# Bias-guided Metrics for Motion Planning Problems

Regina Rex[1], Bonnie Wang[2], Diane Uwacu[3], Shawna Thomas[3], Nancy M. Amato[4⋆]

[1] Department of Mathematics and Computer Science,
University of Wisconsin-Superior, Superior, WI, USA,
rrex@uwsuper.edu
[2] Department of Computer Science,
Columbia University, New York, NY, USA,
bw2551@columbia.edu
[3] Parasol Lab, Department of Computer Science and Engineering,
Texas A&M University, College Station, TX, USA,
{duwacu, sthomas}@tamu.edu
[4] Parasol Lab, Department of Computer Science,
University of Illinois at Urbana-Champaign, IL, USA,
namato@illinois.edu

**Abstract.** We present a method that uses workspace properties like clearance and energy to improve the motion planning exploration process. Our approach is applied to the robotics and Protein-ligand application of motion planning. In robotics, we use the workspace property, clearance, to find safer paths for robots. And in proteins, we use energy levels and clearance to explore possible routes for the ligand. In both applications, our experiments show we find safer paths in a faster time when compared to other methods.

## 1 Introduction

Motion Planning is a geometry problem that is typically used in robotics to study robot navigation. Given a robot with its desired start and goal points, the motion planning task is to find a collision-free path from the start to the goal while respecting the constraints of the robot. Aside from robotics, motion planning is applied in computer-aided designs, computer graphics, and bioinformatics [1]. One of its applications in bioinformatics is its use in studying how a geometric body known as protein changes when it interacts with a rigid body called a ligand. This application is essential as it is used to improve the design of drug molecules.

This geometry problem is challenging to solve for problems involving complex geometries which are hard to represent in a computer. A deterministic approach

⋆ This work was performed at the Parasol Lab, at the University of Illinois at Urbana-Champaign, in Summer 2019 and supported in part through the CRA-WP Distributed Research Experiences for Undergraduates (DREU).

to the motion planning problem has an exponential runtime proportional to the dimension of the geometrical bodies involved, which implies that motion planning problems are PSPACE-hard [1]. Because of this, randomized algorithms are used to address complicated motion planning problems involving high dimensions. However, the trade-off is that randomized algorithms are probabilistically complete, but they are practically more efficient and less time-consuming for complex problems.

The term "workspace" refers to the physical environment of the robot and obstacles. The workspace is dependent on the motion planning problem, and each workspace has specific properties depending on its application and the geometrical bodies included. For example, in the robotics application, a workspace property could be the lighting. Particular passages in the workspace may have different lighting intensity. Another property for this application is clearance value, as various passages may have varying sizes of free-space or passage width. In the bioinformatics application, the energy threshold is the shared workspace property. However, different molecules have diverse ranges for the energy levels that are specific to their molecular structure.

The Bias-guided method is also applied to the Protein-ligand Binding problem in computational biology. Ligands are small drug molecules that when interacted with proteins, change their shape and functionality. These interaction regions are known as binding sites and can be located within the protein's inner hull. We can study the protein as a geometric and energetic problem, applying both the clearance biasing method as well as an additional energy metric biasing method. These metrics can help gain insight on how a ligand navigates around the tunnels of the protein. Only recently, Protein-ligand binding methods began considering the feasible paths a ligand can take to the binding site. Previous work mainly focused on the final fit of the ligand to the binding site, without considering the path the ligand must traverse to reach its goal. Current work studies these possible paths, investigating how the protein's tunnels can regulate the accessibility of certain ligands to the site [2].

Our method strives to understand and evaluate the tunnels of the protein that the ligand would most likely be able to access in order to reach the binding site. Not only is accessibility of a tunnel important, but probability of a tunnel being a true path a ligand could take should also be taken into account. In order to gain insight on meaningful tunnels, we want to find tunnels of high volume and low energy for which the Protein-ligand complex will increase in stability [2].

By comparing the Bias-guided strategy with other non-biasing strategies, we can see improvements in speed and the quality of tunnels found. The experiments presented in this paper shows that by using workspace metrics to bias planning, we generate desirable paths faster for both the robotics and Protein-ligand application of motion planning.

## 2 Related Work

In this section, we would discuss the preliminaries of our work.

### 2.1 The Motion Planning Problem

A configuration represents a complete specification of the position of every point on the robot. And the set of all the possible configurations of the robot is known as the Configuration Space $C_{space}$. The $C_{space}$ is made up of Obstacle Space $C_{obst}$ and Free Space $C_{free}$. $C_{obst}$ is the set of all configurations which lies in one or more obstacles, and $C_{free}$ represents the set of all configurations that are not in $C_{obst}$. The robot is represented as a configuration in $C_{space}$. $C_{space}$ is an abstract model of the workspace and as such, it does not capture all the possible constraints of the workspace [3].

The motion planning problem is defined in $C_{space}$ as follows: Given a start configuration ($q_s \in C_{free}$) and a goal configuration ($q_g \in C_{free}$). Return a continuous path, $p : [0, 1] \rightarrow C_{free}$ such that $p(0) = q_s$ and $p(1) = q_g$. Other non-robotics applications can be formulated as a motion planning problem, if they can be map to the geometric definition given above. For example, the study of Protein-ligand interaction is a motion planning problem, where the robot is the ligand, and the geometric representation of the protein is the workspace.

**Protein-Ligand Binding Problem** A protein is a large structure made up of a chain of amino acids that interacts in several essential reactions in the body [2]. These reactions can include binding with a small drug molecule called a ligand. The region the ligand interacts with on the protein is known as the binding site. Once the interaction occurs, the protein's shape and functionality can change.

Recent research indicates that the molecular tunnels of the protein regulate the accessibility of ligands and is dependent on those ligands' specific characteristics. These tunnels show a connection with how binding site activity behaves. Investigating how the ligand is able to travel to the binding site can provide insight in how Protein-ligand binding works and how to predict certain biological phenomena.

### 2.2 Sampling-based Planning

This planning approach avoids creating an explicit construction of $C_{obst}$. Instead it treats $C_{space}$ as a black box, and randomly samples configurations in $C_{space}$ using a collision detection check (CDC) to define its validity. A roadmap is constructed from the valid samples and is then searched for a path [3]. A roadmap is a topological graph, which every vertex represents a configuration, and each edge is a path connecting two configurations. One problem with this approach is the cost of CDC. CDCs are expensive because they involve translating the problem between $C_{space}$ and workspace. Another bottleneck for this approach is narrow passages. Narrow passages are parts of the environment where the probability of sampling a valid object position is low.

**Rapidly-exploring Random Trees (RRTs)** A Rapidly-exploring Random Tree (RRT) is a sampling-based planner which takes a tree-based approach to

solving the motion planning problem. Tree-based planners are planners that builds a tree roadmap during planning. At each extension attempt of an RRT, it generates a random configuration, $q_{rand}$ which is connected to the nearest configuration, $q_{near}$, If no obstacles exist between $q_{rand}$ and $q_{near}$. RRTs are good for solving single query problems and they tend to explore the workspace better than other sampling-based planners.

**Probabilistic Roadmap (PRM)** PRM takes a graph based approach to solving the motion planning problem [4]. This planning strategy creates a graph or roadmap by repeatedly generating random possible configurations and then attempting to connect them. As a result, the graph that is created will be entirely in free space since the planner will check for collisions when connecting the roadmap. PRM has an advantage over other planning strategies because it is able to handle multiple queries. In other words, PRM can search and find multiple paths with a single generated roadmap.

### 2.3 Guided Motion Planning

**Topological Guidance** Topological guidance involves using the workspace structure to direct how sampling-based planners like PRM and RRT explores the environment. Some approaches to the motion planning problem utilize the workspace decomposition [5, 6] for planning, specifically for targeting narrow passages. The workspace decomposition involves partitioning the workspace into tetrahedral, which are used to bias the sampling process. Another approach is the used of Skeletons made from the workspace decomposition to bias sampling in $C_{space}$. Workspace Skeletons are graphs that capture the topological features of the environment.

The Bias-guided method relies on topological guidance sampling-based planners. The goal of our approach is to exploit the topological properties of such methods using metrics that are beneficial to specific workspace correlated problems. As mentioned in the introduction, our method is applied to DR-RRT and DR-RRG. For both skeleton-guided planners, we use Mean Curvature Skeleton (MCS) for generating our workspace skeleton [7]. MCS constructs the workspace skeleton using a mesh-based algorithm to compute its skeletal representation from the mean curvature flow of surfaces in the workspace.

**Planning** We use different planners to build our roadmap based on the motion planning problem. Different planning methods have advantages in certain environments over others. The Bias-guided method is applied to dynamic region-biased strategies such as Dynamic Region Rapidly-exploring Random Graph (DR-RRG) [8] for proteins and Dynamic Region Rapidly-exploring Random Tree (DR-RRT) [9] for robotics.

Dynamic region-biased strategies guide the planner to sample only from particular regions based on the workspace topology. A workspace skeleton is created to direct regions to sample from. These regions will then be created, explored,

and then destroyed during planning. DR-RRT is a skeleton-guided RRT which uses the workspace skeleton to bias RRT growth. In other words, the planner will grow the tree from samples generated inside the selected region and will advance to the next region once those regions have been represented. DR-RRT is faster and returns lower collision detection calls when compared to basic RRT. While DR-RRG is a skeleton-guided strategy that combines PRM and RRT. Like the DR-RRT strategy, the planner will choose a region to sample from using RRT; however, it will then attempt to connect these samples to form a roadmap to increase connectivity. One advantage of DR-RRG over DR-RRT is that it supports multiple queries on the roadmap.

## 3 Method

As mentioned in section 2.3, our method is designed for skeleton-guided RRTs and applied to DR-RRT in the robotics application and DR-RRG in the protein application.

**Algorithm 1** Bias-guided DR-RDMP

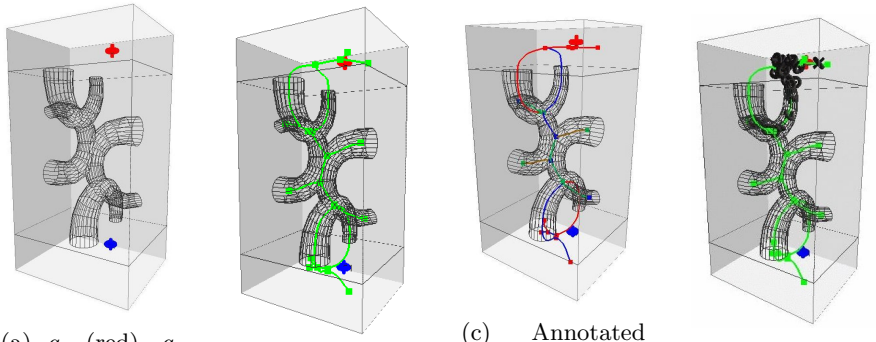
**Input:** Environment $env$, Start $s$, Goal $g$,
Bias Metric (min/max, clearance/energy) $biasMetric$
**Output:** Path $p$
1: $WS \leftarrow$ `GetWorkspaceSkeleton`($env$)
2: $AS \leftarrow$ `AnnotateSkeleton`($WS$)
3: $g \leftarrow s$
4: $r \leftarrow$ `GetInitialRegion`($AS, s$)
5: **while** $\neg done$ **do**
6: $C_r \leftarrow$ `GetChildren`($r$)
7: $r \leftarrow$ `SelectRegion`($C_r$, $biasMetric$)
8: $R \leftarrow$ `GrowRDMP`($r$)
9: **end while**
10: $p \leftarrow$ `Query`($R, g$)
11: **return** $p$

### 3.1 Algorithm Overview

The Bias-guided method utilizes workspace properties such as clearance and energy to guide how skeleton-guided planners generate roadmaps. Algorithm 1 details an application-agnostic Bias-guided strategy to simplify the method explanation. Figure 1 would be used as an example to demonstrate the execution of the method.

The Bias-guided algorithm takes in the environment, query $(q_s, q_g)$, and bias metric which specifies if the planner should favor the maximum or minimum energy or clearance regions. Next, the method generates a workspace skeleton

(Figure 1b) for the given environment. The workspace skeleton is then annotated with workspace properties such as clearance and energy, as seen in Figure 1c.



(a) $q_s$ (red), $q_g$ (blue)

(b) Skeleton

(c) Annotated skeleton

(d) Planning

Fig. 1: Example execution of the Bias-guided method: (a) env with $q_s$ and $q_g$ (b) MazeTunnel Workspace Skeleton (green) (c) Workspace Skeleton annotated with clearance value ( red $\rightarrow$ highest clearance, blue $\rightarrow$ low clearance) (d) Minimum clearance-bias exploration process (half-way).

The planning process is directed using the annotated skeleton. We start sampling close to $q_s$ to ensure the algorithm explores as many paths or tunnels that lead to the goal (line 3). Our method biases roadmap construction by selecting the region to explore from the annotated skeleton using the biasMetric. It then samples and grows from that region to build the roadmap (line 5-8). In Figure 1d, roadmap construction is biased towards regions with minimum clearance. After roadmap construction, we query the resulting roadmap for a path to the goal (line 10).

### 3.2 Metrics

This section covers detailed description of the metrics used for the Bias-guided method.

**Computing Clearance** Clearance is defined as the size of free space between obstacles in the environment. Given a workspace skeleton, we compute clearance using the skeleton node and the point closest to the nearest obstacle. The distance between this point and the skeleton node is what we define as its clearance value.

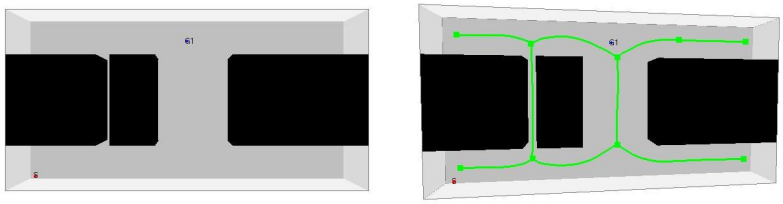
To annotate the workspace skeleton with its clearance value, we do the following: Calculate the clearance value of every node in the workspace skeleton. Then, the edge clearance value is assigned as the maximum or minimum node

clearance value, which makes up the edge. We pick the maximum node clearance for minimum Bias-guidance and minimum node clearance for maximum Bias-guidance. This way, we ensure that we select the best option available for each step of the planning process.

**Computing Energy-value** Our energy function calculates the energy of each region based on Van der waals interactions, which are weak intermolecular forces between two atoms or molecules [10]. How strong the attraction between two atoms are determined by the distance they are from one another; as the atoms move closer the energy of the system decreases, but if they come too close the atoms start to repel each other and dramatically increase the energy. The energy values the function outputs for is determined by first discretizing the workspace into grid cells and determining the influence of the force of every atom of the protein on the ligand.

### 3.3 Visualization Tools

Parasol lab's motion planning visualization tool, Vizmo++, is used to interact and visualize motion planning problems. To test how well our biasing strategy will influence the construction of the roadmap, we added a feature to color the workspace skeleton based on its annotated values. The feature is compatible with both clearance and energy annotations and is adaptable for any given range of values. Figure 1c depicts a workspace skeleton annotated with clearance value, with red indicating high clearance and blue for low clearance. This visualization feature is also handy for debugging purposes and visualizing the performance of the Bias-guided method.



(a) Obstacles with $q_s$, $q_g$ (b) Obstacles with MCS

Fig. 2: Obstacles robotics environment

## 4 Experiments

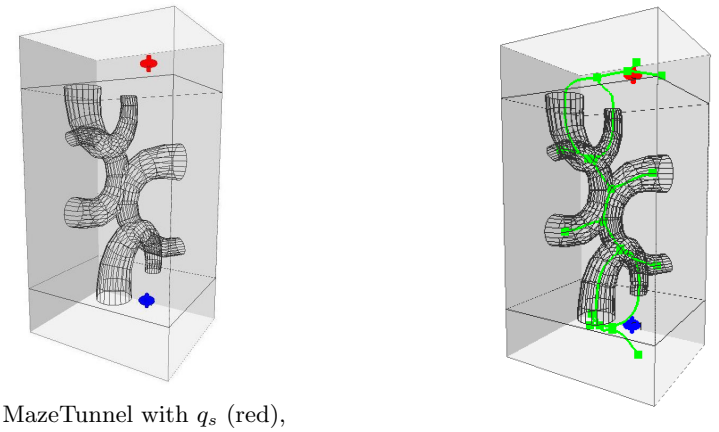
The Bias-guided method is implemented in C++ using the Parasol Motion Planning Library (PMPL) and Standard Template Adaptive Parallel Library

(STAPL); both libraries are developed in the Parasol Lab. We also use Vizmo++ to visualize the simulated experiments. And all the experiments were run on a 3.40GHz Intel Core i7-3770 CPU.

| Environment | Strategy | Runtime | CDC | Nodes | Edges | Path Clearance |
|---|---|---|---|---|---|---|
| Obstacles | Regular DR-RRT | 0.33 | 87,841 | 267 | 531 | 5.05 |
| | Max Clearance-bias | **0.18** | **40,510** | **131** | **260** | **6.12** |
| | Min Clearance-bias | 0.22 | 52,287 | 177 | 353 | 3.40 |
| MazeTunnel | Regular DR-RRT | 0.46 | 11,553 | 98 | 194 | **0.93** |
| | Max Clearance-bias | **0.18** | **6,158** | **65** | **127** | 0.92 |
| | Min Clearance-bias | 1.45 | 28,077 | 88 | 173 | 0.85 |

Table 1: Robotics experiment results. Bolded values indicate the best value for each environment. The best value for Runtime, CDC, Nodes, and Edges is the lowest number in the column, while the best value for the Path Clearance is the highest number, which indicates the safest path.

### 4.1 Robotics Experiments



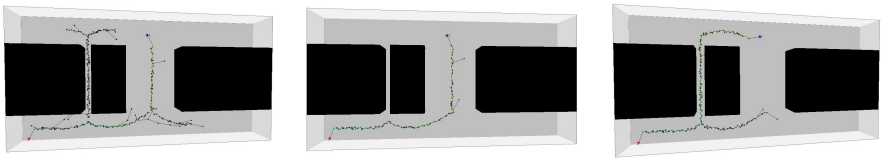
(a) MazeTunnel with $q_s$ (red), $q_g$ (blue)

(b) MCS for MazeTunnel

Fig. 3: MazeTunnel robotics environment

**Environment Setup** In the robotics experiments, we compare our method to the regular DR-RRT. We used DR-RRT because it is a skeleton-guided RRT that utilizes the workspace decomposition for directing RRT growth.

We ran our experiment in MazeTunnel (Figure 3a), and Obstacles (Figure 2a) using holonomic robots. Both environments have wide and narrow passage options. We selected these 3D environments to demonstrate the process of our method and its advantages in such situations. We use MCS for our experiments because it provides a better workspace skeleton for environments with convex bodies like the MazeTunnel environment. Each experiment ran until the query was solved. We performed trials using ten random seeds for each strategy in both environments, and we averaged the following over all the trials: runtime, nodes, edges, CDC, and path clearance.



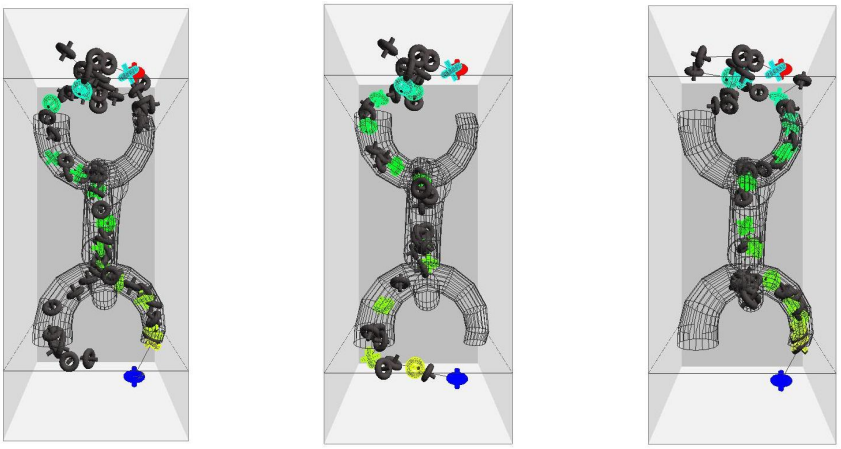
(a) Regular DR-RRT (b) Max Clearance-bias (c) Min Clearance-bias

Fig. 4: The resulting roadmap and path from running the different strategies in Obstacles.

**Result** Table 1 contains the data collection from the robotics experiment. Runtime is the time (in seconds) taken for each strategy to find a path. CDC is the amount of collision detection checks performed while the strategy runs. Nodes and Edges refer to the number of nodes and edges in the resulting roadmap. And Path Clearance is the clearance value of the path return by the strategy. Runtime, CDC, Nodes, Edges, and Path Clearance are averaged over all the ten trials performed. Figure 4 and 5 shows the roadmap (in black) and the resulting path (highlighted) from running the experiments in Obstacles and MazeTunnel environment.

**Discussion** In Obstacles, the results Table 1 shows that both minimum and maximum clearance-bias have lower CDC than regular DR-RRT. The resulting roadmaps with these strategies also have fewer nodes and edges when compared to regular DR-RRT. Minimum clearance-bias returns the path with the lowest clearance value, while maximum clearance-bias returns the path with the widest clearance. The Bias-guided method also returns the safest path in the quickest time using maximum clearance-bias in Obstacles.

Maximum clearance-bias returns the widest path in MazeTunnel with lower CDC, nodes, and edges when compared to regular DR-RRT. And it is the fastest strategy in MazeTunnel. Minimum clearance-bias was the slowest, but it had fewer nodes and edges than regular DR-RRT. As seen from our experiments, biasing roadmap construction using a workspace property such as clearance,

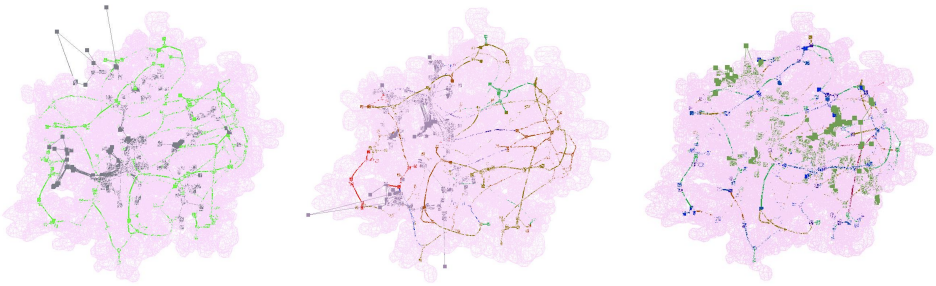


(a) Regular DR-RRT (b) Max Clearance-bias (c) Min Clearance-bias

Fig. 5: The resulting roadmap and path from running the different strategies in MazeTunnel.

help speed up the planning process in robotics environments like Obstacles and MazeTunnel.

### 4.2 Protein Experiments



(a) Regular DR-RRG (b) Min Energy-bias (c) Max Clearance-bias

Fig. 6: The resulting roadmaps from running the different strategies in 3fbw

**Environment Setup** The strategies used for the protein efcinvironments are the Bias-guided method and regular DR-RRG for comparison. We biased towards

minimum energy for the energy-bias method and maximum clearance for the clearance-bias method. MCS is used to generate the skeleton for all strategies.

The data for the protein environments were obtained from the Protein Data Bank (PDB) [11], and their corresponding geometric model was extracted using UCSF Chimera [12]. We also got the ligand data from PDB and modeled it geometrically using a custom script.

All the strategies were tested in four different proteins (3fbw, 4fwb, 3rk4, hzg), all of which binds to the same type of ligand (3KP). We performed trials using ten random seeds, and each experiment ran until a specified number of nodes are built outside the hull of the protein. We also observed and analyzed the tunnels found in the resulting roadmaps.

| Protein | Strategy | Runtime | Nodes | Edges | Tunnels |
|---|---|---|---|---|---|
| 3fbw | default | 173 | 644 | 2878 | **12** |
| | energy | **121** | **413** | **2534** | **12** |
| | clearance | 160 | 616 | 2546 | 10 |
| 3rk4 | default | **273** | 772 | 3202 | **12** |
| | energy | 318 | 693 | 2683 | 11 |
| | clearance | 329 | **428** | **1904** | 9 |
| 4fwb | default | 125 | 559 | 2421 | 8 |
| | energy | 125 | 492 | **2183** | 9 |
| | clearance | **122** | **472** | 2389 | **12** |
| 4hzg | default | **123** | 455 | 1907 | 8 |
| | energy | 169 | **350** | **1649** | **11** |
| | clearance | 141 | 454 | 1984 | **11** |

Table 2: Protein experiment results. Bolded values indicate the best value for each environment. The best value for Runtime, Nodes, and Edges is the lowest number in the column, while the best value for the Tunnels is the highest number.

**Result** Table 2 shows the median runtime, number of nodes, edges, and tunnels found over all the trials performed in each environment.

**Discussion** Although the runtime for the strategies is similar in Table 2, the Bias-guided strategies produced a smaller and tighter roadmap as opposed to the roadmaps returned by regular DR-RRG. We can see that both the clearance and energy biased strategies explored favorable regions (red and orange colors of Figure 6b and 6c) of the skeleton. The Bias-guided roadmaps are congregated around low energy, high volume areas, which demonstrates that our method finds more favorable tunnels than the regular DR-RRG.

The number of tunnels found throughout the ten seed runs is relatively around the same across all strategies. We notice that the biasing strategies are more efficient in finding specified tunnels in less time.

## 5 Conclusions

As noticed from the experiments above, the Bias-guided method is dependent on the workspace skeleton, and it performs best on motion planning problems that have strong relations to its workspace topology. Biasing exploration using workspace metrics allows us to find the most desirable paths faster, while also targeting exploration to narrow passages.

In summary, we introduced using workspace metrics such as clearance and energy to improve the exploration process for motion planning in robotics and the study of Protein-ligand binding. Our main contribution is the use of workspace properties available to skeleton-guided planners to improve the planning process. More generally, workspace related metrics can be designed for any motion planning problem that would benefit from exploiting such metrics during planning.

### 5.1 Future Work

Future research might consider the potential applications of the clearance metric to other motion planning applications like character navigation in animation and Image-guided Medical Needle Steering [13]. We also plan to conduct real-world application experiments to demonstrate the correctness of our method.

## Acknowledgement

We want to express our gratitude to Nancy M. Amato (faculty mentor), Shawna Thomas (research specialist advisor), and Diane Uwacu (graduate student mentor), for their support and guidance throughout this project. This research is supported in part by the Computing Research Association and the Computer Science Department at the University of Illinois at Urbana-Champaign.

## References

1. J. C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *Int. Journal of Robotics Research*, vol. 18, no. 11, pp. 1119–1128, 1999.
2. S. Kaushik, S. M. Marques, P. Khirsariya, K. Paruch, L. Libichov, J. Brezovsky, Z. Prokop, R. Chaloupkova, and J. Damborsk, "Impact of the access tunnel engineering on catalysis is strictly ligandspecific," *The FEBS Journal*, vol. 285, 02 2018.
3. S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006. Available at http://planning.cs.uiuc.edu/.
4. L. Kavraki, M. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, pp. 3020–3025, 1996.
5. H. Kurniawati and D. Hsu, "Workspace importance sampling for probabilistic roadmap planning," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, vol. 2, pp. 1618–1623, sept.-2 oct. 2004.

6. H. Kurniawati and D. Hsu, "Workspace-based connectivity oracle - an adaptive sampling strategy for prm planning," in *Algorithmic Foundation of Robotics VII*, pp. 35–51, Berlin/Heidelberg: Springer, 2008. Book contains the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), New York City, 2006.
7. A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, "Mean curvature skeletons," *Eurographics Symposium on Geometry Processing*, p. 27(1), 2012.
8. R. Kala, "Rapidly exploring random graphs: motion planning of multiple mobile robots," *Advanced Robotics*, vol. 27, no. 14, pp. 1113–1122, 2013.
9. J. Denny, R. Sandstrom, A. Bregger, and N. M. Amato, "Dynamic region-biased exploring random trees," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, (San Francisco, CA), December 2016.
10. M. Levitt, "Protein folding by restrained energy minimization and molecular dynamics," *Journal of Molecular Biology*, vol. 170, no. 3, pp. 723–764, 1983.
11. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 01 2000.
12. E. Pettersen, T. Goddard, C. Huang, G. Couch, D. Greenblatt, E. Meng, and T. Ferrin, "Ucsf chimera: A visualization system for exploratory research and analysis," *Journal of Computational Chemistry*, vol. 25, no. 13, pp. 1605 – 1612, 2004.
13. R. Alterovitz, M. Branicky, and K. Goldberg, "Motion planning under uncertainty for image-guided medical needle steering," *The International journal of robotics research*, vol. 27.