Springer, New York, NY, 1996.
4. J. Barraquand and J.-C. Latombe. A Monte-Carlo algorithm for path planning with many degrees of freedom. In *IEEE Int. Conf. Robot. & Autom.*, pages 1712–1717, 1990.
5. R. Bohlin. Path planning in practice; lazy evaluation on a multi-resolution grid. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2001.
6. R. Bohlin and L. Kavraki. Path planning using Lazy PRM. In *IEEE Int. Conf. Robot. & Autom.*, 2000.
7. R. Bohlin and L. Kavraki. A randomized algorithm for robot path planning based on lazy evaluation. In S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim, editors, *Handbook on Randomized Computation*. Kluwer Academic, 2001.
8. V. Boor, N. H. Overmars, and A. F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *IEEE Int. Conf. Robot. & Autom.*, pages 1018–1023, 1999.
9. M. Branicky, S. M. LaValle, K. Olsen, and L. Yang. Quasi-randomized path planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1481–1487, 2001.
10. B. R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artif. Intell.*, 31:295–353, 1987.
11. B. R. Donald, P. G. Xavier, J. Canny, and J. Reif. Kinodynamic planning. *Journal of the ACM*, 40:1048–66, November 1993.

⁴ However, the samples in the sequence are not uncorrelated, as in the case of a uniform, random sequence.

12. B. Faverjon. Obstacle avoidance using an octree in the configuration space of a manipulator. In *IEEE Int. Conf. Robot. & Autom.*, pages 504–512, 1984.
13. B. Faverjon and P. Tournassoud. A local based method for path planning of manipulators with a high number of degrees of freedom. In *IEEE Int. Conf. Robot. & Autom.*, pages 1152–1159, 1987.
14. C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in PRM planners. In *IEEE Int. Conf. Robot. & Autom.*, pages 1408–1413, 2000.
15. D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In et al. P. Agarwal, editor, *Robotics: The Algorithmic Perspective*, pages 141–154. A.K. Peters, Wellesley, MA, 1998.
16. D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Int. J. Comput. Geom. & Appl.*, 4:495–512, 1999.
17. Y. K. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Trans. Robot. & Autom.*, 8(1):23–32, February 1992.
18. L. E. Kavraki. Computation of configuration-space obstacles using the Fast Fourier Transform. *IEEE Trans. Robot. & Autom.*, 11(3):408–413, 1995.
19. L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. & Autom.*, 12(4):566–580, June 1996.
20. F. Lamiroux and J.-P. Laumond. On the expected complexity of random path planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 3306–3311, 1996.
21. J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
22. S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In B. R. Donald, K. M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics: New Directions*, pages 293–308. A K Peters, Wellesley, MA, 2001.
23. P. Leven and S. Hutchinson. Real-time motion planning in changing environments. In *Proc. International Symposium on Robotics Research*, 2000.
24. S. R. Lindemann and S. M. LaValle. Incremental low-discrepancy lattice methods for motion planning. 2003. Submitted to IEEE International Conference on Robotics and Automation.
25. J. Matousek. *Geometric Discrepancy*. Springer-Verlag, Berlin, 1999.
26. E. Mazer, J. M. Ahuactzin, and P. Bessière. The Ariadne’s clew algorithm. *J. Artificial Intell. Res.*, 9:295–316, November 1998.
27. E. Mazer, G. Talbi, J. M. Ahuactzin, and P. Bessière. The Ariadne’s clew algorithm. In *Proc. Int. Conf. of Society of Adaptive Behavior*, Honolulu, 1992.
28. H. Niederreiter. *Random Number Generation and Quasi-Monte-Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1992.
29. H. Niederreiter and C. P. Xing. Nets, (t,s)-sequences, and algebraic geometry. In P. Hellekalek and G. Larcher, editors, *Random and Quasi-Random Point Sets, Lecture Notes in Statistics, Vol. 138*, pages 267–302. Springer-Verlag, Berlin, 1998.
30. C. Pisula, K. Hoff, M. Lin, and D. Manoch. Randomized path planning for a rigid body based on hardware accelerated Voronoi sampling. In *Proc. Workshop on Algorithmic Foundation of Robotics*, 2000.
31. G. Sánchez and J.-C. Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *Int. Symp. Robotics Research*, 2001.
32. T. Simeon, J.-P. Laumond., and C. Nissoux. Visibility based probabilistic roadmaps for motion planning. *Advanced Robotics Journal*, 14(6), 2000.
33. Thierry Simeon, 2002. Personal communication.
34. A. G. Sukharev. Optimal strategies of the search for an extremum. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 11(4), 1971. Translated from Russian, *Zh. Vychisl. Mat. i Mat. Fiz.*, 11, 4, 910-924, 1971.
35. X. Wang and F. J. Hickernell. Randomized halton sequences. *Math. Comp. Modelling*, 32:887–899, 2000.
36. S. A. Wilmarth, N. M. Amato, and P. F. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *IEEE Int. Conf. Robot. & Autom.*, pages 1024–1031, 1999.