Choosing Good Distance Metrics and Local Planners for Probabilistic Roadmap Methods*

Nancy M. Amato O. Burchan Bayazit Lucia K. Dale
Christopher Jones Daniel Vallejo

Technical Report 98-010
Department of Computer Science
Texas A&M University
May 12, 1998

Abstract

This paper presents a comparative evaluation of different distance metrics and local planners within the context of probabilistic roadmap methods for motion planning. Both C-space and Workspace distance metrics and local planners are considered.

The study concentrates on cluttered three-dimensional Workspaces typical, e.g., of mechanical designs. Our results include recommendations for selecting appropriate combinations of distance metrics and local planners for use in motion planning methods, particularly probabilistic roadmap methods. Our study of distance metrics showed that the importance of the translational distance increased relative to the rotational distance as the environment become more crowded. We find that each local planner makes some connections than none of the others do — indicating that better connected roadmaps will be constructed using multiple local planners. We propose a new local planning method we call *rotate-at-s* that outperforms the common straight-line in C-space method in crowded environments.

*A preliminary version of this paper appeared in the *Proc. of the IEEE International Conference on Robotics and Automation* [2]. This research supported in part by NSF CAREER Award CCR-9624315 (with REU Supplement), NSF Grant IRI-9619850, and by the Texas Higher Education Coordinating Board under grant ARP-036327-017. Bayazit is supported in part by the Turkish Ministry of Education, Dale is supported in part by a GE Foundation Graduate Fellowship, and Vallejo is on leave from Universidad de las Americas-Puebla, Mexico, and is supported in part by a Fulbright-CONACYT scholarship.

1 Introduction

Automatic motion planning has application in many areas such as robotics, virtual reality systems, and computer-aided design. Although many different motion planning methods have been proposed, most are not used in practice since they are computationally infeasible except for some restricted cases, e.g., when the robot has very few degrees of freedom (dof) [23, 35]. Indeed, there is strong evidence that any complete planner (one that is guaranteed to find a solution or determine that none exists) requires time exponential in the number of dof of the robot [47].

For this reason, attention has focused on randomized or probabilistic motion planning methods. Notable among these are randomized potential field methods. The Randomized Path Planner (RPP) [7] uses random walks to attempt to escape local minima; it often works well for many dof robots, but there are simple situations in which it performs poorly [10, 30]. Researchers have proposed various potential functions (e.g., [6]) and other techniques, such as learning [15], for escaping local minima. In general, these methods can be quite effective when the configuration space (C-space) is relatively uncluttered, but there exist common situations in which they can fail [10, 30].

PRMs: Probabilistic Roadmap Methods. Recently, a new class of randomized motion planning methods has gained much attention [1, 3, 4, 7, 5, 20, 21, 22, 28, 30, 32, 42, 43]. These methods, known as *probabilistic roadmap methods* (PRMs), use randomization during preprocessing to construct a graph in C-space (a *roadmap*). Roadmap nodes correspond to collision-free configurations of the robot. Two nodes are connected by an edge if a path between the two corresponding configurations can be found by a 'local' planning method. Motion planning queries are processed by connecting the initial and goal configurations to the roadmap, and then finding a path in the roadmap between these two connection points. For each component in the outline below, there are various strategy and implementation options, and the success and efficiency of the PRM depends heavily on these choices (and also, of course, on the nature of the problems to be solved).

PRMs: PROBABILISTIC ROADMAP METHODS
I. PREPROCESSING: ROADMAP CONSTRUCTION
  1. NODE GENERATION (find collision-free configurations)
  2. CONNECTION (connect nodes to form roadmap)
    - **distance metric** selects candidate node pairs
    - **'dumb' local planner** checks connection
  3. ROADMAP ENHANCEMENT (connectivity/quality)
    - e.g., generate more nodes, use **'smart'** local planner
II. QUERY PROCESSING
  1. CONNECT START AND GOAL TO ROADMAP
    - **distance metric** selects roadmap nodes
    - **'dumb'** and then **'smart' local planner**
  2. FIND PATH IN ROADMAP BETWEEN CONNECTION NODES

PRMs have been shown to perform well in practice. In particular, after the roadmap is constructed during preprocessing, many difficult planning queries can be answered very quickly (in fractions of seconds) [3, 4, 32]. Although these methods are particularly suitable when multiple queries must be performed in the same static environment, the general PRM strategy can be used to solve single queries by only constructing 'useful' portions of the roadmap [22, 43]. A related idea is to use a sample of free points to specify promising subgoals for planning [14, 17].

PRMs can differ according to the high-level strategy decisions used during roadmap construction, that is, how nodes are generated and connected, and how the initial roadmaps are improved. The first PRM methods [32] generate nodes by uniformly sampling C-space (retaining collision-free configurations); roadmaps are enhanced by further sampling in 'difficult' regions. Some methods use information about the environment

to guide node generation. For example, the obstacle-based PRM, or OBPRM [3, 4], samples points near C-obstacle surfaces, the method in [20] employs random reflections at C-obstacle surfaces, and a technique called geometric node adding is used in [43] to generate robot configurations near obstacle boundaries. In [21], a roadmap is first built in a dilated free-space, and then efforts are made to "push" it into the true free-space by resampling.

Although PRMs can vary in terms of high-level strategy, they all share several important primitive operations – and the selection of appropriate techniques and implementations for these operations can very well prove to be the determining factor for the efficiency and success of the PRM. In addition to collision detection, which is perhaps the most important primitive operation in any motion planner, all PRMs make heavy use of *distance computations* and so-called *local planners*.

Local planners are used to make connections between roadmap nodes when building the roadmap, and also between the start and goal configurations and the roadmap when processing queries. There is a trade-off between the ability of a local planner to make connections and its running time. The general strategy of PRMs is to use a "dumb" local planner during initial roadmap construction, that is, a planner that is fast, perhaps not very powerful, and deterministic. The reason for this is that the faster the planner, the more connections can be attempted – and most connection attempts fail when environments are crowded. The planner should be deterministic so that paths don't have to be saved (they can be re-generated when needed). A full PRM will also need a "smart" local planner, that is, a planner that is slower, but is more likely to make connections. Smart local planners might be used during enhancement to connect different connected components of the roadmap, or during query processing to connect the start and goal to the roadmap; these planners could be randomized since they will be invoked fewer times and their paths can be saved.

Distance metrics play a less obvious, but still very crucial role in PRMs. As it would be infeasible to attempt to connect all possible pairs of nodes, distance metrics are used to determine which pairs one should try to connect when building the roadmap. Similarly, to obtain fast query times, it is necessary to limit the number of connection attempts from the start and goal to the roadmap. Thus, the connectivity/quality of the roadmap and the success of queries depends intimately on the distance metric. Therefore, in the context of PRMs, a good distance metric should indicate the relative probability that two configurations can be connected (which is clearly highly dependent upon the local planner being used to attempt that connection).

1.1 Our study

As discussed above, the choice of local planner and distance metric can crucially impact the success and efficiency of a PRM. Unfortunately, due to the high complexity and sometimes counter-intuitive nature of C-space, it is not always clear which are the most appropriate methods to use in a given situation.

The purpose of this paper is to provide guidance for selecting good (combinations of) distance metrics and local planners for PRMs. In particular, we present a comparative evaluation of distance metrics and local planners. We consider both C-space and Workspace distance metrics and local planners. Our study concentrates on cluttered three-dimensional Workspaces typical, e.g., of mechanical designs [11]. The moving objects (robots) are rigid, non-articulated objects yielding six-dimensional C-spaces.

The results of our study include recommendations for selecting appropriate combinations of distance metrics and local planners for use in PRMs. Our study of distance metrics showed that the importance of the translational distance increased relative to the rotational distance as the environment become more crowded. We find that each of the local planners makes some connections than none of the others do — which indicates that better connected roadmaps will be constructed using multiple local planners. We also propose a new local planning method, called *rotate-at-s*, that outperforms the common straight-line in C-space method in many of the environments we studied.

Although our study is motivated by our interest in PRMs, we believe our conclusions will be useful for other motion planning approaches as well – particularly those employing 'local planners' (e.g., [8, 38, 50]).

We remark that the goal of our study is to provide empirical evidence that certain combinations of distance metrics and local planners perform well for PRMs for certain types of problems. We seek empirical evidence due to the randomized nature of PRMs, which makes them difficult to analyze. Recently, a number of attempts have been made to theoretically explain the success of PRMs (see, e.g., [22, 29, 31]). However, most of these analyzes make simplifying assumptions regarding the nature of the C-space and/or the PRM components (e.g., local planner), and therefore are not applicable to all PRMs (e.g., they cannot be applied to OBPRM [3, 4].)

1.2 Outline of the paper

We describe the various distance metrics and local planners evaluated, including relevant previous work, in Sections 2 and 3, respectively. Implementation details and experimental results are presented in Section 4, and our recommendations and conclusions are summarized in Section 5.

2 Distance Metrics

Distance metrics are used in PRMs to determine which nodes one should attempt to connect using a local planner. Thus, they play a crucial role in both the efficiency and success of the PRM. That is, a good metric will limit the number of calls to the local planner by classifying enough connectable nodes as close and will produce a well-connected roadmap. Note that it must also be fast to compute, since distance calculations are one of the most numerous operations in a PRM. The determination of good metrics is hampered by the fact that C-space is not a Euclidean space, and thus our intuitive notions of closeness are not necessarily meaningful.

2.1 Previous work

Intuitively, a good measure of the distance between two configurations is a measure of the workspace region swept by the robot as it moves between them (the *swept volume* [51]). Unfortunately, the computation of such a metric would be prohibitively expensive for a PRM (and would also rely explicitly on the local planner).

Perhaps the simplest, and also most common, approach has been to consider C-space as a Cartesian space, and to use Euclidean distance in this space (see, e.g., [4, 22, 29]). Similar, but slightly more sophisticated approaches use weighted Euclidean distances, that is, Euclidean distances where each coordinate is given a 'suitable' weight (see, e.g., [17, 34]); this is a common approach for articulated robots, where links closer to the base are given larger weights.

The metric used in [22] is an upperbound on the maximum distance traveled by any point on the robot as it moves along a straight-line path between the two configurations. In [1], a distance is computed based on the differences between unit vectors on the object's local reference frame in the two configurations.

A way to define the distance between two configurations is to find the minimum distance between any point on robot in both configurations. In [16], a fast algorithm to find the minimum distance between two convex objects is given. In [46] a similar distance metric is defined for non-convex objects that can be computed using a hierarchical bounding representation. Other approaches to minimum distance can be found in [13, 27, 36, 37, 40]

Many metrics have been defined that have not (yet) been used with PRMs. For example, Hausdorff distance [18] which is a Euclidean based metric. Given two objects in contact but not collision, the rotation distance [52] is defined as the shortest contact-constrained rotation of the first object that collides with the second. Another interesting example is the *growth distance* defined for convex polyhedra in [41] which is the amount objects must be grown from their internal seed points until their surfaces touch. Although this

C-SPACE DISTANCE METRICS STUDIED			
METRIC	FORMULA	PARAMETERS	μsecs
Euclidean	$d(c_1, c_2) = \left(\sum_{k=x,y,z} P_{12}(k)^2 + \sum_{k=\alpha,\beta,\gamma} Q_{12}(k)^2 \right)^{\frac{1}{2}}$	none	4.6
Scaled Euclidean	$d(c_1, c_2) = \left(s \sum_{k=x,y,z} P_{12}(k)^2 + (1-s) \sum_{k=\alpha,\beta,\gamma} Q_{12}(k)^2 \right)^{\frac{1}{2}}$	$0 \leq s \leq 1$	4.7
Minkowski	$d(c_1, c_2) = \left(\sum_{k=x,y,z} P_{12}(k)^r + \sum_{k=\alpha,\beta,\gamma} Q_{12}(k)^r \right)^{\frac{1}{r}}$	$r > 0$	11.6
Modified Minkowski	$d(c_1, c_2) = \left(\sum_{k=x,y,z} P_{12}(k)^{r_1} + \sum_{k=\alpha,\beta,\gamma} Q_{12}(k)^{r_2} \right)^{\frac{1}{r_3}}$	$r_1, r_2, r_3 > 0$	24.2
Manhattan	$d(c_1, c_2) = \sum_{k=x,y,z} P_{12}(k) + \sum_{k=\alpha,\beta,\gamma} Q_{12}(k)$	none	4.4
WORKSPACE DISTANCE METRICS STUDIED			
METRIC	DESCRIPTION		μsecs
Center of Mass	Euclidean distance between c_1 and c_2's center of mass		18.4
Bounding Box	max Euclidean distance between vertices of c_1 and c_2's bounding boxes		33.7

Table 1: Distance metrics studied. In the formulas above, c_1, $c_2 \in \mathbb{R}^6$ are two configurations in C-space, $P_{12}(k) = |c_2(k) - c_1(k)|$, where $k \in \{x, y, z\}$, and $Q_{12}(k) = n_{12}|c_2(k) - c_1(k)|$, where $k \in \{\alpha, \beta, \gamma\}$ and n_{12} is the normalization factor.

metric has only been applied to convex objects, it is interesting since it yields both separation and penetration distances.

Another common metric used for rigid body kinematic analysis and robot trajectory planning is the Riemannian metric. In [48], C-Space is viewed as a Riemannian manifold by defining an inner product on every tangent space in C-space. In general, Riemannian metrics can be seen as the means by which familiar Euclidean concepts like lengths, angles and volumes can be extended to abstract differentiable manifolds. In [44, 45], the concept of metrics invariant with respect to the choice of coordinate axis for the Euclidean group in three-dimensions (SE(3)) is investigated, and a left invariant Riemannian metric is suggested. A Riemannian metric provides a notion of the length of a curve on a manifold. By definition the Riemannian metric is everywhere positive definite. A similar but indefinite metric is the semi-Riemannian metric [9] in which a curve can be seen as an energy curve. Curves that minimize the energy between two points are called *geodesics*. Geodesics can be considered a generalization of straight lines in Euclidean space to Riemannian manifolds. The existence of a Riemannian metric whose geodesics is screw motions are further investigated in [53, 54].

2.2 Distance metrics evaluated

The moving objects (robots) considered in our study were rigid objects in three-space. We represent configurations using six-tuples $(x, y, z, \alpha, \beta, \gamma)$, where the first three coordinates define the position and the last three the orientation. The orientation coordinates are the degrees in radians, represented as values in $[0 - 1)$.

Our study considered seven distance metrics (See Table 1). Five of these metrics are directly applied to 6-dimensional C-space configurations, and two of the metrics operate on points in the three-dimensional Workspace. Since we represent configurations in C-space coordinates, the Workspace metrics first transform the C-space configurations to Workspace coordinates before computing a distance.

Normalization. Remember that the orientation coordinates have values in the range [0-1), while the position coordinates have no range limits. Since all of the distance metrics compare the values of each coordinate, it is necessary to normalize the orientation coordinates with respect to the position coordinates. We suggest two normalization factors for this purpose. The first, *Free Space Normalization Factor*, is used if the configuration is far away from the C-space obstacles and the second, *Surface Normalization Factor*, is used if the configuration is close to C-space obstacles. We defined these normalization factors using the bounding box

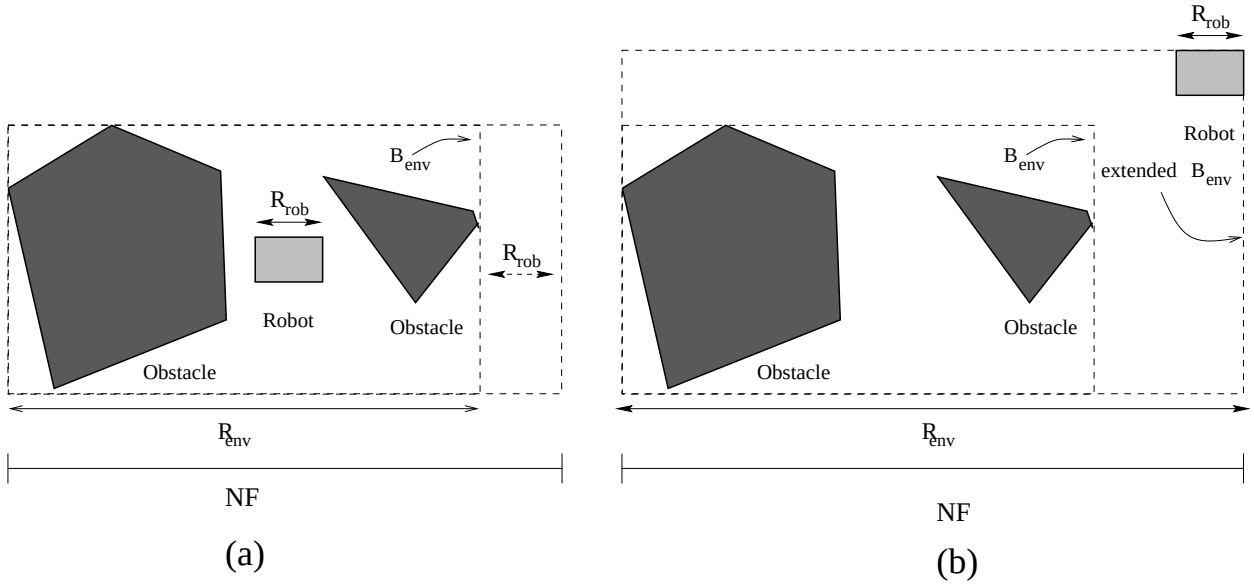


Figure 1: The calculation of the *Normalization Factor*. There are two different cases: (a) the robot is close to the obstacles, (b) the robot is far from the obstacles.

for the environment, B_{env}, and a bounding box for the robot, B_{rob}. These bounding boxes give a range for each coordinate axis (x, y, z). We choose the maximum of these ranges as the range of the corresponding bounding box and refer to them as R_{env} and R_{rob}, respectively. If one of the configurations is far from the C-space obstacles, B_{env} is temporally extended to include that configuration and R_{env} is calculated using for the new B_{env}. The *Normalization Factor*, or *NF*, is defined as (see Figure 1):

$$NF = \begin{cases} R_{env} + R_{rob} & \text{if the configuration is close to obstacles} \\ R_{env} & \text{otherwise} \end{cases}$$

C-space metrics. The C-space metrics we considered are listed in Table 1; orientational differences are always measured in the shortest direction. The regular Euclidean distance in $\mathbb{R}^6$ gives both position and orientation the same importance. The Scaled Euclidean distance changes the relative importance of the position and orientation components through the scale parameter s. The Minkowski distance is the generalized form of the Euclidean distance which uses a parameter r in place of the 2; as with Euclidean, both position and orientation are given the same importance. The Modified Minkowski metric distinguishes between the position and orientation coordinates using the parameters r_1 (position) and r_2 (orientation). The Manhattan metric is the usual Manhattan distance in $\mathbb{R}^6$.

Note that the Minkowski metric approaches the Manhattan distance as r tends to infinity. Also note that while both the Scaled Euclidean and the Modified Minkowski metrics enable us to weight the position and orientation coordinates differently, the two metrics do not behave the same. In particular, the impact of a change in the s parameter of the Scaled Euclidean metric is much greater than changes in the relative values of the r_1 and r_2 parameters of the Modified Minkowski metric.

Workspace metrics. The Workspace metrics we chose to study are all simple metrics based on Euclidean distances in the Workspace (see Table 1). Although there are many other types of metrics, such as Riemannian [48], we did not select them since they require expensive computations and are not suitable for PRMs. Similarly, for efficiency considerations, we did not use any of the more sophisticated Euclidean-based metrics, such as the Hausdorff distance [18] (its complexity depends on the number of vertices).

The first Workspace metric we studied is the Euclidean distance between the center of mass of the robot in the two configurations; the center of mass is approximated by averaging all object vertices. The second Workspace metric makes use of the bounding box of the robot, and finds the maximum of the Euclidean distances between any vertex of the bounding box in one configuration and its corresponding vertex in the other configuration. Both these metrics were selected as rough approximations of the minimum distance between the two objects.

3 Local Planners

Local planners are used in PRMs to make connections between nodes when building the roadmap, and also between the start and goal and the roadmap when processing queries. Our study concentrates on the so-called "dumb" local planners, which are fast (so many connections can be attempted), perhaps not very powerful (a trade-off, since they must be fast), and deterministic (so paths don't have to be saved). As mentioned in Section 1, a full PRM will also need a "smart" local planner to enhance the roadmap and for use during query processing.

3.1 Previous work

We mention here just a few of the many methods that could potentially be used as (dumb) local planners in PRMs (see, e.g., [35]). Perhaps the most used local planner is the straight-line in C-space which tests configurations (at a sufficient resolution) along that line for collision (see, e.g., [22, 29, 49]). Another approach is to move on Manhattan paths (one dof at a time) in C-space and to test for collision after moving each dof. *Rebound* methods reverse the direction of the current dof when a collision is found (e.g., [1, 19, 20, 38]).

Many planners have been based on the A^* search strategy (see, e.g., [26, 33, 39]). These methods are not always suitable for use as dumb planners since they may have large running times. One class of A^* planners, the so-called *local search with slide* methods (see, e.g., [12, 17, 24, 25, 50]), iteratively tries to move between two configurations c_1 and c_2 by first moving to a neighbor c' of c_1 that is one 'step' (in one dimension) closer to c_2 and has maximum clearance from obstacles. Next, the slide step considers each dimension in turn, and replaces c' with its neighbor c'' (one step away in that direction) if c'' is closer to c_2 than c_1 (under the L_∞-norm), and has a larger clearance.

3.2 Local planners evaluated

Our study considered several common planning methods and some new variants on them. All of them can be used with any of the distance metrics described in Section 2. We denote the start and goal configurations of the robot by $c_1 = (x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1)$ and $c_2 = (x_2, y_2, z_2, \alpha_2, \beta_2, \gamma_2)$, respectively.

The paths tested by all our local planners consist of a sequence of configurations that differ from their predecessors and successors by at most some fixed *resolution* (in each coordinate). The resolution values differed for position and orientation coordinates, and also varied according to the environment. Generally, given a pair of start and goal configurations (c_1, c_2), and a resolution for each coordinate, our planners first determine a (minimum) number *nsteps* of intermediate configurations to test (the maximum coordinate-wise difference divided by that coordinate's resolution), and then calculate an *increment* value for each coordinate (the coordinate-wise difference divided by *nsteps*) (see Figure 3(a)). This *increment vector* I is used to determine neighboring configurations tested by the planners.

Straight-line in C-space. Our first planner is the common *straight-line* in C-space method, which interpolates without bias along a six-dimensional straight line from configuration c_1 to c_2, and checks all points at

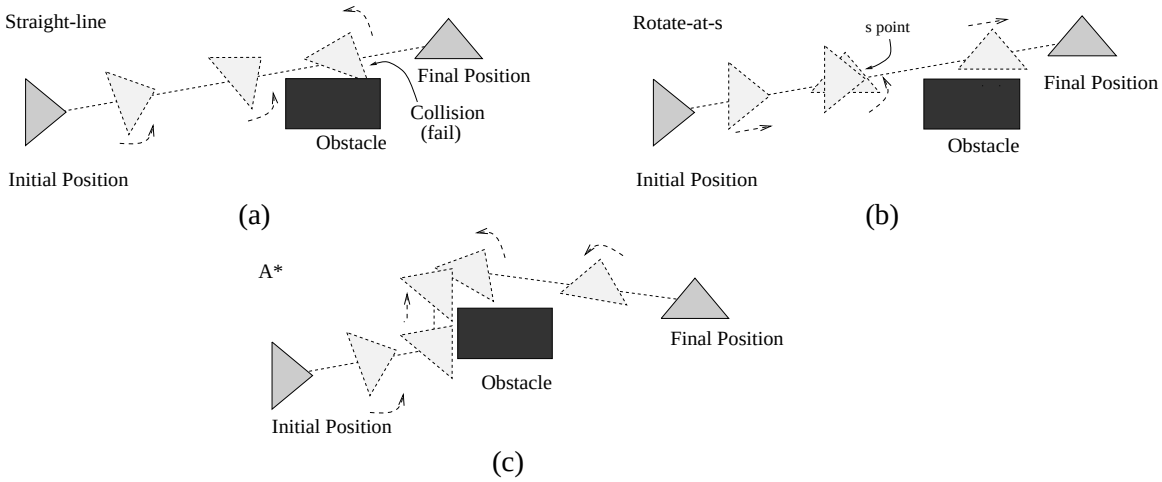


Figure 2: Paths tested by different local planners. The local planners are: (a) straight-line, (b) rotate-at-s, and (c) A* local planners.

STRAIGHT-LINE(c_1, c_2)
INPUT: c_1, c_2
OUTPUT:YES/NO
  nsteps := no. intermediate configs
  $I[1,6]$:= increment for each coord
  $c := c_1$
  for step := 1 to nsteps
    $c := c + I$
    if (c in collision) return NO
  endfor
  return YES
end

(a)

ROTATE-AT-S($c_1, c_2, s,$)
INPUT: c_1, c_2, s
OUTPUT:YES/NO
  $(x', y', z') := s(x_2 - x_1, y_2 - y_1, z_2 - z_1)$
  $c' := (x', y', z', \alpha_1, \beta_1, \gamma_1)$
  $c'' := (x', y', z', \alpha_2, \beta_2, \gamma_2)$
  $p1$:= STRAIGHT-LINE(c_1,c')
  $p2$:= STRAIGHT-LINE(c',c'')
  $p3$:= STRAIGHT-LINE(c'',c_2)
  if ($p1, p2, p3$ are all YES) return YES
    else return NO
end

(b)

A^*-PLANNER($c_1, c_2,$)
INPUT: c_1, c_2
OUTPUT:YES/NO
  $c := c_1$
  while ($c \neq c_2$) or ($\leq$ MAXITERS)
    find collision-free neighbors of c
    evaluate each neighbor
    $c :=$ most promising neighbor
  endwhile
  if ($c = c_2$) return YES
    else return NO
end

(c)

Figure 3: Local planners evaluated. The configurations are $c_i = (x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i)$, $i = 1, 2$.

some fixed resolution on that line for collision. (See Figures 2(a) and 3(a).) Orientational distance calculations and movements are in the shortest direction, e.g., from 0.1 back to, instead of forward to, 0.9. Note that in this planner, the robot simultaneously translates and rotates to the goal.

Rotate–at–s. We call our second (parameterized) family of local planners *rotate–at–s*, where $0 \leq s \leq 1$. This planner first translates from c_1 to an intermediate configuration c', rotates to a second intermediate configuration c'', and finally translates to c_2. The parameter s represents the fractional part of the translational distance between c_1 and c_2 that the robot travels from c_1 to c'. The straight-line planner is used to plan between each pair of configurations, that is, between (c_1, c'), between (c', c''), and between (c'', c_2). (See Figures 2(b) and 3(b).)

Note that when $s = 0$ (1), the intermediate configuration c' (c'') does not exist, e.g., if $s = 0$, then the robot will first rotate to c_2's orientation, and then translate to c_2's position. Note that this will have a smaller swept-volume than the straight-line local planner (and may have fewer collisions).

A^* planners. The basic A^*-strategy is to compute a set of neighbors of c_1, select the most promising neighbor c', and iterate with c'. The process terminates either when c_2 is reached or when no further movement is possible. Since A^*-methods can have unbounded running times, planning is often halted after some set maximum number of iterations. (See Figures 2(c) and 3(c).)

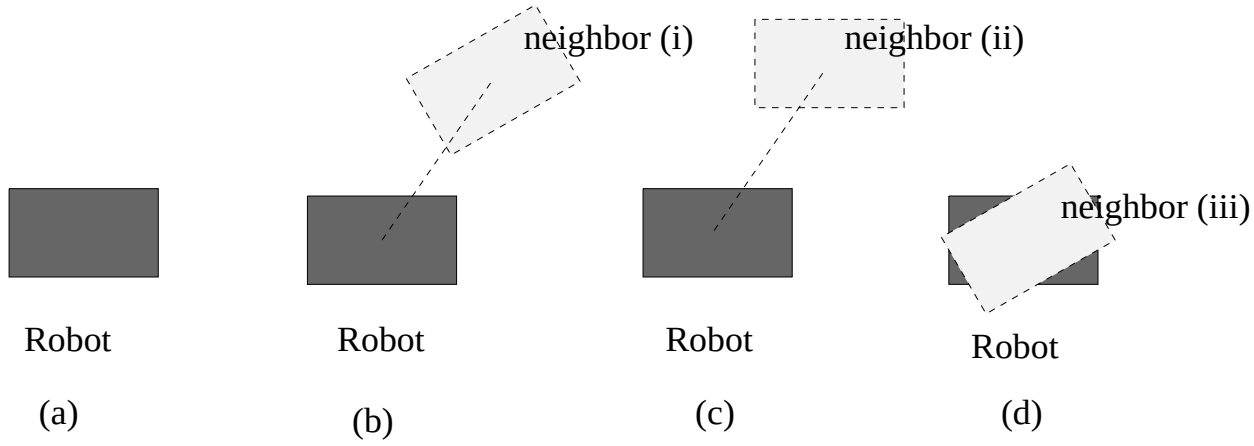


Figure 4: The robot and its neighbors in the workspace

Most A^*-like planning methods are 'stand-alone' planners that are too slow to be used as a dumb local planner in a PRM (see, e.g., [26, 33, 39]). However, they can be made faster by decreasing the number of neighbors checked and/or the maximum number of steps in the search. Our A^*-like methods do both. We set the maximum number of steps to be some small constant (e.g., 6) times the number of steps taken by the straight-line planner, and consider just three neighbors – the configurations where (i) both the position and orientation, (ii) only the position, and (iii) only the orientation, coordinates are incremented towards the goal (Figures 4(b), 4(c) 4(d), respectively).

Both our A^*-like methods move to neighbor (i) if it is collision-free. They differ, however, in their evaluation functions for selecting between neighbors (ii) and (iii). ***A**-clearance** selects the neighbor with maximum *clearance* from the workspace obstacles, where clearance is defined as the minimum distance between the robot's center of mass at c and any obstacle's center of mass. ***A**-distance** selects the neighbor that has minimum distance to the goal configuration.

4 Evaluating the Planners and Metrics

As previously discussed, the most important criteria for selecting distance metrics and local planners for PRMs are effectiveness and running time. A good distance metric (for a given local planner) will yield small distances between configurations that can be connected by the local planner. A good local planner/distance metric combination will make many connections between nodes selected by that metric. The computation times of these operations are also crucially important since typically more than 95% of a PRM's preprocessing time is spent in the connection phase, and the connection phase is dominated by distance computations (selecting candidates for connection) and local planning (making connections).

The experiments described below were designed to evaluate the metrics and planners in terms of these requirements in cluttered Workspaces. All programs were written in C, and run on SGI machines.

4.1 Experimental Design

4.1.1 Environments

Our study considers three basic environments representative of cluttered three-dimensional Workspaces. The moving objects (robots) are rigid, non-articulated objects yielding six-dimensional C-spaces. We consider three versions of each basic environment: easy, moderate, and hard. The hard versions are designed to have

THE PERCENTAGE OF SUCCESSFUL CONNECTIONS IN EACH ENVIRONMENT (60,000 TOTAL)									
ENVIRONMENTS	LOCAL PLANNERS								
	STRAIGHT-LINE			ROTATE-AT-0			ROTATE-AT-$\frac{1}{2}$		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
Cube	20.20%	14.23%	10.57%	16.35%	13.01%	11.22%	22.20%	16.06%	11.35%
Alpha	11.60%	10.25%	8.62%	12.10%	10.98%	9.42%	14.15%	13.80%	11.84%
Ring	10.81%	8.33%	3.28%	24.12%	16.10%	12.53%	5.95%	4.81%	2.99%

THE PERCENTAGE OF SUCCESSFUL CONNECTIONS IN EACH ENVIRONMENT (60,000 TOTAL)									
ENVIRONMENTS	LOCAL PLANNERS								
	ROTATE-AT-1			A*-CLEARANCE			A*-DISTANCE		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
Cube	3.54%	3.36%	2.17%	24.12%	16.10%	12.53%	24.03%	15.95%	12.49%
Alpha	6.10%	4.76%	4.63%	25.63%	22.28%	19.66%	25.53%	22.21%	19.53%
Ring	3.47%	2.49%	1.43%	14.27%	10.84%	5.12%	13.98%	10.52%	4.99%

Table 2: The percentage of successful connections (by different local planners).

narrow corridors in C-space; these corridors are wider in the moderate version, and wider still in the easy version. (See Figures 5–7.)

Although one can often intuitively decide whether one environment is harder than another, it is difficult to determine a rigorous and meaningful characterization of an environment's difficulty. Recently, some good efforts have been made in this direction, e.g., $\epsilon - goodness$ [5, 31] and $expansiveness$ [22]. Briefly, an environment is $\epsilon - good$ if every configuration can "see" (i.e. be connected to using the chosen local planner) an $\epsilon - fraction$ of the free C-space. The larger an environment's ϵ, the easier it is. $Expansiveness$ is a related, slightly more complex characterization.

Unfortunately, these approaches are mainly of theoretical interest since the computation of ϵ is usually not possible except for very simple spaces (although approximations can be made). To determine the relative difficulty of our environments we compared the total number of connections made (by each of the local planners) in each version. In each case, the same number of connections were attempted. The percentage of connections made can be seen as a rough approximation of ϵ. As can be seen in Table 2, for each environment, and for each local planner, the total number of successful connections decreases as the environment gets harder.

Remember, the hardness we define is not necessarily related to the connectivity of the resulting roadmap. For example, although the total number of the connections in the hard 6-cube environment is close to the total number of connections in the easy alpha-puzzle environment, it is much more difficult to create a well connected roadmap for the alpha-puzzle.

6-cube environment

The first set of environments consisted of seven cubes of unit size; six cubes were obstacles and one was the moving object. Each cube consisted of 12 triangles. The obstacle cubes were placed with their centers on three parallel planes, one each on the front and back planes, and four in the middle plane. They were arranged so that the six cubes surrounded a cubical region centered on the middle plane. The hardness of the problem was controlled by varying the distance between the front, middle, and back planes. These distances were 1.1 (hard), 1.6 (moderate), and 2 (easy). As the inter-plane distance decreases, it becomes harder to maneuver the moving cube into and out of the central cubical region (thus implying the existence of a narrow C-space region). (See Figure 5.)

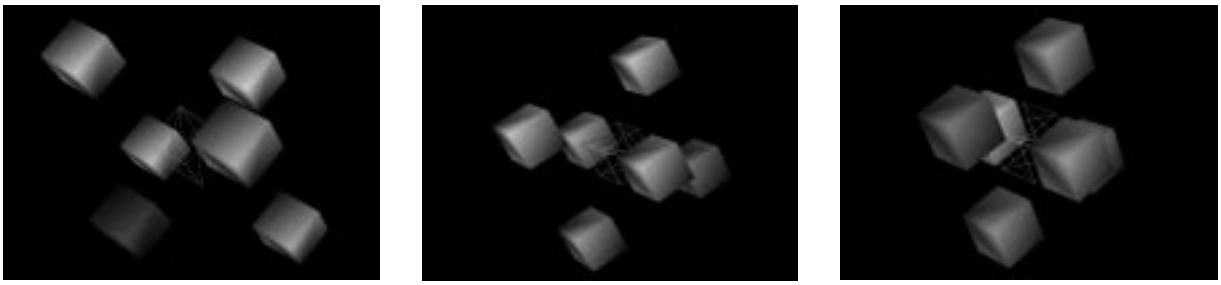


Figure 5: 6-cube environments, easy, moderate, and hard

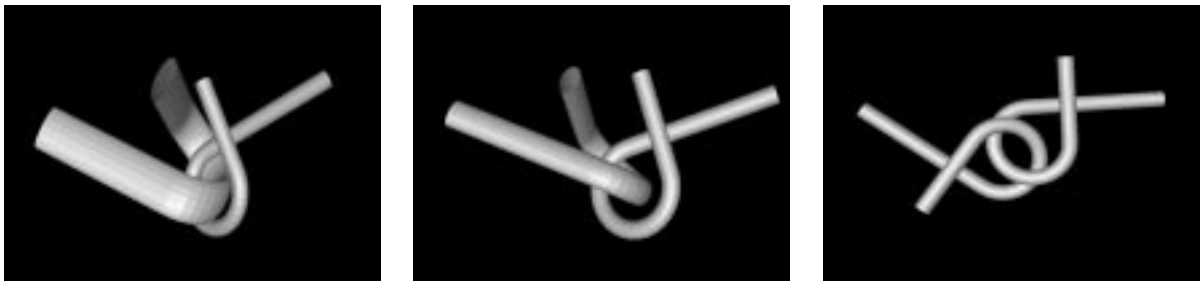


Figure 6: alpha-puzzle environments, easy, moderate, and hard

Alpha-puzzle environment

The second set of environments contained two tubes, each twisted into an α shape; one tube is considered an obstacle and the other the moving object. Each tube consisted of 1008 triangles. The reader might be familiar with a puzzle involving such tubes where the objective is to separate the intertwined tubes by a complicated sequence of rotations and translations. The 'hardness' of this problem was controlled by scaling the obstacle tube in one dimension, which had the effect of shrinking or widening the gap between the two prongs of the α. The hard version was the original problem, the moderate and easy versions scaled the obstacle tube by 1.5 and 2.5 respectively. (See Figure 6.)

The scaling also affects the shape of the tube's cross-section – it is transformed from a circle to an increasingly thinner and longer ellipsoid. Thus, while the 'easy' environment is easier since the gap between the alpha's 'prongs' is wider (leaving more space for the moving tube to escape when they are intertwined), it is also harder since the obstacle tube is larger and its cross-section is thinner and longer (leaving less space for the other tube to maneuver when they are intertwined). Another factor is that the obstacle tube is larger in the 'easy' environment than in the 'hard' environments, and so the moving tube is more likely to collide with it during local planning.

Rings environment

The last set of environments is made up of two "rings", hooked through each other; one ring is considered an obstacle and the other the moving object. The hardness of the problem is defined by the number of vertices of the rings. The "rounder" rings result in easier problems. The objects have the shape of a triangle (hard), a rectangle (moderate) and a pentagon (easy). The triangular object has 24 triangles, the rectangular one has 32, and the pentagonal one has 40. Although the shape has very little effect on the difficulty of translational movement, rotational movement becomes harder as the rings become less round. This set was designed to study the efficiency of the A^* local planners since most of the problems in such an environment are hard to solve by the other proposed local planners. (See Figure 7.)

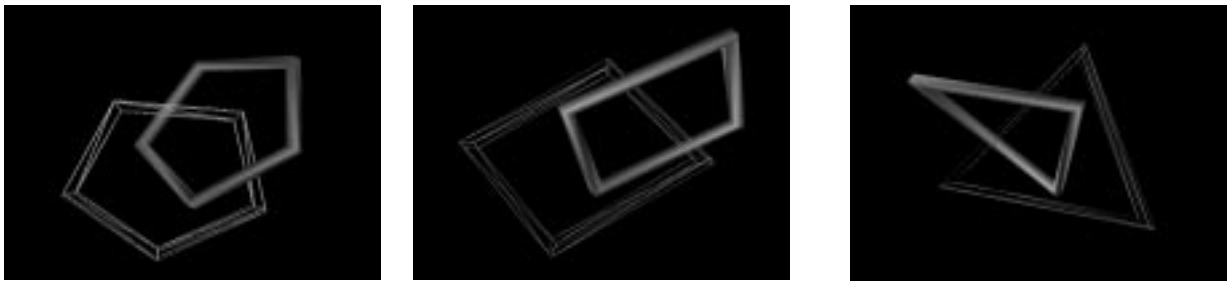


Figure 7: rings environments, easy, moderate, and hard

4.1.2 Experiments

As stated earlier the most expensive step of the roadmap construction is the connection phase. Ideally, every node would be checked for connection with every other node. However, this would require O(n^2) connection attempts and is computationally infeasible. The solution is to assume that for a given node, the k "closest" nodes to that node are more likely to be connected to it than are the other nodes. So a good roadmap connection strategy must first identify those k closest nodes to each node and then it can check if they can be connected to it.

Recall that our objective is to determine which local planners and distance metrics are most effective in cluttered environments. So, in our experiments we investigated whether a randomly chosen node can be connected to its k closest nodes (where the closeness is defined by the distance metric) by a specific local planner. Note that a local planner which performs well in one environment may fail in another one. We implemented our experiments in the nine environments described in Section 4.1.1. They were designed to study the following issues:

- *Selecting the parameters for distance metrics:* As some of the metrics we used were parametric (see Section 2.2), we designed a set of experiments to choose the best parameter values for each local planner.
- *Selecting metrics for local planners:* For each environment we identified the best distance metric for each local planner. We also investigated the effect of choosing a different number of closest nodes (k).
- *Selecting local planners for environments:* We compared the number of connections made by each local planner/distance metric combination in each environment.
- *Using multiple local planners:* Finally, we investigated if more than one local planner would be beneficial, which local planners would be redundant, and which combinations of local planners work well.

Sections 4.1.2 through 4.2.5 describe this process in detail.

Test Design

The motivation behind our experiments is to find which distance metrics are suitable for a given local planner and which local planner/distance metric combinations are best for a given environment. A naive idea would be the experiment shown in Figure 8. We have nine configurations on a C-Space obstacle (`RdmpCfgs` $= \{c_1, c_2, \ldots, c_9\}$), two randomly chosen points (`TestCfgs` $= \{a, b\}$) in C-free, a local planner lp and two different distance metrics dm_1 and dm_2. Furthermore, the local planner lp can connect test configuration

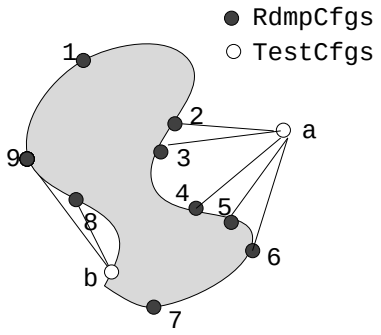


Figure 8: An example of the test environment

a to configurations $\{c_2, c_3, c_4, c_5, c_6\}$ and test configuration b to $\{c_8, c_9\}$. Finally, the distance between `TestCfgs` and `RdmpCfgs` is ordered by each distance metric as follows:

Sorted distances between a and `RdmpCfgs` :

- using dm_1: $\{c_1, c_5, c_6, c_7, c_4, c_8, c_3, c_2, c_9\}$
- using dm_2: $\{c_2, c_3, c_4, c_5, c_6, c_1, c_7, c_8, c_9\}$

Sorted distances between b and `RdmpCfgs` :

- using dm_1: $\{c_7, c_3, c_4, c_6, c_1, c_8, c_9, c_2, c_5\}$
- using dm_2: $\{c_8, c_9, c_7, c_3, c_4, c_5, c_2, c_6, c_1\}$

Suppose we consider only the three closest configurations defined by each metric. For `TestCfg` a, dm_1 selects c_1, c_5, c_6, and dm_2 selects c_2, c_3, c_4 as the three closest configurations, respectively. In this case, we would say dm_2 performed better for this local planner, since lp connected all three closest `RdmpCfgs` whereas only two of dm_1's were connected. However, one test point (a) would not be representative of all configurations in C-free. We repeat this process for the next test point b. This time two of the configurations chosen by dm_2 can be connected, while none of the configurations chosen by dm_1 can be. Overall, for a total of six connection attempts using two `TestCfgs`, dm_2 identified five connectable nodes and dm_1 identified only two connectable nodes. As a result we would say that in this type of the environment dm_2 is better for the local planner lp.

We applied this general idea to each of the nine environments we studied. We had 7 distance metrics (described in Section 2.2) and 6 local planners (described in Section 3.2). First, we generated 600 free configurations (`RdmpCfgs`) near C-obstacle surfaces (simulating roadmap nodes). We also generated 100 free configurations (`TestCfgs`) as test nodes; 50 of the `TestCfgs` were near C-obstacle surfaces and 50 were generated at random (generally not near C-obstacles, referred to as free configurations). For the 6-cube environments, we generated 100 `RdmpCfgs` near each cube, and the cubes for the 50 surface `TestCfgs` were selected at random. For the alpha-puzzle the two tubes were intertwined in roughly half the `RdmpCfgs` and were separated in the other half, and similarly for the 50 surface `TestCfgs`. The rings were intertwined in all `RdmpCfgs` and `TestCfgs`. `RdmpCfgs` and the 50 surface `TestCfgs`were uniformly distributed around, but contacting, each edge. The free `TestCfgs` were also intertwined but contact was not required. There are a number of roadmap node generation strategies (e.g., [3, 4, 7, 20, 28, 32, 42]). We used the method described in [3, 4] to generate configurations near the surfaces of C-Space obstacles, and the method described in [32] for free space nodes.

Next, for each local planner, we tried to connect each configuration in `TestCfgs` to every configuration in `RdmpCfgs`.

LOCAL PLANNER	Successful		Failed	
	Time(μsec)	Coll. Checks	Time(μsec)	Coll. Checks
straight-line	25470	1025	9960	56
rotate-at-0	32050	1038	8022	42
rotate-at-$\frac{1}{2}$	30108	1035	11148	80
rotate-at-1	30840	1030	9982	78
A^*-clearance	107940	4010	22841	143
A^*-distance	90870	3755	25321	170

Table 3: Computational requirements of local planning. This table shows average local planning times and number of collision detections

```
for each c in TestCfgs
  for each local planner lp
    for each c' in RdmpCfgs
      try to connect c to each c' using lp
```

An ideal distance metric for a particular local planner would assign the lowest distances to pairs of nodes that can be connected using that planner. In the context of PRMs, one typically would only attempt to connect a given node to the k closest nodes, where k is a relatively small value. Thus, to rate a metric (for a given local planner), we computed the distances using that metric between each node in `TestCfgs` and every node in `RdmpCfgs`, sorted these distances, and *analyzed those connections made to the closest k nodes.* We implemented our experiments for k=25, 50, and 100.

```
for each local planner lp
  for each distance metric dm
    for each c in TestCfgs
      sort RdmpCfgs c' by dm(c,c')
      evaluate lp/dm combination for k closest nodes
```

4.2 Experimental Results

4.2.1 Computational requirements

As previously mentioned, since PRMs perform a huge number of distance computations and local planning queries the computation times of these operations are important factors to consider when choosing among them. Another important factor for the local planners is the number of collision detection checks they make.

Times for evaluating the distance metrics are shown in Table 1; values shown are averages of 10,000 computations. Since the local planner computation times vary according to the problem instance, it is difficult to compare them. However, a general idea of their relative costs can be inferred from the algorithms. For example, we would expect the straight-line planner to be the fastest (among the successful planners). Also, when both rotate-at-0 and rotate-at-1 are successful, we would expect comparable running times.

Some times for example local planning queries are shown in Table 3; values shown are averages of approximately 100 computations. For these tests we consider the hard alpha-alpha environment. We generated 10 `StartCfgs` of the robot on C-obstacle surfaces. For each $c_s \in$ `StartCfgs` we randomly generated 20 `GoalCfgs` at approximately the same distance from c_s (so most `GoalCfgs` are removed from C-obstacle surfaces). We then tried to connect c_s to each $c_g \in$ `GoalCfgs` using each local planner. (See Figure 9.) We report the successful and unsuccessful queries separately in Table 3. Running times and the number of collision detection calls are averages over all relevant connection attempts. As can be seen, the running times conform to our expectations for the successful queries. The A* local planners are about 3 times slower than the other planners in this example (when the distance between the two configurations increases this

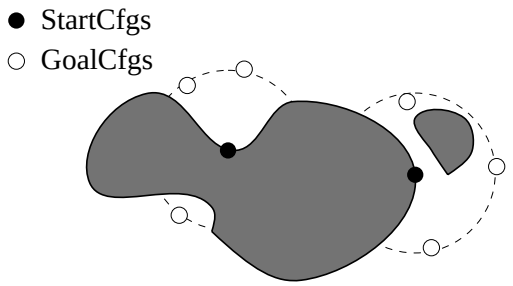


Figure 9: Generation of `StartCfgs` and `GoalCfgs` for the evaluation of the local planner computational requirements

difference increases). Note that for unsuccessful queries, the rotate-at-0 planner has the smallest number of collision detection checks. This is because the we chose our test connections as surface to free space connections, which are more likely to fail at the beginning for the rotate-at-0 local planner. Also since the A* local planners consider alternative paths, it takes longer for them to fail.

4.2.2 Selecting parameters for distance metrics

Three of our metrics require parameters: scaled Euclidean (s), Minkowski (r), and modified Minkowski ($r_1, r_2,$ and r_3). To determine good values for these parameters, we tested several and selected the best for further evaluation. For scaled Euclidean, values of $s = \{0, .1, .25, .5, .75, .9, 1\}$ were tested. For Minkowski, values of $r = \{.5, 1.5, 2.5, 3, 4, 5, 6, 7, 8\}$ were tested. For modified Minkowski, the values tested were $r_i = \{.5, 1.5, 2.5, 3.5\}$, for $i \in \{1, 2, 3\}$; when one parameter was varied, the other two were fixed at 2, and when both r_1 and r_2 were varied, r_3 was fixed at 2.

In order to evaluate the effect of the parameters for each distance metric, we defined a scoring mechanism. The score is meant to favor connections made between the closer nodes (as determined by that metric with that parameter value). That is, if a local planner makes more connections to the "close" nodes as determined using that parameter (for that metric), then that parameter should score higher than the others. For each local planner, we evaluated each distance metric parameter. In order to evaluate the score of a parameter, the nodes were partitioned into five sets according to relative distance: 1 (closest 10), 2 (next 15), 3 (next 25), 4 (next 50), and 5 (remaining 500). The score of a metric for a specific local planner was

$$4(\frac{n_1}{10}) + 3(\frac{n_2}{15}) + 2(\frac{n_3}{25}) + 1(\frac{n_4}{50}),$$

where n_i is the number of connections in set i. Next, we examined two-dimensional and three-dimensional graphs to analyze the scores. In the graphs, the y-axis shows the score. In the two-dimensional graphs the x-axis shows the varying parameter (s, r, r_1, r_2, r_3), while in the three-dimensional graphs, the x-axis shows the r_1 parameter and the z-axis shows the r_2 parameter (r_3 is fixed at the value 2.) Representative graphs for each parameter can be seen in Figures 10–42. In the two-dimensional graphs the average value is the average for all 100 `TestCfgs`, the surface value is the average for the 50 surface `TestCfgs` and the free value is the average for the 50 free `TestCfgs`. To simplify the three-dimensional graphs, we show the values for the surface and the free scores in separate graphs.

A general observation was that the relative importance of the translational distance between the two configurations increased and the importance of rotational distance decreased as the environments became harder. It was also interesting to observe that the hard 6-cube environments display similar characteristics to the easy alpha-puzzle environments — indicating that the alpha-puzzle environments can be considered more difficult than the 6-cube environments. Some local planners displayed similar behaviors, e.g., straight-line and rotate-at-$\frac{1}{2}$ were similar and the two A* local planners were similar. The $r3$ parameter of the modified Minkowski metric had almost no effect, so we omit it in our graphs.

THE BEST SCALED EUCLIDEAN VALUES FOR FREE TO SURFACE CONNECTIONS												
ENVIRONMENTS	LOCAL PLANNERS											
	SL		R-AT-0		R-AT-$\frac{1}{2}$		R-AT-1		A*-CLEAR		A*-DIST	
	Score	s	Score	s	Score	s	Score	s	Score	s	Score	s
Six-Cubes Easy	8.38	.90	8.64	.90	8.66	.90	1.59	.50	8.61	.90	8.59	.90
Six-Cubes Mod.	8.35	.90	8.72	.90	8.69	.90	1.81	.50	8.79	.90	8.73	.90
Six-Cubes Hard	8.58	.90	9.11	.90	9.10	.90	1.75	.50	9.14	.90	9.10	.90
Alpha-Puzzle Easy	4.61	.75	5.34	.90	5.67	.75	2.60	.75	8.08	.75	7.97	.75
Alpha-Puzzle Mod.	4.47	.75	5.44	.90	5.80	.90	3.01	.75	8.28	.75	8.20	.75
Alpha-Puzzle Hard	4.33	.75	5.36	.90	5.47	.90	3.07	.75	7.89	.75	7.73	.75
Rings Easy	3.84	.25	2.83	.25	2.56	.25	2.25	.50	4.18	.25	4.11	.25
Rings Mod.	3.69	.25	2.60	.25	2.37	.25	1.99	.50	3.78	.25	3.71	.25
Rings Hard	1.62	.50	0.82	.50	0.69	.50	0.77	.50	1.52	.50	1.51	.50
THE BEST SCALED EUCLIDEAN VALUES FOR SURFACE TO SURFACE CONNECTIONS												
ENVIRONMENTS	LOCAL PLANNERS											
	SL		R-AT-0		R-AT-$\frac{1}{2}$		R-AT-1		A*-CLEAR		A*-DIST	
	Score	s	Score	s	Score	s	Score	s	Score	s	Score	s
Six-Cubes Easy	2.71	.25	1.35	.25	2.82	.50	1.26	.25	3.38	.50	3.34	.50
Six-Cubes Mod.	2.86	.75	1.64	.75	2.98	.75	1.56	.75	3.13	.75	3.05	.75
Six-Cubes Hard	1.23	.75	0.96	.75	0.95	.75	0.95	.90	1.84	.75	1.80	.75
Alpha-Puzzle Easy	2.38	.90	2.88	.90	1.01	.75	1.90	.75	4.71	.75	4.70	.75
Alpha-Puzzle Mod.	1.77	.90	3.23	.90	0.91	.75	1.13	.90	3.96	.90	3.98	.90
Alpha-Puzzle Hard	1.51	.90	2.58	.90	0.82	.90	1.17	.90	3.56	.90	3.51	.90
Rings Easy	4.04	.50	2.30	.50	2.42	.25	2.51	.50	4.56	.50	4.52	.50
Rings Mod.	3.48	.50	2.02	.50	2.29	.50	2.06	.50	3.99	.50	3.97	.50
Rings Hard	2.48	.90	2.53	.90	1.16	.90	1.30	.90	3.07	.75	3.06	.75
Local Planners Used												
SL straight-line **R-at-0** rotate-at-0 **R-at-$\frac{1}{2}$** rotate-at-$\frac{1}{2}$ **R-at-1** rotate-at-1 **A*-Clear** A*-Clearance **A*-Dist** A*-Distance												

Table 4: The best scaled Euclidean values. The best s parameter value for the Scaled Euclidean distance in each environment and each local planner for free to surface (top) and surface to surface connections (bottom).

6-cubes

For the scaled Euclidean metric in the 6-cube environments (Table 4 and Figures 10–15), note that the optimal value for the scaled Euclidean parameter depends on whether the test points are near C-obstacle surfaces (recall all `RdmpCfgs` are near C-obstacle surfaces). For example, Figure 10 shows that for the easy environment the rotational distance between the two configurations is most important when both configurations are near the surface ($s = .25$), but the translational distance is more important for free to surface connections ($s = .9$). This difference decreases, however, as the environment becomes harder (Figure 11). This behavior also can be observed in the Table 4, where the optimal s increases as the environment gets harder. It is clear from the tables and the graphs that all of the local planners show similar behavior with respect to the s parameter. The only exception is rotate-at-1 where the free to surface connections reached the peak at $s = 0.5$. This may be related to the fact that rotate-at-1 rotates the robot at the goal position, which, in this case is a surface node with little space to maneuver. Similarly the low score for the free to surface connections shows that they could not make many "close" connections. We will see a similar behavior of rotate-at-1 in Section 4.2.3. Table 4 and Figures 10–15 indicate that $s = 0.25, 0.5, 0.75$ and 0.9 are significant to investigate further.

Figures 16–18 show some graphs for the r parameter for the Minkowski metric. In general, almost all the local planners reached their peak value at $r = 4$ for the free to surface connections, regardless of the environment. After reaching their peak, the scores stay around that value as r increases. The graphs for the other local planners are similar to Figure 16. However, again we see that rotate-at-1 has a different shape. It reaches a peak at $r = 1.5$ and then decreases. It is noticeable that in Figures 17 and 18 as the environment

THE BEST MINKOWSKI VALUES FOR FREE TO SURFACE CONNECTIONS												
ENVIRONMENTS	LOCAL PLANNERS											
	SL		R-AT-0		R-AT-$\frac{1}{2}$		R-AT-1		A*-CLEAR		A*-DIST	
	Score	r	Score	r	Score	r	Score	r	Score	r	Score	r
Six-Cubes Easy	7.84	3.0	8.03	4.0	8.03	4.0	1.60	1.5	8.02	4.0	8.03	4.0
Six-Cubes Mod.	7.83	4.0	8.09	4.0	8.07	4.0	1.86	1.5	8.23	4.0	8.21	3.0
Six-Cubes Hard	8.17	4.0	8.55	4.0	8.53	4.0	1.74	1.5	8.66	4.0	8.66	4.0
Alpha-Puzzle Easy	4.48	2.5	5.08	4.0	5.53	2.5	2.58	2.5	8.01	4.0	7.92	2.5
Alpha-Puzzle Mod.	4.42	2.5	5.13	3.0	5.45	3.0	3.02	3.0	8.06	2.5	7.99	2.5
Alpha-Puzzle Hard	4.23	3.0	5.08	3.0	5.24	3.0	2.94	3.0	7.70	3.0	7.58	3.0
Rings Easy	3.74	3.0	2.59	2.5	2.34	3.0	2.25	3.0	3.98	3.0	3.90	3.0
Rings Mod.	3.73	3.0	2.60	6.0	2.31	6.0	1.99	3.0	3.80	6.0	3.71	3.0
Rings Hard	1.64	3.0	0.82	5.0	0.69	3.0	0.78	1.5	1.53	3.0	1.52	3.0
THE BEST MINKOWSKI VALUES FOR SURFACE TO SURFACE CONNECTIONS												
ENVIRONMENTS	LOCAL PLANNERS											
	SL		R-AT-0		R-AT-$\frac{1}{2}$		R-AT-1		A*-CLEAR		A*-DIST	
	Score	r	Score	r	Score	r	Score	r	Score	r	Score	r
Six-Cubes Easy	2.79	5.0	1.34	1.5	3.0	8.0	1.19	1.5	3.45	5.0	3.40	5.0
Six-Cubes Mod.	2.74	3.0	1.56	1.5	2.88	3.0	1.45	1.5	2.93	1.5	2.89	1.5
Six-Cubes Hard	1.15	1.5	0.95	1.5	0.94	2.5	0.87	1.5	1.74	1.5	1.71	1.5
Alpha-Puzzle Easy	2.07	1.5	2.31	0.5	0.98	2.5	1.79	1.5	4.24	1.5	4.24	1.5
Alpha-Puzzle Mod.	1.35	0.5	2.54	1.5	0.86	1.5	0.98	1.5	3.11	1.5	3.12	1.5
Alpha-Puzzle Hard	1.16	0.5	1.98	0.5	0.73	2.5	1.10	1.5	3.02	1.5	3.0	1.5
Rings Easy	4.01	3.0	2.34	4.0	2.38	3.0	2.48	1.5	4.57	3.0	4.53	3.0
Rings Mod.	3.47	3.0	2.06	3.0	2.32	4.0	2.05	1.5	4.0	3.0	3.98	3.0
Rings Hard	2.11	1.5	2.22	4.0	1.06	2.5	1.11	1.5	2.78	2.5	2.77	2.5
Local Planners Used												
SL straight-line **R-at-0** rotate-at-0 **R-at-$\frac{1}{2}$** rotate-at-$\frac{1}{2}$ **R-at-1** rotate-at-1 **A*-Clear** A*-Clearance **A*-Dist** A*-Distance												

Table 5: The best Minkowski values. The best r parameter value for the Minkowski distance in each environment and each local planner for free to surface (top) and surface to surface connections (bottom).

gets harder, the slope of the curve decreases. Table 5 shows that most of the local planners reach their highest score for the surface to surface connections at $r = 1.5$. Our analysis of the graphs generally showed that if $r = 1.5$ was not the highest score, then it was the second best. The graphs and the tables suggest that $r = 1.5$ and 4 would be good choices for r.

The graphs (Figures 19, and 20) for the r_2 parameter of the modified Minkowski metric have the opposite shape as the s parameter graphs, but indicate the same relation. In Figure 19 as the r_2 parameter increases, the score for free to surface connections decreases while the score for surface to surface connections increases. This shows that in the easy 6-cube environment, the free to surface connections give more importance to translational distance and the surface to surface connections give more importance to rotational distance. However, as the environment gets harder, the importance moves to translational distance for both types of connections.

In Figures 22 and 23 we see the parameter graphs for the r_1 and r_2 parameters for surface to surface and free to surface connections in the easy environment. It is clear from the graphs that in the easy environments the free to surface connections favor the r_1 parameter (Figure 23), while the surface to surface connections favor the r_2 parameter (Figure 22). This also indicates that the free to surface connections give more importance to the translational distance (related to r_1) and surface to surface connections give more importance to rotational distance (related to r_2). Finally, while these results support those for the s parameter for the scaled Euclidean metric we see, as noted in Section 2.2, larger values of r_1 are required to achieve analogous results. Based on the graphs 19 - 23 and Table 6, our suggestion for the modified Minkowski values is $(r_1, r_2, r_3) = \{(2, 1.5, 2), (2, 2.5, 2)\}$. Note that the first triple favors translational distance while the second triple favors rotational distance.

THE BEST MODIFIED MINKOWSKI VALUES FOR FREE TO SURFACE CONNECTIONS						
ENVIRONMENTS	LOCAL PLANNERS					
	SL		R-AT-0		R-AT-$\frac{1}{2}$	
	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$
Six-Cubes Easy	8.63	{2.5,0.5,2.0}	8.64	{2.0,0.5,2.0}	8.66	{2.0,0.5,2.0}
Six-Cubes Mod.	8.34	{2.0,0.5,2.0}	8.69	{2.0,0.5,2.0}	8.67	{2.0,0.5,2.0}
Six-Cubes Hard	8.60	{2.0,0.5,2.0}	9.10	{2.0,0.5,2.0}	9.09	{2.0,0.5,2.0}
Alpha-Puzzle Easy	5.33	{1.5,0.5,2.0}	5.37	{2.0,1.5,2.0}	5.64	{2.0,1.5,2.0}
Alpha-Puzzle Mod.	5.50	{1.5,0.5,2.0}	5.50	{1.5,0.5,2.0}	5.82	{2.0,1.5,2.0}
Alpha-Puzzle Hard	4.32	{2.0,1.5,2.0}	5.36	{2.0,1.5,2.0}	5.46	{2.0,1.5,2.0}
Rings Easy	3.95	{3.5,2.0,2.0}	2.90	{3.5,2.0,2.0}	2.86	{3.5,1.5,2.0}
Rings Mod.	3.78	{3.5,2.0,2.0}	2.67	{3.5,2.0,2.0}	2.40	{3.5,2.0,2.0}
Rings Hard	1.65	{3.5,2.0,2.0}	0.83	{3.5,2.0,2.0}	0.83	{3.5,2.5,2.0}

THE BEST MODIFIED MINKOWSKI VALUES FOR FREE TO SURFACE CONNECTIONS						
ENVIRONMENTS	LOCAL PLANNERS					
	R-AT-1		A*-CLEAR		A*-DIST	
	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$
Six-Cubes Easy	8.63	{2.5,0.5,2.0}	8.63	{2.5,0.5,2.0}	8.63	{2.5,0.5,2.0}
Six-Cubes Mod.	2.93	{1.5,0.5,2.0}	8.77	{2.0,0.5,2.0}	8.72	{2.0,0.5,2.0}
Six-Cubes Hard	2.72	{0.5,0.5,2.0}	9.13	{2.0,0.5,2.0}	9.09	{2.0,0.5,2.0}
Alpha-Puzzle Easy	5.33	{1.5,0.5,2.0}	8.05	{2.0,1.5,2.0}	7.92	{2.0,1.5,2.0}
Alpha-Puzzle Mod.	5.50	{1.5,0.5,2.0}	8.32	{2.0,1.5,2.0}	8.22	{2.0,1.5,2.0}
Alpha-Puzzle Hard	4.11	{1.5,0.5,2.0}	7.86	{2.0,1.5,2.0}	7.68	{2.0,1.5,2.0}
Rings Easy	2.86	{3.5,1.5,2.0}	4.27	{3.5,2.0,2.0}	4.19	{3.5,2.0,2.0}
Rings Mod.	2.16	{0.5,1.5,2.0}	3.86	{3.5,2.0,2.0}	3.79	{3.5,2.0,2.0}
Rings Hard	0.83	{3.5,2.5,2.0}	1.54	{3.5,2.0,2.0}	1.53	{3.5,2.0,2.0}

THE BEST MODIFIED MINKOWSKI VALUES FOR SURFACE TO SURFACE CONNECTIONS						
ENVIRONMENTS	LOCAL PLANNERS					
	SL		R-AT-0		R-AT-$\frac{1}{2}$	
	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$
Six-Cubes Easy	2.79	{2.0,2.5,2.0}	1.34	{1.5,2.0,2.0}	2.89	{2.0,2.5,2.0}
Six-Cubes Mod.	2.98	{3.5,2.0,2.0}	1.65	{3.5,2.0,2.0}	3.16	{3.5,2.0,2.0}
Six-Cubes Hard	1.23	{2.0,0.5,2.0}	0.97	{2.0,1.5,2.0}	0.98	{3.5,2.0,2.0}
Alpha-Puzzle Easy	2.85	{3.5,2.5,2.0}	2.90	{2.0,1.5,2.0}	2.85	{3.5,2.5,2.0}
Alpha-Puzzle Mod.	1.74	{2.0,1.5,2.0}	3.21	{2.0,1.5,2.0}	0.88	{2.5,2.0,2.0}
Alpha-Puzzle Hard	1.60	{3.5,2.0,2.0}	2.67	{3.5,2.0,2.0}	0.83	{3.5,2.0,2.0}
Rings Easy	4.04	{2.0,2.0,1.5}	2.37	{3.5,2.0,2.0}	2.39	{3.5,2.0,2.0}
Rings Mod.	3.51	{2.5,2.0,2.0}	2.07	{2.5,2.0,2.0}	2.34	{3.5,2.0,2.0}
Rings Hard	2.22	{2.0,0.5,2.0}	2.30	{2.0,0.5,2.0}	1.52	{2.5,0.5,2.0}

THE BEST MODIFIED MINKOWSKI VALUES FOR SURFACE TO SURFACE CONNECTIONS						
ENVIRONMENTS	LOCAL PLANNERS					
	R-AT-1		A*-CLEAR		A*-DIST	
	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$	Score	$\{r_1, r_2, r_3\}$
Six-Cubes Easy	1.24	{1.5,2.0,2.0}	3.45	{2.0,2.5,2.0}	3.41	{2.0,2.5,2.0}
Six-Cubes Mod.	1.53	{3.5,2.0,2.0}	3.18	{3.5,2.0,2.0}	3.11	{3.5,2.0,2.0}
Six-Cubes Hard	0.93	{2.0,0.5,2.0}	1.80	{2.0,0.5,2.0}	1.75	{2.0,0.5,2.0}
Alpha-Puzzle Easy	2.85	{3.5,2.5,2.0}	4.77	{2.5,2.0,2.0}	4.75	{2.5,2.0,2.0}
Alpha-Puzzle Mod.	1.10	{2.0,1.5,2.0}	3.88	{2.0,1.5,2.0}	3.91	{2.5,2.0,2.0}
Alpha-Puzzle Hard	1.17	{2.0,1.5,2.0}	3.54	{2.0,1.5,2.0}	3.49	{2.0,1.5,2.0}
Rings Easy	2.52	{2.0,1.5,2.0}	4.56	{2.0,3.5,2.0}	4.52	{2.0,3.5,2.0}
Rings Mod.	2.06	{2.0,1.5,2.0}	4.04	{2.5,2.0,2.0}	4.01	{2.5,2.0,2.0}
Rings Hard	1.52	{2.5,0.5,2.0}	2.86	{2.0,0.5,2.0}	2.85	{2.0,0.5,2.0}
Local Planners Used						
SL straight-line **R-at-0** rotate-at-0 **R-at-$\frac{1}{2}$** rotate-at-$\frac{1}{2}$ **R-at-1** rotate-at-1 **A*-Clear** A*-Clearance **A*-Dist** A*-Distance						

Table 6: The best modified Minkowski values. The best $\{r_1, r_2, r_3\}$ parameters value for the Modified Minkowski distance in each environment and local planner for free to surface (top) and surface to surface connections (bottom).

Alpha-puzzle

The alpha-puzzle environments (Figures 24–31) display similar trends as the 6-cube environments. It can also be seen that the graphs for the hard 6-cube environment are very similar in shape to those for the easy alpha-puzzle environment, which indicates, as mentioned above, that the alpha-puzzle environments are harder.

The graphs (Figures 24 and 25) for the s parameter for the scaled Euclidean distance have similar shapes for each local planner. The surface to surface connections always peak around $s = 0.9$. However, depending on the environment, the score is highest either at $s = 0.75$ or $s = 0.9$. We choose both values for our metrics, as well as $s = 0.25$, in order to see the effect of a small s in this environment. Note that in Table 4 it is clearly shown that the optimal s value gets higher as the environment gets harder.

The Minkowski parameter graphs (Figures 26 and 27) and Table 5 display two common trends in the alpha-puzzle environment. The free to surface connection score always reaches the highest value at either $r = 2.5$ or $r = 3$ while the highest value for surface to surface connections is around $r = 1.5$. Since the score for $r = 2.5$ was in general close to $r = 3$, we choose that value as well as $r = 1.5$.

In Figures 28–29, the r_2 parameter for alpha-puzzle environment behaves similarly to the r_2 parameter for the hard 6-cube. Finally in the $r_1 - r_2$ graphs (Figures 30 and 31), both surface to surface connections and free to surface connections show that the alpha-puzzle environments favor r_1, i.e., the translational distance. Our choice for the modified Minkowski values are $(r_1, r_2, r_3) = \{(2, 1.5, 2), (2, 2.5, 2), (2.5, 2, 2)\}$. The first and the last triples favor the translational distance and the second triple favors rotational distance.

Rings

The first thing to notice for the rings environments in Figures 32–41 is that the surface to surface connections usually have better scores. This may be explained by the fact that the surface nodes have more rotational freedom than the free nodes due to the shape of the objects .

Some interesting graphs for the s parameter of scaled Euclidean are shown in Figures 32–35. The surface to surface connections usually have the shape shown in the Figures 32 and 33. However, a few of the graphs have the shape shown in Figure 35. In Figure 34 we can see that both surface to surface and free to surface connections reach their peak at $s = 0.25$. In Table 4 the $s = 0.75$ value is only best for A*-clearance and A*-distance in the hard rings environment. However, the score for $s = 0.9$ was a very close second. We chose $s = 0.25, 0.5, 0.75$ and 0.9 for our experiments. The peak value for the free to surface connections was at most $s = 0.5$ in all of the environments. This fact can be seen in Table 4. We also noticed that in the easy and moderate environments the s value is always less than 0.5. Our conclusion is that for the ring environments the free to surface connections always give importance to rotational distance. Surface to surface connections favor the same in easier environments, but favor the translational distance in harder environments.

The graphs (Figures 36 and 37) for the r parameter of the Minkowski metric show three important values of r. In 36, the surface to surface connections have the highest score at $r = 3$, whereas the free to surface connections have the highest value at $r = 2.5$. However in Figure 37, both reach their peak at $r = 1.5$. We chose $r = 3$ and $r = 1.5$ for our experiments, since if they were not always the best, they were competitive with the others.

One interesting behavior of the modified Minkowski metric can be seen in Figure 38 where the r_2 value starts decreasing at $r_2 = 1.5$ for the free to surface connections , while the value continues to increase for surface to surface connections . So upto some point, the free to surface connections favor r_2, whereas the surface to surface ones always do. Notice that this is consistent with the s graphs, where both types of connections favor rotational distance in easy environments. However, as can be seen in Figure 39, the decrease in surface to surface connection scores says the opposite is true for them, i.e., as the environment gets harder they don't favor rotational distance while free to surface connections continue to favor it until

LOCAL PLANNER	METRICS					
	EASY		MODERATE		HARD	
	good	recom.	good	recom.	good	recom.
SL (cubes)	$\{E, \mathbf{SE}, MM\}$	SE (.5)	$\{\mathbf{SE}, MM\}$	SE (.75)	$\{SE, \mathbf{MM}, CM\}$	SE (.75)
LP1 (alpha)	$\{\mathbf{SE}, M, MM\}$	SE (.75)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, MM, \mathbf{CM}\}$	SE (.9)
(rings)	$\{E, SE, M, \mathbf{MM}\}$	SE (.5)	$\{E, SE, \mathbf{M}\}$	SE (.5)	$\{E, \mathbf{SE}\}$	SE (.75)
R-at-0 (cubes)	$\{SE, \mathbf{MM}, CM\}$	SE (.5)	$\{\mathbf{SE}, MM, CM\}$	SE (.9)	$\{SE, \mathbf{MM}, CM\}$	SE (.9)
LP2 (alpha)	$\{E, \mathbf{SE}, M, MM\}$	SE (.75)	$\{SE, \mathbf{MM}\}$	SE (.75)	$\{SE, \mathbf{MM}\}$	SE (.75)
(rings)	$\{SE, \mathbf{MM}, BB\}$	SE (.25)	$\{E, SE, M, \mathbf{MM}\}$	SE (.5)	$\{SE, MM, \mathbf{BB}\}$	SE (.5)
R-at-$\frac{1}{2}$ (cubes)	$\{SE, \mathbf{M}, MM\}$	SE (.5)	$\{SE, \mathbf{MM}\}$	SE (.75)	$\{SE, \mathbf{MM}.CM\}$	SE (.75)
LP3 (alpha)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, MM, \mathbf{CM}\}$	SE (.9)
(rings)	$\{SE, MM, \mathbf{BB}\}$	SE (.25)	$\{SE, MM, \mathbf{BB}\}$	SE (.25)	$\{E, \mathbf{SE}, MM\}$	SE (.5)
R-at-1 (cubes)	$\{\mathbf{SE}, MM\}$	SE (.25)	$\{E, SE, \mathbf{M}\}$	SE (.75)	$\{\mathbf{E}, SE, M\}$	SE (.75)
LP4 (alpha)	$\{SE, M, \mathbf{MM}\}$	SE (.75)	$\{\mathbf{SE}, MM\}$	SE (.75)	$\{\mathbf{SE}, MM\}$	SE (.75)
(rings)	$\{E, SE, M, \mathbf{MM}\}$	SE (.5)	$\{E, SE, \mathbf{M}, MM\}$	SE (.5)	$\{\mathbf{SE}, M\}$	SE (.75)
A^*-clear (cubes)	$\{\mathbf{SE}, M, MM\}$	SE (.5)	$\{SE, \mathbf{MM}\}$	SE (.75)	$\{SE, \mathbf{MM}\}$	SE (.75)
LP5 (alpha)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, \mathbf{MM}\}$	SE (.9)
(rings)	$\{E, SE, M, \mathbf{MM}\}$	SE (.5)	$\{E, SE, \mathbf{M}, MM\}$	SE (.5)	$\{E, \mathbf{SE}\}$	SE (.5)
A^*-dist (cubes)	$\{E, \mathbf{SE}, M\}$	SE (.5)	$\{SE, \mathbf{MM}\}$	SE (.75)	$\{SE, \mathbf{MM}\}$	SE (.75)
LP6 (alpha)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, \mathbf{MM}\}$	SE (.9)	$\{SE, \mathbf{MM}\}$	SE (.9)
(rings)	$\{SE, \mathbf{M}, MM\}$	SE (.5)	$\{SE, \mathbf{M}\}$	SE (.5)	$\{E, \mathbf{SE}, MM\}$	SE (.5)
Distance Metrics Used						
E Euclidean **SE** Scaled Euclidean **M** Minkowski **MM** Modified Minkowski **CM** Center of Mass **BB** Bounding Box						
Local Planners Used						
SL straight-line **R-at-0** rotate-at-0 **R-at-$\frac{1}{2}$** rotate-at-$\frac{1}{2}$ **R-at-1** rotate-at-1 **A*-clear** A*-Clearance **A*-dist** A*-Distance						

Table 7: Recommended distance metrics. This table matches distance metrics with local planners. The labels associated with the distance metrics are as shown at the bottom of the table. The label in bold is the metric that was best for that local planner in our tests.

some point. The $r_1 - r_2$ graphs of Figures 40 and 41 show this trend clearly. The parameter values we choose for the modified Minkowski metric are $(r_1, r_2, r_3) = \{(2, 1.5, 2), (2, 2.5, 2)\}$.

4.2.3 Selecting metrics for local planners

After selecting interesting parameter values for the various metrics, we were left with a total of 12 different distance metrics. Each local planner was evaluated with these 12 metrics for each of the six environments.

A summary of the results and metric recommendations for each local planner are shown in Table 7. Generally, the best metrics placed more importance on the translational distance than on the rotational distance. Our recommendations take both efficiency and effectiveness into account. When results differed for free to surface and surface to surface connections, a compromise is proposed. When the best metric is computationally expensive, we suggest a more efficient alternative (as long as effectiveness is comparable).

Generally, the metrics performed comparably in all of the environments. The exceptions, however, were the center of mass metric (D11) and the bounding-box metric (D12). The center of mass performs poorly in the rings environment, likely because the center of masses were very close to each other when the rings were intertwined and so it was not useful for differentiating among different configurations. The bounding box did not do as well in the alpha-puzzle environments as it did in the 6-cube and rings environments — likely because the bounding-box is not a very good approximation of the α-shape, whereas it is exact for the cubes and reasonable for the rings. A better metric would likely be to use extreme vertices of the object instead of bounding-box vertices.

Both A* local planners behaved similarly for the various distance metrics. We believe this is because we evaluated only three neighbors, all of which are in the direction of the goal configuration. However, usually A*-clearance performed better. We believe this is because this A* planner tries to avoid the obstacles, and thus gains some advantages over A*-distance with this limited number of neighbors.

Detailed results are shown in Figures 43–77. For each environment, there is one graph for each local

planner, and each set of bars corresponds to one of the three versions of the environment. (Note that the y-scales of the graphs vary.) The bottom black portion of each bar shows the average number of surface `TestCfgs` (surface to surface connections) that were connected to `RdmpCfgs` by that local planner which *that distance metric placed among the closest k nodes where k is either* 25 *or* 100. The top white portion of the bar shows the same for the free `TestCfgs` (free to surface connections).

Note that in some cases fewer connections were made in the easy versions of an environment than in the moderate or hard versions, and that usually the differences are due to fewer connections for surface `TestCfgs` (e.g. Figure 63). The explanation for this is that in the easy 6-cube environment we generated 100 `RdmpCfgs` near each cube, and we only consider connections from each `TestCfg` to the k closest nodes (determined by that metric). Thus, since the cubes are far apart in the easy environment, all surface to surface connections reported were between configurations on or near the same cube – which are harder to connect.

The effect of k - the number of closest configurations

The scoring mechanism we used to select parameters for the distance metrics (Section 4.2.2) places most importance on the first two sets (i.e., the closest 25 nodes). To investigate whether our parameter selection was biased towards the 25 closest nodes at the expense of the closest 100 nodes, we performed experiments for $k = 25$, 50 and 100.

Recall that we concluded in Section 4.2.2 that the optimal s value for surface to surface connections in the easy 6-cube environment was $s = 0.25$. The results for $k = 25$ shown in Figure 43 support this conclusion (the black portion is surface to surface connections). In contrast, the results for $k = 100$ shown in Figure 60 indicate that optimal value would be $s = 0.75$ or $s = 0.9$. To investigate this issue further, we examined the case for $k = 50$ (Figure 87). It is clear from the figures that as k approaches 25 we see the behavior predicted by our parameter scores (rotational distance increases in importance), but as k grows larger then the translational distance takes on greater importance (larger s).

We see the same trend for the modified Minkowski metric (D8) with $(r_1, r_2, r_3) = (2.0, 0.5, 2)$. The efficiency of this metric in the easy 6-cube environment decreases with decreasing values of k. This suggests that the rotational distance is important for the closest nodes. However, the moderate and hard 6-cube environments do not seem to be affected as much by the larger k values.

A similar situation exists for the alpha-puzzle environments (Figures 53 and 70). The efficiency of the scaled Euclidean metric (D4) with $s = 0.90$ decreases as k decreases. In Figures 43, 53, 60 and 70 (in both environments) we also see that the performance of the center of mass metric (D11) decreases with k. This suggests that as the nodes get closer the efficiency of this metric declines.

In the rings environment we don't see much difference for different values of k. In some cases, the center of mass metric (D11) performs a little worse. The reason for this insensitivity to k may be that the nodes other than the k closest may already be very close so that they are affected by both the rotational and translational variances.

Despite the differences mentioned above, it was generally true that variations in k were only slightly correlated with changes in metric performance. Moreover, although there were sometimes slight changes in the best metric for a given environment, the best metric for $k = 25$ still performed well for $k = 50$ or $k = 100$.

Another observation was that the performance of rotate-at-$\frac{1}{2}$ increases with k. This may be related to the surface nodes. Recall that the rotate-at-$\frac{1}{2}$ local planner rotates halfway between the two configurations. When k is small, the two surface configurations will be close together and the rotation point will also be close to the surface. In this case, a collision is likely to occur. However, when k is large, the two surface nodes may be farther apart and perhaps the rotation point is farther from the surface.

4.2.4 Selecting local planners for environments

After determining which distance metrics were best suited for each local planner in each type of environment, we then compared the local planner and distance metric combinations. The straight-line and rotate-at-$\frac{1}{2}$ planners had similar behavior, while the A* local planners were generally the best planners.

6-cubes

Figures 78 and 79 show the results for the 6-cube environments. Figure 78 shows that when $k = 25$ the best local planners in all the 6-cube environments are the A^* planners. Figure 79 shows the same for $k = 100$, except the moderate environment where rotate-at-$\frac{1}{2}$ is slightly better. In general, rotate-at-$\frac{1}{2}$ outperforms the common straight-line planner. This can be explained by the fact that the swept volume of the rotate-at-$\frac{1}{2}$ planner is generally considerably less than for the straight-line planner. We note that the performance of A^* planners would improve if their maximum number of iterations is increased.

The dramatic difference between the seemingly symmetrical rotate-at-0 and rotate-at-1 can be explained by the fact that the goal configuration is always near a C-obstacle surface. Thus, rotations at the goal ($s = 1$) will likely involve collision, but rotations at the start ($s = 0$) are less likely to cause collisions for free to surface connections. This reasoning also explains why we do in fact see similar behavior for $s = 0$ and $s = 1$ when the test configurations are near C-obstacle surfaces – rotations at the start or goal are both likely to cause collision.

Notice that the best distance metrics in the easy environment do not contain the metrics that favor rotational distances. This is because we choose the overall performance, where the free to surface connections are also taken into consideration.

Alpha-puzzle

Figures 80 and 81 show the results for the the alpha-puzzle environments. The results are similar to the 6-cube environments. That is, the A* planners perform the best in all versions for both $k = 25$ and $k = 100$. Here the rotate-at-$\frac{1}{2}$ planner outperformed the rotate-at-0 and straight-line planners for $k = 25$, but rotate-at-0 is best for $k = 100$ (as previously discussed).

Rings

Figures 82 and 83 represent our final set of the environments. As mentioned in Section 4.2.3, variations in k have almost no effect in these environments. These graphs again prove the power of the A^* planners which perform best in all cases. Here the straight-line local planner made more connections than the rotate-at-s planners. Perhaps this is because the rotate-at-s planners completely rotate at the s point, while the straight-line rotates just a little bit at each step and, due to the shape of the rings, it is difficult for them to completely rotate at any particular configuration.

4.2.5 Using multiple local planners

Our results indicate that if only one local planner is to be used, then probably the best choice is rotate-at-$\frac{1}{2}$. However, a PRM could easily utilize more than one local planner. Would this be useful? That is, do different local planners make different connections?

Individual local planners

The results shown in Table 8 indicate that the answer is yes. In each row (environment), the number of connections made *only* by that local planner is shown. The percentage of the connections uniquely made by

CONNECTIONS MADE BY ONLY ONE LOCAL PLANNER					
ENVIRONMENT	ROTATE-AT-0	ROTATE-AT-$\frac{1}{2}$	ROTATE-AT-1	A*-CLEARANCE	A*-DISTANCE
Six-Cubes Easy	2105 (11.52%)	198 (1.08%)	385 (2.11%)	73 (0.40%)	69 (0.38%)
Six-Cubes Moderate	1518 (8.70%)	260 (1.49%)	440 (2.52%)	106 (0.61%)	46 (0.26%)
Six-Cubes Hard	218 (1.68%)	166 (1.28%)	302 (2.32%)	51 (0.39%)	36 (0.28%)
Alpha-Puzzle Easy	83 (0.48%)	649 (3.79%)	554 (3.24%)	73 (0.43%)	155 (0.91%)
Alpha-Puzzle Moderate	116 (0.68%)	1984 (11.70%)	808 (4.77%)	108 (0.64%)	155 (0.91%)
Alpha-Puzzle Hard	403 (2.43%)	1661 (10.01%)	979 (5.90%)	172 (1.04%)	159 (0.96%)
Rings Easy	113 (1.23%)	281 (3.05%)	130 (1.41%)	101 (1.10%)	32 (0.35%)
Rings Moderate	114 (1.59%)	329 (4.58%)	108 (1.51%)	122 (1.70%)	15 (0.21%)
Rings Hard	92 (2.06%)	879 (19.66%)	242 (5.41%)	74 (1.65%)	14 (0.31%)

Table 8: Connections made by only one local planner. This table shows the number of connections that each planner found where others failed. Straight Line is not shown since, every connection that can be found by straight-line local planner, is found by A*'s.

CONNECTIONS ONLY MADE BY A* LOCAL PLANNERS		
ENVIRONMENT	A*-CLEARANCE	A*-DISTANCE
Six-Cubes Easy	4038 (22.09%)	4034 (22.07%)
Six-Cubes Moderate	6866 (39.37%)	6806 (39.03%)
Six-Cubes Hard	5601 (43.05%)	5586 (42.94%)
Alpha-Puzzle Easy	4074 (23.80%)	4156 (24.28%)
Alpha-Puzzle Moderate	5180 (30.56%)	5227 (30.84%)
Alpha-Puzzle Hard	6440 (38.80%)	6427 (38.72%)
Rings Easy	3448 (37.42%)	3379 (36.67%)
Rings Moderate	2644 (36.85%)	2537 (35.35%)
Rings Hard	1787 (39.96%)	1727 (38.62%)

Table 9: Connections only made by A* planners. This table shows the number of connections that were only made by the A* planners.

that local planner with respect to the total number of connections made using all six local planners is shown in parenthesis. Note that the A* local planners have very low values; this is because they both made many of the same connections. Similarly, the rotate-at-$\frac{1}{2}$ values are often low because the same connections were made by the straight-line or the A* planners. Finally, the straight-line values were all 0 (not shown) because all straight-line connections were made by the A* planners as well.

The efficiency of the A* planners can be seen in Table 9. In each column (local planner) of the table, the number of connections uniquely made by that local planner is shown. However, in this table the nodes connected by the other A* planner are not considered. Each column shows the comparison with the indicated A* planner and the straight-line and rotate-at-s planners. It is clear from the table that in every environment, at least 22% of the total connections are made by one of the A* local planners. Also, notice that as the environment gets harder, this percentage increases.

A similar table (Table 10), shows the same statistics for the rotate-at-s local planners. Each local planner in the table is compared with the A* local planners. Although the results are not as impressive as Table 9, they show that the rotate-at-s planners made connections that the A* planners could not. Note the relatively low numbers for rotate-at-$\frac{1}{2}$ are because the straight-line planner can make many of the same connections.

CONNECTIONS ONLY MADE BY ROTATE-AT-S LOCAL PLANNERS			
ENVIRONMENT	ROTATE-AT-0	ROTATE-AT-$\frac{1}{2}$	ROTATE-AT-1
Six-Cubes Easy	3130 (17.12%)	978 (5.35%)	753 (4.12%)
Six-Cubes Moderate	7024 (40.28%)	5533 (31.73%)	1501 (8.61%)
Six-Cubes Hard	4974 (38.24%)	4882 (37.53%)	997 (7.66%)
Alpha-Puzzle Easy	325 (1.90%)	893 (5.22%)	561 (3.28%)
Alpha-Puzzle Moderate	562 (3.32%)	2398 (14.15%)	968 (5.71%)
Alpha-Puzzle Hard	1928 (11.62%)	3098 (18.66%)	1371 (8.26%)
Rings Easy	200 (2.17%)	365 (3.96%)	136 (1.48%)
Rings Moderate	208 (2.90%)	417 (5.81%)	121 (1.69%)
Rings Hard	255 (5.70%)	1012 (22.63%)	344 (7.69%)

Table 10: Connections only made by rotate-at-s planners. This table shows the number of connections that were only made by the rotate-at-s planners.

Combinations of two local planners

Tables 11–17 show the comparison of individual local planners. In Table 11, the first column shows connections made by the straight-line but not by the rotate-at-$\frac{1}{2}$ planner and in the second column vice versa. The third column shows the number of connections made by both planners and the last column total number of connections that were made by either the straight-line or the rotate-at-$\frac{1}{2}$ planner. Notice that the combined performance of these two local planners is very high, usually more than 50% of all connections.

Noticeable among these is Table 16 which shows that usually more than 85% of all connections are made by the rotate-at-$\frac{1}{2}$ and A*-clearance planner combination. In the alpha-puzzle and rings environments, combinations of the rotate-at-0 and A*-clearance planners look even more powerful. However, we believe that this is an artifact of the free to surface connections, since our analysis of the free to surface connections showed that rotate-at-0 performed better for them in these environments (since it rotated at the start which is free), while its connection rate for surface to surface connections was very poor (see Figures 85 and 86).

Combinations of three local planners

Tables 11–17 consider combination two local planners. We also investigated combinations of three local planners using pie charts as shown in Figures 88–93.

Each chart shows the connections made by various combinations of the three local planners with respect to the total connections *they* made. In the graphs all combinations of the three planners were considered, and any set that was smaller than about 2% was collected into the *other* field. Figure 88 shows that each local planner (straight-line, rotate-at-0 and rotate-at-$\frac{1}{2}$) finds a considerable amount of connections that the others did not. Also, the rotate-at-0 and rotate-at-$\frac{1}{2}$ combination found a quarter of all connections that were found by the three local planners.

In Figure 89, we see that in the moderate alpha-puzzle environment each rotate-at-s local planner individually found connections that the others did not. The graph also shows that rotate-at-1 was less successful than other two, which found 34% of the same connections. We believe this number is high because of the free to surface connections (which favor rotate-at-0 and rotate-at-$\frac{1}{2}$ over rotate-at-1).

Figure 90 shows that the A*-distance planner is far more efficient than rotate-at-0 and rotate-at-1. In Figure 91 we see that in the hard 6-cube environment the rotate-at-$\frac{1}{2}$ planner is nearly as powerful as the A* local planners and that the two A* planner made almost the same connections.

Figure 92 shows that rotate-at-0, rotate-at-$\frac{1}{2}$ and A*-clearance find different connections and that the A* planner is more powerful than the other two.

COMPARISON OF STRAIGHT-LINE AND ROTATE-AT-$\frac{1}{2}$ LOCAL PLANNERS				
ENVIRONMENT	STRAIGHT-LINE	ROTATE-AT-$\frac{1}{2}$	BOTH	TOTAL
Six-Cubes Easy	626 (3.42%)	4193 (22.94%)	9124 (49.92%)	13943 (76.28%)
Six-Cubes Moderate	190 (1.09%)	7462 (42.79%)	2168 (12.43%)	9820 (56.31%)
Six-Cubes Hard	145 (1.11%)	5219 (40.12%)	1589 (12.21%)	6953 (53.45%)
Alpha-Puzzle Easy	2769 (16.18%)	3126 (18.26%)	4132 (24.14%)	10027 (58.59%)
Alpha-Puzzle Moderate	1682 (9.92%)	3008 (17.75%)	3574 (21.08%)	8264 (48.75%)
Alpha-Puzzle Hard	1054 (6.35%)	3492 (21.04%)	2159 (13.01%)	6705 (40.39%)
Rings Easy	3246 (35.23%)	570 (6.19%)	2998 (32.53%)	6814 (73.94%)
Rings Moderate	2425 (33.79%)	567 (7.90%)	2320 (32.33%)	5312 (74.02%)
Rings Hard	925 (20.68%)	301 (6.73%)	362 (8.09%)	1588 (35.51%)

Table 11: Comparison of the Straight-line and Rotate-at-$\frac{1}{2}$ planners.

COMPARISON OF ROTATE-AT-0 AND ROTATE-AT-$\frac{1}{2}$ LOCAL PLANNERS				
ENVIRONMENT	ROTATE-AT-0	ROTATE-AT-$\frac{1}{2}$	BOTH	TOTAL
Six-Cubes Easy	357 (1.95%)	3863 (21.13%)	9454 (51.72%)	13674 (74.81%)
Six-Cubes Moderate	341 (1.96%)	2165 (12.41%)	7465 (42.81%)	9971 (57.18%)
Six-Cubes Hard	187 (1.44%)	266 (2.04%)	6542 (50.29%)	6995 (53.77%)
Alpha-Puzzle Easy	3347 (19.56%)	2113 (12.35%)	5145 (30.06%)	10605 (61.96%)
Alpha-Puzzle Moderate	3435 (20.26%)	1735 (10.24%)	4847 (28.59%)	10017 (59.09%)
Alpha-Puzzle Hard	2694 (16.23%)	1457 (8.78%)	4194 (25.27%)	8345 (50.27%)
Rings Easy	1071 (11.62%)	1607 (17.44%)	1961 (21.28%)	4639 (50.34%)
Rings Moderate	880 (12.26%)	1353 (18.85%)	1534 (21.38%)	3767 (52.49%)
Rings Hard	1430 (31.98%)	298 (6.66%)	365 (8.16%)	2093 (46.80%)

Table 12: Comparison of the Rotate-at-0 and Rotate-at-$\frac{1}{2}$ planners.

Finally, Figure 93 suggests that combinations of straight-line, rotate-at-$\frac{1}{2}$ and A* planners may be useful, since the A* planner and the rotate-at-$\frac{1}{2}$ planner find different connections.

5 Conclusion

The main goal of our study was to determine good combinations of distance metrics and local planners for use by PRMs in cluttered environments.

Our results, as reported in Table 7, include recommendations for selecting distance metrics for various local planners in different types of environments. Generally, a good choice is the Scaled Euclidean metric, where more weight is placed on the position coordinates as the environment becomes more cluttered. Although it was not always the absolute best metric, its performance was comparable and it is quite efficient to compute.

The most powerful local planners were the A* planners. Among the less expensive planners, the rotate-at-$\frac{1}{2}$ planner was the best, outperforming the common straight-line local planner. However, we also found that each of the tested local planners made some connections that the others did not. Thus, roadmap connectivity will be enhanced if the PRM uses multiple local planners. Based on our experience, we would recommend the following order. First, the rotate-at-$\frac{1}{2}$ and straight-line planners. Next, the rotate-at-0 and rotate-at-1 planners (recall they perform comparably when both configurations are near C-obstacle surfaces). And finally, the A^* planners.

Acknowledgment

We would like to thank the robotics group at Texas A&M, especially Jeff Trinkle, Li Han and Steve Wilmarth for their suggestions regarding this work. We are also grateful to John Canny, Shane Chang, Lydia Kavraki,

COMPARISON OF ROTATE-AT-0 AND ROTATE-AT-1 LOCAL PLANNERS				
ENVIRONMENT	ROTATE-AT-0	ROTATE-AT-1	BOTH	TOTAL
Six-Cubes Easy	8597 (47.03%)	912 (4.99%)	1214 (6.64%)	10723 (58.66%)
Six-Cubes Moderate	6646 (38.11%)	858 (4.92%)	1160 (6.65%)	8664 (49.68%)
Six-Cubes Hard	5849 (44.96%)	421 (3.24%)	880 (6.76%)	7150 (54.96%)
Alpha-Puzzle Easy	7486 (43.74%)	2655 (15.51%)	1006 (5.88%)	11147 (65.13%)
Alpha-Puzzle Moderate	7295 (43.04%)	1868 (11.02%)	987 (5.82%)	10150 (59.88%)
Alpha-Puzzle Hard	6032 (36.34%)	1924 (11.59%)	856 (5.16%)	8812 (53.09%)
Rings Easy	2358 (25.59%)	1406 (15.26%)	674 (7.31%)	4438 (48.16%)
Rings Moderate	1909 (26.60%)	989 (13.78%)	505 (7.04%)	3403 (47.42%)
Rings Hard	1597 (35.71%)	658 (14.71%)	198 (4.43%)	2453 (54.85%)

Table 13: Comparison of the Rotate-at-0 and Rotate-at-1 planners.

COMPARISON OF ROTATE-AT-0 AND STRAIGHT-LINE LOCAL PLANNERS				
ENVIRONMENT	STRAIGHT-LINE	ROTATE-AT-0	BOTH	TOTAL
Six-Cubes Easy	1617 (8.85%)	1678 (9.18%)	8133 (44.49%)	11428 (62.52%)
Six-Cubes Moderate	408 (2.34%)	5856 (33.58%)	1950 (11.18%)	8214 (47.10%)
Six-Cubes Hard	140 (1.08%)	5135 (39.47%)	1594 (12.25%)	6869 (52.80%)
Alpha-Puzzle Easy	1811 (10.58%)	3402 (19.88%)	5090 (29.74%)	10303 (60.20%)
Alpha-Puzzle Moderate	1128 (6.65%)	4154 (24.51%)	4128 (24.35%)	9410 (55.51%)
Alpha-Puzzle Hard	758 (4.57%)	4433 (26.71%)	2455 (14.79%)	7646 (46.06%)
Rings Easy	4004 (43.45%)	792 (8.59%)	2240 (24.31%)	7036 (76.35%)
Rings Moderate	3092 (43.09%)	761 (10.60%)	1653 (23.04%)	5506 (76.73%)
Rings Hard	826 (18.47%)	1334 (29.83%)	461 (10.31%)	2621 (58.61%)

Table 14: Comparison of the Rotate-at-0 and Straight-line planners.

Jean-Claude Latombe, Donald H. House and Donald K. Friesen for useful discussions and comments. The alpha puzzle was designed by Boris Yamrom of the Computer Graphics & Systems Group at GE's Corporate Research & Development Center. GE also provided us with *Product Vision*, their CAD animation package, which was used to produce the environment snapshots shown in this paper.

References

[1] J. M. Ahuactzin and K. Gupta. A motion planning based approach for inverse kinematics of redundant robots: The kinematic roadmap. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 3609–3614, 1997.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. Choosing good distance metrics and local planners for probabilistic roadmap methods. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, 1998.

[3] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proc. Intern. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.

[4] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 113–120, Minneapolis, MN, April 1996.

[5] J. Barraquand, L.E. Kavraki, J.C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan. A random sampling scheme for path planning. *Int. J. of Rob. Res*, 16(6):759–774, 1997.

[6] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Trans. Sys., Man, Cybern.*, 22(2):224–241, 1992.

[7] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Internat. J. Robot. Res.*, 10(6):628–649, 1991.

[8] P. Bessiere, J. M.Ahuactzin, E.-G. Talbi, and E. Mazer. The Ariadne's Clew Algorithm: Global planning with local methods. In *Proc. IEEE Internat. Conf. Intel. Rob. Syst. (IROS)*, volume 2, pages 1373–1380, 1993.

[9] R.L. Bishop and S.I. Goldberg. *Tensor Analysis and manifolds*. Dover Publications, New York, 1980.

COMPARISON OF ROTATE-AT-$\frac{1}{2}$ AND ROTATE-AT-1 LOCAL PLANNERS				
ENVIRONMENT	ROTATE-AT-$\frac{1}{2}$	ROTATE-AT-1	BOTH	TOTAL
Six-Cubes Easy	11693 (63.97%)	502 (2.75%)	1624 (8.88%)	13819 (75.60%)
Six-Cubes Moderate	8171 (46.85%)	559 (3.21%)	1459 (8.37%)	10189 (58.43%)
Six-Cubes Hard	5887 (45.25%)	380 (2.92%)	921 (7.08%)	7188 (55.25%)
Alpha-Puzzle Easy	6017 (35.16%)	2420 (14.14%)	1241 (7.25%)	9678 (56.55%)
Alpha-Puzzle Moderate	5458 (32.20%)	1731 (10.21%)	1124 (6.63%)	8313 (49.04%)
Alpha-Puzzle Hard	4636 (27.93%)	1765 (10.63%)	1015 (6.11%)	7416 (44.68%)
Rings Easy	2586 (28.06%)	1098 (11.92%)	982 (10.66%)	4666 (50.63%)
Rings Moderate	2145 (29.89%)	752 (10.48%)	742 (10.34%)	3639 (50.71%)
Rings Hard	395 (8.83%)	588 (13.15%)	268 (5.99%)	1251 (27.97%)

Table 15: Comparison of the Rotate-at-$\frac{1}{2}$ and Rotate-at-1 planners.

COMPARISON OF ROTATE-AT-$\frac{1}{2}$ AND A*-CLEARANCE LOCAL PLANNERS				
ENVIRONMENT	ROTATE-AT-$\frac{1}{2}$	A-STAR-CLEARANCE	BOTH	TOTAL
Six-Cubes Easy	3148 (17.22%)	4300 (23.52%)	10169 (55.63%)	17617 (96.38%)
Six-Cubes Moderate	7028 (40.30%)	7060 (40.48%)	2602 (14.92%)	16690 (95.71%)
Six-Cubes Hard	4986 (38.33%)	5694 (43.77%)	1822 (14.01%)	12502 (96.10%)
Alpha-Puzzle Easy	328 (1.92%)	8445 (49.34%)	6930 (40.49%)	15703 (91.75%)
Alpha-Puzzle Moderate	567 (3.34%)	7353 (43.38%)	6015 (35.48%)	13935 (82.21%)
Alpha-Puzzle Hard	1944 (11.71%)	8090 (48.74%)	3707 (22.33%)	13741 (82.78%)
Rings Easy	209 (2.27%)	5202 (56.45%)	3359 (36.45%)	8770 (95.17%)
Rings Moderate	215 (3.00%)	3834 (53.43%)	2672 (37.24%)	6721 (93.66%)
Rings Hard	255 (5.70%)	2666 (59.62%)	408 (9.12%)	3329 (74.44%)

Table 16: Comparison of the Rotate-at-$\frac{1}{2}$ and A*-clearance planners.

[10] D. J. Challou, M. Gini, and V. Kumar. Parallel search algorithms for robot motion planning. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 2, pages 46–51, 1993.

[11] H. Chang and T. Y. Li. Assembly maintainability study with motion planning. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 1012–1019, 1995.

[12] P. C. Chen and Y. K. Hwang. SANDROS: A motion planner with performance proportional to task difficulty. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 2346–2353, 1992.

[13] D.P. Dobkin, J. Hershberger, D.G. Kirkpatrick, and S. Suri. Computing the intersection depth of polyhedra. *Algorithmica*, 9:518–533, 1993.

[14] B. R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artif. Intell.*, 31(3):295–353, 1987.

[15] B. Faverjon and P. Tournassoud. A practical approach to motion-planning for manipulators with many degrees of freedom. In H. Miura and S. Arimoto, editors, *Robotics Research 5*. The MIT Press, 1990.

[16] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex robots in three-dimensional space. *IEEE Trans. Rob. and Aut.*, (4):193–203, 1988.

[17] B. Glavina. Solving findpath by combination of directed and randomized search. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 1718–1723, 1990.

[18] B. Grünbaum. *Convex Poyltopes*. Wiley-Intersciences, 1967.

[19] K. K. Gupta and Z. Guo. Motion planning for many degrees of freedom: Sequential search with backtracking. *IEEE Trans. Robot. Automat.*, 11(6):897–906, 1995.

[20] T. Horsch, F. Schwarz, and H. Tolle. Motion planning for many degrees of freedom – random reflections at c-space obstacles. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 3318–3323, 1994.

[21] D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. Intern. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.

[22] D. Hsu, J-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 2719–2726, 1997.

COMPARISON OF ROTATE-AT-0 AND A*-CLEARANCE LOCAL PLANNERS				
ENVIRONMENT	ROTATE-AT-0	A-STAR-CLEARANCE	BOTH	TOTAL
Six-Cubes Easy	993 (5.43%)	5651 (30.92%)	8818 (48.24%)	15462 (84.59%)
Six-Cubes Moderate	5536 (31.74%)	7392 (42.39%)	2270 (13.02%)	15198 (87.15%)
Six-Cubes Hard	4894 (37.62%)	5681 (43.67%)	1835 (14.11%)	12410 (95.40%)
Alpha-Puzzle Easy	897 (5.24%)	7780 (45.46%)	7595 (44.38%)	16272 (95.07%)
Alpha-Puzzle Moderate	2403 (14.18%)	7489 (44.18%)	5879 (34.68%)	15771 (93.04%)
Alpha-Puzzle Hard	3121 (18.80%)	8030 (48.38%)	3767 (22.69%)	14918 (89.87%)
Rings Easy	372 (4.04%)	5901 (64.04%)	2660 (28.87%)	8933 (96.94%)
Rings Moderate	423 (5.89%)	4515 (62.92%)	1991 (27.75%)	6929 (96.56%)
Rings Hard	1012 (22.63%)	2291 (51.23%)	783 (17.51%)	4086 (91.37%)

Table 17: Comparison of the Rotate-at-0 and A*-clearance planners.

[23] Y. Hwang and N. Ahuja. Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):219–291, 1992.

[24] Y. K. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Trans. Robot. Automat.*, 8(1):23–32, 1992.

[25] Y. K. Hwang and P. C. Chen. A heuristic and complete planner for the classical mover's problem. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 729–736, 1995.

[26] Yong K. Hwang and Pang C. Chen. A heuristic and compete planner for the classical mover's problem. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 1, pages 729–736, 1995.

[27] P. Jimenez, F. Thomas, and C. Torras. Collision Detection algorithms for Motion Planning. *Robot Motion Planning and Control*. (ed.) J.P. Laumond. Lecture Notes in Control and Info. Sc., 229, Springer, New York, NY, 1998.

[28] L. Kavraki. *Random Networks in Configuration Space for Fast Path Planning*. PhD thesis, Stanfort Univ, Comp.Sc.Dept., 1995.

[29] L. Kavraki, M. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic roadmaps for path planning. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 4, pages 3020–3025, 1996.

[30] L. Kavraki and J. C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 2138–2145, 1994.

[31] L. Kavraki, J. C. Latombe, R. Motwani, and P. Raghavan. Randomized query preprocessing in robot path planning. In *Proc. ACM Symp. Theory of Computing (STOC)*, pages 353–362, 1995.

[32] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[33] K. Kondo. Motion planning with six degrees of freedom by multistrategic bidirectional heuristic free space enumeration. *IEEE Trans. Robot. Automat.*, 7(3):267–277, 1992.

[34] E. Kruse, R. Gutsche, and F. Wahl. Efficient, iterative, sensor based 3-d map building using rating functions in configuration space. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 2, pages 1067–1072, 1996.

[35] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[36] M. Lin and J. Canny. A fast algorithm for incremental distance computation. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 602–608, 1994.

[37] M. Lin, D. Manocha, J. Cohen, and S. Gottschalk. *Collision Detection: Algorithms and Applications*. Algorithmic foundations of Robotics, Goldberg et al. (Eds.), A K Peters, Ltd., 1995.

[38] A. McLean and I. Mazon. Incremental roadmaps and global path planning in evolving industrial environments. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 1, pages 101–107, 1996.

[39] C. Mirolo and E. Pagello. A practical motion planning strategy based on plane-sweep approach. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 2705–2712, 1997.

[40] B. Mirtich. V-clip: A fast and robust polyhedral collision detection. Technical Report TR97-05, Mitsubishi El. Res. Lab., Cambridge, MA, 1997.

[41] C. J. Ong and E. G. Gilbert. Growth distances: New measures for object separation and penetration. *IEEE Trans. Robot. Automat.*, 12(6):888–903, 1996.

[42] M. Overmars. A random approach to path planning. Technical Report RUU-CS-92-32, Computer Science, Utrecht University, The Netherlands, 1992.

[43] M. Overmars and P. Svestka. A probabilistic learning approach to motion planning. In *Proc. Workshop on Algorithmic Foundations of Robotics*, pages 19–37, 1994.

[44] F.C. Park. Distance metrics on the rigid-body motions with applications to mechanism design. *ASME Journal of Mechanical Design*, 117(1):48–54, 1995.

[45] F.C. Park and R.W. Brockett. Kinematic dexterity of robotic mechanisms. *Internat. J. Robot. Res.*, 13(1):1–15, 1994.

[46] S. Quinlan. Efficient distance computation between non-convex objects. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 3324–3330, 1994.

[47] J. Reif. Complexity of the piano mover's problem and generalizations. In *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, pages 421–427, 1979.

[48] M. Spivak. *Comprehensive Introduction to Differential Geometry*. Publish or Perish, Wilmington, DE, 1979.

[49] M. Tarokh. Implementation of a fast path planner on an industrial manipulator. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 1, pages 436–441, 1996.

[50] P. Watterberg, P. Xavier, and Y. Hwang. Path planning for everyday robotics with sandros. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 1170–1175, 1997.

[51] P. Xavier. Fast swept-volume distance for robust collision detection. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 1162–1169, 1997.

[52] J. Xiao and L. Zhang. Computing rotation distance between contacting polyhedra. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, volume 1, pages 791–797, 1996.

[53] M. Zefran, V. Kumar, and C. Croke. Metrics and connections for rigit body kinematics. Submitted to Internat. J. Robot. Res., May 1996.

[54] M. Zefran, V. Kumar, and C. Croke. Choice of riemannian metrics for rigid body kinematics. In *Proc. ASME Design Eng. Tech. Conf. and Comp. in Eng. Conf.*, 1996.

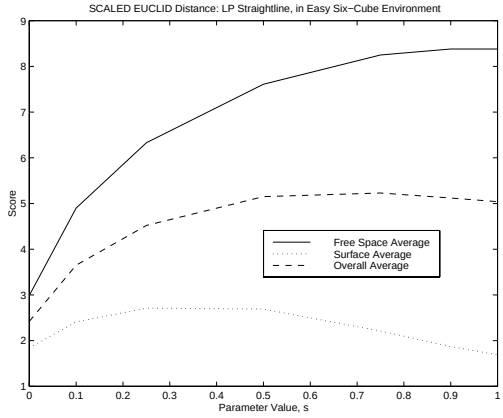


Figure 10: s parameter graph for Scaled Euclidean metric, straight-line LP, easy 6-cube.

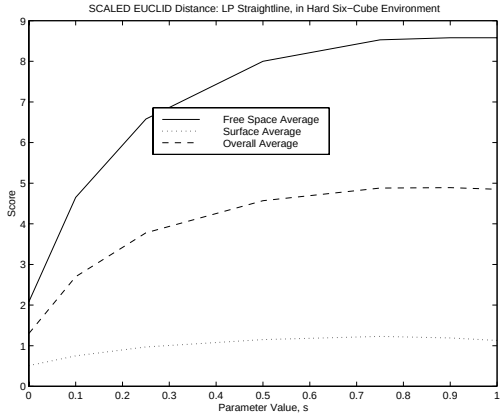


Figure 11: s parameter graph for Scaled Euclidean metric, straight-line LP, hard 6-cube.

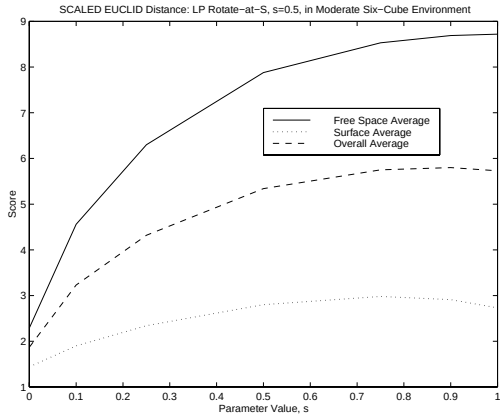


Figure 12: s parameter graph for Scaled Euclidean metric, rotate-at-$\frac{1}{2}$ LP, moderate 6-cube.

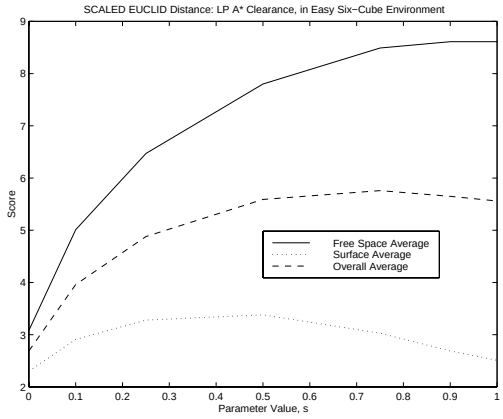


Figure 13: s parameter graph for Scaled Euclidean metric, a-star-clearance LP, easy 6-cube.

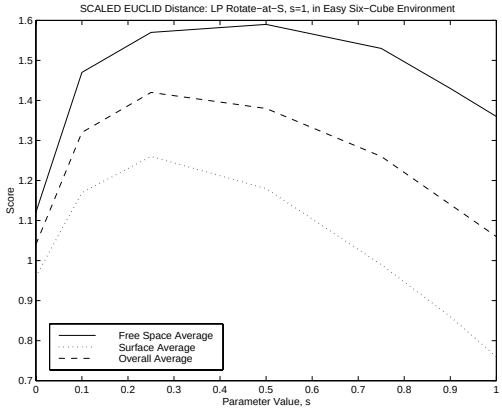


Figure 14: s parameter graph for Scaled Euclidean metric, rotate-at-1 LP, easy 6-cube.

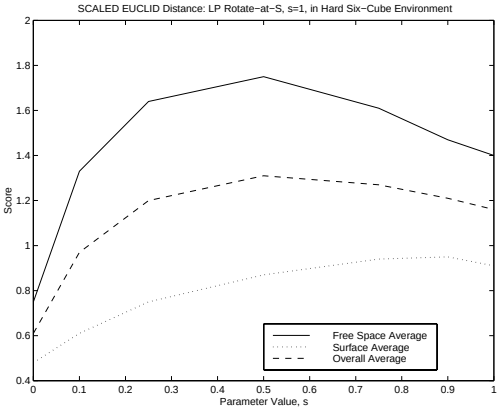


Figure 15: s parameter graph for Scaled Euclidean metric, rotate-at-1 LP, hard 6-cube.

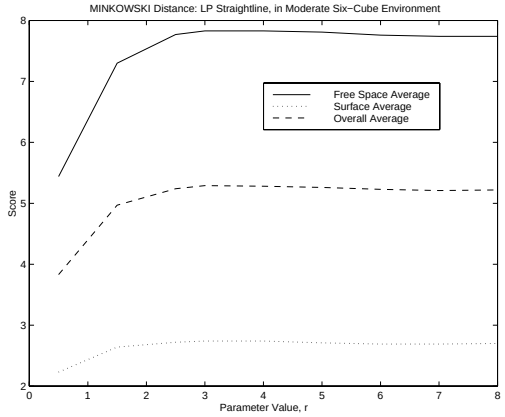


Figure 16: r parameter graph for Minkowski metric, straight-line LP, moderate 6-cube.

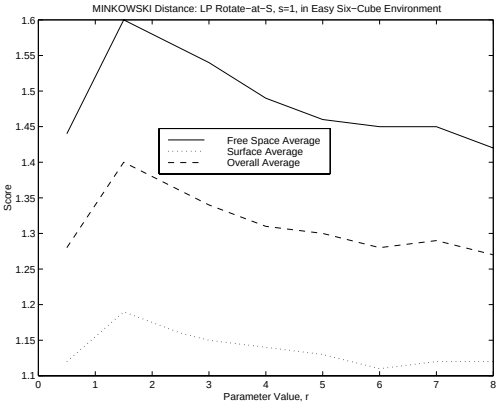


Figure 17: r parameter graph for Minkowski metric, rotate-at-1 LP, easy 6-cube.

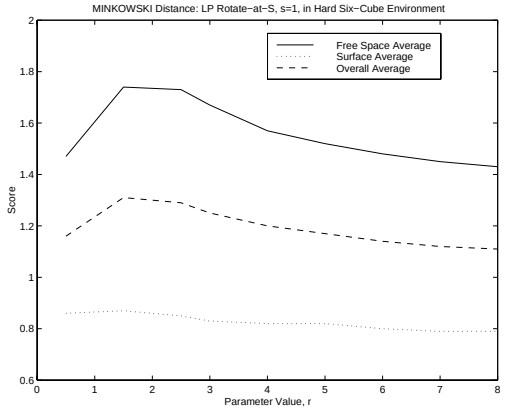


Figure 18: r parameter graph for Minkowski metric, rotate-at-1 LP, hard 6-cube.

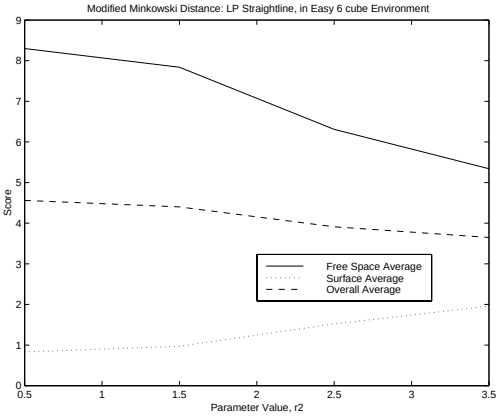


Figure 19: r_2 parameter graph for modified Minkowski metric, straight-line LP, easy 6-cube.

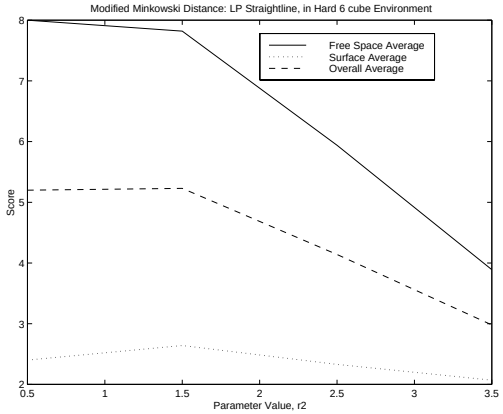


Figure 20: r_2 parameter graph for modified Minkowski metric, straight-line LP, hard 6-cube.

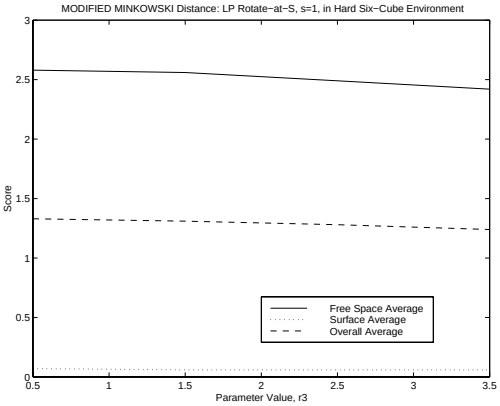


Figure 21: r_3 parameter graph for Modified Minkowski metric, rotate-at-0 LP, hard 6-cube

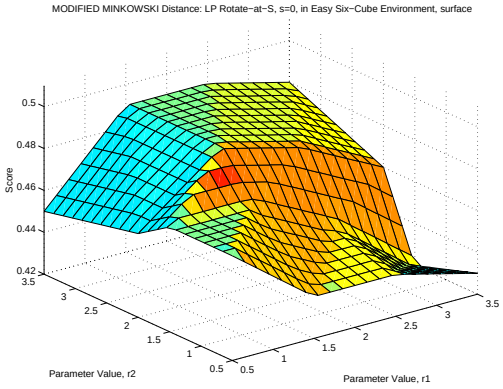


Figure 22: r_1-r_2 parameters graph for Modified Minkowski metric, rotate-at-0 LP, easy 6-cube

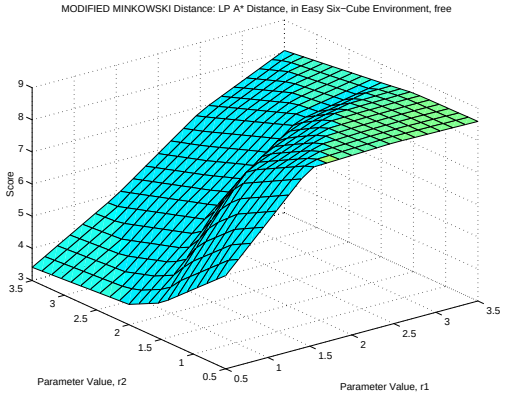


Figure 23: r_1-r_2 parameters graph for Modified Minkowski metric, a-star-distance LP, easy 6-cube

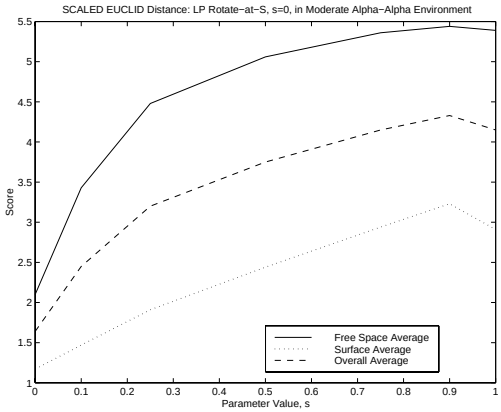


Figure 24: s parameter graph for Scaled Euclidean metric, rotate-at-0 LP, moderate alpha-puzzle.

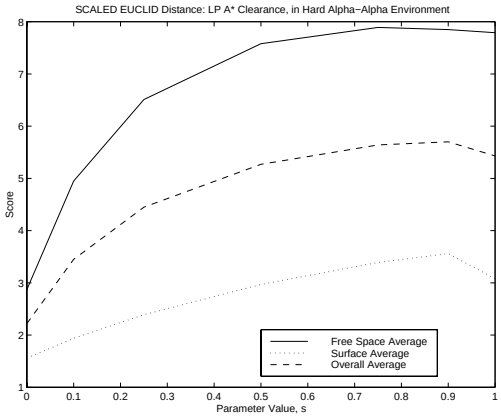


Figure 25: s parameter graph for Scaled Euclidean metric, a-star-clearance LP, hard alpha-puzzle.

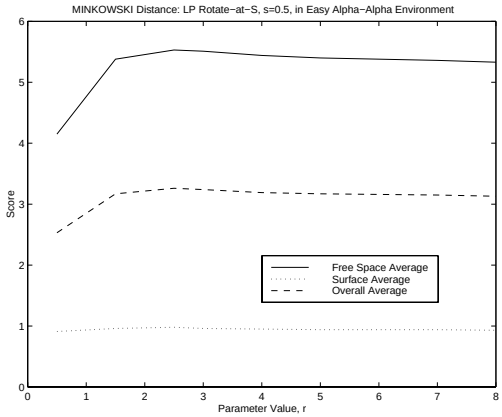


Figure 26: r parameter graph for Minkowski metric, rotate-at-s=5 LP, easy alpha-puzzle.

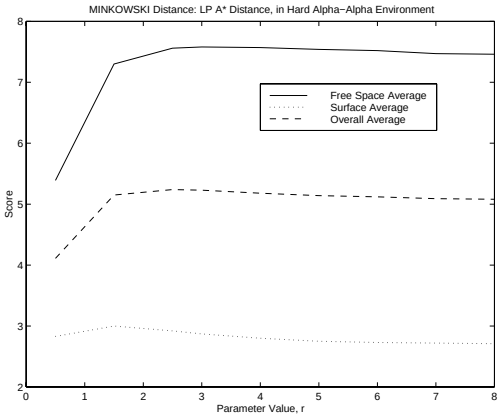


Figure 27: r parameter graph for Minkowski metric, a-star-distance LP, hard alpha-puzzle.

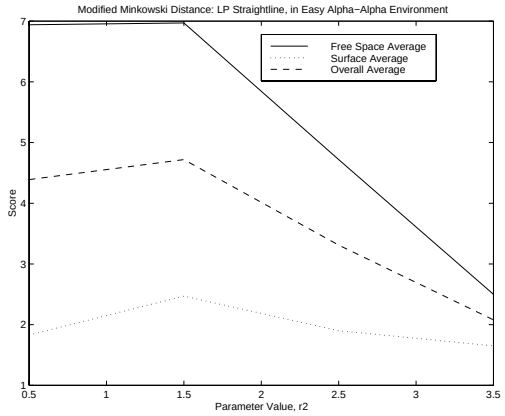


Figure 28: r_2 parameter graph for Modified Minkowski metric, straight-line LP, easy alpha-puzzle.

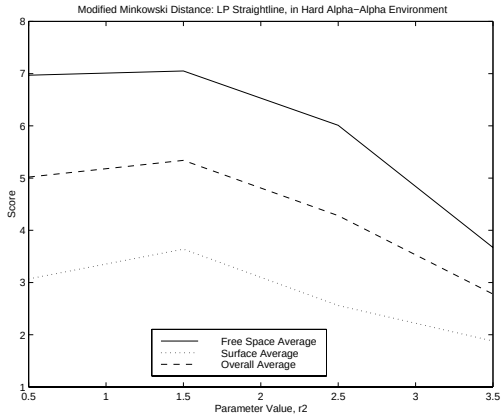


Figure 29: r_2 parameter graph for Modified Minkowski metric, straight-line LP, hard alpha-puzzle.

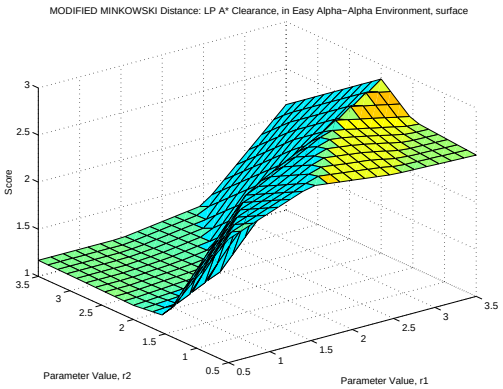


Figure 30: r_1-r_2 parameters graph for Modified Minkowski, a-star-clearance LP, easy alpha-puzzle

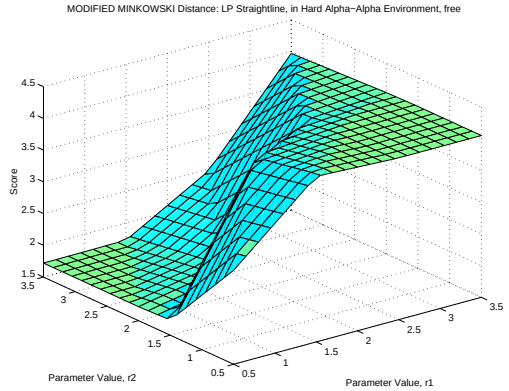


Figure 31: r_1-r_2 parameters graph for Modified Minkowski, straight-line LP, hard alpha-puzzle

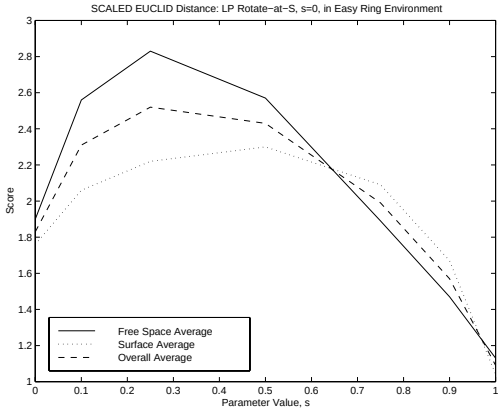


Figure 32: s parameter graph for Scaled Euclidean metric, rotate-at-0 LP, easy rings.

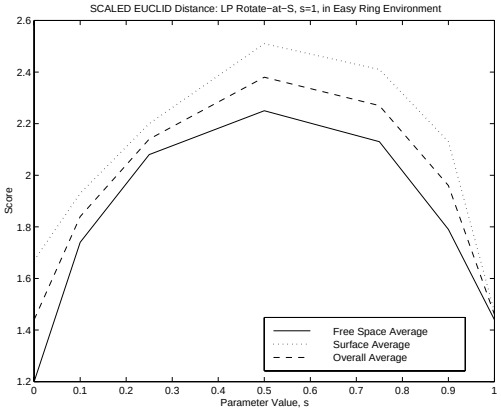


Figure 33: s parameter graph for Scaled Euclidean metric, rotate-at-1 LP, easy rings.

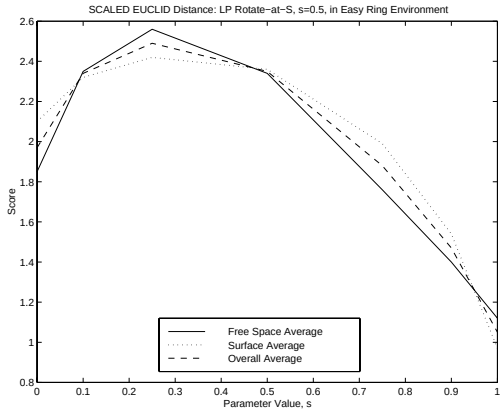


Figure 34: s parameter graph for Scaled Euclidean metric, rotate-at-$\frac{1}{2}$ LP, easy rings.

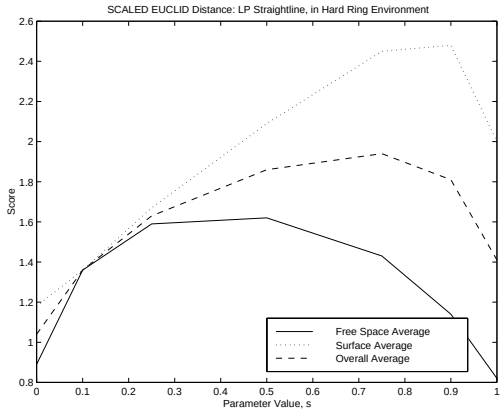


Figure 35: s parameter graph for Scaled Euclidean metric, straight-line LP, easy rings.

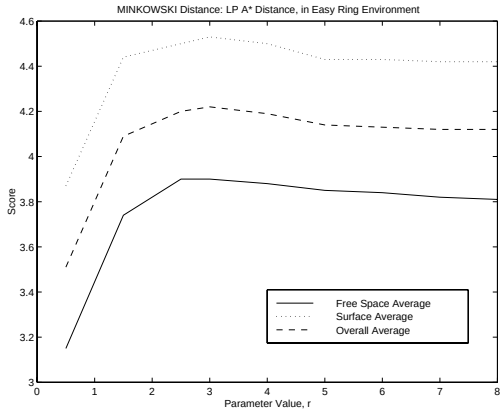


Figure 36: r parameter graph for Minkowski metric, a-star-distance LP, easy rings.

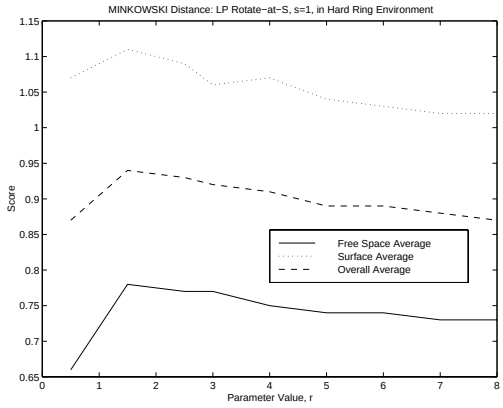


Figure 37: r parameter graph for Minkowski metric, rotate-at-1 LP, hard rings.

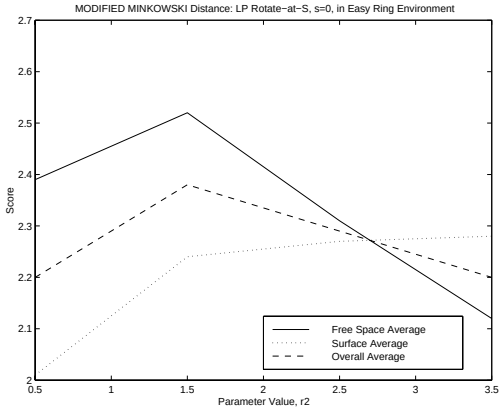


Figure 38: r_2 parameter graph for Modified Minkowski metric, rotate-at-0 LP, easy rings.

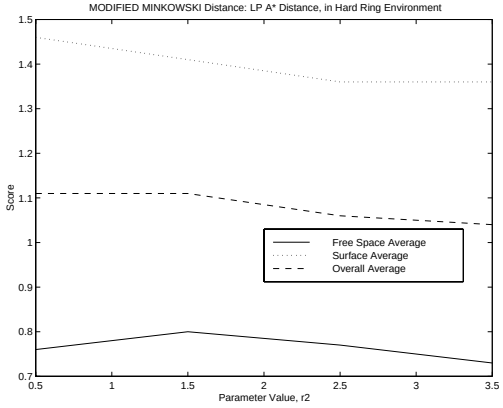


Figure 39: r_2 parameter graph for Modified Minkowski metric, a-star-distance LP, hard rings.

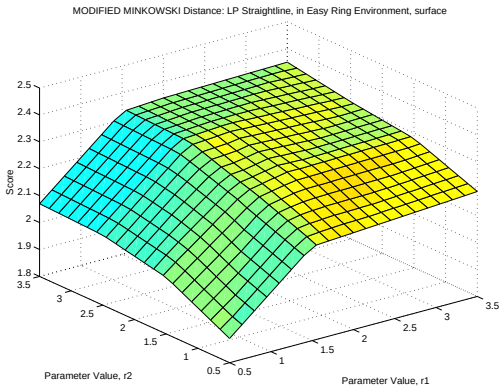


Figure 40: r_1-r_2 parameters graph for Modified Minkowski metric, straight-line LP, easy alpha-puzzle

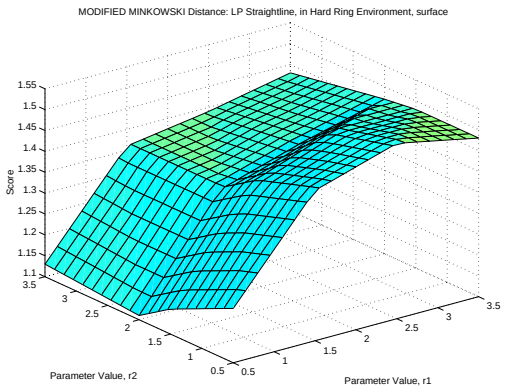


Figure 41: r_1-r_2 parameters graph for Modified Minkowski metric, straight-line LP, hard alpha-puzzle

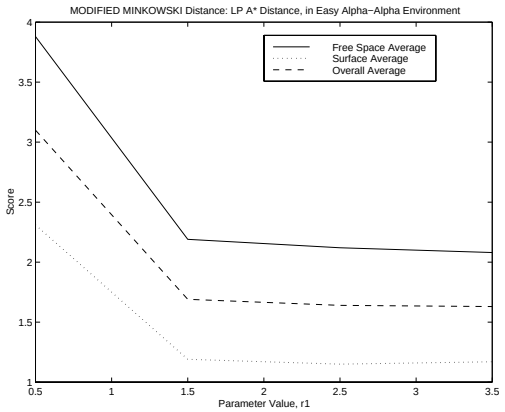


Figure 42: r_1 parameter graph for Modified Minkowski metric, a-star-distance LP, easy alpha-puzzle.

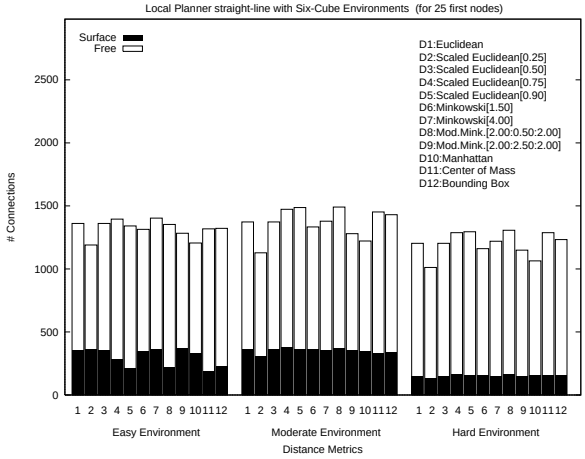


Figure 43: Straight-line LP, 6-cube (for 25 closest nodes).

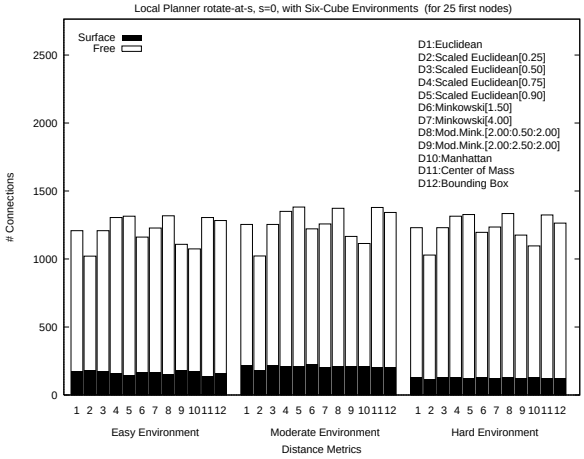


Figure 44: Rotate-at-0 LP, 6-cube (for 25 closest nodes).

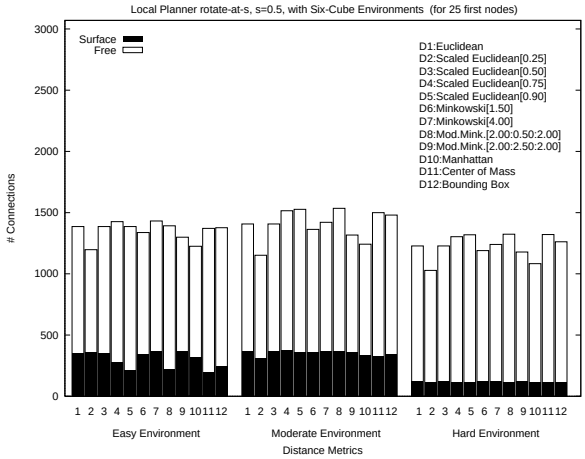


Figure 45: Rotate-at-$\frac{1}{2}$ LP, 6 cube (for 25 closest nodes).

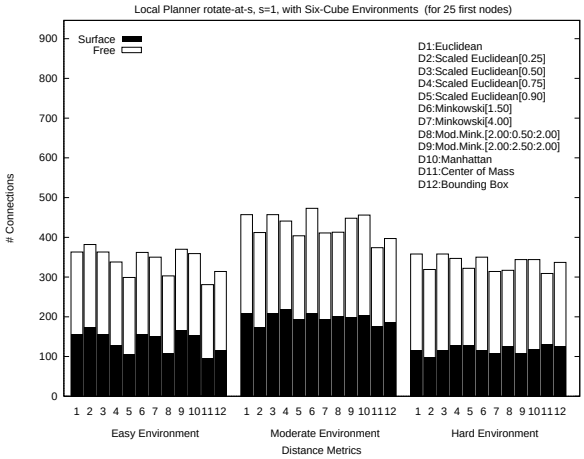


Figure 46: Rotate-at-1 LP, 6-cube (for 25 closest nodes).

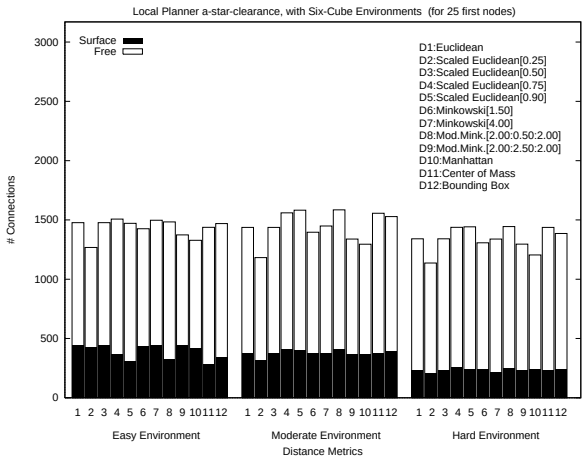


Figure 47: A*-clearance LP, 6-cube (for 25 closest nodes).

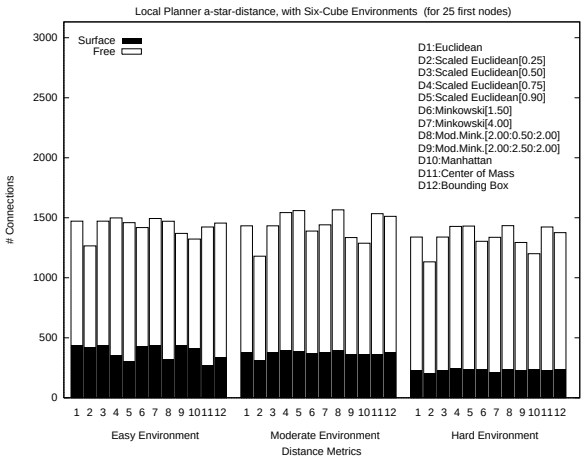


Figure 48: A*-distance LP, 6-cube (for 25 closest nodes).

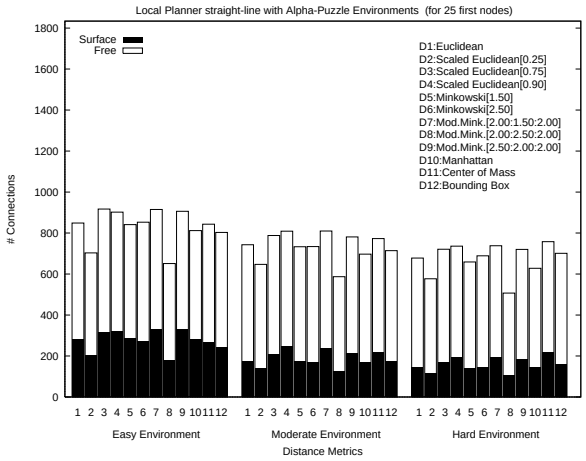


Figure 49: Straight-line LP, alpha-puzzle (for 25 closest nodes).

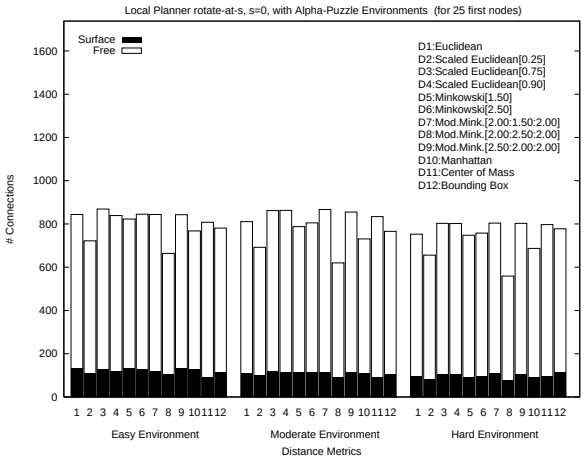


Figure 50: Rotate-at-0 LP, alpha-puzzle (for 25 closest nodes).

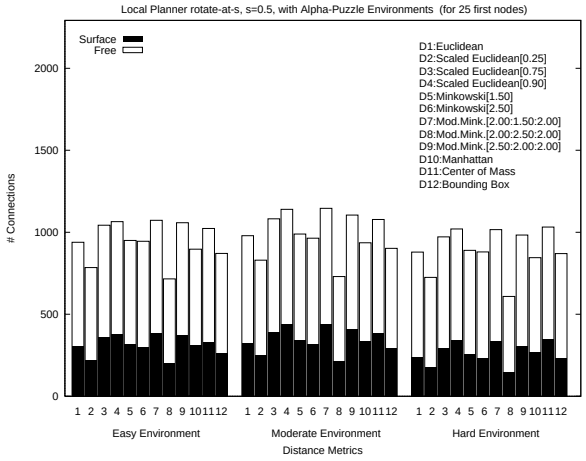


Figure 51: Rotate-at-$\frac{1}{2}$ LP, alpha-puzzle (for 25 closest nodes).

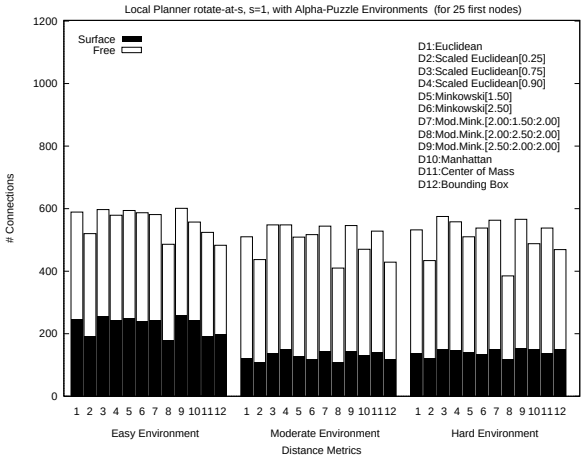


Figure 52: Rotate-at-1 LP, alpha-puzzle (for 25 closest nodes).

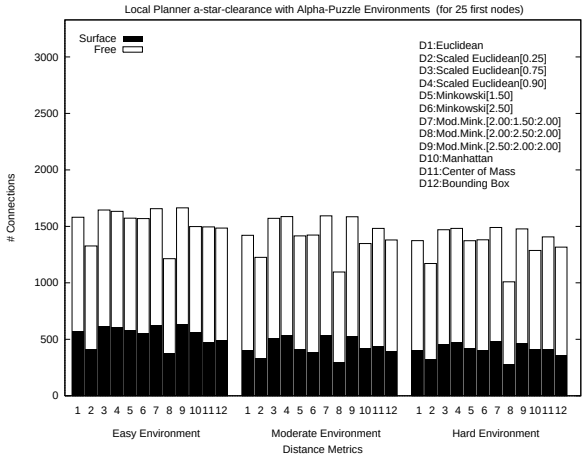
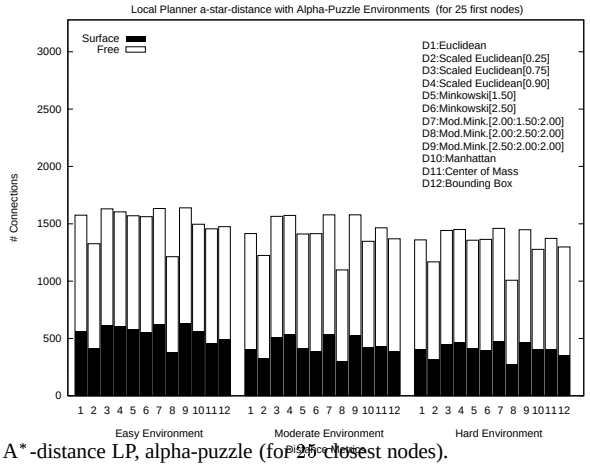


Figure 53: A*-clearance LP, alpha-puzzle (for 25 closest nodes).



A*-distance LP, alpha-puzzle (for 25 closest nodes).

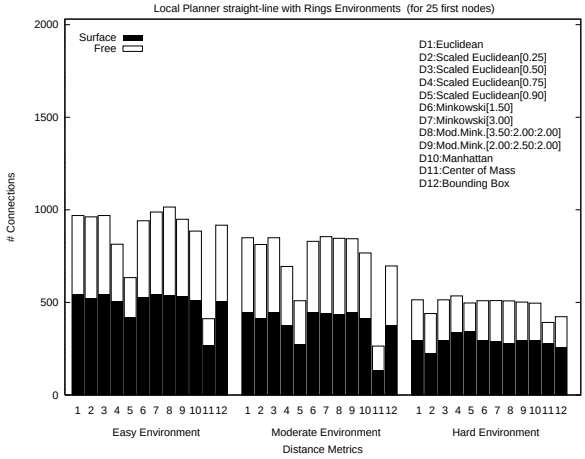


Figure 54: Straight-line LP, rings (for 25 closest nodes).

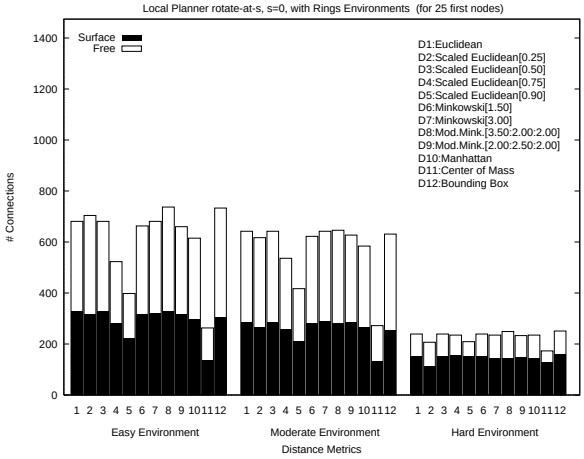


Figure 55: Rotate-at-0 LP, rings (for 25 closest nodes).

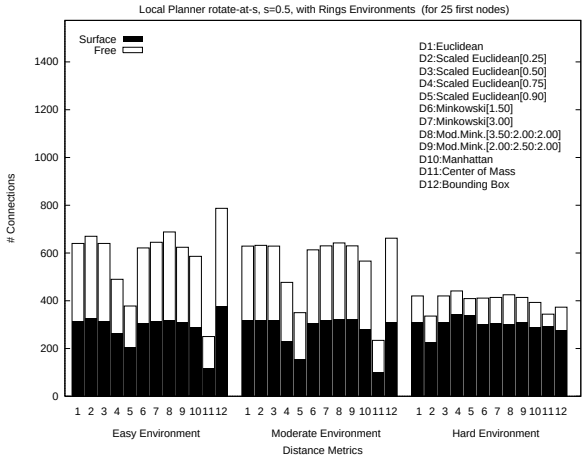


Figure 56: Rotate-at-$\frac{1}{2}$ LP, rings (for 25 closest nodes).

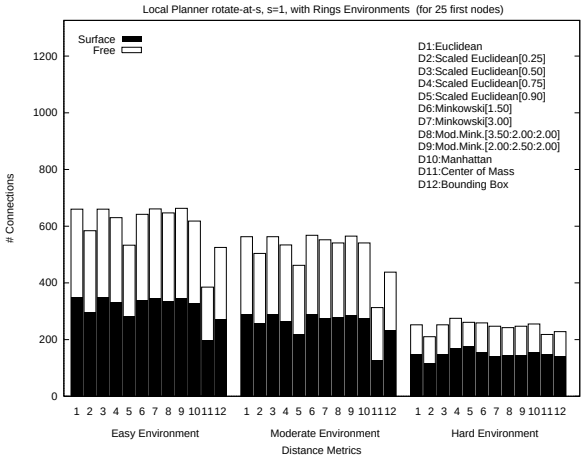


Figure 57: Rotate-at-1 LP, rings (for 25 closest nodes).

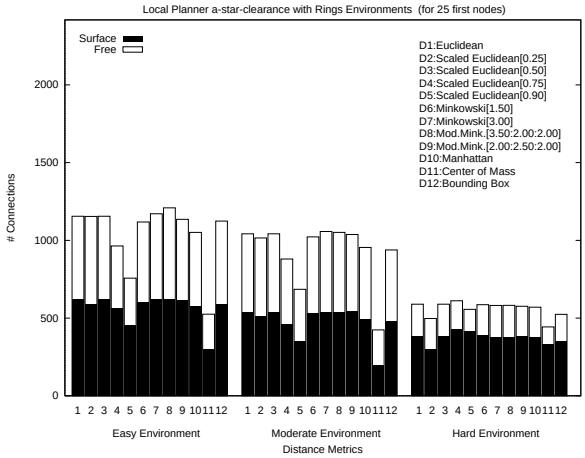


Figure 58: A*-clearance LP, rings (for 25 closest nodes).

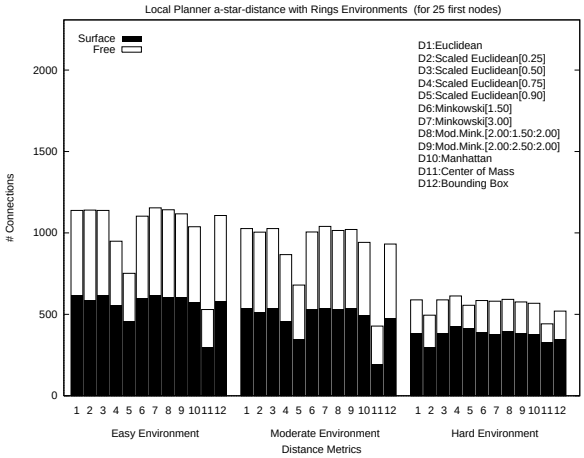


Figure 59: A*-distance LP, rings (for 25 closest nodes).

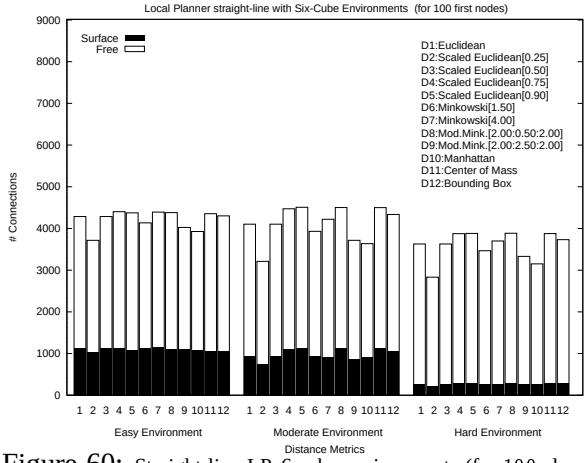


Figure 60: Straight-line LP, 6 cube environments (for 100 closest nodes).

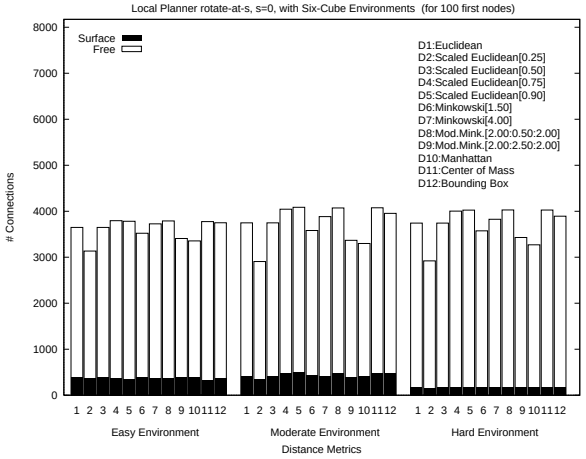


Figure 61: Rotate-at-0 LP, 6 cube (for 100 closest nodes).

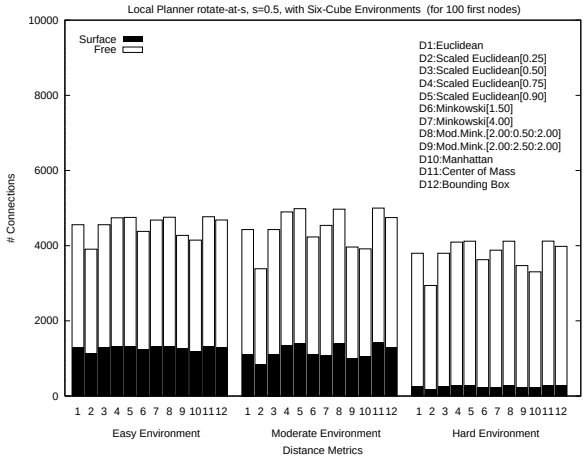


Figure 62: Rotate-at-$\frac{1}{2}$ LP, 6 cube (for 100 closest nodes).

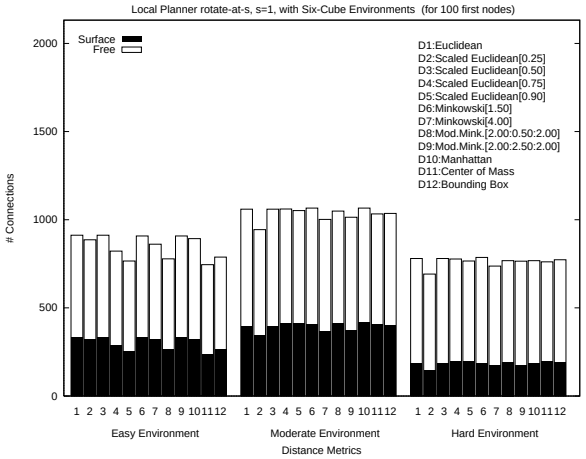


Figure 63: Rotate-at-1 LP, 6 cube (for 100 closest nodes).

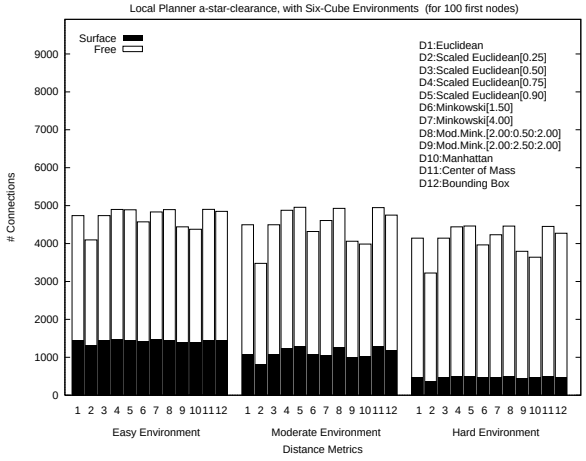


Figure 64: A*-clearance LP, 6 cube (for 100 closest nodes).

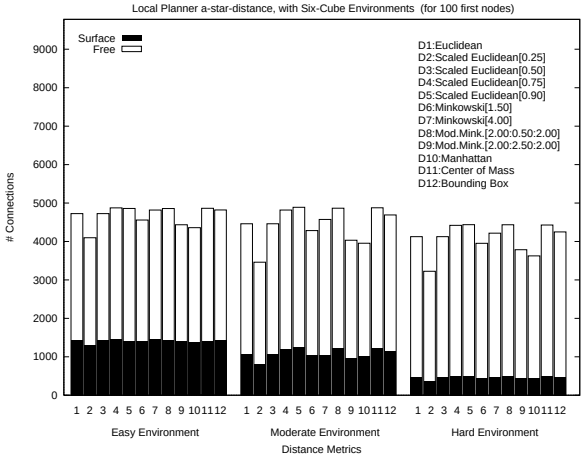


Figure 65: A*-distance LP, 6 cube (for 100 closest nodes).

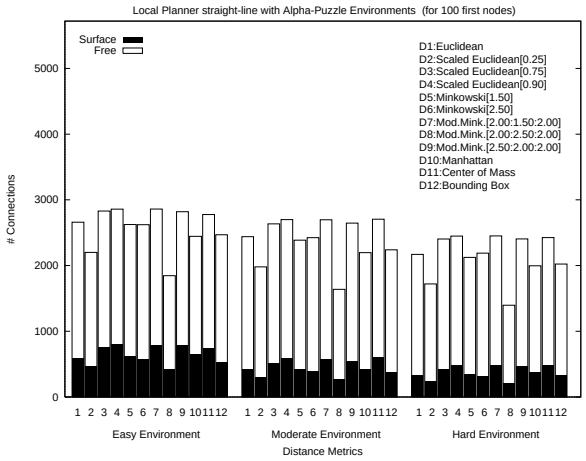


Figure 66: Straight-line LP, alpha-puzzle (for 100 closest nodes).

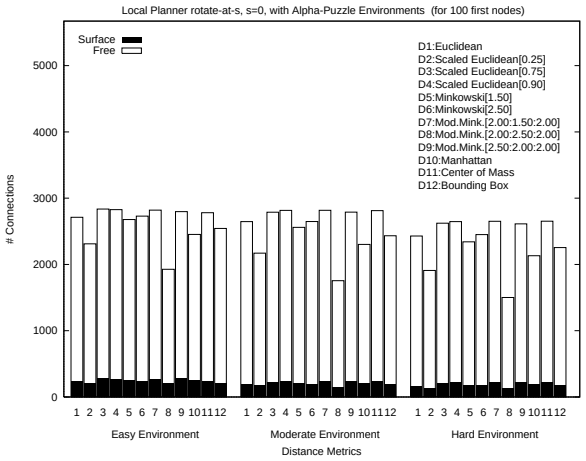


Figure 67: Rotate-at-0 LP, alpha-puzzle (for 100 closest nodes).

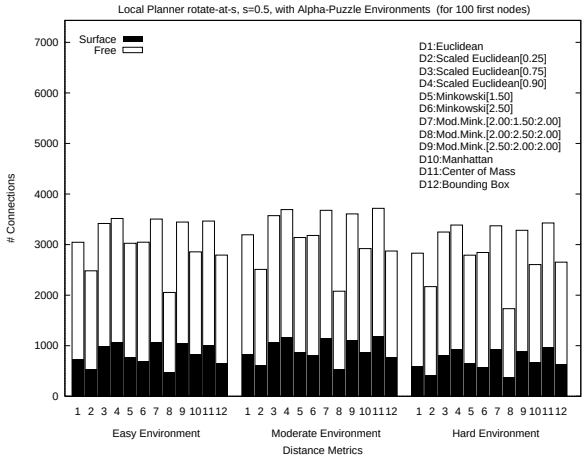


Figure 68: Rotate-at-$\frac{1}{2}$ LP, alpha-puzzle (for 100 closest nodes).

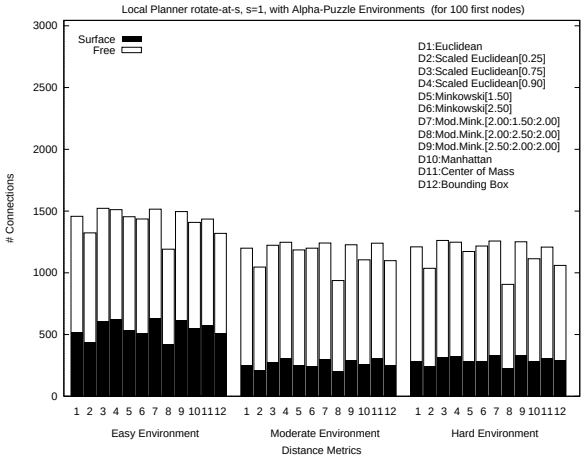


Figure 69: Rotate-at-1 LP, alpha-puzzle (for 100 closest nodes).

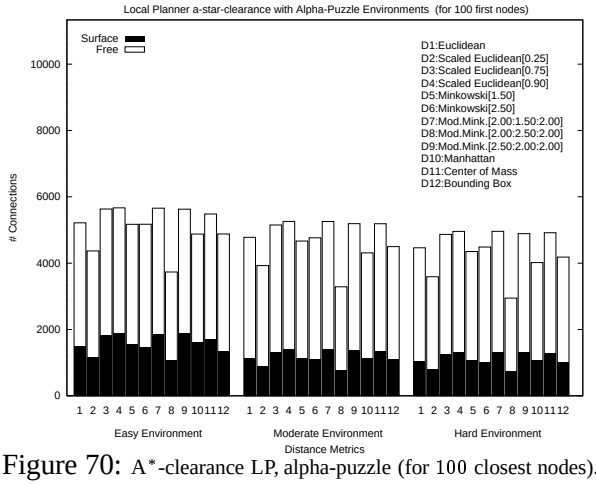


Figure 70: A*-clearance LP, alpha-puzzle (for 100 closest nodes).

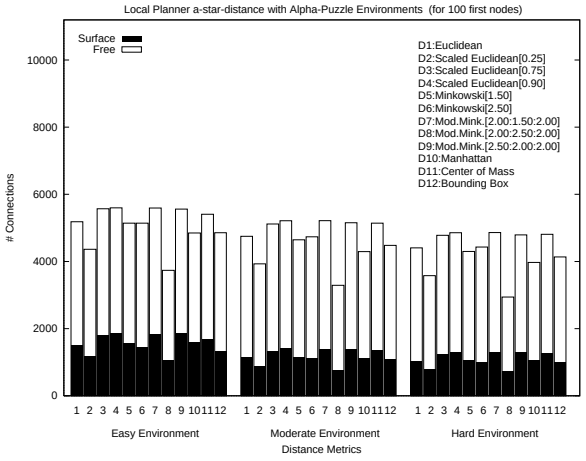


Figure 71: A*-distance LP, alpha-puzzle (for 100 closest nodes).

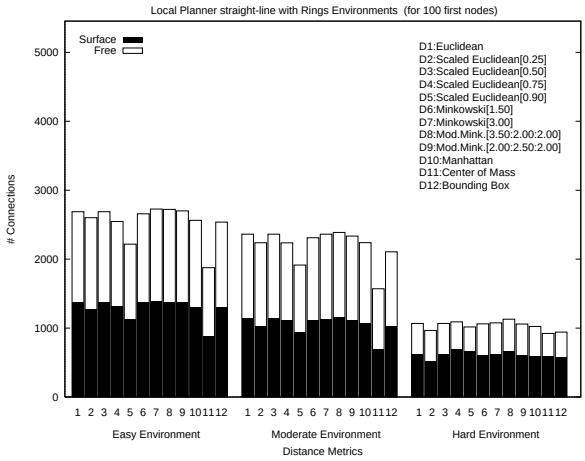


Figure 72: Straight-line LP, rings (for 100 closest nodes).

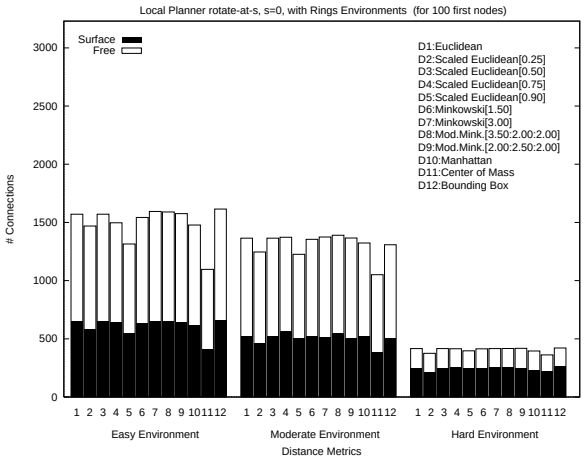


Figure 73: Rotate-at-0 LP, rings (for 100 closest nodes).

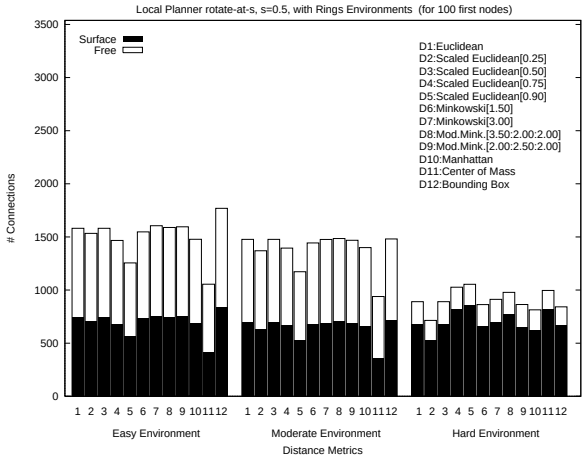


Figure 74: Rotate-at-$\frac{1}{2}$ LP, rings (for 100 closest nodes).

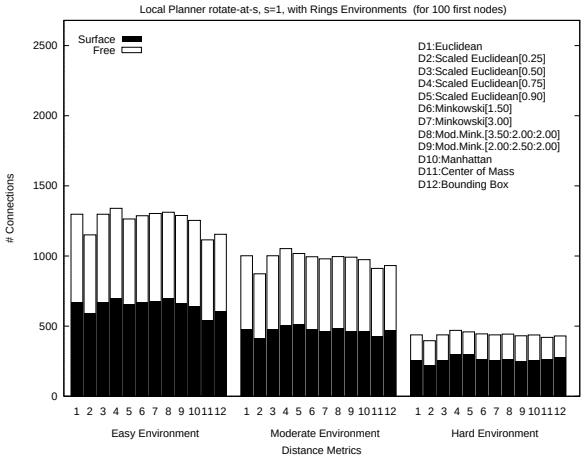


Figure 75: Rotate-at-1 LP, rings (for 100 closest nodes).

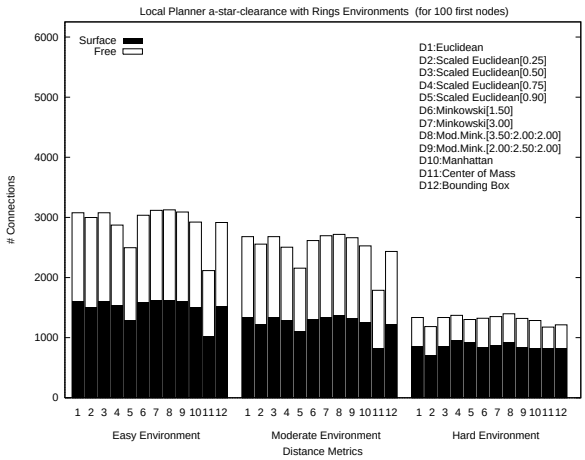


Figure 76: A*-clearance LP, rings (for 100 closest nodes).

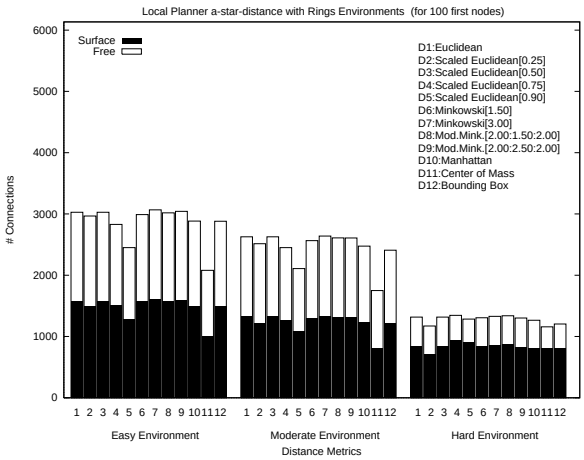


Figure 77: A*-distance LP, rings (for 100 closest nodes).

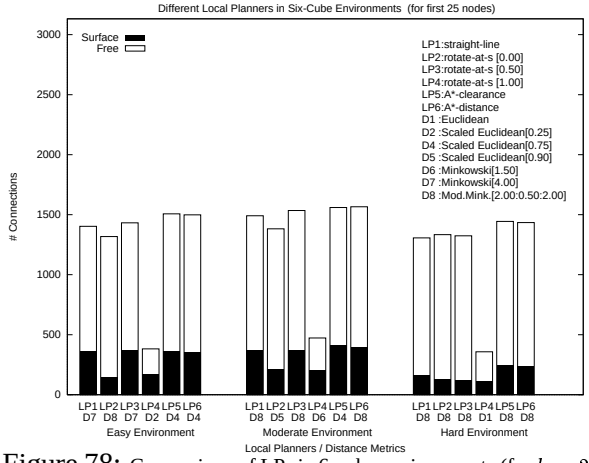


Figure 78: Comparison of LPs in 6-cube environments (for $k = 25$ closest nodes).

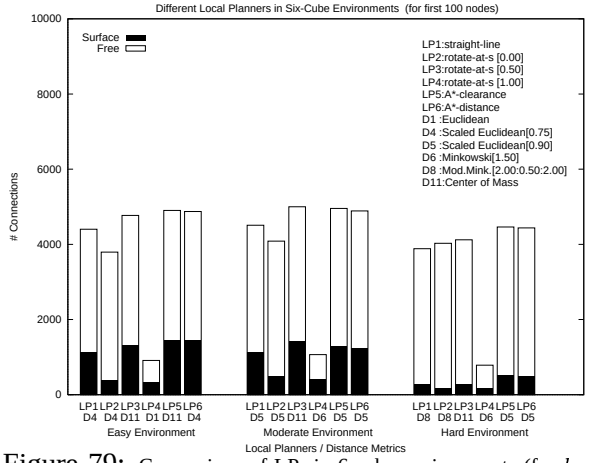


Figure 79: Comparison of LPs in 6-cube environments (for $k = 100$ closest nodes).

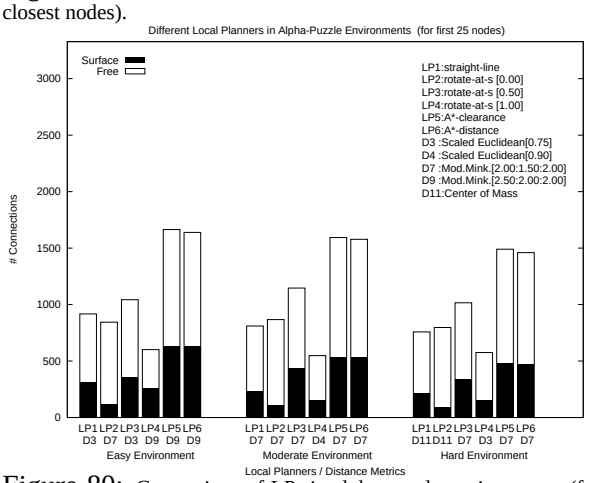


Figure 80: Comparison of LPs in alpha-puzzle environments (for $k = 25$ closest nodes).

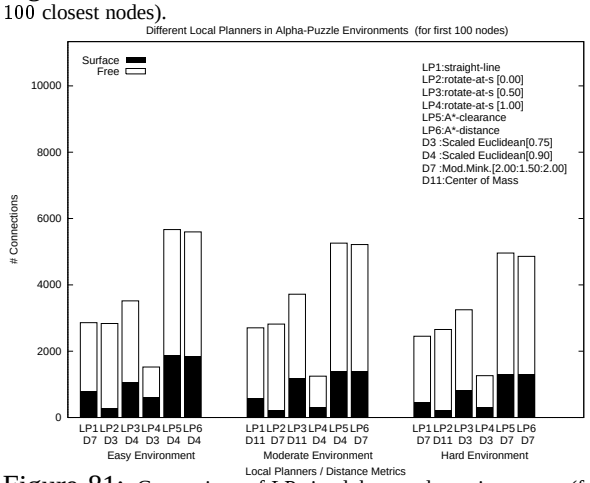


Figure 81: Comparison of LPs in alpha-puzzle environments (for $k = 100$ closest nodes).

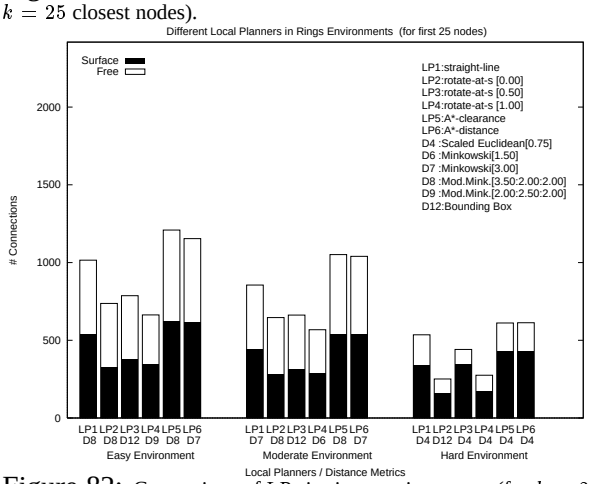


Figure 82: Comparison of LPs in rings environments (for $k = 25$ closest nodes).

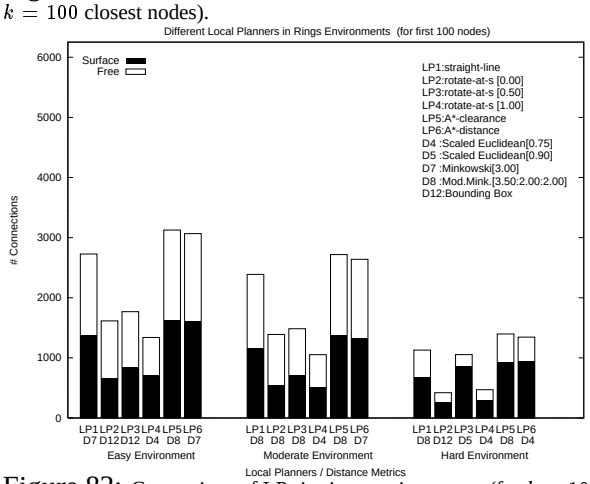


Figure 83: Comparison of LPs in rings environments (for $k = 100$ closest nodes).

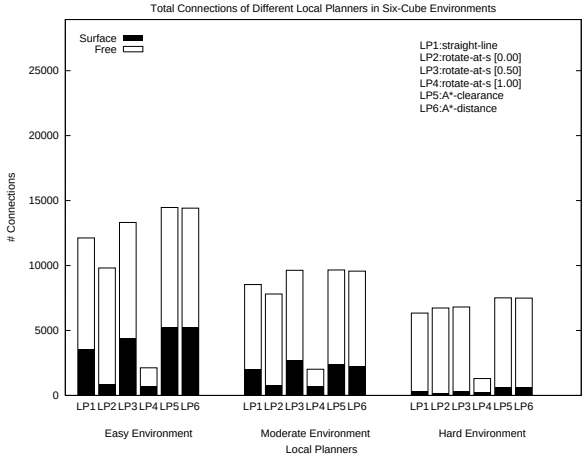


Figure 84: 6-cube Environment

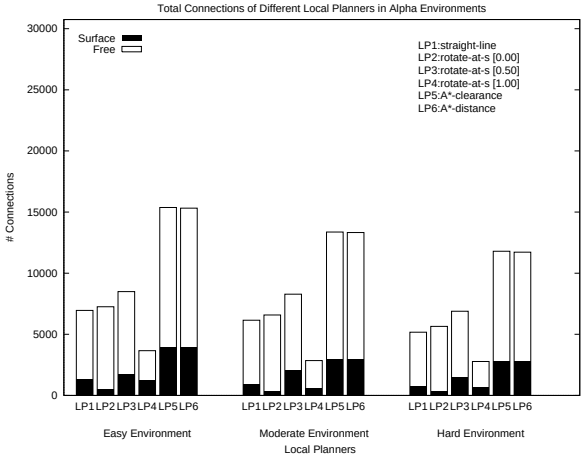


Figure 85: Alpha-Puzzle Environment

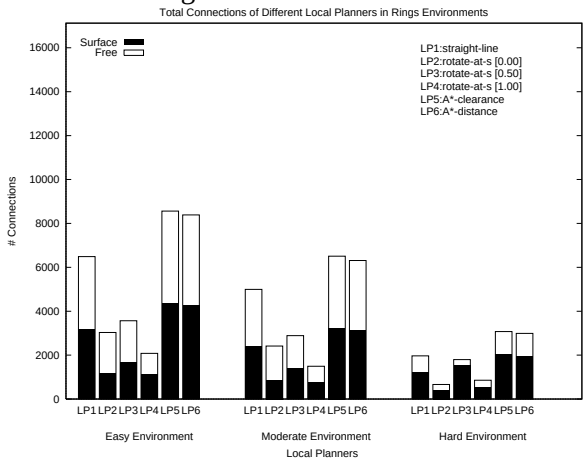


Figure 86: Rings Environment

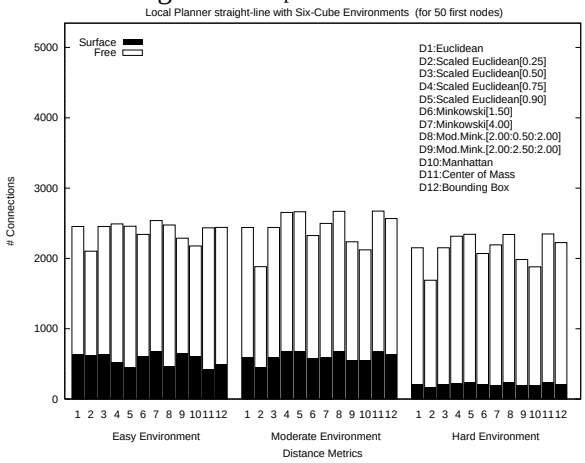


Figure 87: Straight-line LP, 6-cube (for 50 closest nodes).

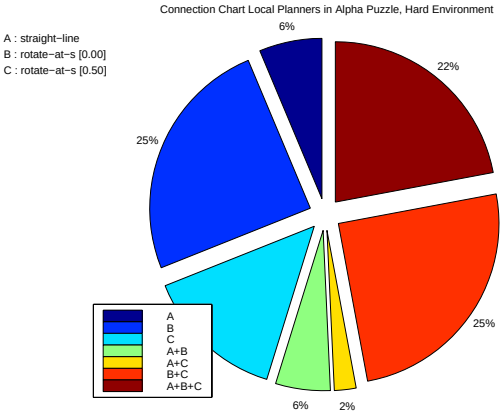


Figure 88: LPs: SL, R-at-0, R-at-$\frac{1}{2}$ in hard alpha-puzzle.

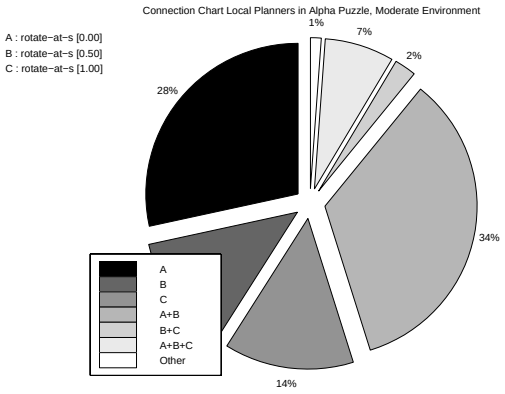


Figure 89: LPs: R-at-0, R-at-$\frac{1}{2}$, R-at-1 in moderate alpha-puzzle.

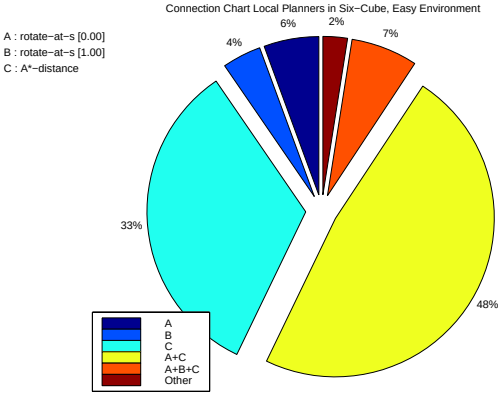


Figure 90: LPs: R-at-0, R-at-1, A*-Dist in easy 6-cube.

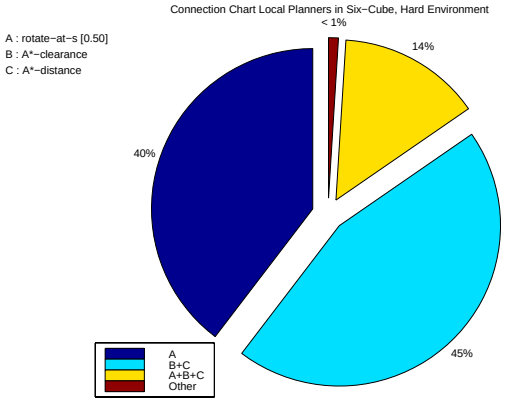


Figure 91: LPs: R-at-$\frac{1}{2}$, A*-Clear, A*-Dist in hard 6-cube.

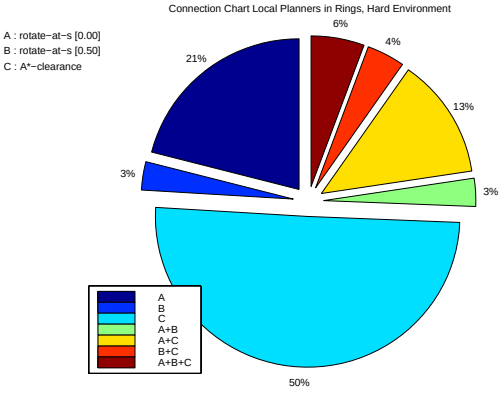


Figure 92: LPs: R-at-0, R-at-$\frac{1}{2}$, A*-Clear in hard rings.

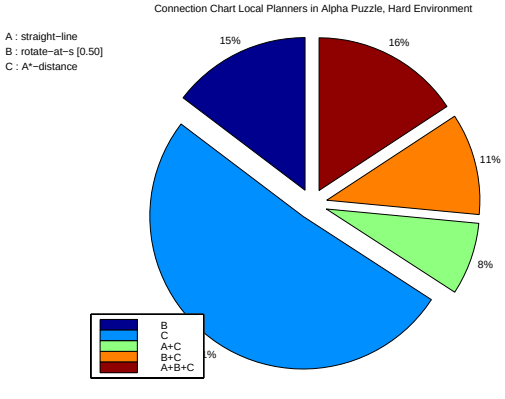


Figure 93: LPs: SL, R-at-$\frac{1}{2}$, A*-Dist in hard alpha-puzzle.