METRICS FOR SAMPLING-BASED MOTION PLANNING

A Dissertation

by

MARCO ANTONIO MORALES AGUIRRE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2007

Major Subject: Computer Science

METRICS FOR SAMPLING-BASED MOTION PLANNING

A Dissertation

by

MARCO ANTONIO MORALES AGUIRRE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Nancy M. Amato
Committee Members,	Ricardo Gutiérrez-Osuna
	Donald H. House
	John Keyser
Head of Department,	Valerie E. Taylor

December 2007

Major Subject: Computer Science

ABSTRACT

Metrics for Sampling-Based Motion Planning. (December 2007)

Marco Antonio Morales Aguirre, B.S., Universidad Nacional Autónoma de México;
M.S., Universidad Nacional Autónoma de México
Chair of Advisory Committee: Dr. Nancy M. Amato

A motion planner finds a sequence of potential motions for a robot to transit from an initial to a goal state. To deal with the intractability of this problem, a class of methods known as sampling-based planners build approximate representations of potential motions through random sampling. This selective random exploration of the space has produced many remarkable results, including solving many previously unsolved problems. Sampling-based planners usually represent the motions as a graph (e.g., the Probabilistic Roadmap Methods or *PRMs*), or as a tree (e.g., the Rapidly exploring Random Tree or RRT). Although many sampling-based planners have been proposed, we do not know how to select among them because their different sampling biases make their performance depend on the features of the planning space. Moreover, since a single problem can contain regions with vastly different features, there may not exist a simple exploration strategy that will perform well in every region. Unfortunately, we lack quantitative tools to analyze problem features and planners performance that would enable us to match planners to problems.

We introduce novel metrics for the analysis of problem features and planner performance at multiple levels: node level, global level, and region level. At the node level, we evaluate how new samples improve coverage and connectivity of the evolving model. At the global level, we evaluate how new samples improve the structure of the model. At the region level, we identify groups or regions that share similar features. This is a set of general metrics that can be applied in both graph-based and tree-based planners. We show several applications for these tools to compare planners, to decide whether to stop planning or to switch strategies, and to adjust sampling in different regions of the problem.

To my wife Aimée, blessing love and fortitude:
tonehuan ticeixtli ticeyollotl (you and me, one face, one heart)

To my parents Marcos and Esperanza:
nianix nianyollo (I hold your face and heart)

ACKNOWLEDGMENTS

I would like to thank Dr. Nancy Amato. She is a great mentor and research leader. Her guidance was key to find the right way in this journey countless times. It has been a privilege and joy to be part of her research group.

Thanks to Dr. Ricardo Gutiérrez-Osuna, Dr. Donald House, and Dr. John Keyser, for their comments that enriched this research and for their support. Also, thanks to the many professors I met at A&M, for their classes, talks, and discussions.

Thanks to the members of the Parasol Lab for their great work that supports our research, for the many discussions, and for the rewarding collaborations, especially: Burchan Bayazit, Jyh-Ming Lien, Olga Pearce, Roger Pearce, Samuel Rodríguez, Guang Song, Xinyu Tang, Lydia Tapia, Shawna Thomas, Aimée Vargas, and Dawen Xie. Also, thanks to the great students I could meet at A&M.

Thanks to the Comission México-United States for Educational and Cultural Exchange for the Fulbright-Garcia Robles Scholarship that allowed me to pursue a PhD. Also, thanks to the Mexican Government and its Consejo Nacional de Ciencia y Tecnología , and to the U.S. Government, its Department of State, and the Institute of International Education.

Thanks to Marcos and Esperanza, my parents. They showed me how to search for my way in life, they carefully nurtured my curiosity, they let me go, and they have always been there for me. Also, thanks to Rocío, my sister, for her encouragement and friendship.

My most wholehearted thanks to Aimée, my loved wife and comrade. Her courage fed my enthusiasm to embark in this adventure. Her support has kept me afloat when my strength has dwindled. Her tenderness has warmed my heart in the worst times. Her smile is the tastiest delicacy. We are truly one face and one heart.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

II SAMPLING-BASED MOTION PLANNING 5

A. Configuration Space 6
1. Validity 6
2. Visibility 7
3. Visibility or Covered Region 8
4. Connectability 8
5. Homotopy 8
B. Sampling-Based Motion Planning Techniques 9
1. Roadmap-Based Planners 10
2. Incrementally-Exploring Planners 13
3. Adaptive Planners 15

III NEW METRICS FOR SAMPLING-BASED PLANNERS 16

A. Traditional Evaluation of Planners 16
B. New Node, Region, and Global Metrics for Sampling-Based Planners 18
1. Coverage 19
2. Connectivity 19
3. Topology 20
4. Estimating the Evolution of Planner Learning Ability 20
5. Extensions to Other Types of C-Space Models 21

IV EVALUATION STRATEGY 23

A. Planners 23
B. Motion Planning Problems 25
C. Experimental Setup 28

V NODE-LEVEL METRICS 32

A. Type and Amount of Improvement Produced by a New Node 32
B. Population Distribution of Node Types 40
C. Visibility Around Growth Sites 43

CHAPTER Page

D. Overhead of Node-Level Metrics 45

VI GLOBAL-LEVEL METRICS 47

A. Changes in Motion Pathways Produced by New Nodes . . 47
B. Detection of Relative Change of Global-Level Metrics . . . 53
C. Overhead of Global-Level Metrics 55

VII REGION-LEVEL METRICS 58

A. Region Construction 58
1. Features 59
2. Clustering Strategies 59
a. Axis-Aligned Regions 59
b. Simple-Feature Regions 60
c. Coverage Regions 61
3. Frequency of Region Updates 63
4. Region Statistics 63
B. Overhead of Region-Level Metrics 64

VIII APPLICATIONS AND EXPERIMENTS 66

A. Learning Process of Planners 66
1. Evolution of the Node-Level Metrics 66
2. Evolution of the Global-Level Metrics 69
B. Stages of the Learning Process of Planners 76
1. How Stage Transitions Can Be Detected? 80
2. What Can Be Done When *Learning Decay* Starts? . . 83
C. Distribution of Nodes in the C-Space 85
1. What Is the Population Distribution of Regions for Different Planners? 85
2. How Effective Are Planners in Biasing towards Highly-Constrained Regions? 90
3. How Do Coverage Regions Evolve in Incremental Planners? 90
4. How Can We Adapt Planning Based on Region Complexity? 92
D. Comparison of Metrics at the Start of Learning Decay for Different Roadmap-Based Planners 94
E. Metrics in High-DOF Problems 97

IX CONCLUSIONS 102

Page

REFERENCES 106

VITA 114

LIST OF TABLES

TABLE Page

I Visibility of Growth Sites in *serial-hook-5* 44

II Node-Level Metrics Overhead in *rigid-maze* 46

III *Basic-PRM* on *rigid-windows.* Robot Width = 1 51

IV Global-Level Metrics Overhead in *rigid-maze* 57

V Region-Level Metrics Overhead in *rigid-maze* 65

VI Population Distribution of Nodes in *rigid-maze.* Deviation 71

VII Population Distribution of Nodes in *rigid-walls.* Deviation 71

VIII Population Distribution of Visibility Regions in *rigid-maze.* Deviation 88

IX Population Distribution of Visibility Regions in *rigid-walls.* Deviation 90

X Number of Samples Needed for Planners to Get to Goal Region . . . 92

XI Planners at Start of *Learning Decay* on *rigid-maze* 95

XII Planners at Start of *Learning Decay* on *rigid-hook* 97

XIII Planners at Start of *Learning Decay* on *serial-hook-5* 98

LIST OF FIGURES

FIGURE Page

1 Validity, visibility, visible region, connectability, and homotopy in (a) the C-Space of a point robot moving in the plane. (b) A valid configuration in $\mathcal{F}$ and an invalid configuration in $\mathcal{O}$. (c) Visibility with straight-line local planner: a can see b but it can not see c. (d) The visible region for a, with straight-line local planner, in light blue. (e) a and c are connectable ($\tau(a,c) = \{a,b,c\}$). (f) Two homotopy classes between a and c: $\tau_1 = \{a,b,c\}$ and $\tau_2 = \{a,b,d,c\}$ can be continuously transformed into each other; 2. $\tau_3 = \{a,e,f,c\}$ cannot be transformed into τ_1 nor τ_2. 7

2 Witness queries between "witness" configurations *1* and *2*. (a) Only witness *1* is connected to the model, there has not been found any path between witnesses. (b) Both witnesses are connected to the model, one path between them has been found. (c) Both witnesses are connected to the model, two paths between them have been found. The witness evaluation does not make a distinction between case (b), with only one pathway between witnesses, and case (c), with two pathways between witnesses. 17

3 Problem: *rigid-maze*. (a) solid view. (b) wire-view shows the internal tunnels. (c) close-up view of the robot. 26

4 Problem: *rigid-windows*. The robot has 3 translational degrees of freedom. Four pathways of different sizes allow the robot to cross from the front to the back. The start and goal configurations are at each side of the leftmost wall. 26

5 Problem: *rigid-hook*. In order to get through the passages, the 6-*DOF* robot needs to perform translations and rotations. 27

6 Problem: *rigid-walls*. Incremental planners find their in increments as they find their way through narrow passages. 28

FIGURE Page

7 Problem: *serial-hook-5*. The 10-*DOF* robot can fold and unfold to get through the opening that divides the environment. This is a variation of the *rigid-hook* problem. 29

8 Problem: *serial-spring-98*. The 103-*DOF* robot folds and unfolds to get above the wall that divides the environment. 29

9 Classification of new nodes when modeling the C-Space of a point robot moving in the plane shown in (a). (b) The first sample in the model with its visibility region. (c) A new sample lying outside the visibility region of any other sample creates another component with its own visibility region. (d) A new sample lying in the overlap of the visibility region of two components allows to merge them. (e) A new sample lying inside the visibility region of one component expanding its visibility: cc-expand. (f) A new sample lying inside the visibility region of one component without changing its visibility: cc-oversample. 33

10 An implementation of classification of new nodes. (a) State of M before a new node is added. (b) New node increases the number of components: *cc-create*. (c) New node reduces the number of components: *cc-merge*. (d) New node v cannot connect to any neighbor of v': *cc-expand*. (e) New node v cannot connect to 50% of the neighbors of v': *cc-expand* for $E_t = 0.5$. (f) New node v can connect to all the neighbors of v': *cc-oversample*. 35

11 Absolute error in the classification of *cc-expand* nodes for different approximations ($p = 0.0$ no additional connection test, $p = 0.1$, and $p = 0.5$) with respect to a full test ($p = 1.0$) vs. nodes in the model for roadmap-based planners applied to the *rigid-maze* problem. (a) *Gauss-PRM*. (b) *OBPRM*. Each line represents statistics from four different runs with standard deviations are below 5% before 1000 nodes in all the experiments. *Basic-PRM* and *MAPRM* are similar to *Gauss-PRM*, and *Bridge-Test* is similar to *OBPRM*. Overheads are shown in Table II. 37

FIGURE Page

12 Absolute error in the classification of *cc-expand* nodes for different approximations ($p = 0.0$ no additional connection test, $p = 0.1$, and $p = 0.5$) with respect to a full test ($p = 1.0$) vs. nodes in the model for incremental planners applied to the *rigid-maze* problem. (a) *RPP*. (b) *RRT*-Expand. Each line represents statistics from four different runs with standard deviations are below 5% before 1000 nodes in all the experiments. *EST* shows error amounts in between *RPP* and *RRT*-Expand, and *RRT*-Connect is similar to *RRT*-Expand. Overheads are shown in Table II. 39

13 Population distribution of node types produced by individual instances of planners when modeling the *serial-hook-5* problem. (a) *Basic-PRM*. (b) *RPP*. 41

14 Population distribution of node types produced by eight instances of *Basic-PRM* when modeling the *serial-hook-5* problem. (a) Average populations with nodes in logarithmic scale to better see evolution in initial iterations. (b) Standard deviation of populations with both nodes and proportion of nodes shown in logarithmic scale, the standard deviations for all node types fall below 10% before 50 nodes. 42

15 Visibility ratio of growth sites a, c, and e as they are added to the model and connected with a bidirectional local planner. (a) new node a is added, $V_a = 2/3$. (b) new node c is added $V_c = 0/3$, the visibility of a needs to be updated $V_a = 2/4$. (c) new node e is added $V_e = 3/3$, the visibility of a needs to be updated $V_a = 3/5$. . . 44

16 Global changes in model topology resulting from new nodes and connections. (a) State of M before adding new nodes, in the C-Space there is one component and two homotopy classes, or distinct pathways, between a and b; in contrast, the model has three components and no pathway between a and b. (b) Two new nodes and their connections improve the topology of the model to have one component and one homotopy class between a and b. (c) one more node and its connections improve the topology of the model to have one component and two homotopy classes between a and b. 48

FIGURE Page

17 Evolution of global-level metrics for one instance of *Basic-PRM* on the *rigid-windows* problem. (a) Number of components. (b) *sum-diameter* and *max-diameter* correlate to new pathways found through the windows and on each side of the wall. (c) Population distribution of node types, learning stages correlate to changes in diameters. The witness query solved at 9 nodes is marked on both plots. 50

18 Population distribution of node types for one instance of *Basic-PRM* on the *rigid-windows* problem. This distribution corresponds to the same instance discussed in Figure 17. 52

19 Rate of change of *max-diameter* and *sum-diameter* for the same instance of *Basic-PRM* on the *rigid-windows* problem discussed in Figure 17. Size of bins $n = 1$, only changes in each bin are considered. 55

20 Average rate of change of *max-diameter* and *sum-diameter* for the same instance of *Basic-PRM* on the *rigid-windows* problem discussed in Figure 19. (a) Size of bins $n = 1$, averaged bins $k = 4$. (b) Size of bins $n = 10$, averaged bins $k = 8$. 56

21 Visibility regions in a model produced by *Basic-PRM* in the *rigid-maze* problem. (a) 1000-node model. (b) Low-visibility region: *visibility* $< 1/3$. (c) Medium-visibility region: $1/3 <=$ *visibility* $< 2/3$. (d) High-visibility region: $2/3 <=$ *visibility*. 61

22 Visibility regions in a model produced by *OBPRM* in the *rigid-maze* problem. (a) 1000-node model. (b) Low-visibility region: *visibility* $< 1/3$. (c) Medium-visibility region: $1/3 <=$ *visibility* $<$ $2/3$. (d) High-visibility region: $2/3 <=$ *visibility*. 62

23 Average population distribution of node types produced by the roadmap-based *Basic-PRM* planner on the *rigid-maze* problem. 4 runs of 3000 nodes. 67

24 Average population distribution of node types produced by different roadmap-based planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner. . . 68

FIGURE Page

25 Average population distribution of node types produced by different incremental planners on the *rigid-walls* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes were run for each planner. . 70

26 Average *max-diameter* and *sum-diameter* of models produced by *Basic-PRM* on the *rigid-maze* problem. 4 runs of 3000 nodes. 72

27 Average *max-diameter* and *sum-diameter* of models produced by different planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner. 73

28 Average *max-diameter* and *sum-diameter* of models produced by different planners on the *rigid-maze* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes for each planner. 74

29 Average *max-diameter* and *sum-diameter* of models produced by *EST* on the *rigid-maze* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes. 75

30 Standard deviation of *max-diameter* and *sum-diameter* of models produced by the roadmap-based *Basic-PRM* planners on the *rigid-maze* problem. 4 runs of 3000 nodes. 76

31 Standard deviation of *max-diameter* and *sum-diameter* of models produced by different roadmap-based planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner. 77

32 Standard deviation of *max-diameter* and *sum-diameter* of models produced by different incremental planners on the *rigid-walls* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes for each planner. 78

33 Standard deviation of *max-diameter* and *sum-diameter* of models produced by different incremental planners on the *rigid-walls* problem. (a) *EST*. (b) *RRT*-Expand. 8 runs of 3000 nodes for each planner. 79

FIGURE Page

34 Average change in *max-diameter* and *sum-diameter* of models produced by the roadmap-based *Basic-PRM* planner on the *rigid-maze* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. 4 runs of 3000 nodes. 81

35 Average change in *max-diameter* and *sum-diameter* of models produced by different roadmap-based planners on the *rigid-maze* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner. . . 82

36 Average change in *max-diameter* and *sum-diameter* of models produced by the incremental *RRT*-Connect planner on the *rigid-walls* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. 8 runs of 3000 nodes. 83

37 Average change in *max-diameter* and *sum-diameter* of models produced by different incremental planners on the *rigid-walls* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. (a) *EST*. (b) *RRT*-Expand. 8 runs of 3000 nodes for each planner. 84

38 Average population distribution of visibility regions in models produced by the roadmap-based *Basic-PRM* planner on the *rigid-maze* problem. 4 runs of 3000 nodes. 86

39 Average population distribution of visibility regions in models produced by different planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner. . . 87

40 Average population distribution of visibility regions in models produced by different incremental planners on the *rigid-walls* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes for each planner. 89

FIGURE Page

41 Distribution of low-visibility nodes (*visibility* $< 1/3$) in models produced by one run of different planners on the *rigid-maze* problem after global-level metrics have converged. (a) *Basic-PRM.* (b) *OBPRM.* (c) *Gauss-PRM.* 91

42 Coverage rate of incremental planners *RRT*-Expand, *EST*, *RRT*-Connect, and *RPP*when mapping two problems. (a) *rigid-walls.* (b) *rigid-maze*, *EST* is close to 0 most of the time. 93

43 Visibility regions in a model produced by *RRT*-Connect in the *rigid-walls* problem. (a) 2000-node model. (b) Low-visibility region: *visibility* $< 1/3$. (c) Medium-visibility region: $1/3 <=$ *visibility* $< 2/3$. (d) High-visibility region: $2/3 <=$ *visibility*. 94

44 Average population distribution of node types produced by different planners on the *rigid-maze* problem at the beginning of the *learning decay* stage. 95

45 Average population distribution of node types produced by different planners on the *rigid-hook* problem at the beginning of the *learning decay* stage. 96

46 Average population distribution of node types produced by different planners on the *serial-hook-5* problem at the beginning of the *learning decay* stage. 98

47 Population distribution of node types produced by *RRT*-Connect when modeling the *serial-spring-98* problem. 99

48 Evolution of global-level metrics for one instance of *RRT*-Connect on the *serial-spring-98* problem. (a) *max-diameter* and *sum-diameter*. (b) changes in *max-diameter* and *sum-diameter* (also shown, the moment when the rate of change of both diameters fall below thresholds $t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$). 100

49 Regions found in the *serial-spring-98* problem with *RRT*-Connect (at about 400 nodes when the expanding trees join together). (a) Population distribution of visibility regions. (d) Low-visibility region at 400 nodes. (c) High-visibility region at 400 nodes. 101

CHAPTER I

INTRODUCTION

A motion planner finds a sequence of motions for an object (the robot) to move from its initial state to a goal state while satisfying any constraints specified on its motions. Since the motion planning problem is considered intractable [55, 57, 13], research on heuristic approaches has flourished [5, 29, 50, 51, 2, 10, 64, 31, 8, 34, 23]. These heuristics include sampling-based approaches that have enabled us to address many important motion planning problems that were previously impractical. Instead of computing an exact representation of the planning space, the sampling-based approach samples and tests motions in the space formed by the robot configurations called configuration space (C-space) [40]. The result is an approximate model that encodes representative robot motions. This general methodology has extended its original applications in robotics to diverse areas, such as the study of protein folding in biology and chemistry [4, 61, 7, 59, 63], virtual prototyping in manufacturing and mechanical design[6, 14], and the simulation of characters for animation and games [38, 39].

Much work has been done to improve sampling-based planners, especially on heuristics to bias sampling towards regions of the space that model highly constrained robot motions. As a result, there are many sampling-based planners to choose from, but we do not know how to select among them. The sampling bias of each planner makes its performance depend on the features of the planning space. Moreover, since

This dissertation follows the style of the *IEEE Transactions on Automatic Control.*

a single problem can contain regions with vastly different features, there may not exist a simple exploration strategy that will perform well in every region. Unfortunately, we lack quantitative tools that would enable us to match planners to problems while building a motion model through the analysis of problem features and planner performance. This lack has recently led researchers to investigate mechanisms to dynamically adapt the planning strategy to the features discovered in each problem instance [11, 44, 46, 12, 26, 27, 65]. We need metrics that gather relevant information about the planning process in order to make effective decisions to adapt the planning strategy.

Metrics that have been used to compare sampling-based planners typically evaluate computational efficiency [3, 18, 32] such as the number of basic operations needed to compute the model, or the amount of information used to model the planning space, or the minimum number of samples to solve a particular set of queries. However, these metrics provide limited information that would help to dynamically adapt the planning strategy to each problem instance.

Also, some work has focused on defining properties of the planning space of problems, called C-Space, such as ϵ-goodness [28] and $(\epsilon, \alpha, \beta)$-expansiveness [25], or simply expansiveness. The planning space of a problem instance is ϵ-good if it is composed of samples that are ϵ-good, meaning that they can be connected to a set of samples that cover at least a fraction ϵ of the volume of the valid space [28] (the set of configurations that satisfy all the robot constraints). Problem instances that have ϵ-good spaces can be easily modeled and a planner has been designed specifically for such problems [25]. $(\epsilon, \alpha, \beta)$-expansiveness is a property for ϵ-good spaces that requires that every subset of the valid planning space which is a fraction α of the volume of the C-Space, can also be connected to a fraction β of the C-Space, and the larger the values of α and β, the more expansive is the space [25]. Unfortunately, these features are not practical to compute for most interesting problems and there are many important problems which do not satisfy them.

We introduce a set of novel metrics for the analysis of problem features and planners' performance during the construction of motion models. Instead of comparing the model achieved by the planner with the underlying C-Space, which would be intractable, we propose a set of metrics that provide insight into the ability of the planner to sustain its learning about problems and into the features of the models obtained. These metrics operate at multiple levels: node level, global level, and region level. At the node level, we evaluate how new samples improve the coverage and connectivity of the model. At the global level, we evaluate how new samples improve the structure of the model. At the region level, we identify groups or regions that share similar features. This way we do not measure the success of the planner in modeling the underlying C-Space, but its effectiveness in increasing its knowledge about it.

For simplicity, in the presentation of these metrics we assume motion models where potential robot configurations satisfy a binary validity test. In these models, robot configurations and potential motions are either valid or invalid. Nevertheless, extensions for other types of models, such as those that make lazy evaluations of validity or that use probability functions instead of binary validity tests, are straightforward.

We show some applications of these metrics. We identify three phases that planners go through when building C-Space models: quick learning (rapidly building a coarse model), model enhancement (refining the model), and learning decay (oversampling – most new samples do not provide additional information). We compare planners to gain insight into their strengths and weaknesses. We measure the amount of structural improvement of the models to decide whether to stop planning or to switch strategies. We propose a strategy to group samples in order to adjust sampling in different regions of the problem.

This work is the result of our continued research. In [44], we proposed a feature-sensitive motion planner, this was one of the first adaptive planners and one of our main motivations to investigate metrics to evaluate the planning process. In [46], we studied and refined two steps of the feature-sensitive motion planner: the subdivision of the problem, and the integration of partial solutions. In [45], we proposed the first set of node-level metrics and we identified the different learning stages followed by sampling-based motion planners. In [65] we developed the first global-level metrics and we applied them to decide when to finish the planning process and to improve adaptive learning. In [62], we applied the node-level and global-level metrics to study a new motion planner. In [42], we developed the first set of region-level metrics to study sampling-based planners that explore the space incrementally.

This document is organized as follows: chapter II describes the configuration space (C-Space), and defines some concepts and C-Space properties that will be used in the rest of this work, it provides an overview on sampling-based motion planners, then it describes traditional methods to evaluate planners; chapter III briefly introduces the metrics at the node, global, and region levels; chapter IV describes the strategy we followed to evaluate the metrics, the planners and problems studied, and the experimental setup; chapter V describes the node-level metrics in detail; chapter VI describes the global-level metrics in detail; chapter VII describes the region-level metrics in detail; chapter VIII shows applications and the experiments performed to evaluate the metrics at all levels; and chapter IX presents some concluding observations.

CHAPTER II

SAMPLING-BASED MOTION PLANNING

Sampling-based planners address the intractability of the motion planning problem [55, 57, 13] by using sampling to build an approximate model of potential robot motions. These planners are not intended to provide a complete solution, that is, to find a path or report that none exists. Nevertheless, some sampling-based planners have been proved to be probabilistically resolution complete [5], meaning that the probability of finding a path, if one exists, increases with the effort spent searching for it.

The motion model produced by sampling-based planners is usually represented as a graph whose vertices represent feasible configurations and whose edges represent potential transitions or motions between the corresponding configurations. Without loss of generality, the techniques discussed here assume that configurations and potential motions are either valid or invalid. Nevertheless, these techniques can be extended to other types of models, such as those that make lazy evaluations of motion validity [9, 48], or those that evaluate potential motions with probability functions instead of binary validity tests [4].

First, this chapter describes an important abstraction for sampling-based motion planning called the configuration space (C-Space) and some of its properties that will be used in further definitions. Second, it provides an overview of different sampling-based motion planners.

A. Configuration Space

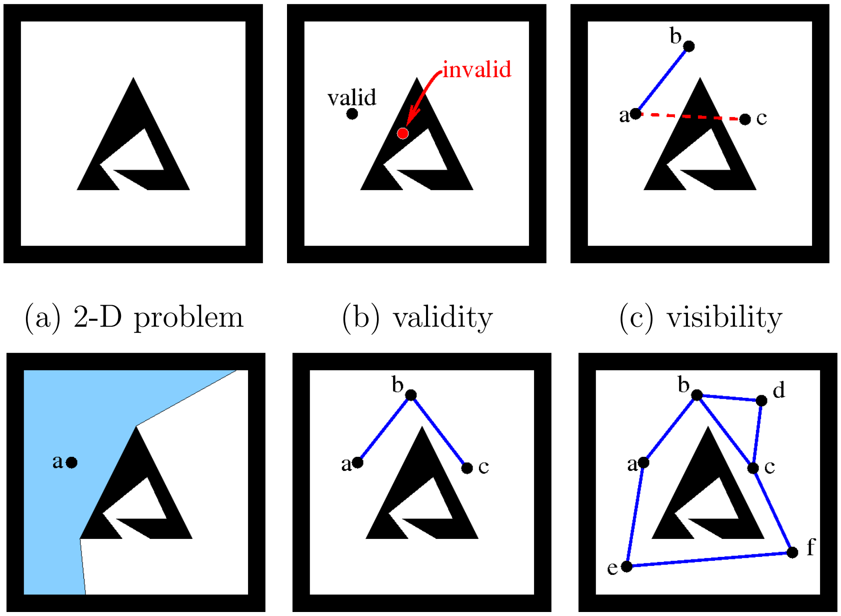
Configuration space (C-Space) is an abstraction that allows us to apply the same basic planning framework for every kind of robot [40]. A configuration q encodes the placement of every component of the robot as a point $q = (x_1, ..., x_d)$. Each of the d parameters, or degrees of freedom ($DOFs$), in q corresponds to an independent motion ability of the robot (e.g., base translations and rotations, link angles and displacements, etc.). The set of all the robot configurations in the given problem instance form the d-dimensional C-Space $\mathcal{C}$ [40].

Thus, the motion planning problem consists of finding a valid trajectory in the C-Space between the start and goal configurations. Unfortunately, any complete planner, one that finds a path or reports that no such path exists, would need to compute the entire C-Space. Indeed, there is strong evidence that this will require exponential time in the number of *DOFs* of the robot [55, 57, 13].

Let us define some properties related to configurations q and q' that will be used in the discussion of sampling-based planners and of the properties of the C-Space in the following sections. Figure 1 illustrates these definitions in a problem for a point robot moving in 2-dimensional space.

1. Validity

The boolean function $valid(q)$ is true if q satisfies the constraints of the robot and problem instance, and false otherwise. The subset of valid configurations in C-Space is the C-Free space $\mathcal{F}$, and the subset of invalid configurations in C-Space is the C-Obstacle space $\mathcal{O}$ such that $\mathcal{C} = \mathcal{F} \bigcup \mathcal{O}$. Figure 1(b) shows a valid configuration in $\mathcal{F}$ and an invalid configuration in $\mathcal{O}$.



(a) 2-D problem (b) validity (c) visibility

(d) visible region (e) connectability (f) homotopy

Fig. 1. Validity, visibility, visible region, connectability, and homotopy in (a) the C-Space of a point robot moving in the plane. (b) A valid configuration in $\mathcal{F}$ and an invalid configuration in $\mathcal{O}$. (c) Visibility with straight-line local planner: a can see b but it can not see c. (d) The visible region for a, with straight-line local planner, in light blue. (e) a and c are connectable ($\tau(a,c) = \{a,b,c\}$). (f) Two homotopy classes between a and c: $\tau_1 = \{a,b,c\}$ and $\tau_2 = \{a,b,d,c\}$ can be continuously transformed into each other; 2. $\tau_3 = \{a,e,f,c\}$ cannot be transformed into τ_1 nor τ_2.

2. Visibility

The boolean function $visible(q,q')$ is true if some specified method, typically called a *local planner*, can produce a *path* τ consisting of a continuous sequence of adjacent (at a required resolution) configurations $\tau = (q_1, q_2, \ldots, q_n)$ where $q_1 = q$, $q_n = q'$, and $valid(q_i) = true \ \forall q_i \in \tau$. For example, the straight-line local planner will decide that q can see q' if the straight line segment $\overline{q,q'}$ is composed only of valid configurations as shown in Figure 1(c) where configuration a can see configuration b, but it can not see configuration c. Note that symmetry is determined by the local planner.

3. Visibility or Covered Region

Based on [49], we define the *visibility* or *covered* region for a valid configuration q as the subset of configurations $Visibility(q) = \{q' \in \mathcal{F} | valid(q') = true, visible(q, q') = true\}$. Figure 1(d) shows the visibility region for one configuration using the straight-line local planner.

4. Connectability

Based on [25], we define the boolean function $connectable(q, q')$ as true if there exists a *trajectory* consisting of a sequence of configurations $\tau(q, q') = (q_1, q_2, \ldots, q_n)$ where $q_1 = q$, $q_n = q'$, and $visible(q_i, q_{i+1}) = true$, $1 \leq i < n$. For example, configurations a and c in Figure 1(e) are connectable through $\tau(a, c) = (a, b, c)$. Note that connectability does not imply visibility.

5. Homotopy

A *homotopy class* $\mathcal{H}$ is a set of *similar* paths between a pair of connectable configurations. Based on [17], we define a *homotopy class* between q and q' such that $connectable(q, q') = true$ as the set of paths $\mathcal{H}_{q,q'} = \{\tau_1(q, q'), \tau_2(q, q'), \ldots, \tau_h(q, q')\}$ where for any two paths $\tau_i(q, q'), \tau_j(q, q') \in \mathcal{H}_{q,q'}$ there exists a continuous deformation that transforms $\tau_i(q, q')$ into $\tau_j(q, q')$ such that the validity of all the configurations in the deformation remains the same. Thus, for any pair of connectable configurations there may be more than one homotopy class. In Figure 1(f), there are two homotopy classes between configurations a and c: the pathways $\tau_1(a, c) = (a, b, c)$ and $\tau_2(a, c) = (a, b, d, c)$ can be transformed into each other without passing through obstacles, so they belong to one homotopy class, whereas $\tau_3(a, c) = (a, e, f, c)$ cannot be transformed into $\tau_1(a, c)$ nor $\tau_2(a, c)$, so it belongs to another homotopy class.

B. Sampling-Based Motion Planning Techniques

Since it is intractable to explicitly compute the C-Space in order to completely solve a motion planning instance, we can use randomized algorithms to find approximate solutions [47]. Sampling-Based Motion Planners approximate the connectivity of the valid C-Space by sampling configurations and searching for valid trajectories between them. In particular, we can often determine whether a configuration is valid or not quite efficiently (e.g., by performing a collision detection test in the workspace), and this is also the basic operation in visibility tests.

Sampling-Based Motion Planners trade completeness for efficiency by pursuing the weaker condition of probabilistic resolution completeness [5]. Under this condition, the probability of finding a path increases with the effort spent searching for it. However, the planner is not guaranteed to terminate when there is no path. In fact, the density, complexity, and distribution of the obstacles reduces the probability of sampling configurations or making connections in some regions of the C-Space. This is the case of the narrow passage problem where relatively open areas are connected by a small-volume passage that runs through a dense and complex area. Many sampling-based planners have been developed with different bias strategies intended to sample more configurations that are likely to lie inside narrow passages.

In general, sampling-based planners produce motion models that consist of a subset V of configurations selected from the C-Free $\mathcal{F}$ and a subset E of pairs of configurations selected among all the visible configuration pairs $V \times V$ to form a motion graph $M = (V, E)$.

The metrics introduced in this work assume that $valid(q) = true$, $\forall q \in V$ and that $visible(q, q') = true$, $\forall (q, q') \in E$, but these concepts can be extended to cover other cases (such as the lazy methods described below) as explained briefly in chapter

III.

We can classify sampling-based planners based on their strategy for exploring the C-Space. Roadmap-based planners make a global exploration of the space. Incrementally-exploring planners start from one or two configurations and explore the space incrementally from these initial configurations. Adaptive planners dynamically adjust the planning strategy or parameters based on features discovered while sampling the space. A brief discussion of these classes of sampling-based planners follows.

1. Roadmap-Based Planners

Roadmap-based planners [29, 50, 51], or Probabilistic Roadmap Methods (*PRMs*), build a C-Space model in two main steps: node generation and node connection. Node generation consists of randomly sampling configurations, testing for visibility between them, and keeping the valid ones as roadmap vertices. Edge generation consists of selecting pairs of nearby configurations, testing them with a local planner, and keeping the valid transitions as roadmap edges. The resulting roadmap represents the connectivity of the C-Space and can be queried as many times as needed. Additional steps can be performed to refine the roadmap, such as connection attempts focused on joining roadmap components using incrementally-exploring methods [43, 1].

The most researched aspect of roadmap-based planners is the node generation. The main focus has been to improve the chances to produce samples inside the narrow passages. Some of the node generation techniques include the following:

- *Basic-PRM* [29], the original Probabilistic Roadmap Method, samples configurations in the C-Space with a uniform distribution retaining those that are valid.

- *OBPRM* [2], the obstacle-based *PRM* generates invalid configurations and pushes them in random directions to generate valid configurations around the boundaries of C-Obstacles.

- *Gauss-PRM* [10] performs uniform sampling to produce valid samples within distance d from invalid samples. Valid samples have a gaussian distribution around obstacle boundaries.

- *Bridge-Test* [22] performs uniform sampling to produce valid configurations, or bridges, between pairs of invalid samples separated a distance d.

- *MAPRM* [64] performs uniform sampling and retracts every valid and invalid configuration towards the medial axis of the free space. Although MAPRM can be implemented practically only for rigid bodies in three-dimensional space, an approximate version has performed well for high-dof problems [37].

- The Visibility Roadmap [31] performs uniform sampling and adds each valid new sample to the roadmap only when the new sample is not visible from any other node in the roadmap or when it is visible from at least two nodes in different components of the roadmap. The number of nodes in the roadmap is minimized, but as new nodes are added, the cost of connection evaluations increases.

It is worth noting that some of these techniques achieve their bias by *selecting* uniformly sampled configurations that satisfy the properties of interest (e.g., *Gauss-PRM* and the visibility roadmap) while others *manipulate* configurations in order to satisfy the properties of interest (e.g., *OBPRM* and *MAPRM*).

The selection of pairs of nodes that will be tested for connection allows us to limit the number of costly feasibility tests. Among the pair selection strategies we have the following:

- *k-closest* [29] is the most common strategy. For each node, it selects the k-closest nodes to attempt connections.

- *Components* [2] selects pairs of nodes in different roadmap components.

In order to select pairs of nodes, we may need to evaluate distances between configurations. Common distance metrics include Euclidean, Minkowski, Manhattan, and scaled Euclidean [3, 52].

The test for visibility between a pair of nodes (v, v') is performed by the local planner, with successful tests resulting in edges. Edge weights are usually a distance estimation based on the number of steps interpolated by the local planner. Perhaps the most commonly used local planner is the *straight-line* planner which examines the configurations along a straight line that is interpolated between v and v' at a given resolution. There are other local planners, such as those based on the A* search or the rotate-at-s local planner [3] which separates translational and rotational motions into piecewise straight lines.

Some roadmap-based techniques defer some or all validations for configurations and potential motions until after vertex and edge generation.

- Lazy PRM [9] tests nodes for validity and it initially places edges for all selected pairs without validating them for visibility. Edges are tested only at query time when the shortest path in the roadmap is found. Invalid edges are removed from the roadmap and the search is repeated until a valid path is found or until the start and the goal lie in different roadmap components.

- Fuzzy PRM [48] tests nodes, then it weights the edges with an estimation of the probability of the pair of nodes to be visible. When a query is made, the path with the highest probability of being valid is selected and its edges are validated and updated as needed. This process may change the probability of the edges to represent valid connections.

- C-PRM [60] allows for partial or full validation of nodes and edges. Then, particular constraints can be defined at the query time, such as clearance, smoothness, potential, or ranges in the *DOFs*. Nodes and edges that do not satisfy the required constraints are removed from the roadmap.

2. Incrementally-Exploring Planners

Incrementally-Exploring planners explore the space to find new samples following some strategy to expand the model [5, 8, 41, 35, 23]. They usually root a tree at each valid configuration of some set (typically the start and goal) and then they expand the trees in increments. At each increment, they select a *growth site* among nodes in the model and they explore the vicinity of each growth site to expand the model. The incremental exploration of the space makes some incremental planners particularly well-suited for problems with differential constraints (kino-dynamic motion planning). Among the incremental planners we have the following:

- *RPP* [5], the Randomized Path Planner is the earliest sampling-based planner. It selects the vertex with the smallest value for a potential function or a randomly picked ancestor if the selected vertex is a local minima, then it explores in a random walk towards smaller potentials until reaching the goal or finding a local minimum from where it performs some Brownian motions to try to escape. The potential function used is based on an estimation of the distance to the goal.

- The Ariadne's Clew Algorithm [8, 41] selects a trajectory that minimizes the distance between the goal and the last valid configuration in the trajectory, then it spreads new nodes that are reachable from the nodes in the trajectory. The new nodes are selected such that they maximize the distance between every node and the last node in the trajectory. Selection and exploration are performed through genetic algorithms.

- *RRT* [35, 30], the Rapidly-Exploring Random Tree, initially had two versions: *RRT*-Expand [35] selects as growth sites the nearest node to a random configuration x, then it explores by walking in the direction of x a maximum distance d; *RRT*-Connect [30] selects nodes in an *RRT*-Expand way from one of two trees (one rooted at the start and another rooted at the goal), then it explores in an *RRT*-Expand way and it tries connections to the other tree. Trees alternate turns for node selection. *RRTs* iteratively break Voronoi regions of the explored areas into smaller Voronoi regions. They have been applied for single-query problems [30] and in kino-dynamic motion planning [34, 36].

- *EST* [23], the Expansive-Spaces Tree, favors selection of isolated nodes, then it explores by randomly sampling in the vicinity of isolated areas. This approach has been extended to kino-dynamic motion planning [24].

- Ray Shooting [21] selects a random node from either of two trees (one rooted at the start and the other rooted at the goal), then explores towards the other tree by shooting a random ray that “bounces” in a random direction every time it hits an obstacle while adding “bouncing” nodes to the tree. Variants of this approach have been tried to refine roadmaps [29, 43].

3. Adaptive Planners

Despite all efforts, there is no simple, general sampling-based motion planning strategy that performs well in every problem instance. Instead, we have many sampling-based planners to choose from, but we do not know how to select among them. This has led to the development of adaptive strategies that dynamically adapt the planning strategy to the features discovered while sampling the C-Space [11, 44, 46, 12, 26, 27, 65]. In order to make decisions, adaptive planners need metrics to evaluate the performance of the planning process over time and in the different areas of the C-Space.

CHAPTER III

NEW METRICS FOR SAMPLING-BASED PLANNERS

Traditional planner evaluation based on computational efficiency is insufficient to allow us to address several challenges in motion planning. We need better criteria to choose among the many available planners and to make decisions in adaptive planning. Also, since planners make approximate models of representative motions, we need metrics that allow us to look into the types of motions modeled for each problem.

This chapter discusses the properties of C-Space that give insight into the planning process and into the quality of motion models. Then, it introduces the metrics focus of this work. Finally, it briefly discusses mechanisms to extend these metrics to models that are not based on binary validity tests.

A. Traditional Evaluation of Planners

Traditionally, the evaluation of planners has relied on metrics that evaluate computational efficiency (e.g., [3, 18, 32, 33]). These metrics include the number of basic operations (tipically collision detections as validity tests) and time needed to compute the model, the amount of information needed to model the planning space, the minimum number of samples to solve a particular set of queries. However, these metrics provide limited information about the planning process that would help to dynamically adapt the planning strategy to each problem instance.

In particular, evaluating the ability of the model to solve a particular set of *witness queries* between user-specified start and goal configurations in interesting problems is very common. Under this evaluation, a successful planner solves more

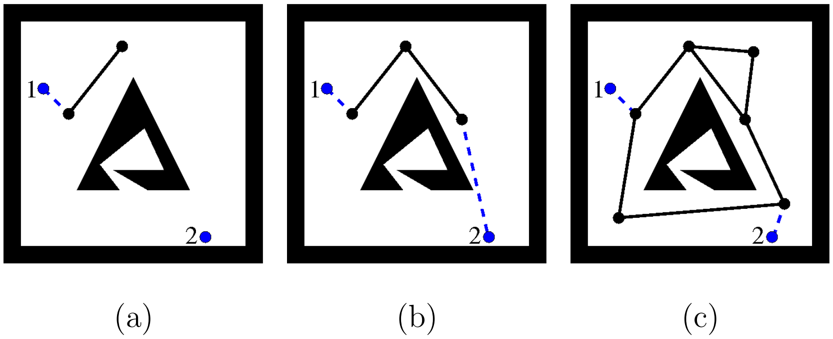


Fig. 2. Witness queries between "witness" configurations *1* and *2*. (a) Only witness *1* is connected to the model, there has not been found any path between witnesses. (b) Both witnesses are connected to the model, one path between them has been found. (c) Both witnesses are connected to the model, two paths between them have been found. The witness evaluation does not make a distinction between case (b), with only one pathway between witnesses, and case (c), with two pathways between witnesses.

witness queries as shown in Figure 2. Unfortunately, this is not necessarily a good evaluation metric because it cannot evaluate paths in different homotopy classes, it can only tell us if the model can solve the queries between a specific set of configurations, and it cannot be automatically adapted to different problems. Moreover, *witness* queries depend heavily on the understanding of the problem by the user to place the *witness* configurations. For example, if configuration *2* in Figure 2(b) were slightly to the left, the evaluation would produce very different results.

As more planners appear, we need better criteria to choose among them. Research on the qualitative aspects of the models and C-Space provides us with insight into the ability of planners to solve different kinds of problems. Reachability analysis evaluates different *PRM*-based planners by exhaustively comparing the coverage they achieve with the underlying C-Space [19]. We also know that a problem whose configurations can be connected to a set of samples that cover at least a fraction ϵ of the volume of the valid space is considered to be ϵ-good [28] and it can be easily modeled. In addition, ϵ-good spaces where every subset of the C-Space, a fraction α

of the volume of the C-Space, can be connected to a fraction β of the C-Space, are considered to be $(\epsilon, \alpha, \beta)$-expansive, or simply expansive. The larger the values of α and β the more expansive is the space and the easier it is to model it [25] (e.g., with the *EST* planner).

Unfortunately, reachability analysis has only been possible in simple problems with few *DOFs*. Similarly, ϵ-goodness and $(\epsilon, \alpha, \beta)$-expansiveness are not practical to compute for most interesting problems and there are many problems which do not satisfy them. As a result, the *witness* query evaluation is still the most commonly used.

B. New Node, Region, and Global Metrics for Sampling-Based Planners

Sampling-based motion planning is facing new challenges. The diversity of planners and of the features in problem instances has led researchers to develop adaptive planners that require powerful, flexible, efficient, and general evaluation methods to make effective decisions in order to build better C-Space models. Also, motion planning has extended its applications to aid in the study of motions from robotics to other disciplines (e.g., to model and understand motions of molecules in biology and biochemistry). These new applications need effective methods to evaluate the types of motions involved in the processes modeled.

Planners build models whose representation of the properties of the C-Space is reflected on their ability to solve general motion queries. The *coverage*, *connectivity*, and *topology* are properties of the C-Space that planners capture with different accuracy levels depending on their sampling bias and on the features of the motion planning instance. Computing these properties would be tantamount to the unfeasible computation of the C-Space. Nevertheless, we can define metrics that give insight into the changes of the representation of *coverage*, *connectivity*, and *topology* during model construction.

Next, we define *coverage*, *connectivity*, and *topology*, and we provide a high-level overview of the strategy introduced in this work to evaluate them.

1. Coverage

The *coverage* of a set of configurations $\mathcal{Q} = \{q_1, \ldots, q_n\}$ is the union of their visibility or covered regions:

$$Coverage(\mathcal{Q}) = \bigcup_{i=1}^{n} Visibility(q_i) \tag{3.1}$$

In an ideal representation of coverage, every valid configuration in the C-Free $\mathcal{F}$ should be visible from and can see at least one vertex in V.

2. Connectivity

A connected component $\mathcal{CC}$ of the C-Free $\mathcal{F}$ is a set of configurations:

$$CC = \{q_1, q_2, \ldots, q_k | \forall (q, q') \in CC, connectable(q, q') = true\} \tag{3.2}$$

and its coverage is the set of configurations in CC: $Coverage(CC) = \bigcup_{i=1}^{k} q_i$. Thus, the C-Free $\mathcal{F}$ consists of m components $\mathcal{F} = \bigcup_{i=1}^{m} CC_i$.

In an ideal representation of connectivity, every valid motion between configurations q and q' should have a corresponding path in M between vertices v and v' such that $visible(q, v) = visible(v', q') = true$. Good coverage is a precondition for good connectivity.

3. Topology

The set of Homotopy classes or similar paths between every pair of connectable configurations in the C-Free $\mathcal{F}$ represents all the potential motions.

In an ideal representation of topology, each component $CC \in \mathcal{F}$ should have a corresponding graph component CC_m and for every homotopy class $H \in \mathcal{F}$ there should be at least one graph path $\tau_m \in H$. Good coverage and connectivity are preconditions for good topology.

4. Estimating the Evolution of Planner Learning Ability

Instead of computing the connectivity, coverage, and topology of a C-Space model, we propose to estimate the ongoing learning achieved during its construction. This can be achieved by extracting metrics that approximate changes in coverage, connectivity, and topology of the model at multiple levels: node level, global level, and region level. These approximate metrics can be applied to any sampling-based motion planner and do not depend on the dimensionality of the problem.

- **Node-Level Metrics** — Enable the measurement of ongoing changes in coverage and connectivity due to each new node and its connections. These metrics can be applied to the analysis and comparison of the learning mechanisms of different planning strategies.

- **Global-Level Metrics** — Enable the measurement of ongoing changes to the global structure of the model that reflects its coverage, connectivity, and topology. These metrics can be applied to evaluate the global progress of sampling to decide when to switch planners or when to stop planning.

- **Region-Level Metrics** — Enable the identification of groups or regions in the model that share similar features. Metrics at this intermediate level between the node-level and the global-level can be applied to identify the important regions of a problem and to adjust sampling in each discovered region.

These metrics are intended to be used in conjunction. Each type of metric provides insight into a separate aspect of the planning process. Together, they allow us to make decisions regarding when and where to use a planner, and to compare models obtained with different sampling-based planners. The following chapters provide a more detailed discussion of these issues.

5. Extensions to Other Types of C-Space Models

In order to apply the metrics introduced here to models that are not based on binary validity tests we need to extend the definitions of *validity*, *visibility*, *visibility region*, *connectability*, and *homotopy*.

For example, in the case of models that use probability functions to evaluate configurations and potential motions [4], we can redefine them as follows: $valid(q) \in [0,1]$ becomes the probability that q is valid; $visible(q,q') \in [0,1]$ becomes the probability of the transition from q to q'; the visibility region $Visibility(q,t) = \{q' | valid(q') >= t, visible(q,q') >= t\}$, for a given threshold t; $connectable(q,q') \in [0,1]$ becomes the minimum conditional probability of going from q to q' through the edges in a path between q and q'; and, the homotopy class becomes the set of paths $\mathcal{H}_{q,q'} = \{\tau_1, \tau_2, \ldots, \tau_h\}$ between q and q' such that $connectable(q,q') > t$ and that for any two paths $\tau_i, \tau_j \in \mathcal{H}$ there is a continuous deformation that transforms τ_i into τ_j such $connectable(q,q') > t$ for the whole transformation and for all $q \in \tau_i$, $valid(q) >= t$. The new definitions are generalizations of the original definitions that are still valid for the cases described in this dissertation. The node-level, global-level, and region-level metrics described here would need to use these new definitions.

In the case of models that make lazy evaluations of motion validity [9, 48], we can also make lazy evaluations of the metrics described here.

CHAPTER IV

EVALUATION STRATEGY

We are interested in the learning process at the node, global, and region levels exhibited by different planners when modeling different problems. In particular, we are interested in the learning process followed by planners when building a C-Space model, in their ability to improve their representation of coverage, connectivity, and topology evaluated through node-level, global-level, and region-level metrics, and in the potential applications of these metrics to improve planning.

This chapter describes the problem instances and planners that we will use to illustrate the metrics that are discussed in detail in the subsequent chapters. It also provides details of the experimental setup for evaluating the metrics.

A. Planners

Throughout this work we study several roadmap-based and incrementally-exploring planners. Choosing a diverse set of planners allows us to apply and observe the metrics in different situations to assess their effectiveness in capturing the different processes followed by each planner.

Our implementation of the *PRM* framework works in an incremental fashion. Every new node in the roadmap is tried for connections to the k-closest nodes already in the roadmap (with $k = 20$, unless otherwise noted) using the straight-line local planner with binary search (unless otherwise noted). The distance metrics employed was scaled Euclidean with 50% weight for translational *DOFs* and 50% weight for rotational *DOFs* (unless otherwise noted). The resolution for the local planner was automatically computed for each problem based on its bounding box. One node was generated per iteration (for a maximum of ten attempts). We study the following node generation strategies for roadmap-based *PRM* planners that are described in Section II.B.1: *Basic-PRM* [29], *OBPRM* [2], *Gauss-PRM* [10], *MAPRM* [64], and *Bridge-Test* [22].

The set of roadmap-based planners evaluated represents bias mechanisms that allow us to observe diverse cases. The baseline among *PRMs* is *Basic-PRM* with its uniform sampling. *OBPRM* and *Gauss-PRM* have the same goal of sampling more densely around obstacles, but they do it differently, *OBPRM* by manipulating samples and *Gauss-PRM* by filtering them. Our implementation of *OBPRM* generates a random configuration, if it lies in C-Free it pushes it towards a C-Obstacle, otherwise it pushes it towards C-Free. *MAPRM* and *Bridge-Test* have the same goal of sampling between obstacles, but they also do it differently, *MAPRM* by manipulating samples and *Bridge-Test* by filtering them. Our implementation of *MAPRM* is based on an approximation of the clearance and penetration depth to set the retraction direction of configurations towards the medial axis of C-Free.

We implemented a general framework for incremental planning as described in [42]: a growth site is selected in each iteration to explore the surrounding area according to the rules of each method. We study the following incrementally-exploring planners that are described in Section II.B.2: *RPP* [5], *RRT*-Expand [35], *EST* [23], and *RRT*-Connect [30].

The set of incremental planners evaluated have exploration rules that allow us to observe diverse cases. The baseline among incremental planners is *RPP* the first sampling-based planner, which biases the exploration towards the goal with a simple strategy. *RRT*-Expand and *EST* have the same goal of expanding the volume explored by the planner. *RRT*-Connect and *RPP* are both goal-biased and are applicable to single-query problems.

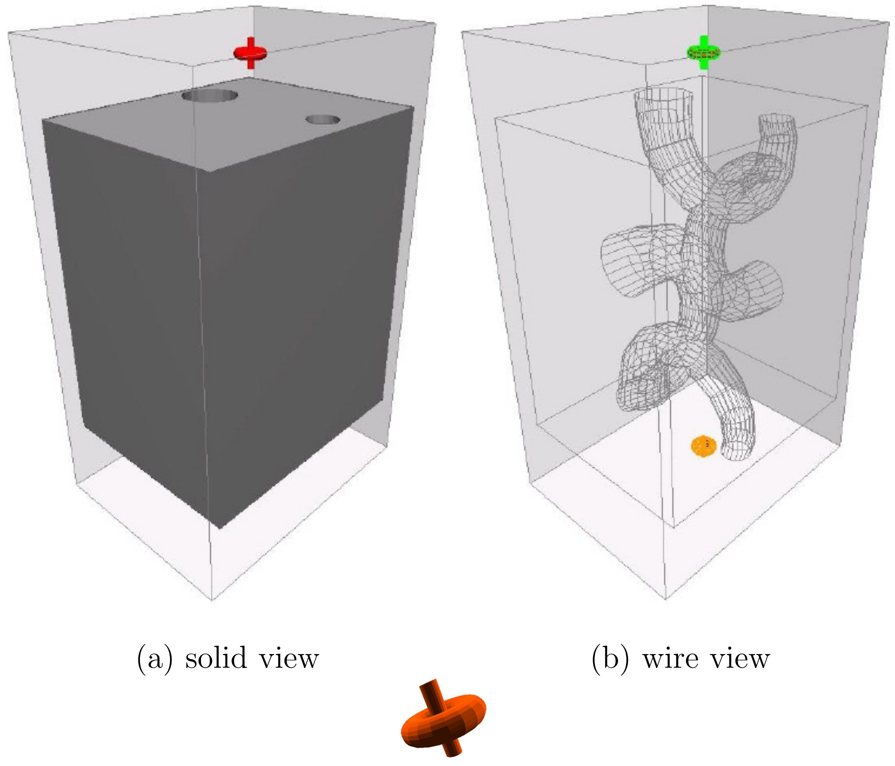
B. Motion Planning Problems

Throughout this work we apply planners and evaluate metrics for several instances of the motion planning problem. These are a diverse set of problems with different densities of C-Obstacles, open spaces and narrow passages. The motion planning instances evaluated are the following:

- The *rigid-maze* problem (Figure 3), has a 6-*DOF* rigid-body robot that should pass through a series of tunnels with some dead-ends from the top to the bottom. This problem is interesting because its C-Space resembles the workspace, it has two clear free areas, the tunnels form a long and narrow passage with dead ends, and the obstacle occupies the majority of the planning space. Translational-*DOF* ranges are: x [-8,7], y [-16.5,16.5], and z [-9,11]. Rotational *DOFs* are not bounded.

- The *rigid-windows* problem (Figure 4) has a 3-*DOF* translational rigid-body cube robot that should pass through any of the four windows in the wall that splits the environment into two halves. This problem is interesting because it has four different pathways from one side to the other and finding one is not enough to achieve the best model. From left to right, the first window is 1.5 times as long as the robot, the second window is 1.75 times as long as the robot, the third window is 2.5 times as long as the robot, and the fourth window is 3.5 times as long as the robot. The width of the wall is the same as the length of the robot. Translational-*DOF* ranges are: x [-10,6], y [-0.5,0.5], and z [-2.0, 2.0]. Rotational *DOFs* are not bounded.

- The *rigid-hook* problem (Figure 5) has a 6-*DOF* rigid-body hook robot that should pass through the narrow openings in the two walls that divide the envi-



(a) solid view

(b) wire view

(c) robot close-up

Fig. 3. Problem: *rigid-maze*. (a) solid view. (b) wire-view shows the internal tunnels. (c) close-up view of the robot.

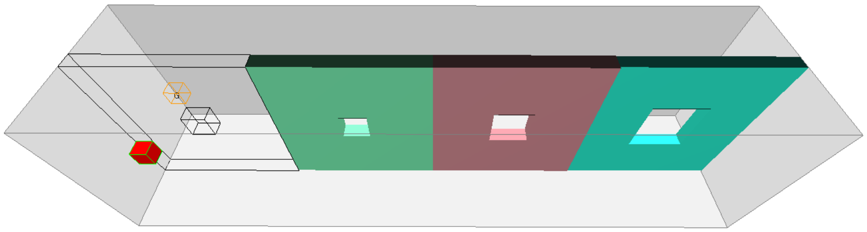


Fig. 4. Problem: *rigid-windows*. The robot has 3 translational degrees of freedom. Four pathways of different sizes allow the robot to cross from the front to the back. The start and goal configurations are at each side of the leftmost wall.

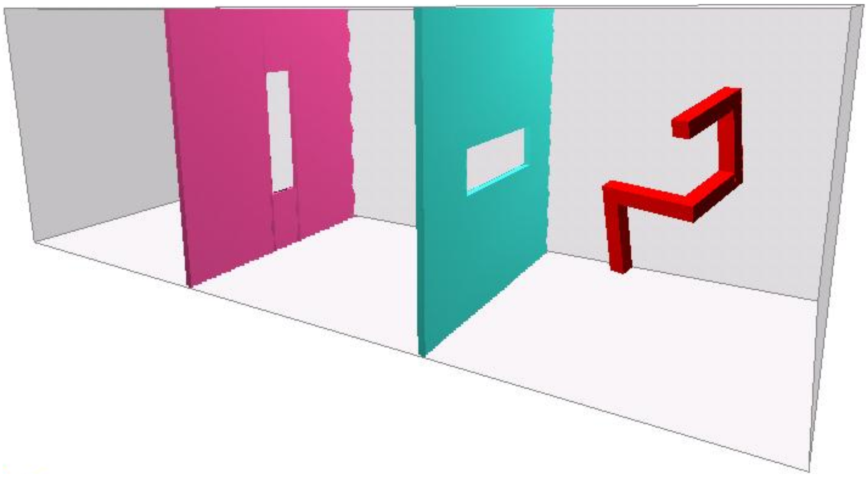


Fig. 5. Problem: *rigid-hook*. In order to get through the passages, the 6-*DOF* robot needs to perform translations and rotations.

ronment into three chambers from one side to the other side of the environment. This is an interesting and difficult problem that requires simultaneous translational and rotational motions. Translational-*DOF* ranges are: x [-100,100], y [-100,100], and z [-400,200]. Rotational *DOFs* are not bounded.

- The *rigid-walls* problem (Figure 6) has a 6-*DOF* rigid-body box robot that should pass through the small openings (slightly larger than the robot) in the walls that divide the environment into five chambers from one side to the other side. This problem has a C-Space that is similar to its workspace, with four narrow passages and open spaces in between. Incremental planners increase their coverage in stages as they find their way through the passages. Translational-*DOF* ranges are: x [0,4], y [0,4], and z [-5,14]. Rotational *DOFs* are not bounded.

- The *serial-hook-5* problem (Figure 7) has five links that form a ten-*DOF* articulated robot that should pass through the opening in the wall that divides the

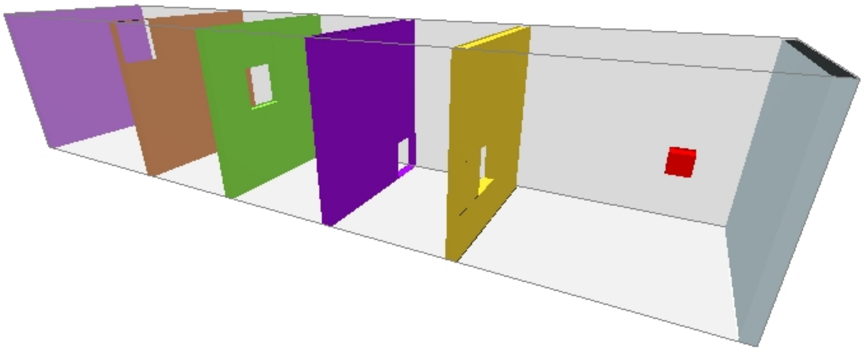


Fig. 6. Problem: *rigid-walls*. Incremental planners find their in increments as they find their way through narrow passages.

environment into two chambers from one side to the other side of the environment. This is an articulated version of the *rigid-hook* problem that has a higher number of *DOFs* than the other problems. Translational-*DOF* ranges are: x [-100,100], y [-100,100], and z [-400,200]. Rotational *DOFs* are not bounded.

- The *serial-spring-98* problem (Figure 8) has ninety eight links that form a 103-*DOF* articulated robot whose start and goal configurations resemble springs of different widths. The robot should pass above a wall that divides the environment into two areas folding and unfolding. The high number of *DOFs* makes this problem much harder than any of the other problems discussed. Translational-*DOF* ranges are: x [-500,500], y [-500,500], and z [-500,500]. Rotational *DOFs* are not bounded.

C. Experimental Setup

All the experiments were performed in individual processors of the IBM HPC cluster 1600 of Texas A&M University. This cluster runs the 64-bit version of AIX (version 5.3), and it has 40 p5-575 nodes, each with 16 Power5+ processors at 1.9 GHz and

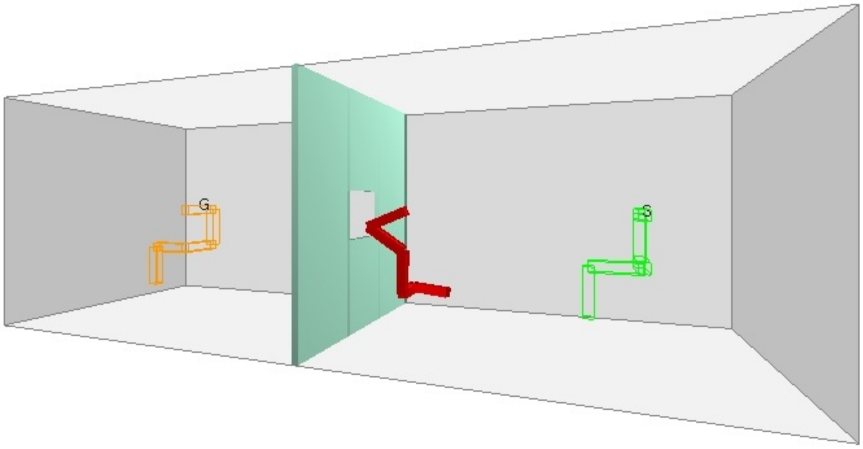


Fig. 7. Problem: *serial-hook-5*. The 10-*DOF* robot can fold and unfold to get through the opening that divides the environment. This is a variation of the *rigid-hook* problem.

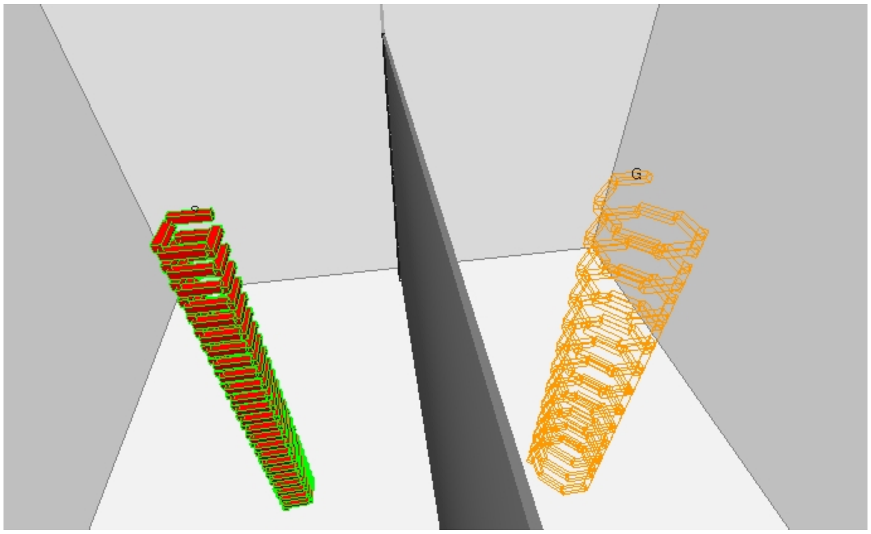


Fig. 8. Problem: *serial-spring-98*. The 103-*DOF* robot folds and unfolds to get above the wall that divides the environment.

32 GBytes of DDR2 DRAM in a shared-memory configuration (SMP). This resource was used for over 600 hours including trial runs and the final experiments shown in this work.

All the techniques were implemented in C++ within the Parasol Lab Motion Planning Library. Validity was evaluated using the RAPID collision detection package [20].

Executions are split into subsets, or bins, of n consecutive iterations to gather statistics and evaluate performance. Times were tracked individually for model generation (node generation and connection in roadmap-based planners, and expansion in incrementally-exploring planners), metrics computation (node-level, global-level, and region-level), witness-query evaluation, and input/output (to store C-Space models).

As mentioned previously, we evaluated the following roadmap-based planners: *Basic-PRM*, *OBPRM*, *Gauss-PRM*, *MAPRM*, and *Bridge-Test*. Also, we evaluated the following incrementally-exploring planners: *RRT*-Expand, *EST*, *RRT*-Connect, and *RPP*. All these planners are described in detail in chapter II. For each planner, we used the parameters that yielded the best performance for each method and problem in preliminary experiments. We applied each planner to an instance of the problem eight times using different seeds for the random number generator, and then we aggregated statistics of the eight runs.

In Section V.A, we evaluate the error with different approximation levels that can be specified in some of the node-level metrics. We executed all the planners in all the problems up to sixteen different times. We noted that even with four different runs we obtained standard deviations smaller than 5%.

In Section VI.A, we illustrate the global-level metrics with the application of *Basic-PRM* on the simple *rigid-windows* problem which can be solved through multiple pathways. The planner was executed ten times, and we showed one of the executions that illustrates better the identification of multiple pathways. Here, the node-level metrics were computed every bin with an expansion threshold of 0.5 and neighbor-probability test for all connections, the global-level metrics were computed every 10 bins, witness queries were performed every bin. Since the structural changes happen at the initial iterations, we only show metrics for the first 200 nodes.

In chapters V, VI, and VII we show the application of the different metrics to evaluate the planners and problems described above. In the roadmap-based *Gauss-PRM* and *Bridge-Test* planners, we used the same value for the d parameter for each problem: $d = 10$ in the *rigid-hook* problem and in the *rigid-maze* problem; $d = 2.16$ in the *serial-hook-5* problem, and $d = 0.2$ in the *rigid-walls* problem. In the incrementally-exploring RRT-Expand and RRT-Connect planners, we used the same value for the q parameter for each problem: $q = 0.06$ in the *rigid-walls* problem, and $q = 0.04$ in the *rigid-hook* and in the *serial-hook-5* problems. In the incrementally-exploring EST planner we used the following parameters: neighborhood radius $q = 0.04$ and number of neighbors to evaluate density $k = 5$ in the *rigid-walls* problem; and $q = 0.08$ and $k = 5$ in the *rigid-hook* and *serial-hook-5* problems. In the incrementally-exploring RPP planner we used the following parameters: step size $q = .05$, maximum escape trials $t = 20$ in the *rigid-walls* problem; and $q = 0.02$, and $t = 10$ in the *rigid-hook* and *serial-hook-5* problems. Each planner was applied 16 times to each problem to gather the metrics. The metrics were computed every 20 bins.

CHAPTER V

NODE-LEVEL METRICS

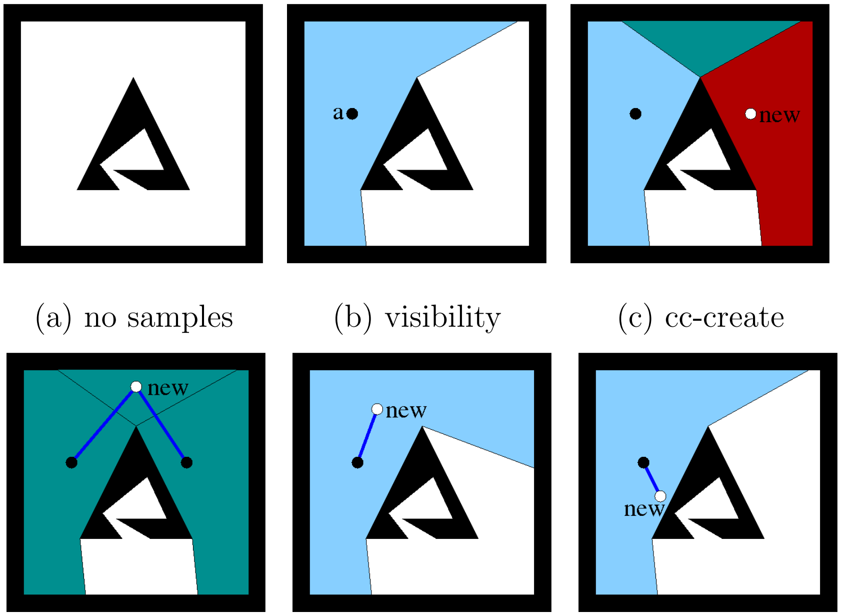
Metrics at the node level allow the estimation of changes in coverage and connectivity of the model with the addition of each new node. We can estimate the type and quantity of improvements in the model due to new nodes based on structural changes of the model and on an estimation of the local visibility of the new node. Node-level metrics provide local information around the nodes in the model, but they do not provide information about global coverage, distribution of samples or topology.

A. Type and Amount of Improvement Produced by a New Node

Given a model M, a planner adds a valid sampled configuration v and a selected subset of its valid connections producing the model M'. This operation changes the connectivity and coverage of the original model M in exactly one of the following ways:

1. **cc-create** — v lies outside the coverage region of all the components in M as seen in Figure 9(c). A new component CC with v as its only node is created. The coverage of M increases by the coverage of v and the connectivity and topology improve due to the new component.

2. **cc-merge** — v lies inside the overlapping coverage region of more than one component of M as seen in Figure 9(d). As a consequence, the components and their coverage regions merge, reducing the number of components. The coverage of M increases only by the coverage of v and its connectivity and topology improve due to the new pathways found.



(a) no samples (b) visibility (c) cc-create

(d) cc-merge (e) cc-expand (f) cc-oversample

Fig. 9. Classification of new nodes when modeling the C-Space of a point robot moving in the plane shown in (a). (b) The first sample in the model with its visibility region. (c) A new sample lying outside the visibility region of any other sample creates another component with its own visibility region. (d) A new sample lying in the overlap of the visibility region of two components allows to merge them. (e) A new sample lying inside the visibility region of one component expanding its visibility: cc-expand. (f) A new sample lying inside the visibility region of one component without changing its visibility: cc-oversample.

3. **cc-expand** — v lies inside the coverage of exactly one component of M and it increases the coverage of the component as seen in Figure 9(e). The coverage of M increases but the connectivity of M remains constant. The amount of model improvement in this case is the increase in coverage.

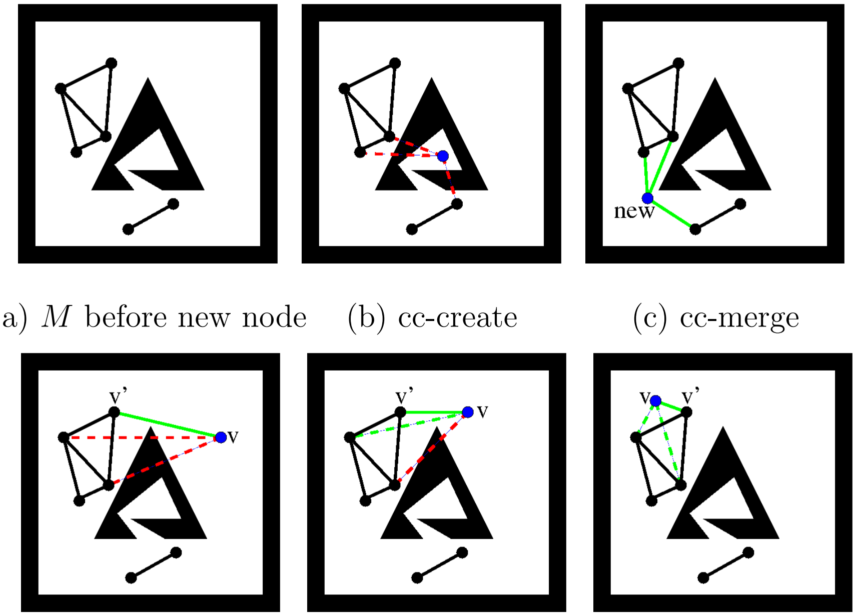
4. **cc-oversample** — v falls inside the coverage of exactly one component CC in M as seen in Figure 9(f). The coverage and connectivity of M remain constant.

Three of these cases improve the representation of the coverage and/or the connectivity of the model: *cc-create*, *cc-merge*, and *cc-expand*. The fourth case, *cc-*

oversample, does not represent an improvement of the model as will be shown in our experimental studies. *cc-expand* and *cc-oversample* nodes occur very frequently while *cc-create* and *cc-merge* nodes are much less frequent. In particular, roadmap-based planners may produce *cc-create* nodes in hard to reach areas and *cc-merge* nodes when paths between disconnected components are found. Also, incremental planners only produce *cc-create* nodes when starting a tree and *cc-merge* nodes when connecting trees or finding a connection to the goal.

In order to accurately classify the nodes, we need to estimate their visibility region. This is unfeasible to compute because it would be as hard as computing the C-Space around the node. However, as we will see, reasonable approximations can be computed efficiently using only local information.

In this work we discuss only one of the many ways in which node classification can be implemented. A node that increases the number of roadmap components is a *cc-create* node as shown in Figure 10(b). A node that causes a reduction in the number of components in the roadmap is a *cc-merge* node as shown in Figure 10(c). In order to distinguish *cc-expand* and *cc-oversample* nodes, we compute the *expansion ratio* $E(v)$ for the node v as follows: a node v that connects to a node v' in the roadmap, but cannot be connected to a percentage $E_{v,v'}$ of v''s neighbors produces an expansion in the proportion of $E_{v,v'}$ as shown in Figure 10(d). We call $E_{v,v'}$ the amount of expansion of v with respect to v'. The expansion ratio $E(v)$ for the node v is the maximum of the expansions produced for all the nodes v' connected from v. A threshold E_t is used to distinguish between *cc-expand* and *cc-oversample* nodes so that a node is *cc-expand* if $E(v) >= E_t$, otherwise it is *cc-oversample*. Most nodes of our preliminary experiments showed an $E(v)$ either close to 0.0 or close to 1.0, so we decided to use a $E_t = 0.5$ in the rest of this work. This way the node v in Figs. 10 (d) and (e) are *cc-expand* while the node v in (f) is *cc-oversample*.



(a) M before new node (b) cc-create (c) cc-merge

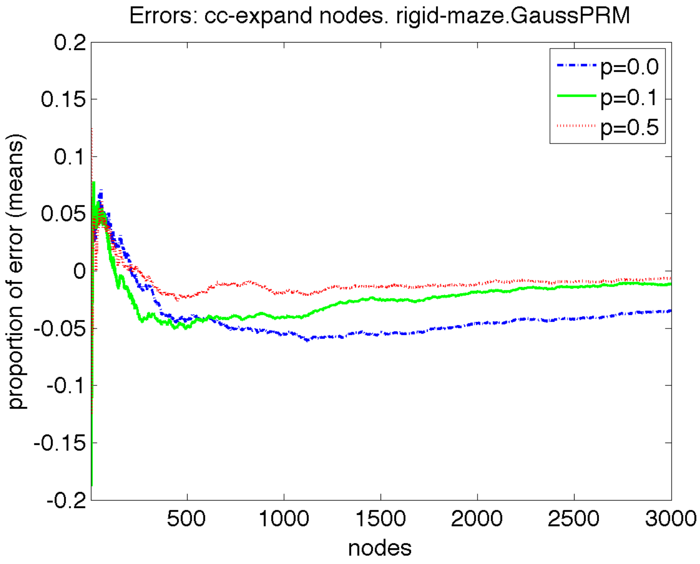
(d) 100% expansion (e) 50% expansion (f) 0% expansion

Fig. 10. An implementation of classification of new nodes. (a) State of M before a new node is added. (b) New node increases the number of components: *cc-create*. (c) New node reduces the number of components: *cc-merge*. (d) New node v cannot connect to any neighbor of v': *cc-expand*. (e) New node v cannot connect to 50% of the neighbors of v': *cc-expand* for $E_t = 0.5$. (f) New node v can connect to all the neighbors of v': *cc-oversample*.

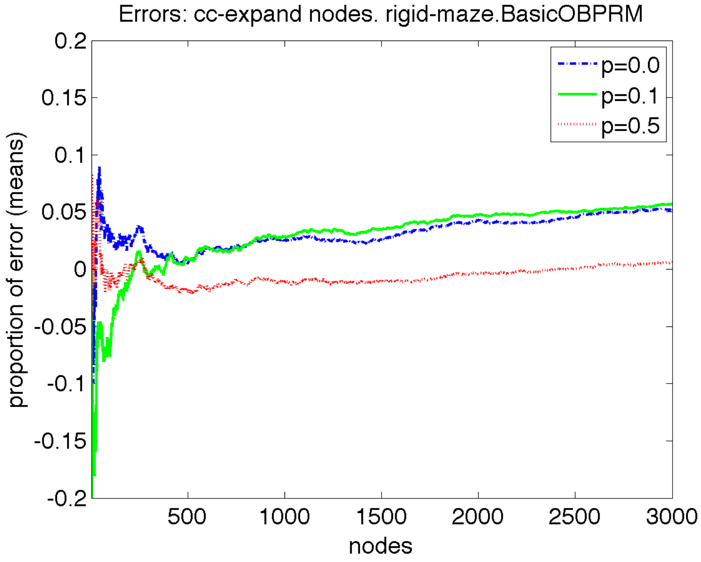
The computation of the expansion ratio $E_{v,v'}$ to distinguish between *cc-expand* nodes and *cc-oversample* incorporates the successes and failures in the connection attempts from the new node v to the neighbors of the connecting node v' tried by the planner. However, the planner may have not selected some of the neighbors of v' for connection from v. We try these additional connections with probability p. This way, when $p = 1$ we attempt to connect v to all the neighbors of v', resulting in the best approximation, and when $p = 0$ we only use the connections tried by the planner, resulting in reduced accuracy. In addition, we stop trying additional connections when the number of connections achieved is larger than E_t times the number of neighbors of v' and larger than E_t times the number of connections attempted because at that moment we have enough information to decide whether $E_{v,v'} >= E_t$.

Node classification in roadmap-based planners can be done even with low values of p to get correct results. Figure 11 shows the absolute error in the classification of *cc-expand* nodes in multiple runs of roadmap-based planners on the *rigid-maze* problem for several values of p with respect to a full test $p = 1.0$. We show results for *Gauss-PRM* and *OBPRM*, and we also evaluated *Basic-PRM* (with similar trends as *Gauss-PRM* but with smaller errors), *MAPRM* (with similar trends as *Gauss-PRM* but with smaller errors) and in *Bridge-Test* (with similar trends as *OBPRM*). The highest error happened in *OBPRM* and *Bridge-Test* whose nodes had much fewer neighbors than *Basic-PRM*, *Gauss-PRM*, and *MAPRM* which were heavily connected. Nevertheless, the quality of the evaluation remains reasonable even for *OBPRM* and *Bridge-Test*.

The overhead of the expansion test in roadmap-based planners is reasonable for low values of p. Table II shows the average computation times for modeling the problems, and the overhead for node-level metrics using different values of p. We notice that the overhead in *Basic-PRM*, *Gauss-PRM*, and *MAPRM* was considerably



(a)



(b)

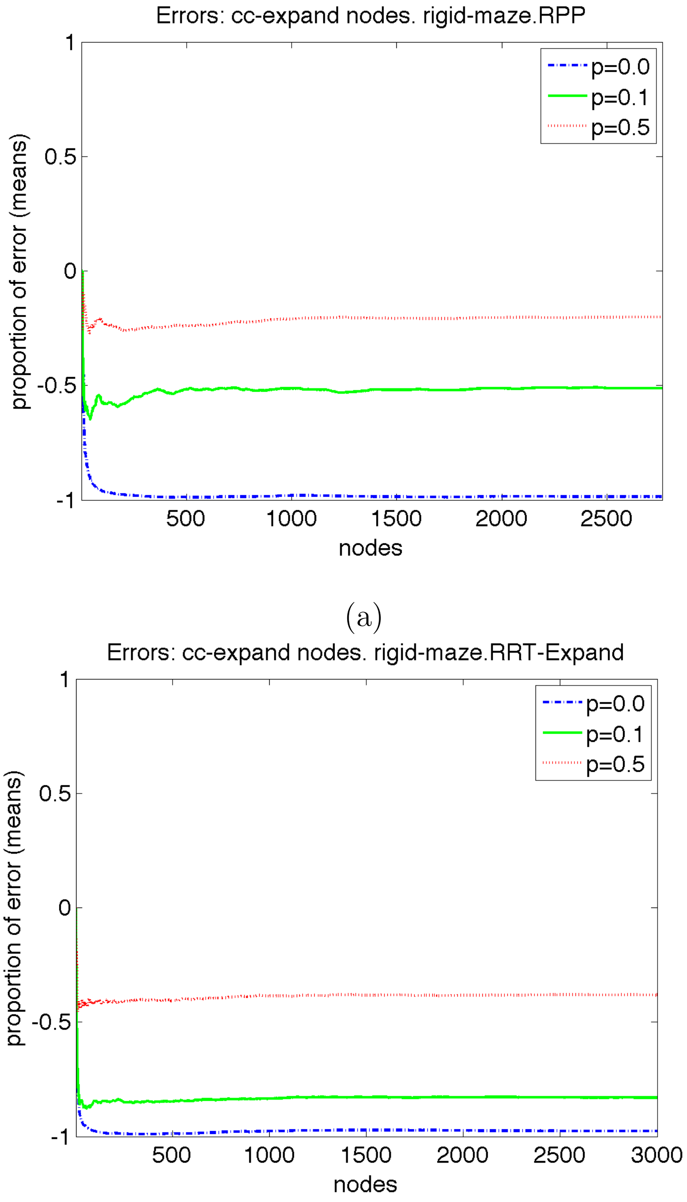
Fig. 11. Absolute error in the classification of *cc-expand* nodes for different approximations ($p = 0.0$ no additional connection test, $p = 0.1$, and $p = 0.5$) with respect to a full test ($p = 1.0$) vs. nodes in the model for roadmap-based planners applied to the *rigid-maze* problem. (a) *Gauss-PRM*. (b) *OBPRM*. Each line represents statistics from four different runs with standard deviations are below 5% before 1000 nodes in all the experiments. *Basic-PRM* and *MAPRM* are similar to *Gauss-PRM*, and *Bridge-Test* is similar to *OBPRM*. Overheads are shown in Table II.

high due to the unnecessary extra connections attempted. This overhead can still be reduced by making p the inverse of the number of neighbors of v. This will reduce the number of tests for nodes that are already heavily connected and are more likely to be *cc-oversample*.

Node classification in incremental planners should be done with high values of p to get correct results. Figure 12 shows the absolute error in the classification of *cc-expand* nodes in multiple runs of incremental planners on the *rigid-maze* problem for several values of p with respect to a full test $p = 1.0$. We show results for RPP and RRT-Expand, and we also evaluated EST and RRT-Connect which show similar trends as RPP and RRT-Expand. The high error for low values of p is due to the small number of neighbors for most nodes in models produced with incremental planners. This increases the impact on the accuracy of the expansion test for each disregarded node.

The overhead of the expansion test in incremental planners is reasonable even for high values of p. In Table II we notice that, except from RPP, the overhead of node-level metrics is smaller than the modeling time even for $p = 1.0$. This is mostly because of the small number of neighbors of each node which limits the number of extra connection attempts.

The errors and the overhead indicate that we can have a good evaluation at a low cost by making additional connection tests only to a small number of neighbors (the planners evaluated tried to connect every node to its k neighbors with $k = 20$). Also, we can use a small value of p in roadmap-based planners to make the classification affordable while in incremental planners we should use high values of p without incurring on excessive costs.



(a)

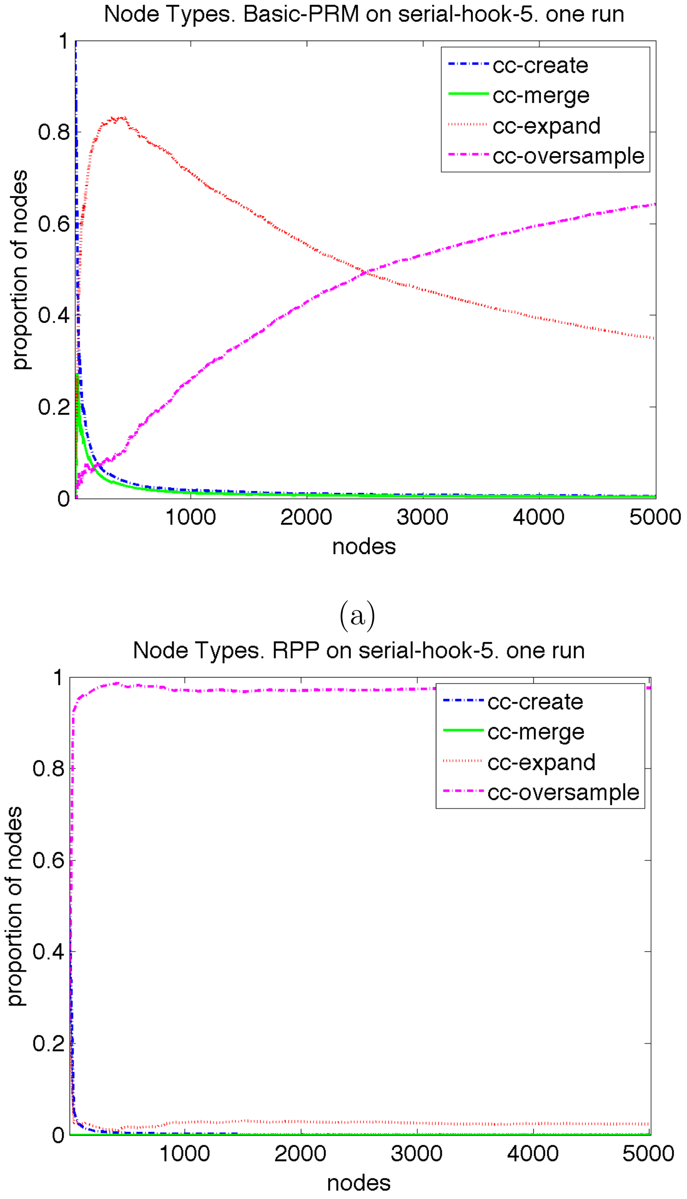
(b)

Fig. 12. Absolute error in the classification of *cc-expand* nodes for different approximations ($p = 0.0$ no additional connection test, $p = 0.1$, and $p = 0.5$) with respect to a full test ($p = 1.0$) vs. nodes in the model for incremental planners applied to the *rigid-maze* problem. (a) RPP. (b) RRT-Expand. Each line represents statistics from four different runs with standard deviations are below 5% before 1000 nodes in all the experiments. EST shows error amounts in between RPP and RRT-Expand, and RRT-Connect is similar to RRT-Expand. Overheads are shown in Table II.

B. Population Distribution of Node Types

The population of each of the different node types as model construction progresses reflects the ability of the planner to improve its knowledge about the C-Space of the problem being modeled. For example, a growing percentage of new *cc-oversample* nodes as the growth of the other types stalls or drops is a good indication that the chances of further improving the model using the same strategy are diminishing. A given population is the proportion of the corresponding type of nodes with respect to the total number of nodes accumulated at a given time. Also, we should notice that node types in incremental planners may show a big amount of *cc-oversample* nodes when they are exploring the open spaces (where samples are easy to connect) and many *cc-expand* nodes when they are exploring dense areas (where samples are harder to connect). Figures 13 and 14 illustrate how the population of *cc-create*, *cc-merge*, *cc-expand*, and *cc-oversample* nodes change when modeling the *serial-hook-5* problem with individual executions of the roadmap-based planner *Basic-PRM* and of the incremental planner *RPP*. We can notice how the distribution changes differently for each planner and that most changes occur at the initial iterations. Figure 14(a) shows the average of eight independent runs of *Basic-PRM* with the nodes axis in logarithmic scale to see better the initial iterations, and Figure 14(b) shows the standard deviation of the distributions in logarithmic scale for both axes. All the sampling strategies studied show similar trends in the distribution of the type of nodes they produce (more results can be found in [45]).

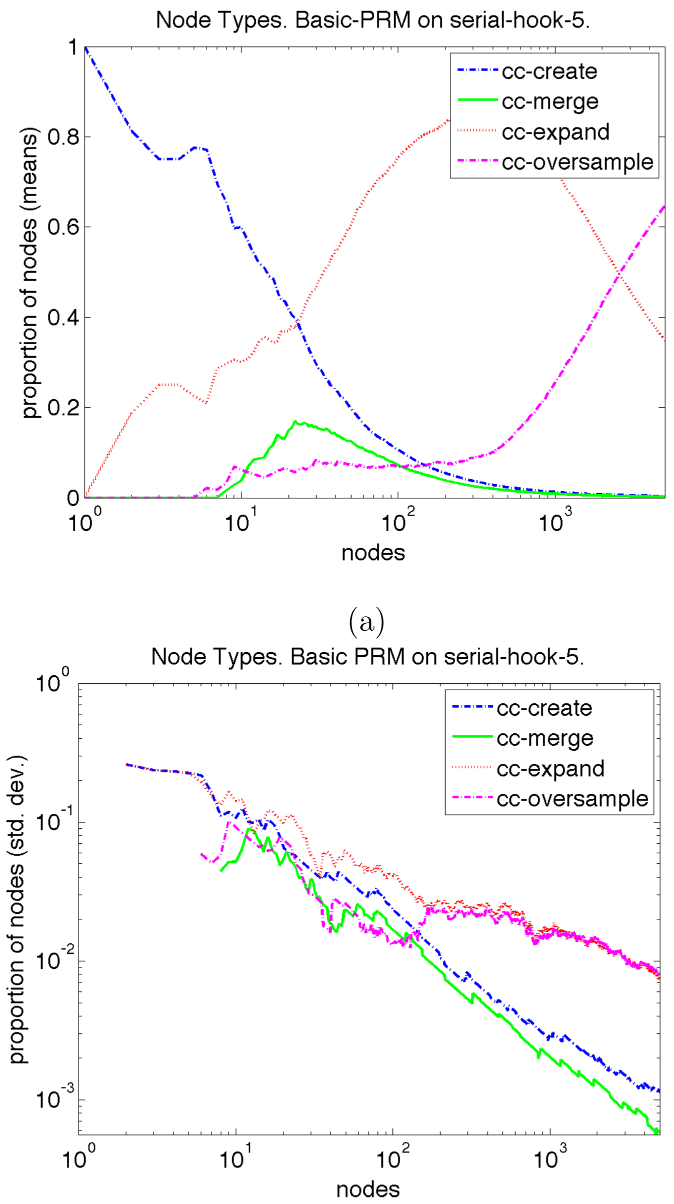
We can identify different stages in the learning process that will be discussed in more detail in chapter VIII. These stages correspond to the way each of the node types change as new nodes are added into the model. We can notice that most of the changes happen in the initial iterations, then they reach a stable value for some time



(a)

(b)

Fig. 13. Population distribution of node types produced by individual instances of planners when modeling the *serial-hook-5* problem. (a) *Basic-PRM*. (b) *RPP*.



(a)

(b)

Fig. 14. Population distribution of node types produced by eight instances of *Basic-PRM* when modeling the *serial-hook-5* problem. (a) Average populations with nodes in logarithmic scale to better see evolution in initial iterations. (b) Standard deviation of populations with both nodes and proportion of nodes shown in logarithmic scale, the standard deviations for all node types fall below 10% before 50 nodes.

until the *cc-oversample* nodes begin a steady increase as seen in Figure 14 (a).

The standard deviations of all the node types fall below 0.1 at less than 100 nodes, and below 0.05 at less than 300 nodes. This indicates that the population distribution is very similar in different executions and that we only need a few hundred nodes to be able to make fair classifications in this type of environment.

C. Visibility Around Growth Sites

Every time that the planner selects a node g to attempt to connect to another node g', the resulting success or failure is new knowledge that can be used to estimate the visibility around g. We call g a *growth site*. When the local planner employed in the connection attempt is bidirectional, both g and g' are *growth sites*. In problems with non-holonomic constraints that are usually addressed with incrementally-exploring planners the local planner is usually not bidirectional and only g is considered a *growth site*.

The visibility around a growth site g can be computed by keeping track of the number of its growth attempts a_g, and the number of its successful growths s_g. We define the *visibility ratio* of g, $V_g = s_g/a_g$. V_g is updated every time g is selected for growth as shown in Figure 15. The more growth attempts from g, the better is the quality of the estimation of V_g.

When the visibility ratio stabilizes at some value, it can be used in several ways. One application is to identify highly constrained regions where nodes have low visibility as will be discussed in chapter VII. For example, in incrementally-expanding planners most growth sites should have a similar number of *growth attempts* if the model is expanding uniformly. Or, in *RRTs*, sites that are repeatedly selected for growth, but their Voronoi region fails to shrink will have a number of growth attempts larger than

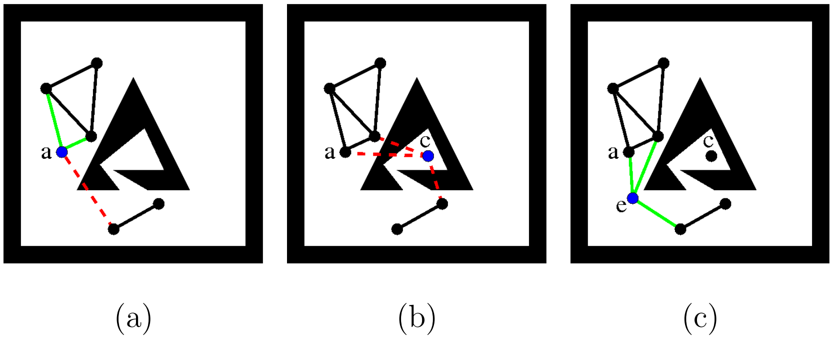


Fig. 15. Visibility ratio of growth sites a, c, and e as they are added to the model and connected with a bidirectional local planner. (a) new node a is added, $V_a = 2/3$. (b) new node c is added $V_c = 0/3$, the visibility of a needs to be updated $V_a = 2/4$. (c) new node e is added $V_e = 3/3$, the visibility of a needs to be updated $V_a = 3/5$.

TABLE I

VISIBILITY OF GROWTH SITES IN *serial-hook-5*

	Growth Sites	**Visibility Range V_g**		
Planner	5000 nodes	$[0, 1/3)$	$[1/3, 2/3)$	$[2/3, 1]$
RPP	97%	31%	13%	53%
RRT-Connect	83%	12%	11%	60%

average. Table I shows the proportion of growth sites used when modeling the *serial-hook-5* environment with the incremental planners *RPP* and *RRT*-Connect after 5000 nodes and the proportion of their growth sites in the ranges $[0, 1/3)$, $[1/3, 2/3)$, and $[2/3, 1]$. We notice that some nodes are not used as growth sites and that the visibility ratio of growth sites have different trends in different planners. The later will be used to identify regions as discussed in chapter VII.

D. Overhead of Node-Level Metrics

Node-Level metrics incur in reasonable overheads. Table II shows the average time for computing node-level metrics for different roadmap-based and incremental planners on the *rigid-maze* problem. The planners were run sixteen times (four for each value of the probability of neighbor visibility tests for the computation of expansion p) and their times averaged. The node-level metrics times shown include the computation of node types which are the most expensive, and the estimation of visibility around growth sites.

The overhead for the computation of the expansion ratio needed to distinguish between *cc-expand* and *cc-oversample* nodes is higher in heavily connected problems. One potential optimization is to make the value of p be inversely proportional to the number of neighbors in order to reduce the number of tests in nodes that are already heavily connected and are more likely to be *cc-oversample*. Also, we can use a low p in roadmap-based planners without affecting the accuracy of the evaluation. In contrast, we need to use a high p in incremental planners, but this is not as expensive in this type of planners.

TABLE II

Node-Level Metrics Overhead in *rigid-maze*

	Modeling	**Average Overhead** [s]			
Planner	(3000 nodes)	$p = 0.0$	$p = 0.1$	$p = 0.5$	$p = 1.0$
Basic-PRM	791.6	132.6	2,441.9	3,560.4	3,668.9
OBPRM	518.6	71.0	265.1	499.1	569.9
Gauss-PRM	639.8	110.7	898.5	1,403.8	1,347.6
MAPRM	299.2	122.2	653.3	948.7	920.6
Bridge-Test	5,607.4	75.8	267.7	468.6	524.1
RPP	68.2	187.9	200.1	211.39	279.88
EST	201.8	60.5	60.6	63.5	67.0
RRT-Expand	294.8	50.2	53.0	55.1	60.9
RRT-Connect	266.3	50.0	51.2	54.1	57.1

CHAPTER VI

GLOBAL-LEVEL METRICS

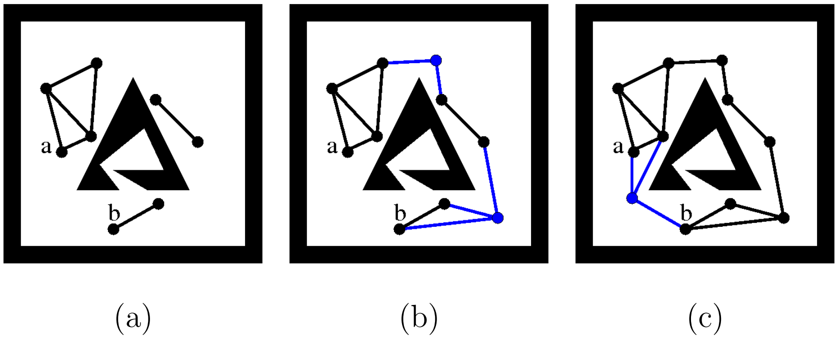
Metrics at the global level allow us to estimate the changes in the topology of the model due to the addition of new nodes by analyzing the structure of the model graph. In particular, we can estimate improvements in the topology represented in the model graph through approximations of graph statistics related to components and pathway lengths. Global-level metrics can be analyzed for sets of one or more new nodes and connections.

A. Changes in Motion Pathways Produced by New Nodes

Given a model M, a planner adds a valid sampled configuration v and a selected subset of its valid connections producing the model M'. This operation changes the homotopy classes or motion pathways of the original model M. As shown in Figure 16, improvements due to new nodes and connections make the number of components in the model and the number of pathways between any two nodes reflect better the structure of the underlying C-Space. Unfortunately, it would be unfeasible to compute the number of pathways between every pair of nodes in M. Alternatively, we can analyze the changes in the structure of M that are often caused by changes in its internal pathways.

Some informative features of the model graph include the number of connected components, the diameter of the graph components, and the total weight of the minimum-spanning tree. Below, we discuss the number and diameter of components.

- **Number of components** — When the number of components in M ap-



(a) (b) (c)

Fig. 16. Global changes in model topology resulting from new nodes and connections. (a) State of M before adding new nodes, in the C-Space there is one component and two homotopy classes, or distinct pathways, between a and b; in contrast, the model has three components and no pathway between a and b. (b) Two new nodes and their connections improve the topology of the model to have one component and one homotopy class between a and b. (c) one more node and its connections improve the topology of the model to have one component and two homotopy classes between a and b.

proaches the number of components of the underlying C-Space, the topology of the model improves. This metric is maintained directly in the graph.

- **Diameter of components** — The diameter of a graph G in M is the length of the shortest path between the most eccentric nodes in G. This metric allows us to trace the changes in some important subsets of motion pathways represented in M. We keep track of the sum of the diameters (*sum-diameter*) of all the components as an approximation of the structural changes happening in the whole model, and the diameter of the largest component (*max-diameter*) which in many problems represents most motion pathways. We have found that tracking these features allows us to capture the most dramatic changes in the model.

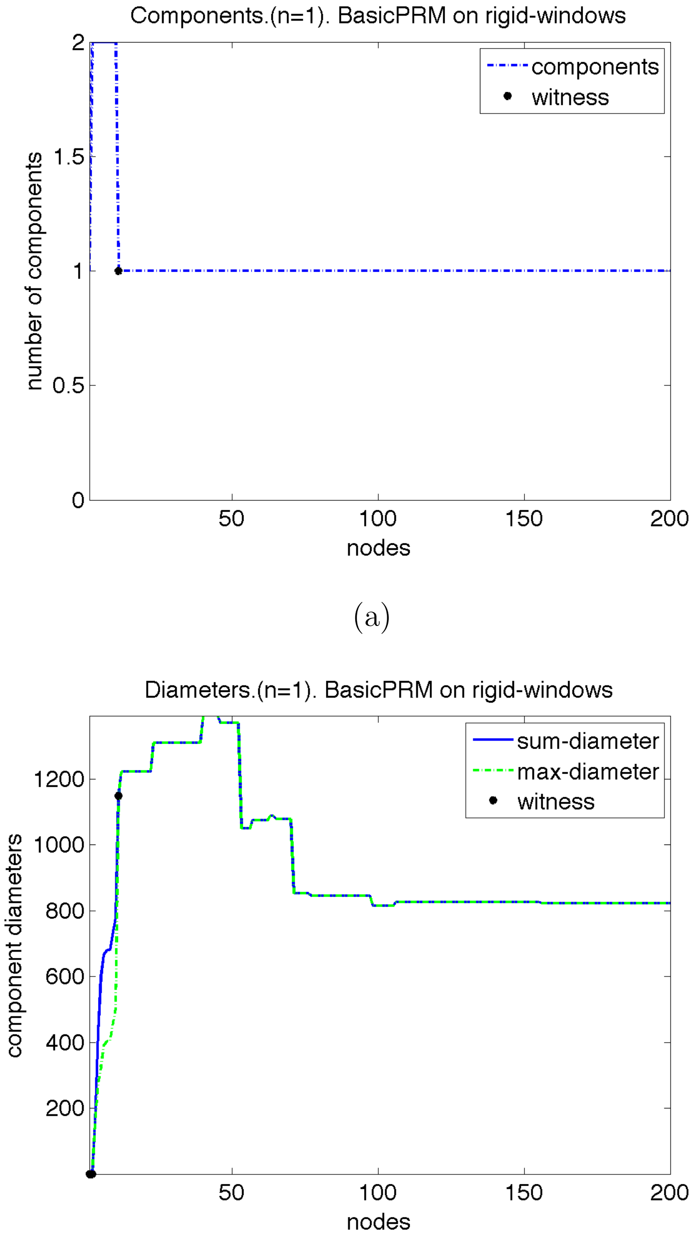
As will be shown shortly, the *max-diameter* and *sum-diameter* have their most dramatic changes at the initial iterations of sampling and they stabilize when the model does not have many more structural improvements. In problems with only one connected component in the free space, these two diameters will stabilize around the same value.

The accurate computation of the diameter is not efficient for our needs. For dense graphs, the all-pairs-shortest-path problem (and thus the diameter problem) can be computed in time $O(M(n) \log n)$ where $M(n)$ is the time to multiply two $n \times n$ matrices of small integers [58]. Matrix multiplication takes time $O(n^2.376)$ [15].

Fortunately, there are efficient approximate algorithms to compute the diameter of a graph. In particular, we use the algorithm presented in [16] that employs Breadth First Search (which takes time $O(|E| + |V|)$ for a graph with edges E and vertices V) to determine a tight bound on the diameter for graphs with no induced cycle greater than k that is no worse than the diameter of the graph minus $\lfloor k/2 \rfloor$. Since we estimate overall changes in diameters rather than in their actual value, this approximation is sufficient for our needs.

In order to illustrate these global-level metrics, we keep track of the number of components, the *max-diameter* and *sum-diameter* during one execution of the *Basic-PRM* planner to model the *rigid-windows* problem. In order to show better the evolution of the pathways in the model we limited the rotational degrees of freedom so that the start and goal have always the same orientation so that there is only one main pathway through each of the windows. Thus, this problem has at least four clear distinct pathways through each of its windows whose widths range from 1.5 times the width of the robot to 3.5 times the width of the robot as shown in Table III. We also introduced a witness-query evaluation to compare it to the node-level and global-level metrics discussed in this dissertation: the start and the goal have the same relative position with respect to the wall and close to window 1.

The largest changes in the components correspond to the generation of the nodes that enable a pathway through each window as shown in Figure 17 and Table III.



(a)

(b)

Fig. 17. Evolution of global-level metrics for one instance of *Basic-PRM* on the *rigid-windows* problem. (a) Number of components. (b) *sum-diameter* and *max-diameter* correlate to new pathways found through the windows and on each side of the wall. (c) Population distribution of node types, learning stages correlate to changes in diameters. The witness query solved at 9 nodes is marked on both plots.

TABLE III

Basic-PRM ON *rigid-windows*. ROBOT WIDTH = 1

Window	**1**	**2**	**3**	**4**
Width	1.50	1.75	2.50	3.50
Nodes to Model Passage	156	71	53	11

The initial nodes are connected in two components accounted for in the *sum-diameter* and *max-diameter*. When node 11 is generated, the first edge through window 4 appears and the two components merge into one making the *sum-diameter* and *max-diameter* increase and join for the rest of the process. When node 53 is generated, the first edge through window 3 appears corresponding to a big drop in the diameters measures. When node 71 is generated, the first edge through window 2 appears corresponding to the latest big drop in the diameters. When node 156 is generated, the first edge through window 1 appears corresponding to the last, although very small, reduction in the diameters. It is worth noting that by the time that a passage through window 1 is found, most of the pathways have been refined, and although the reduction in diameter caused by this specific edge is very small these global-level metrics are able to capture it. The few other noticeable changes in the diameter measures correspond to pathway refinements on either side of the wall.

The witness query was solved at node 11. The path found is actually the longest one through window 4 instead of the shortest one that goes through window 1 and that is only possible after 156 nodes. This shows one of the reasons why the use of witness queries is a poor evaluation metric for problems with multiple pathways between important configurations because witness queries cannot identify the different

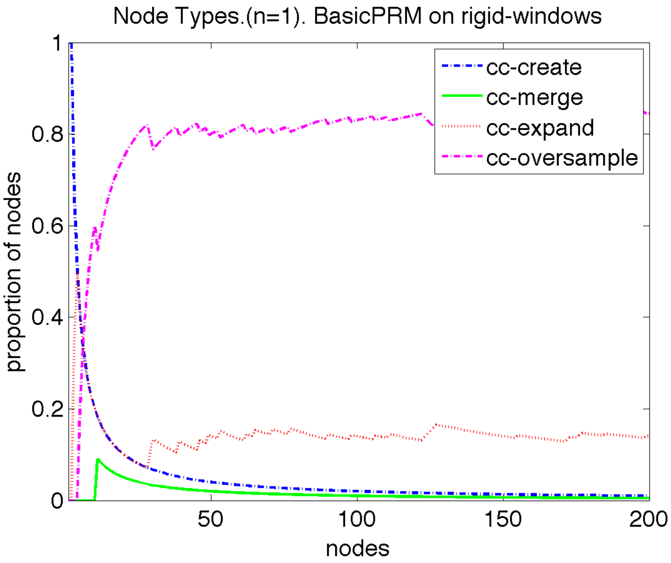


Fig. 18. Population distribution of node types for one instance of *Basic-PRM* on the *rigid-windows* problem. This distribution corresponds to the same instance discussed in Figure 17.

homotopy classes for the pathways in the problem. On the other hand, the changes in the structure captured by the global-metrics discussed here capture some of the changes in homotopy classes for the pathways in the problem.

We also gathered the population distribution of node types as shown in Figure 18. We can notice that the biggest changes in the population distribution of nodes corresponds with the largest changes in the diameters that happen before node 50 when both the diameter and the node types start to stabilize. We can also notice that after around node 75, the *cc-oversample* nodes start rising and the diameter has only small changes. These events correspond to the learning stages that were mentioned in Section V.B and that will be discussed in detail in chapter VIII.

Incrementally-exploring planners will usually not represent the multiple homotopy classes or pathways between pairs of configurations because they do not allow cycles in their models. Nevertheless, their diameters expand while they increase their coverage of the C-Space just as roadmap-based planners do. This allows us to apply the same global-level metrics in both types of planners as we will show in chapter VIII.

B. Detection of Relative Change of Global-Level Metrics

It is likely that the changes caused by sets of successive nodes in the structure of the model are more meaningful than those caused by a single sample. In order to study this, we group sets of recent nodes into bins of n consecutive nodes and then, for a given feature A, we compute the average rate of change of A at bin i relative to the k previous bins as follows:

$$\Delta(i,k)_A = \sum_{j=0}^{k-1} \left| \frac{A(i-j) - A(i-j-1)}{A(i-j-1)} \right| \tag{6.1}$$

A change at bin i has a sustained effect in the computation of Δ for the k consecutive bins. This sustained effect allows us to evaluate changes not only with respect to the previous bin, but with respect to the previous k bins. This gives us a sliding window of k bins where we can monitor the planner's ability in gaining more knowledge about feature A. A reduction in the speed at which the planner gains information about several features, such as the *max-diameter* and *sum-diameter*, eventually leads those features to converge around some value, which corresponds to $\Delta(i,k)_A$ get closer and closer to 0. We can detect this by testing whether $\Delta(i,k)_A$ has fallen bellow a small threshold. When this happens we can either decide to stop planning or switch strategies as in [65].

The size of the bin n allows the user to specify the number of samples to consider before the next evaluation of feature A. Bins should be large enough so we can capture statistically significant changes and small enough so that the overhead can be kept low. In our experiments we obtained consistent results with bins that vary from 20 to 50 nodes.

The size of the sliding window k allows the user to specify the number of evaluations of feature A to consider when making decisions. As discussed before, most of the changes occur in the initial iterations, and later changes are smaller in magnitude and less frequent. Thus, it would be convenient to start with a relatively low k and increase it over time. However, here we only discuss a constant window size. In our experiments we obtained consistent results with sliding windows that vary from 4 to 8 bins.

We evaluate the rate of change of *max-diameter* and *sum-diameter* to identify structural changes in the same instance of *Basic-PRM* on the *rigid-windows* problem discussed before. First, we evaluate at every node (bin size $n = 1$) and we compute $\Delta(i, k = 1)_{max-diameter}$ and $\Delta(i, k = 1)_{sum-diameter}$ with respect to the previous bin (window size $k = 1$) as shown in Figure 19. We note that there are spikes at the nodes that cause structural changes as seen in Figure 17(b) and Table III. Second, we increase the window size to compute $\Delta(i, k = 4)_{max-diameter}$ and $\Delta(i, k = 4)_{sum-diameter}$ with respect to the previous 4 bins (window size $k = 4$). We can see in Figure 20(a) that the detected changes at each bin are averaged for the following 4 bins, but since n is too small the changes oscillate and do not stabilize. Last, we keep the same window size $k = 4$ to compute $\Delta(i, k = 4)_{max-diameter}$ and $\Delta(i, k = 4)_{sum-diameter}$, but we increase the bin size to $n = 10$. As we see in Figure 20(b) both $\Delta(i, k = 4)_{max-diameter}$ and $\Delta(i, k = 4)_{sum-diameter}$ are smoother. We also show when three different thresholds are met for both $k = 4, n = 1$ and $k = 4, n = 10$ and we see that they happen when the diameter is closer to stabilization for $k = 4, n = 10$.

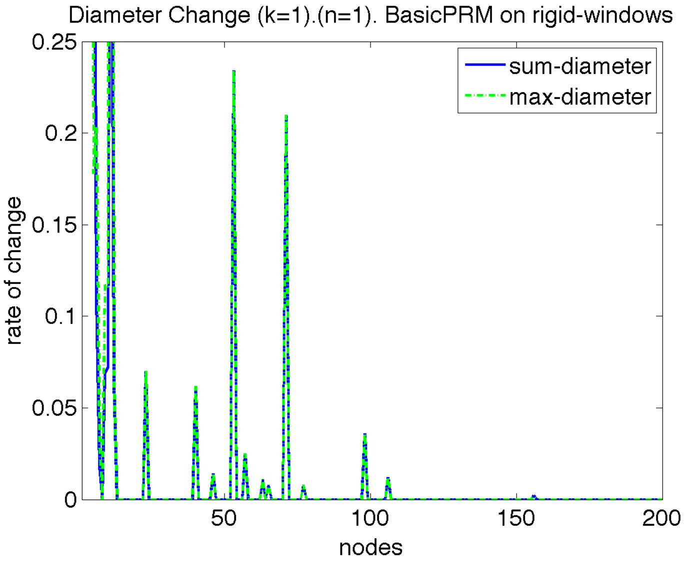
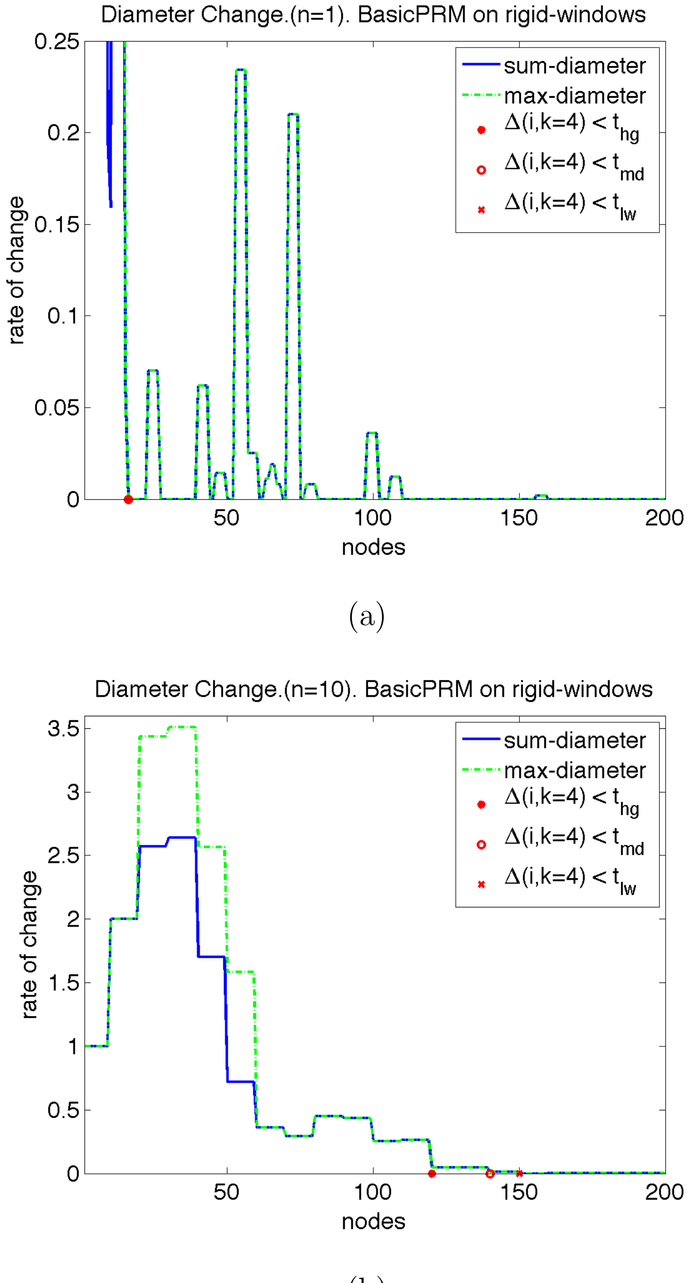


Fig. 19. Rate of change of *max-diameter* and *sum-diameter* for the same instance of *Basic-PRM* on the *rigid-windows* problem discussed in Figure 17. Size of bins $n = 1$, only changes in each bin are considered.

C. Overhead of Global-Level Metrics

The overhead of global-level metrics is very low in comparison to the time to model the problems. Table IV shows the average time for computing global-level metrics every twenty nodes for different roadmap-based and incremental planners executed sixteen times on the *rigid-maze* problem.

The overhead of global-level metrics is similar for all planners because it depends more on the structure of the graphs, although this is influenced by the planning strategy. Incremental planners incur slightly smaller costs because their models are trees with fewer edges than those produced with roadmap-based planners.



(a)

(b)

Fig. 20. Average rate of change of *max-diameter* and *sum-diameter* for the same instance of *Basic-PRM* on the *rigid-windows* problem discussed in Figure 19. (a) Size of bins $n = 1$, averaged bins $k = 4$. (b) Size of bins $n = 10$, averaged bins $k = 8$.

TABLE IV

Global-Level Metrics Overhead in *rigid-maze*

	Modeling	**Average Overhead**	
Planner	(3000 nodes)	[s]	%
Basic-PRM	791.6	18.7	2.4
OBPRM	518.6	11.9	2.3
Gauss-PRM	639.8	15.8	2.5
MAPRM	299.2	17.0	4.3
Bridge-Test	5,607.4	12.2	0.2
RPP	68.2	0.3	0.5
EST	201.8	1.6	0.8
RRT-Expand	294.8	2.7	0.9
RRT-Connect	266.3	2.7	1.0

CHAPTER VII

REGION-LEVEL METRICS

Metrics at the region level allow us to identify distinct sections of C-Space. A region is a set of configurations generated by the planner that are similar with respect to features of interest computed at the node and global levels. For example, we can define regions based on the visibility around nodes: configurations close to the obstacles have a lower visibility than those that are far from obstacles.

Regions allow us to analyze the information gained by the planner in order to evaluate the spatial performance of planners, to adapt the planning strategy to take advantage of the features of the regions by assigning well-suited planners or changing planner parameters. For example, we can identify the degrees of freedom that are restricted in highly-constrained regions to bias local planners. Also, we can evaluate the complexity of the region to decide whether more sampling is required by tracking the population distribution of node types or the ratio of valid to non-valid configurations. Moreover, we can monitor the structural changes in the subgraphs inside each region to decide when to stop planning in each region or when to re-evaluate specific regions.

A. Region Construction

A region is a set of configurations used in the construction of the model M that are similar with respect to a given set of features $A = A_1, A_2, ..., A_n$ according to some clustering strategy. Both, nodes in M and configurations that were used in the generation of nodes without being stored as nodes in M, can be considered to define regions, although here, we only consider nodes in M. If the distance metric used for clustering is based on the configuration distances, then the configurations inside each region may be spatially close-by, but if the distance metric is based on some other feature, then they may be sparsely distributed in the C-Space.

1. Features

The only requirement for clustering features is that they should be based on intrinsic information of the node, such as the configuration parameters, or in local information around each node, such as the visibility ratio. This way each region is formed using only local information, but the collection of regions provides global information about the problem. The distance metric used in the clustering is the difference between feature values for scalar features, and the Euclidean distance for vectors.

2. Clustering Strategies

There are many potential clustering strategies that depend on the application. We discuss axis-aligned regions, simple-feature regions, and coverage regions.

a. Axis-Aligned Regions

Axis-aligned regions split some configuration parameter at some value to make two adjacent regions that incorporate the nodes within range. This simple strategy is a generalization of the one we applied in our machine learning approach to feature-sensitive motion planning [44, 46]. The configuration parameter to split and the value at which to split is chosen based on how diverse the samples are as explained in two subdivision strategies below.

An axis-aligned subdivision strategy based on gaps selects the configuration parameter with the largest gap of valid configurations and splits it at the middle of the gap. The goal of this strategy is to group samples at different sides of obstacles.

An axis-aligned subdivision strategy based on information gain to group samples at different sides of obstacles splits the planning space in a similar way as the decision-tree machine learning algorithm C4.5 [53, 54]. Information gain is computed for several prospective partitions and the one that best separates nodes into two homogeneous regions is selected. Information gain is defined in terms of entropy, a feature that measures the diversity of a set relative to a c-wise classification, in the case of a 2-wise classification for the set S with a proportion of valid configurations p_v and a proportion of invalid configurations p_i it is defined as $Entropy(S) = -p_v \log_2 p_v - p_i log_2 p_i$. Intuitively, a region has higher entropy when the proportion of valid and invalid configurations is similar, and it has lower entropy when there is a bias towards either valid or invalid configurations. The information gain of splitting the region S in parameter D through the point m into two subregions $S_{D,1}$ at the left of m and $S_{D,2}$ at the right of m is $Gain(S, D, m) = Entropy(S) - (|S_{D,1}|Entropy(S_{D,1}) + |S_{D,2}|Entropy(S_{D,2}))|S|^{-1}$.

These naive axis-aligned strategies can identify distinct regions of the space to assign planners in [46], however, they are limited because they depend on the number of *DOFs* of the robot and because it is not likely that complex C-Spaces can be properly subdivided in axis-aligned regions.

b. Simple-Feature Regions

Simple-feature regions cluster nodes with a naive strategy that splits the values of the feature evenly to make a region for low values, a region for medium values, and a region for large values. When applying this strategy using the visibility of the nodes, we can identify regions with different levels of complexity as seen in Figure 21 which shows the 1000-node model produced by *Basic-PRM* in the *rigid-maze* problem and the regions for low, medium, and high visibility and in Figure 22 which shows the 1000-node model produced by *OBPRM* in the same problem and the corresponding

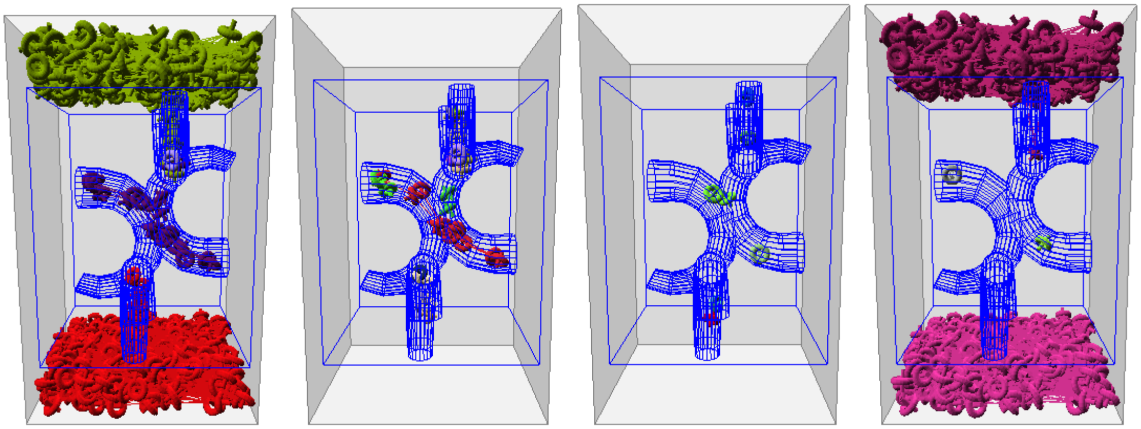


Fig. 21. Visibility regions in a model produced by *Basic-PRM* in the *rigid-maze* problem. (a) 1000-node model. (b) Low-visibility region: *visibility* < 1/3. (c) Medium-visibility region: 1/3 <= *visibility* < 2/3. (d) High-visibility region: 2/3 <= *visibility*.

visibility regions. We can see that the regions capture the areas with different complexities in the problem. They also show the very different sampling distributions produced by *Basic-PRM* and *OBPRM*. One potential application of this strategy is to focus powerful planners in low-visibility regions and find representative nodes of high-visibility regions that reduce the size of the model.

c. Coverage Regions

Coverage regions cluster nodes inside the local neighborhood of growth sites as the planner increases its coverage of the C-Space. These regions allow us to monitor the rate at which planners increase their coverage in unexplored regions. When a new node lies within a pre-defined radius r away from the center of the region, it is incorporated into the region. Growth sites, nodes that are selected for connection to other nodes, are treated in a special way: when a new growth site cannot be incorporated into any previously existing region, a new region centered at the new

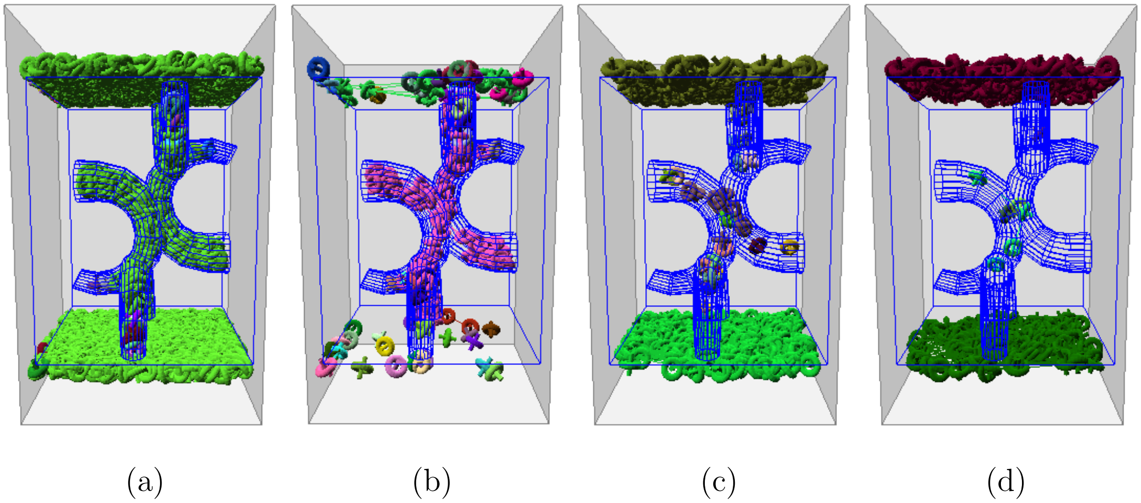


Fig. 22. Visibility regions in a model produced by *OBPRM* in the *rigid-maze* problem. (a) 1000-node model. (b) Low-visibility region: *visibility* $< 1/3$. (c) Medium-visibility region: $1/3 <=$ *visibility* $< 2/3$. (d) High-visibility region: $2/3 <=$ *visibility*.

growth site is created and nodes within the local neighborhood are incorporated. There is no limit on the number of regions of which a node can be part. Limiting the creation of regions around growth sites is particularly useful in incremental planners because the regions allow us to track the sampling distribution over time. The union of all the coverage regions represents the coverage of the C-Space achieved by the planner at a resolution defined by the radius of the regions r. This radius can be defined based on the resolution parameters of the problem. In incremental planners we found it convenient to make it slightly larger than the expansion increment, so that each region incorporates at least two nodes without being excessively large.

A planner is more efficient in increasing its coverage when new growth sites are able to create new coverage regions. We can estimate this efficiency by computing the *coverage rate*: the ratio of the number of coverage regions to the number of growth sites. A high value corresponds to the exploration of new areas, whereas a low value corresponds to exploration of areas previously explored.

3. Frequency of Region Updates

In order to keep the overhead of region computation low, we update them with the same frequency as we compute global changes by grouping sets of recent nodes into bins of n consecutive nodes.

4. Region Statistics

In addition to providing a spatial partitioning of the samples, regions allow us to aggregate additional statistics that may be useful to improve sampling. Some of these statistics include:

- Entropy as described earlier and used in [46]. This feature can also be applied in a similar way to the visibility to identify regions that are close to the surface of the C-obstacles and that are more likely to need additional sampling as in [56] or as a way of measuring the learning achieved in the model as in [11].
- Variability of the DOF parameters. In low-visibility regions the DOF parameters with the smallest variability are very constrained, but parameters with higher variability are not. This information can be used to guide local planners towards DOF parameters with higher variability.
- Diameters in subgraphs formed by nodes and edges in the region. This feature allows us to monitor the changes in the pathways represented in each region in order to make decisions such as biasing the sampling towards regions with more changes.

The evolution of region statistics provides insight into the sampling distribution and pinpoints areas where special kinds of sampling can be applied. When we are only interested in recent events, we can aggregate statistics bin-wise for recent nodes.

For example, a sustained high percentage of successful growths and high visibility in recent nodes indicates that the areas being covered are mostly free.

B. Overhead of Region-Level Metrics

The overhead of region-level metrics is very low compared to the time to model the problems. Table V shows the average time for computing region-level metrics using the naive clustering strategy every twenty nodes for different roadmap-based planners and incremental planners executed sixteen times on the *rigid-maze* problem. The overhead is similar for all planners because it depends on the number of nodes in the problem and on the clustering strategy more than on the method to produce nodes.

TABLE V

Region-Level Metrics Overhead in *rigid-maze*

	Modeling	**Average Overhead**	
Planner	(3000 nodes)	[s]	%
Basic-PRM	791.6	5.9	0.7
OBPRM	518.6	6.1	1.2
Gauss-PRM	639.8	6.0	0.9
MAPRM	299.2	6.0	1.5
Bridge-Test	5,607.4	5.9	0.1
RPP	68.2	0.8	1.1
EST	201.8	2.7	1.3
RRT-Expand	294.8	5.4	1.8
RRT-Connect	266.3	5.3	2.0

CHAPTER VIII

APPLICATIONS AND EXPERIMENTS

We apply the metrics to different problems to investigate several questions about the planning process. First, we study the similarities in the metrics in the modeling process of different planners by looking at their node-level and global-level metrics to identify the stages of the planning process. Second, we discuss how the identification of learning stages can be applied to make decisions about which planners to use. Third, we study the evolution of regions over time, and we use them to compare the coverage achieved by different planners. We also discuss other uses for the regions to influence planning. Fourth, we apply node-level, global-level, and region-level features to compare planners.

A. Learning Process of Planners

We study the evolution of the metrics at the node and global levels during the modeling process.

1. Evolution of the Node-Level Metrics

Planners show three stages in the evolution of the node-level metrics: 1) all node types have their largest changes; 2) there is a temporal stabilization in all node types; 3) there is a slight, but steady increase of *cc-oversample* nodes. We show the average population distribution of node types produced by the roadmap-based *Basic-PRM* (Figure 23), *OBPRM* (Figure 24(a)), and *Gauss-PRM* (Figure 24(b)) when applied to the *rigid-maze* problem. The distributions in *MAPRM* were similar to *Gauss-PRM* and the distributions in *Bridge-Test* were between *OBPRM* and *Gauss-*

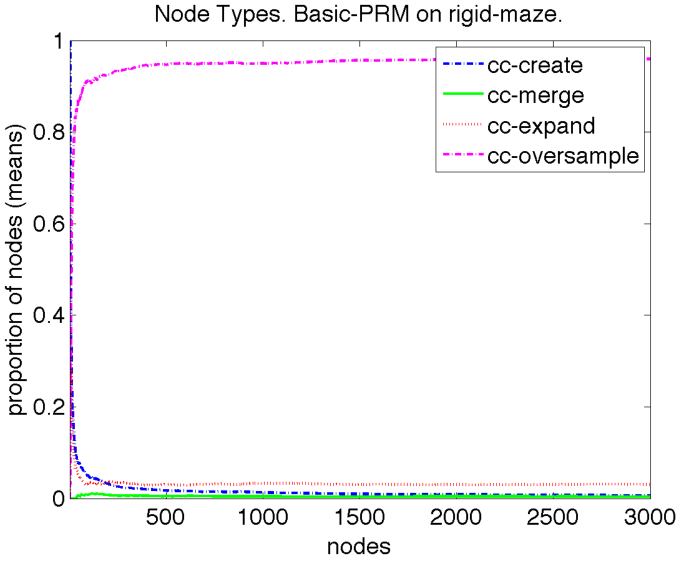
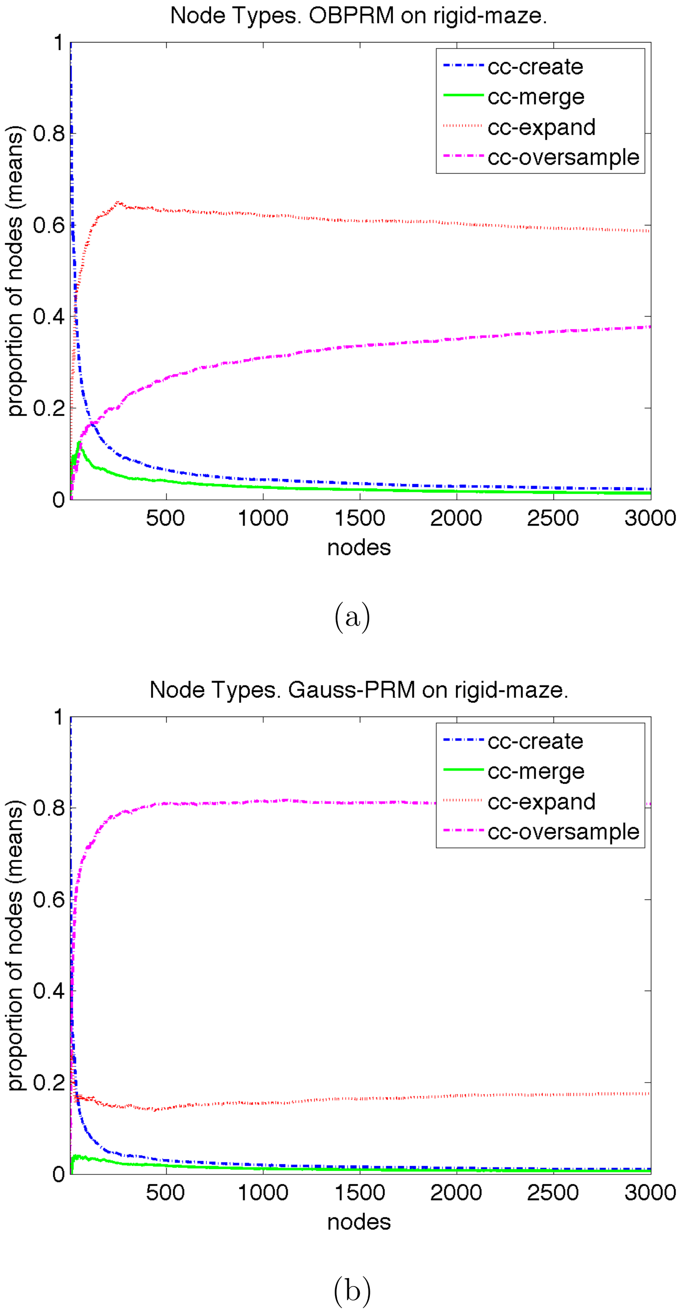


Fig. 23. Average population distribution of node types produced by the roadmap-based *Basic-PRM* planner on the *rigid-maze* problem. 4 runs of 3000 nodes.

PRM. Figure 25 shows the average population distribution of node types produced by the incrementally-exploring *RPP*, and *RRT*-Connect when applied to the *rigid-walls* problem.

Different roadmap-based planners have very distinct profiles of node type distributions. On the other hand, all incrementally-exploring planners have very similar profiles because they produce many *cc-oversample* nodes before they can reach the *cc-expand* nodes in low visibility regions. Also, the distributions produced by each planner change in different problems as we will see later in Sections VIII.D and VIII.E.

In this problem, *OBPRM* showed the highest proportion of *cc-expand* nodes, while *Basic-PRM* had the lowest. The dominant proportion of *cc-oversample* nodes generated by *Basic-PRM* is due to its uniform sampling that gets most of its nodes in the open spaces where they are very easy to connect. In contrast, *OBPRM* produces nodes inside the narrow passage which are not as easy to connect to each other and represent larger expansions. *Gauss-PRM* is in between *Basic-PRM* and *OBPRM*



(a)

(b)

Fig. 24. Average population distribution of node types produced by different roadmap-based planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner.

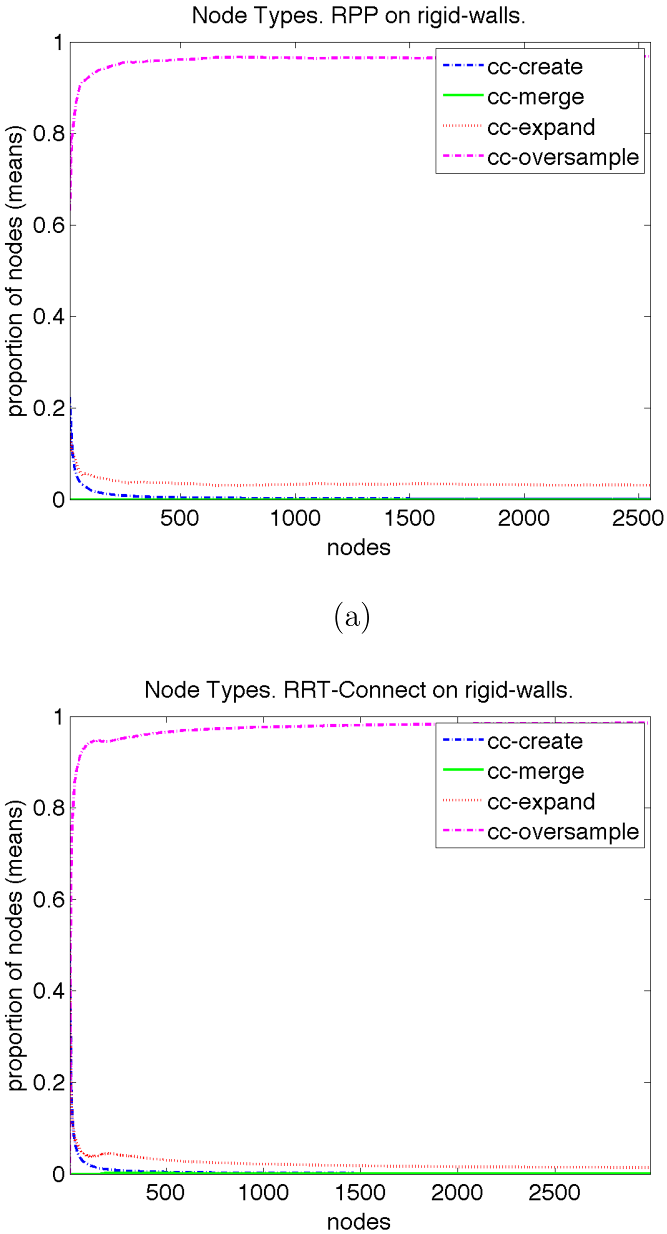
because, although it tries to get nodes in the narrow passages, it does it from uniformly sampled nodes which is not as effective as the *OBPRM* strategy which makes them based on configurations in collision.

Node types in incremental planners may show a big amount of *cc-oversample* nodes when they are exploring the open spaces and many *cc-expand* nodes when they are exploring dense areas. In this problem, both *RPP* and *RRT*-Connect produce a big amount of *cc-oversample* nodes because both of them started sampling from the open spaces where most nodes are easy to connect. *RPP*never makes it through the passages and keeps producing *cc-oversample* nodes. By the time when *RRT*-Connect finds the narrow passage and it starts generating *cc-expand* nodes, the *cc-oversample* nodes it has already generated dominate the population distribution of node types.

All the planners achieve a similar node population in multiple runs as can be noticed in their small standard deviation before a couple hundreds of nodes or much earlier for some planners. Table VI shows the number of nodes after which the standard deviation of the node population distribution falls bellow 0.10 and 0.05 for multiple runs of *Basic-PRM*, *OBPRM*, *Gauss-PRM*, *MAPRM*, and *Bridge-Test* on the *rigid-maze* problem. Similarly, Table VII shows the corresponding standard deviation for runs of *RPP*, *EST*, *RRT*-Expand, and *RRT*-Connect on the *rigid-walls* problem.

2. Evolution of the Global-Level Metrics

The planners also show three stages in the evolution of global-level metrics: 1) the diameter measures have their largest changes when the main components form; 2) the diameter measures undergo many smaller changes, in roadmap-based planners these changes correspond to large components joining together, and in incremental planners they correspond to a reduction in coverage expansion; 3) changes in the diameter



(a)

(b)

Fig. 25. Average population distribution of node types produced by different incremental planners on the *rigid-walls* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes were run for each planner.

TABLE VI

Population Distribution of Nodes in *rigid-maze*. Deviation

	Nodes for Standard Deviation	
Planner	< 0.10	< 0.05
Basic-PRM	10	20
OBPRM	40	80
Gauss-PRM	35	220
MAPRM	25	70
Bridge-Test	60	180

TABLE VII

Population Distribution of Nodes in *rigid-walls*. Deviation

	Nodes for Standard Deviation	
Planner	< 0.10	< 0.05
RPP	25	30
EST	10	15
RRT-Expand	10	20
RRT-Connect	10	15

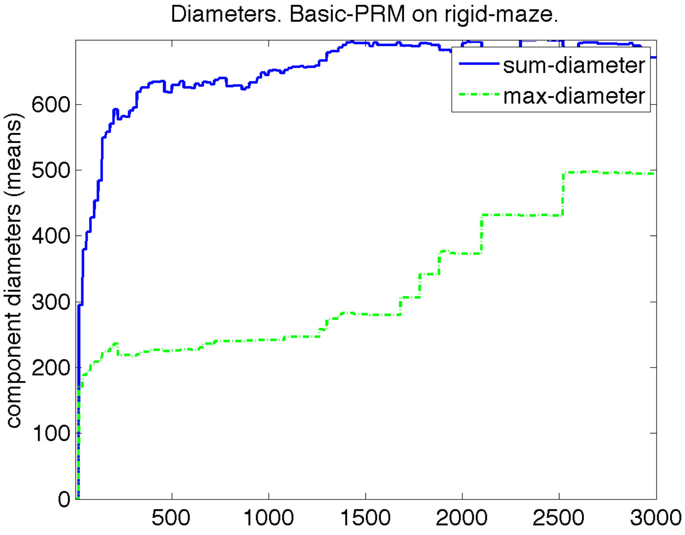
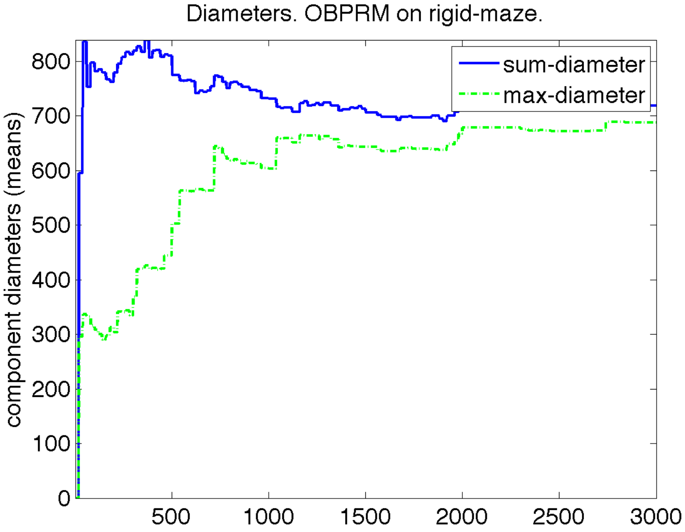
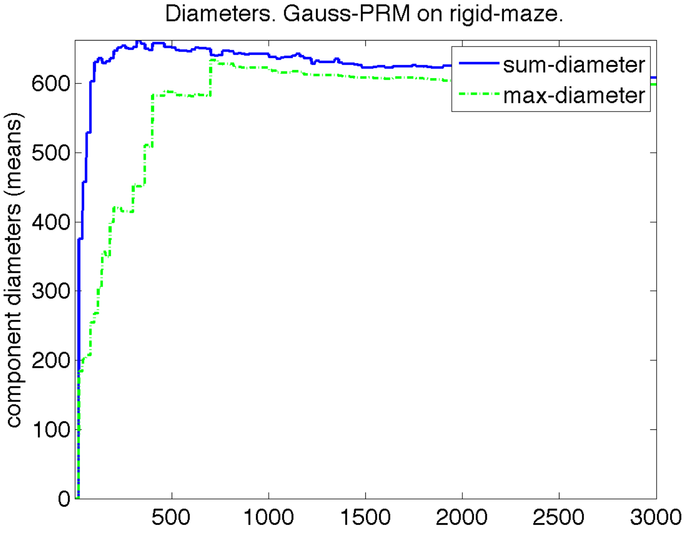


Fig. 26. Average *max-diameter* and *sum-diameter* of models produced by *Basic-PRM* on the *rigid-maze* problem. 4 runs of 3000 nodes.

measures reduce their frequency fluctuating around a more stable value, in roadmap-based planners these changes correspond to improvements in the internal pathways of the components and in incremental planners they correspond to extensions in currently existing pathways. We show the average *max-diameter* and *sum-diameter* of models produced by *Basic-PRM* (Figure 26), *OBPRM* (Figure 27(a)), and *Gauss-PRM* (Figure 27(b)) applied to the *rigid-maze* problem. The diameters in *MAPRM* had similar trends as those in *Gauss-PRM*, and the diameters in *Bridge-Test* had diameters similar in magnitude to *Gauss-PRM* with a stabilization closer to that of *Basic-PRM*. Also, we show the average *max-diameter* and *sum-diameter* of models produced by *RPP* (Figure 28(a)), *RRT*-Connect (Figure 28(b)), and *EST* (Figure 29) when applied to the *rigid-walls* problem. The diameters in *RRT*-Expand show similar trends to those of *RRT*-Connect, but with a slower growth for the only component growing. The stair-like growth of *RPP* happens because of its random expansion, many times from nodes that will not produce any structural expansion.

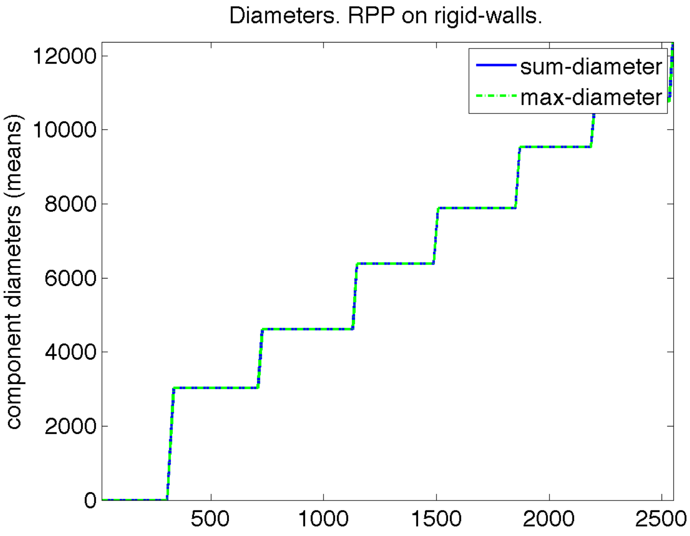


(a)

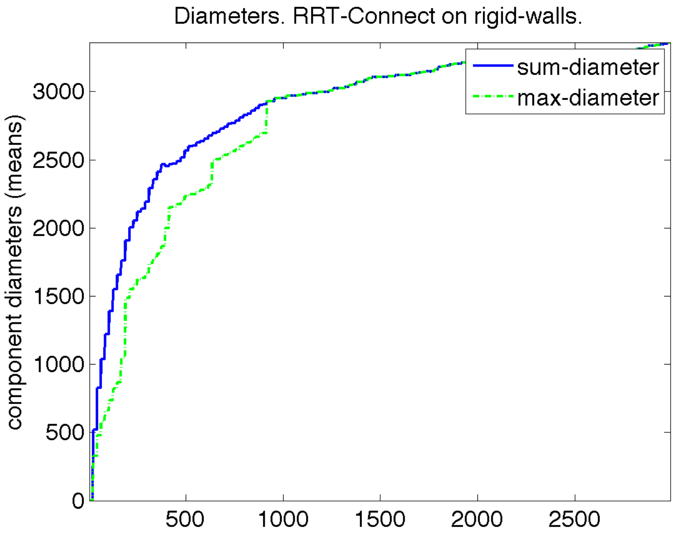


(b)

Fig. 27. Average *max-diameter* and *sum-diameter* of models produced by different planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner.



(a)



(b)

Fig. 28. Average *max-diameter* and *sum-diameter* of models produced by different planners on the *rigid-maze* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes for each planner.

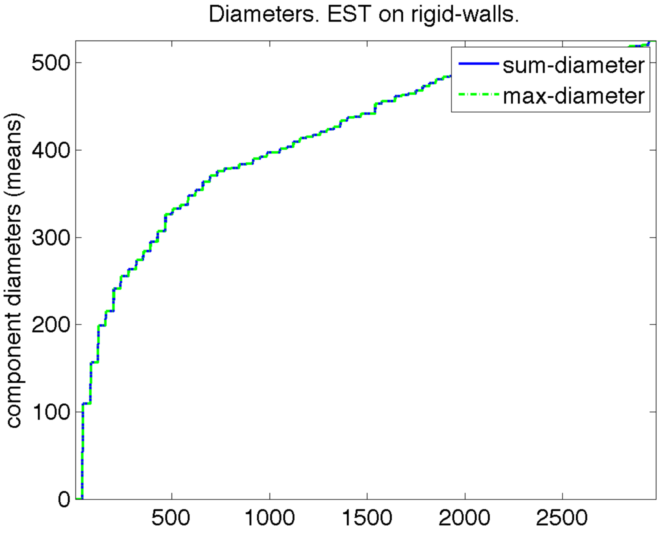


Fig. 29. Average *max-diameter* and *sum-diameter* of models produced by *EST* on the *rigid-maze* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes.

The standard deviations take many more samples to get to a lower value at the global level (Figures 30 and 31 for the roadmap-based planners on the *rigid-maze* and Figures 32 and 33 for the incremental planners on the *rigid-walls*) than at the node level (Table VI for the roadmap-based planners on the *rigid-maze* and Table VII for the incremental planners on the *rigid-walls*). This is mostly due to the higher variance in the ability of planners to find the right samples to find their way through the passages that connect the main components of the problem instance. The standard deviation of *Basic-PRM* is still at a high value at 3000 nodes because only half of the executions managed to stabilize their diameters for a single large component. Nevertheless, these variances go down when most planners have reached a stable value for the components when the planners are only modeling pathways that are internal to the components. The *max-diameter* is more unstable than the *sum-diameter*, but the combined use of both allows us to identify significant changes in the structure as a whole and in the biggest component of the model. Among

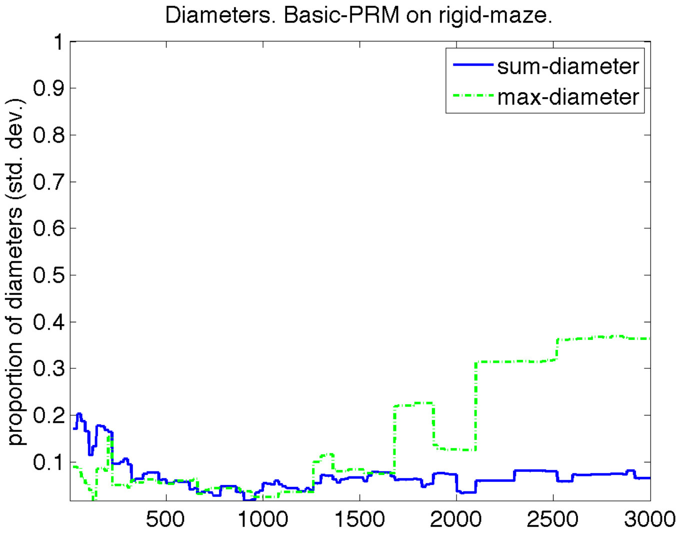


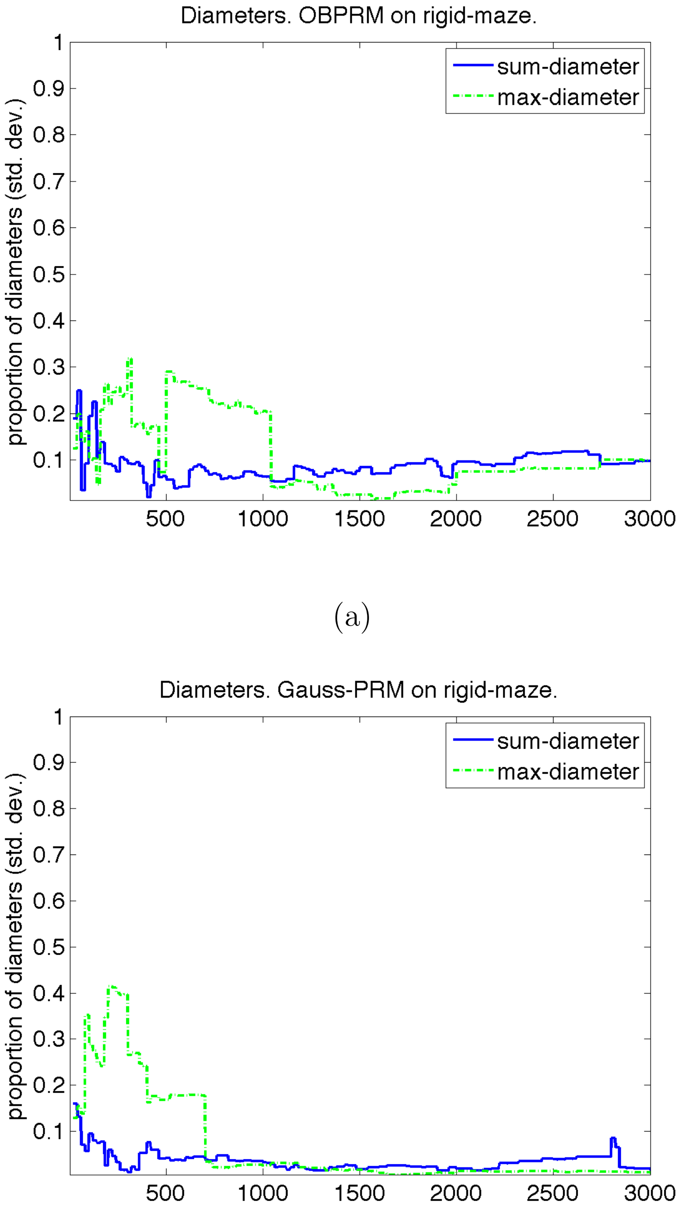
Fig. 30. Standard deviation of *max-diameter* and *sum-diameter* of models produced by the roadmap-based *Basic-PRM* planners on the *rigid-maze* problem. 4 runs of 3000 nodes.

the incremental planners, only *RPP* shows the highest standard deviation after 3000 nodes.

B. Stages of the Learning Process of Planners

As noticed in node-level and global-level metrics, we can identify three stages in the learning process. These three stages correspond to the ability of the planners to increase their knowledge about coverage, connectivity, and topology of C-Free, the subset of valid configurations of the problem. We call these three stages: quick learning, model enhancement, and learning decay.

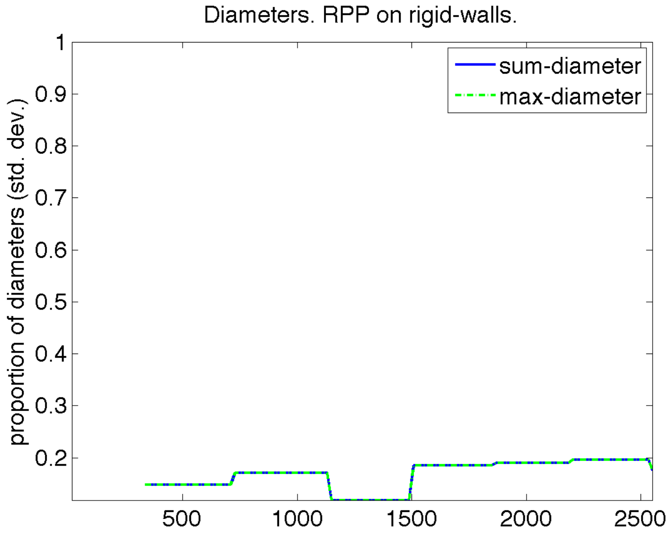
1. **Quick learning** — Coverage, connectivity, and topology of the model are quickly improved: *cc-create* nodes start off high and quickly decline in an exponential drop; in roadmap-based planners, *cc-merge* nodes peak briefly to get down to low values as *cc-create* nodes; *cc-expand* and *cc-oversample* nodes start a continuous growth. At the end of this stage, the planner has reached most of its potential coverage.



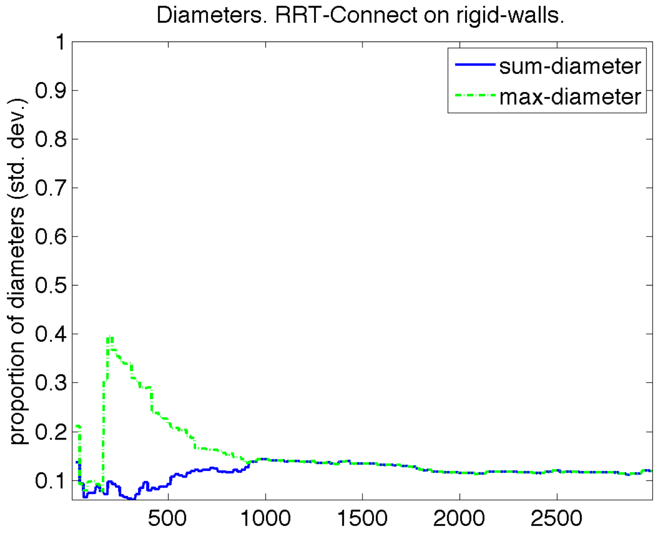
(a)

(b)

Fig. 31. Standard deviation of *max-diameter* and *sum-diameter* of models produced by different roadmap-based planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner.

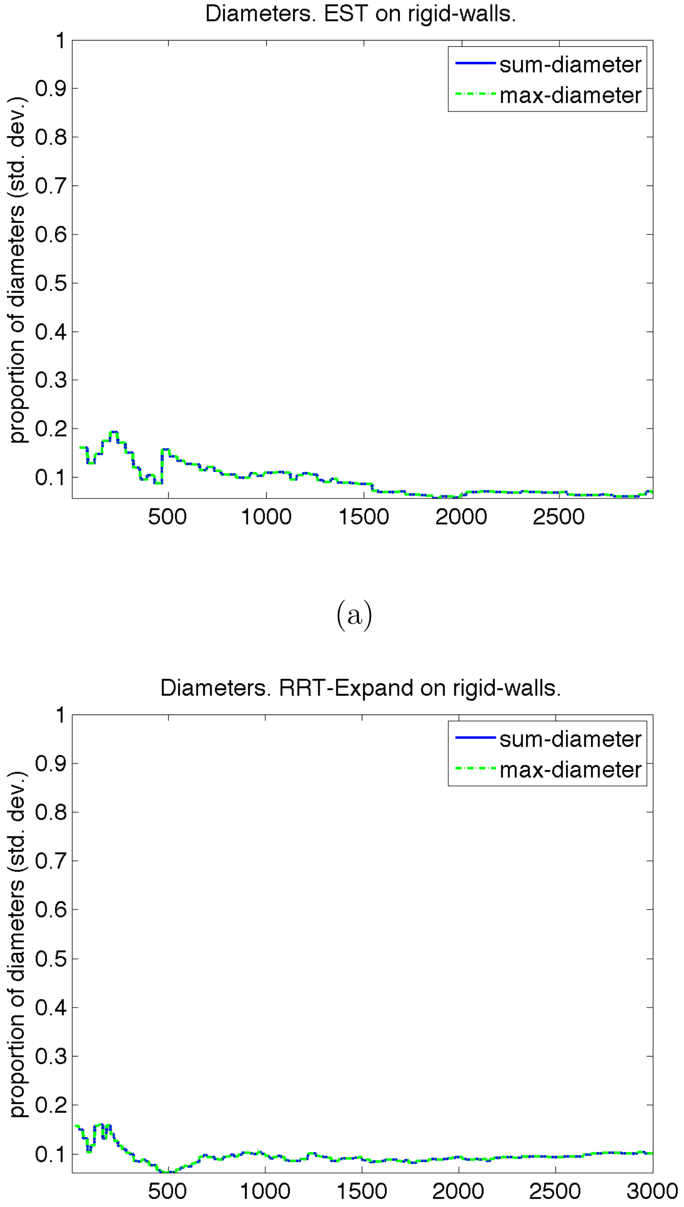


(a)



(b)

Fig. 32. Standard deviation of *max-diameter* and *sum-diameter* of models produced by different incremental planners on the *rigid-walls* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes for each planner.



(a)

(b)

Fig. 33. Standard deviation of *max-diameter* and *sum-diameter* of models produced by different incremental planners on the *rigid-walls* problem. (a) *EST*. (b) *RRT*-Expand. 8 runs of 3000 nodes for each planner.

2. **Model enhancement** — Coverage of the model still improves, but at a slower rate. Connectivity and topology are still undergoing significant changes. The reduction in the rate of learning is because it takes more effort to find the key samples that join together the main pathways than to cover the space. The rate of change in all node types reduces while *cc-expand* nodes slowly get closer to its highest values. The diameter measures also approach a stable value. At the end of this stage, the planner has reached most of its potential connectivity and topology.

3. **Learning decay** — Coverage, connectivity, and topology of the model change at a higher expense, although unexplored areas are still likely to be found. *cc-expand* nodes gradually decline; *cc-oversample* nodes gradually increase. Frequency of changes in diameters slows down.

1. How Stage Transitions Can Be Detected?

In our experiments we noticed trends in multiple runs of node-level and global-level metrics that help us to identify the transitions between planning stages. Metrics at the node level have a very low standard deviation at the beginning of the *model enhancement* stage, their most dramatic changes decrease sooner than the metrics at the global level. We are mostly interested in identifying the start of the *learning decay* when the current planning strategy reduces its chances to make significant improvements in the model. We notice that at the beginning of the *learning decay* stage, metrics at the global level have significantly smaller changes and, also, they have a low standard deviation. Figures 34 and 35 show the means of the change in *max-diameter* and *sum-diameter* of models produced by different roadmap-based planners applied to the *rigid-maze* problem.

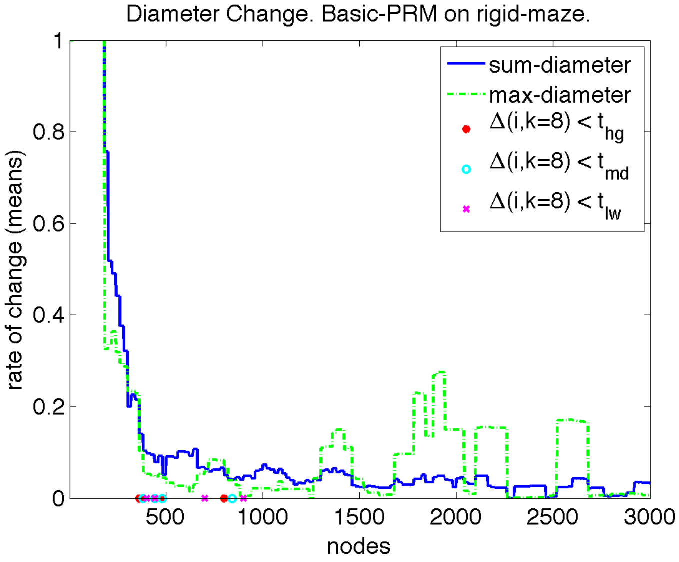
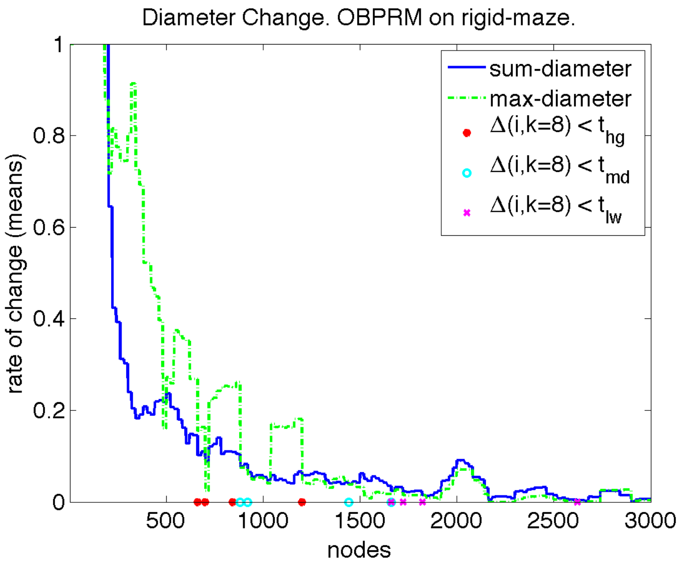


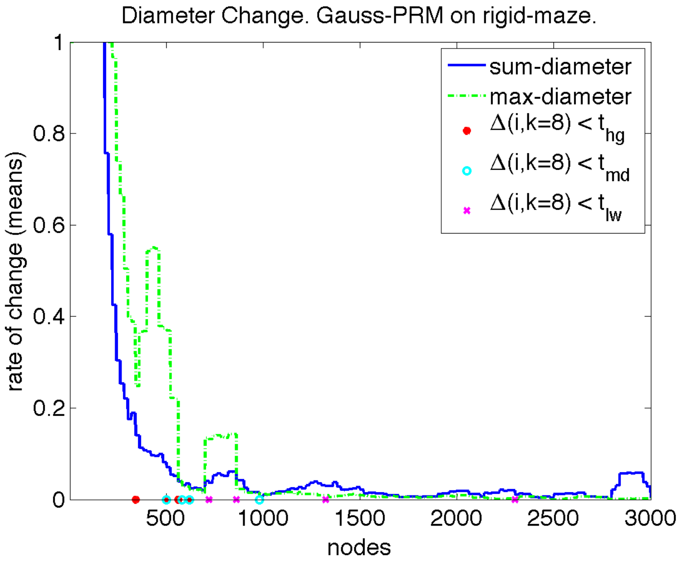
Fig. 34. Average change in *max-diameter* and *sum-diameter* of models produced by the roadmap-based *Basic-PRM* planner on the *rigid-maze* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. 4 runs of 3000 nodes.

The boundary between the stages is fuzzy rather than sharp. During the *quick learning* stage the changes in diameter measures are more dramatic starting in values even larger than 1.0, and quickly declining to values around 0.1. During the *model enhancement* stage the changes in diameters are much smaller between 0.15 and 0.02. During the *model enhancement* stage, the diameters have very small changes, most of the time below 0.05. Figures 34 and 35 mark the nodes where individual runs have changes below $t_{hg} < 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$. We notice that $t_{md} < 0.05$ happens around the end of the *model enhancement* stage and $t_{lw} < 0.02$ happens around the beginning of the *learning decay* stage.

Incremental planners also allow the use of the diameter measures to mark the start of the *learning decay* stage of planning. Figures 36 and 37 show the average change in *max-diameter* and *sum-diameter* of models produced by incremental plan-



(a)



(b)

Fig. 35. Average change in *max-diameter* and *sum-diameter* of models produced by different roadmap-based planners on the *rigid-maze* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner.

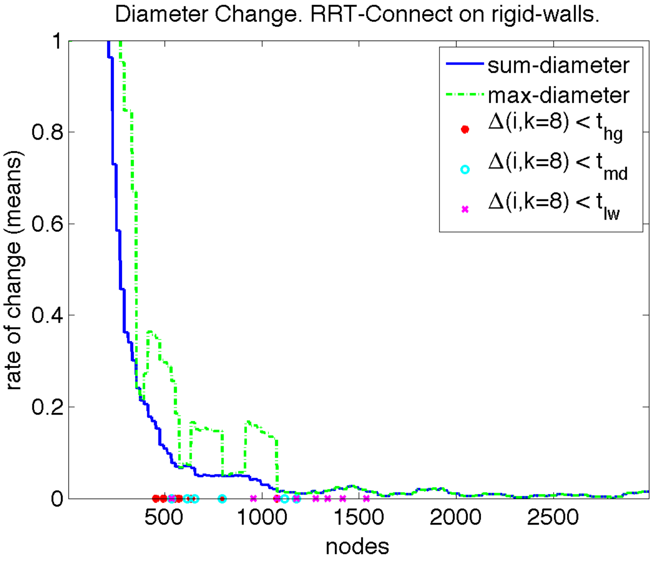
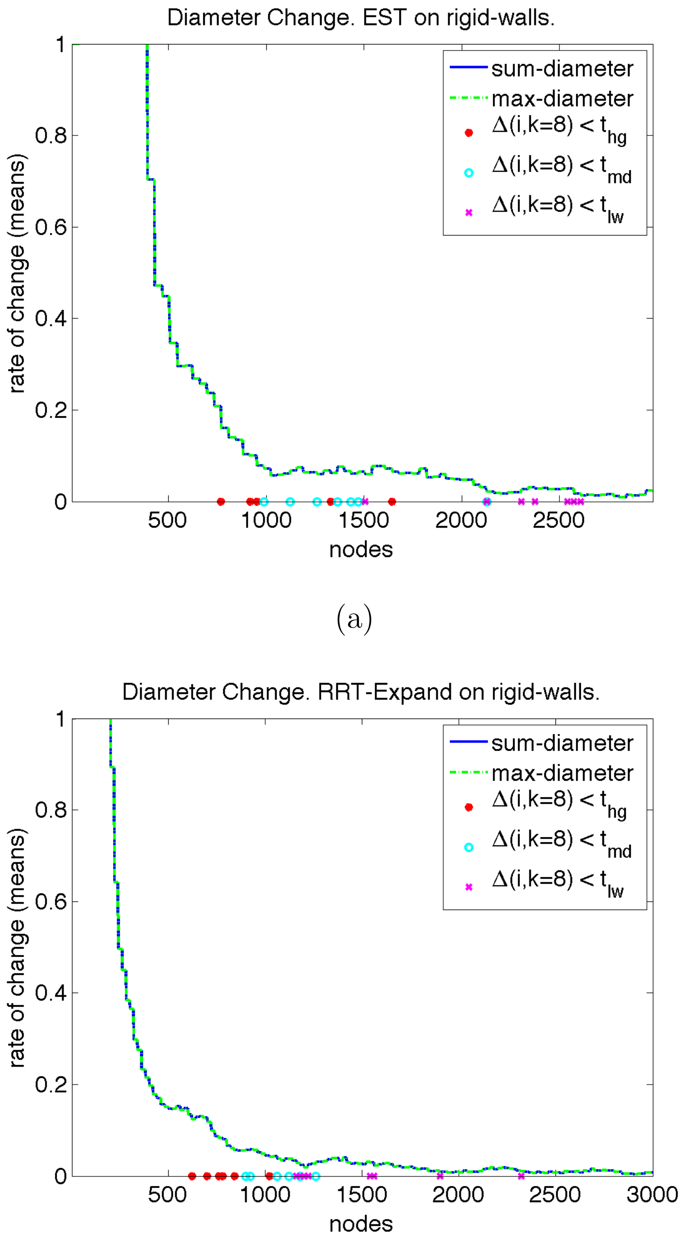


Fig. 36. Average change in *max-diameter* and *sum-diameter* of models produced by the incremental *RRT*-Connect planner on the *rigid-walls* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. 8 runs of 3000 nodes.

ners applied to the *rigid-walls* problem and the nodes where the diameters' change in individual runs fall below $t_{hg} < 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$. Similarly to roadmap-based planners, $t_{md} < 0.05$ happens close to the end of the *model enhancement* stage and $t_{lw} < 0.02$ happens close to the beginning of the *learning decay* stage.

2. What Can Be Done When *Learning Decay* Starts?

One important practical issue in sampling-based planners is to determine how large a roadmap is needed to model a motion planning instance. This issue has been traditionally approached through a time-consuming trial-and-error process which frequently results in model sizes larger than needed. Although during the *learning decay* stage the planner is still potentially able to find samples that improve the model, these samples are found at increasing costs. We can stop planning or switch sampling



(a)

(b)

Fig. 37. Average change in *max-diameter* and *sum-diameter* of models produced by different incremental planners on the *rigid-walls* problem. The moment when the rate of change of both diameters fall below three thresholds ($t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$) for individual runs is shown with dots over the nodes axis. (a) *EST*. (b) *RRT*-Expand. 8 runs of 3000 nodes for each planner.

strategies by evaluating global-level metrics to detect the *learning decay* stage as we did in the Incremental Map Generation (IMG) method [65].

IMG constructs C-Space models iteratively. In each iteration, a group of nodes and connections are sampled and added to the model. Then, a set of evaluations are performed to determine whether a different sampling strategy can be used or if model construction can be stopped. At the core of the evaluation are the diameter global-level metrics which determine if the model is at the start of the *learning decay* as an indication of diminished improvements. This is when planning can be stopped or a different strategy can be tried. The desired rate of change in global-level metrics can be adjusted to essentially specify the duration of the *learning decay* desired.

Roadmap-based planners, incrementally-exploring planners and adaptive planners can be easily incorporated into the IMG framework, as we did with all the planners discussed in this work and with the Hybrid PRM planner [26]. More information can be found in [65].

C. Distribution of Nodes in the C-Space

We identify distinct regions of the C-Space found by different planners by building groups of nodes based on their local information accessed through node-level metrics.

1. What Is the Population Distribution of Regions for Different Planners?

We identify regions with low, medium and high visibility in different planners. The three stages of learning are reflected in the region-level metrics as can be seen in Figures 38 and 39 which show the average population distribution of visibility regions in models produced by *Basic-PRM*, *OBPRM*, and *Gauss-PRM*when applied to the *rigid-maze* problem. *Basic-PRM* is unable to produce a significant amount

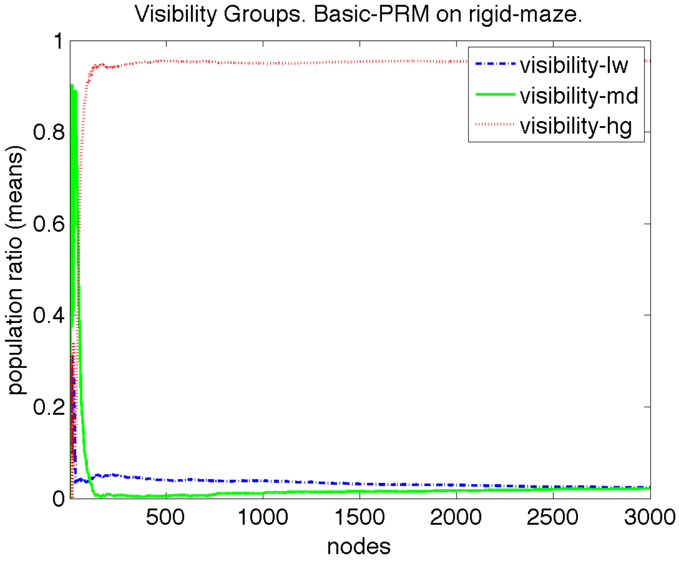
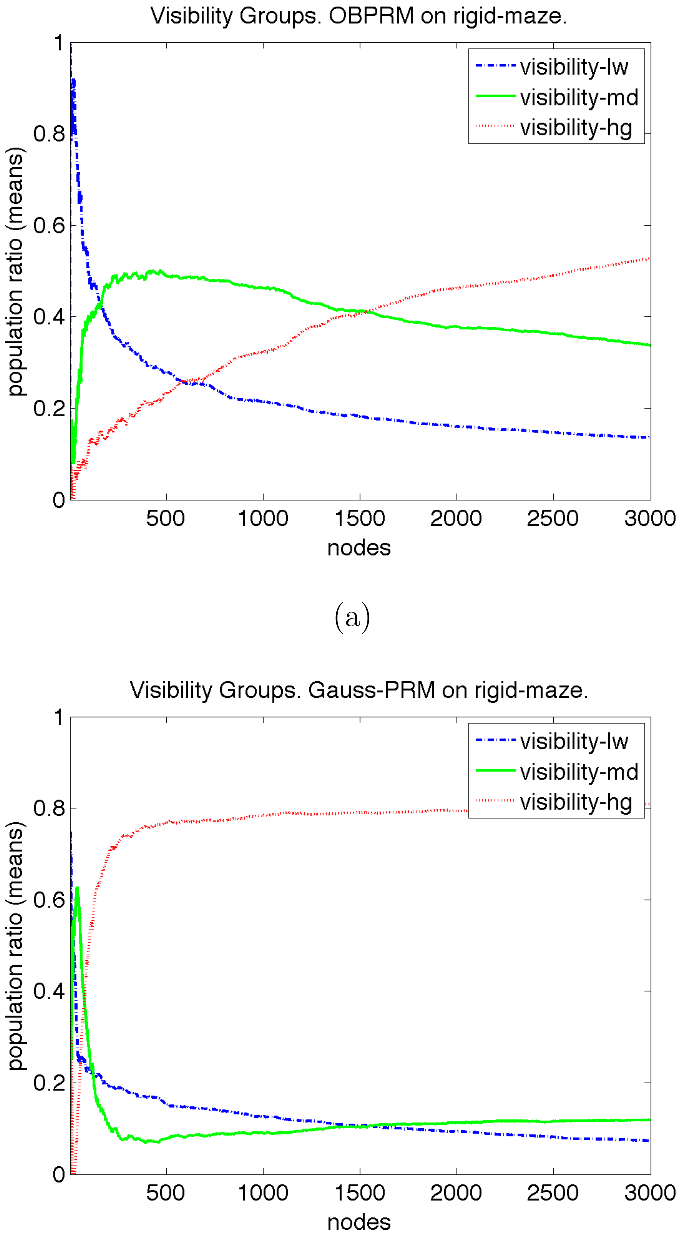


Fig. 38. Average population distribution of visibility regions in models produced by the roadmap-based *Basic-PRM* planner on the *rigid-maze* problem. 4 runs of 3000 nodes.

of low-visibility nodes. On the other hand *OBPRM* produces more high-visibility and medium-visibility nodes. *Gauss-PRM* is between the two. The population of visibility regions in *MAPRM* (not shown) is similar to *Gauss-PRM*, partially because nodes inside narrow passages produced by *MAPRM* have higher visibility than those produced by other strategies. The population of visibility regions in *Bridge-Test* (not shown) is similar to *OBPRM*.

Visibility regions in incrementally exploring planners also depend on their sampling distribution. Figure 40 shows the average population distribution of visibility regions in models produced by *RPP*, and *RRT*-Connect when applied to the *rigid-walls* problem. We notice that both methods have about 40% of high-visibility nodes. In addition, *RPP* visibility groups amount for about 100% of the nodes, meaning that most nodes are used as growth sites. In contrast, *RRT*-Connect only uses about 70% of the nodes as growth sites, with about 10% with medium visibility and less than



(a)

(b)

Fig. 39. Average population distribution of visibility regions in models produced by different planners on the *rigid-maze* problem. (a) *OBPRM*. (b) *Gauss-PRM*. 4 runs of 3000 nodes for each planner.

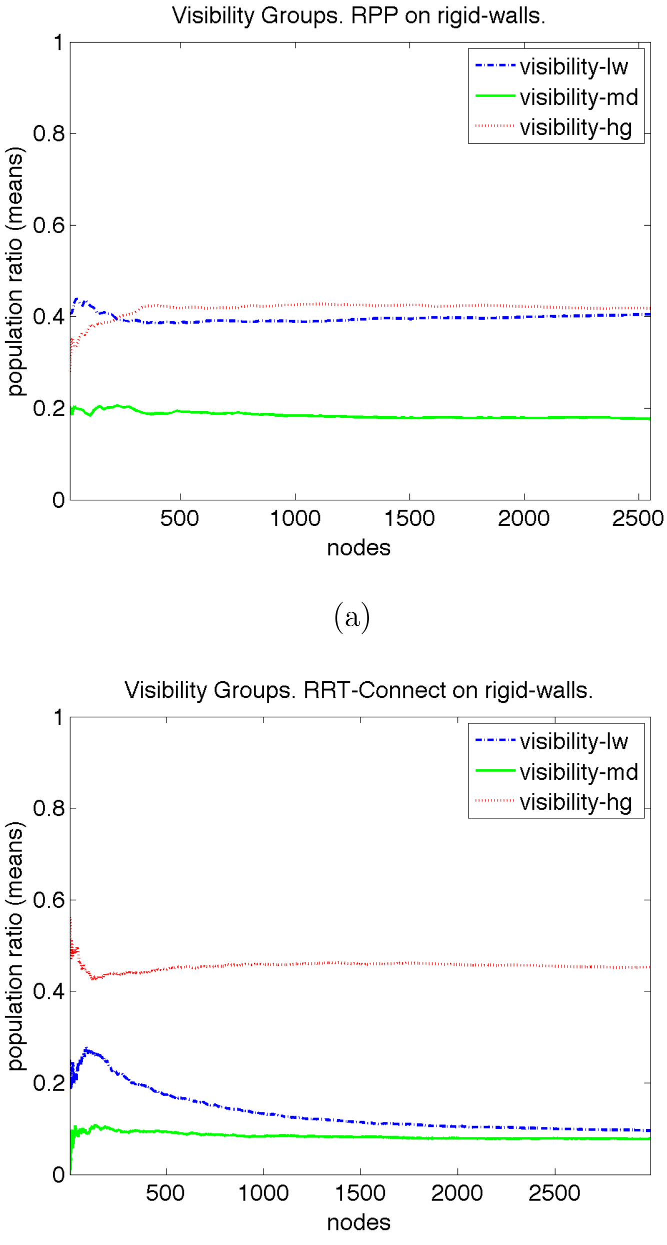
TABLE VIII

Population Distribution of Visibility Regions in *rigid-maze*. Deviation

	Nodes for Standard Deviation	
Planner	< 0.10	< 0.05
Basic-PRM	30	80
OBPRM	55	150
Gauss-PRM	30	140
MAPRM	200	240
Bridge-Test	80	210

15% with low-visibility. Although *RPP* produces many more low-visibility nodes, they are likely to be local minima which produce other low-visibility nodes while trying to escape. In contrast, the Voronoi-region reduction of the *RRT* methods allows them to increase coverage in a more uniform way and to find low-visibility nodes in wider areas of the space. *RRT*-Expand and *EST* has similar trends as *RRT*-Connect, with less medium-visibility and low-visibility nodes.

The variability of the population distribution of regions across runs is very low as can be seen in Table VIII and in Table IX. These tables show the number of nodes after which the standard deviation of the population in all the regions falls below 0.10 and 0.05 for roadmap-based planners (Table VIII) and incremental planners (Table IX).



(a)

(b)

Fig. 40. Average population distribution of visibility regions in models produced by different incremental planners on the *rigid-walls* problem. (a) *RPP*. (b) *RRT*-Connect. 8 runs of 3000 nodes for each planner.

TABLE IX

Population Distribution of Visibility Regions in *rigid-walls*. Deviation

	Nodes for Standard Deviation	
Planner	< 0.10	< 0.05
RPP	15	189
EST	10	40
RRT-Expand	25	166
RRT-Connect	20	108

2. How Effective Are Planners in Biasing towards Highly-Constrained Regions?

Planners that bias their sampling of configurations and motions towards highly constrained regions are more likely to find motions through narrow passages of the space. Figure 41 shows the subgraphs of the low-visibility regions (*visibility* $< 1/3$) in individual runs of *Basic-PRM*, *OBPRM*, and *Gauss-PRM* at 2000 nodes, long after global-level metrics have stabilized. Most low-visibility nodes are inside the narrow passages, but *Basic-PRM* nodes are few and badly distributed, *Gauss-PRM* nodes are more and better distributed in the passage, and *OBPRM* nodes are much better distributed in the narrow passage. It is worth noting that the low visibility nodes generated by *OBPRM* and *Gauss-PRM* outside of the passage might have a higher visibility if there were more nodes in the open space to connect them more easily.

3. How Do Coverage Regions Evolve in Incremental Planners?

We evaluate the *coverage rate* of the *RRT*-Expand, *EST*, *RRT*-Connect, and *RPP* incremental planners when applied to the *rigid-maze*, and *rigid-walls* problems as

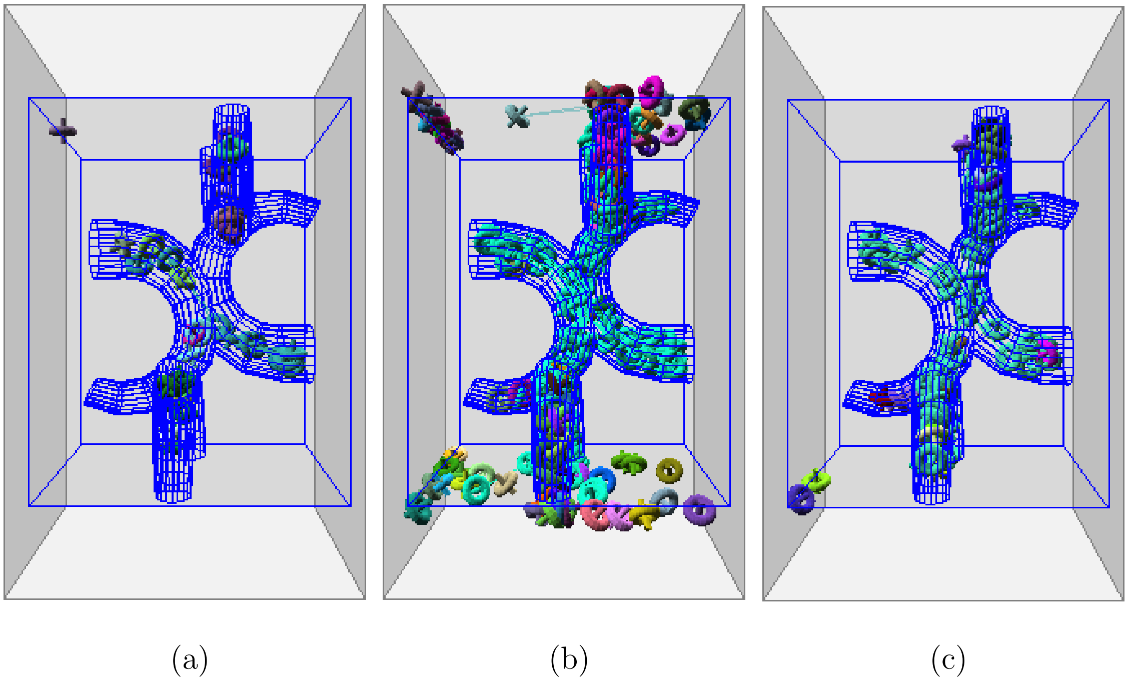


Fig. 41. Distribution of low-visibility nodes (*visibility* $< 1/3$) in models produced by one run of different planners on the *rigid-maze* problem after global-level metrics have converged. (a) *Basic-PRM*. (b) *OBPRM*. (c) *Gauss-PRM*.

shown in Figure 42. We also evaluated the number of samples that each planner needed to produce a node whose positional *DOFs* lie in the same workspace chamber as the goal configuration (note that this does not imply that the goal is reachable) as shown in Table X. All the planners started their search at the start configuration. The goal-biased *RRT*-Connect and *RPP* were provided the goal configuration. Region radius was defined as 15% larger than the expansion step used by the planner.

RRT-Expand and *EST* have very similar expansion philosophies, they try to expand towards unexplored areas of the C-Space. Nevertheless, *RRT*-Expand has a much higher coverage rate than *EST*. This is because *RRT*-Expand is biased towards the biggest unexplored Voronoi regions whereas *EST* is biased towards regions with fewer nodes regardless of whether they have been explored or not. Among the goal-biased planners, *RRT*-Connect is better than *RPP*, but *RPP* has a very good

TABLE X

Number of Samples Needed for Planners to Get to Goal Region

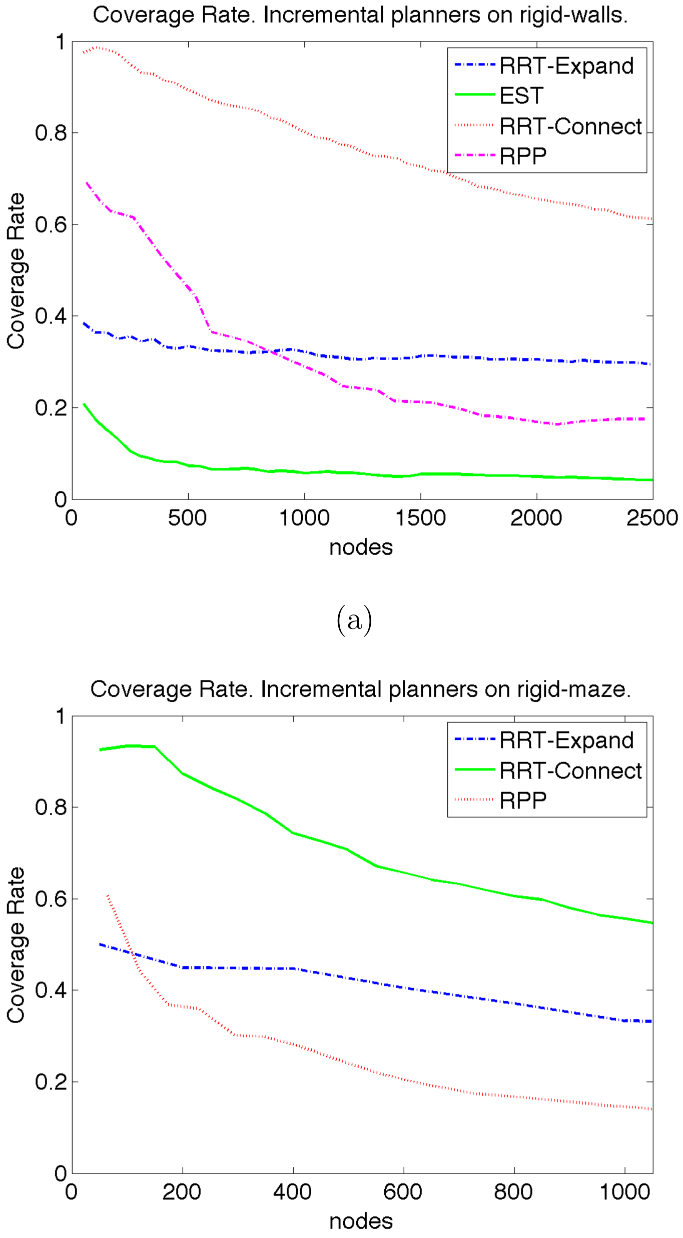
Problem	**Method**	**Samples to Goal Region**
rigid-walls	*RRT*-Expand	1000
	EST	1400
	RRT-Connect	500
	RPP	2300
rigid-maze	*RRT*-Expand	5800
	EST	failed
	RRT-Connect	700
	RPP	1800

performance and is able to make its way through the chambers.

4. How Can We Adapt Planning Based on Region Complexity?

Identifying highly constrained regions allow us to adapt the sampling strategy based on the complexity of the distinct regions of the problem, with additional sampling in low-visibility regions and scarcer sampling in high-visibility regions.

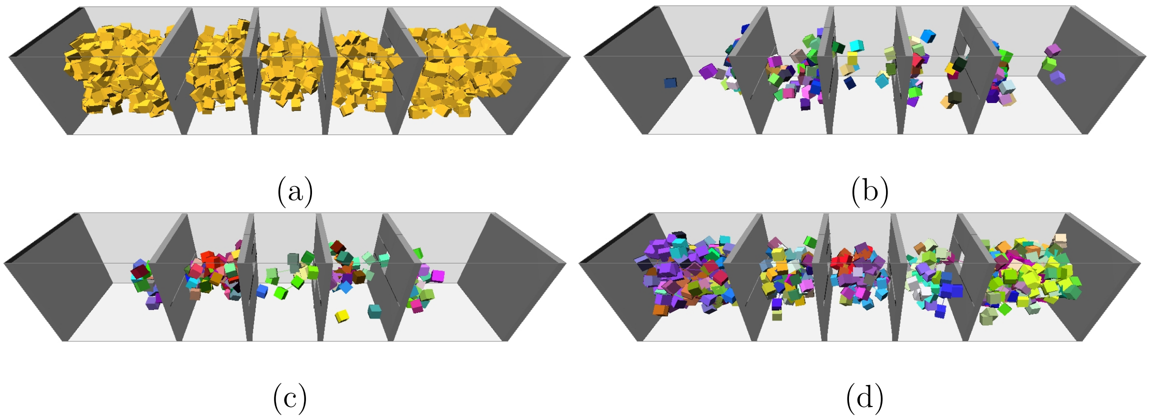
We can use the different regions to dynamically adapt sampling. For example, Figure 43 shows the 2000-node model produced when running *RRT*-Connect on the *rigid-walls* problem. We can see the low-visibility nodes that are close to C-obstacles and inside narrow passages. These low-visibility nodes are good candidates to strengthen the sampling bias when the global-level metrics indicate that the *learning decay* has started, or when the coverage growth is stalled.



(a)

(b)

Fig. 42. Coverage rate of incremental planners *RRT*-Expand, *EST*, *RRT*-Connect, and *RPP*when mapping two problems. (a) *rigid-walls*. (b) *rigid-maze*, *EST* is close to 0 most of the time.



(a)

(b)

(c)

(d)

Fig. 43. Visibility regions in a model produced by *RRT*-Connect in the *rigid-walls* problem. (a) 2000-node model. (b) Low-visibility region: *visibility* < 1/3. (c) Medium-visibility region: 1/3 <= *visibility* < 2/3. (d) High-visibility region: 2/3 <= *visibility*.

D. Comparison of Metrics at the Start of Learning Decay for Different Roadmap-Based Planners

We compare planners at the time of the start of the *learning decay* stage that we detect using the global-level metrics described above. We compute the population distribution for node types produced by *Basic-PRM*, *OBPRM*, *MAPRM*, *Gauss-PRM*, and *Bridge-Test* applied to the *rigid-maze* (Figure 44), *rigid-hook* (Figure 45), and *serial-hook-5* (Figure 46) problems. We also show the number of nodes, modeling time, *max-diameter*, *sum-diameter*, and percentage of witness queries solved at the time of the start of the *learning decay* stage in Table XI for the *rigid-maze* problem, Table XII for the *rigid-hook* problem, and Table XIII for the *serial-hook-5* problem.

In the *rigid-maze* and in the *rigid-hook* problems, *Basic-PRM* produces mostly *cc-oversample* nodes. In contrast, *OBPRM* produces many *cc-expand* nodes. *MAPRM* produces many more *cc-expand* nodes than *Basic-PRM*, but it also produces many *cc-oversample* nodes because of its bias that creates many close-by nodes that are easy to connect locally, it may be possible to filter some of these nodes without affecting the quality of the model. *Gauss-PRM* has medium quality nodes. *Bridge-Test* produces

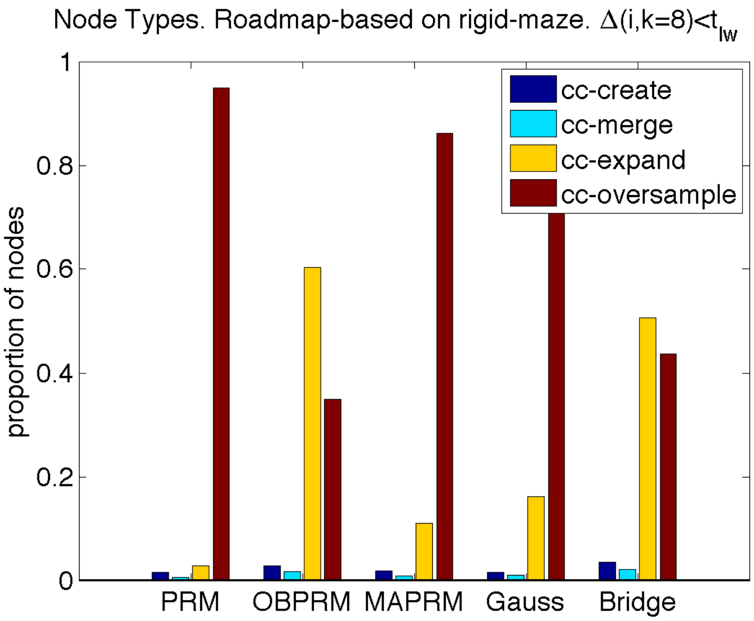


Fig. 44. Average population distribution of node types produced by different planners on the *rigid-maze* problem at the beginning of the *learning decay* stage.

TABLE XI

PLANNERS AT START OF *Learning Decay* ON *rigid-maze*

		Modeling	**Diameters**		**Witness**
Problem	**Nodes**	**Time [s]**	***max***	***sum***	**%**
Basic-PRM	610	171	226	624	0
OBPRM	1,955	311	648	700	100
MAPRM	810	96	600	610	100
Gauss-PRM	1,300	265	612	631	100
Bridge-Test	3,000	5,666	1,155	1774	100

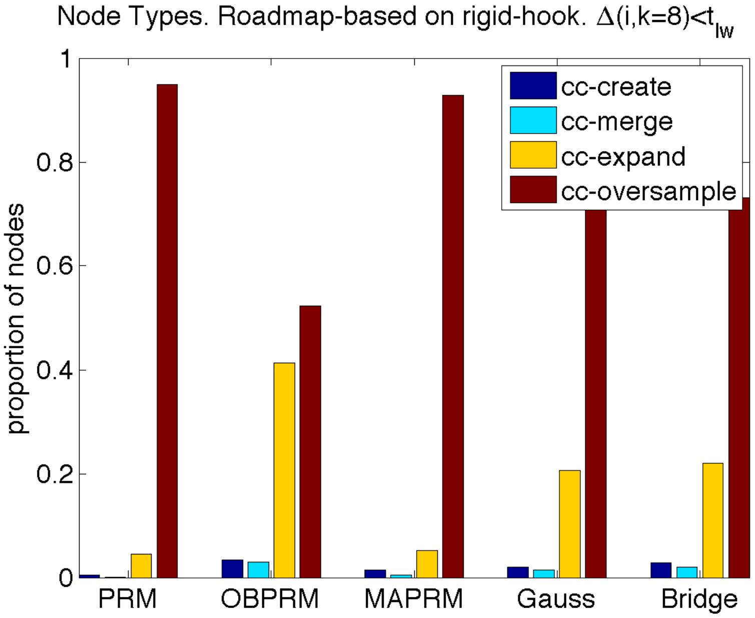


Fig. 45. Average population distribution of node types produced by different planners on the *rigid-hook* problem at the beginning of the *learning decay* stage.

many *cc-expand* nodes but at a very high price (one order of magnitude as large as all the other methods). We also notice that in the *rigid-maze* problem, *Basic-PRM* and *Bridge-Test* reach a *max-diameter* that is still far from the *sum-diameter* when all the other planners managed to make them very similar. In the *rigid-hook* problem only *OBPRM* was able to do the same.

In the *serial-hook-5* problem, *Basic-PRM* surprisingly produces many *cc-expand* nodes, but later well into the *learning decay* the *cc-oversample* nodes take over. We can also notice that *Basic-PRM*is the one whose *max-diameter* is the smallest among all planners. *OBPRM MAPRM*, and *Gauss-PRM* also produce many *cc-expand* nodes with *Gauss-PRM* being the one with the largest proportion, but with *MAPRM* and *OBPRM* having their *max-diameter* being the closest to the *sum-diameter* which reflects better the C-Space of this problem with one large component dominating the problem. *Bridge-Test* produces the smallest number of *cc-expand* nodes and it is also

TABLE XII

Planners at Start of *Learning Decay* on *rigid-hook*

		Modeling	**Diameters**		**Witness**
Problem	**Nodes**	**Time [s]**	***max***	***sum***	**%**
Basic-PRM	2558	87	57	158	0
OBPRM	1998	42	144	147	100
MAPRM	2100	74	85	263	0
Gauss-PRM	2963	543	177	250	60
Bridge-Test	2048	1618	169	200	100

the most expensive.

E. Metrics in High-DOF Problems

In this Section we analyze the *RRT*-Connect planner when mapping the high-*DOF* *serial-spring-98* problem with the metrics introduced in this work. Our results illustrate the power of the metrics to characterize the planning process in high-*DOF* problems just as in the low-*DOF* problems discussed in previous sections. We let the planner keep running even after the two growing trees join together to evaluate the ability of the planner to keep learning about the problem.

Node-level metrics (Figure 47) show that *RRT*-Connect stabilizes at about 60% of *cc-oversample* nodes and about 40% of *cc* create nodes. This indicates that the problem has many hard to connect areas, but still it needed a big percentage of nodes to get there. We also see that the *learning decay* stage started after a few hundreds of nodes.

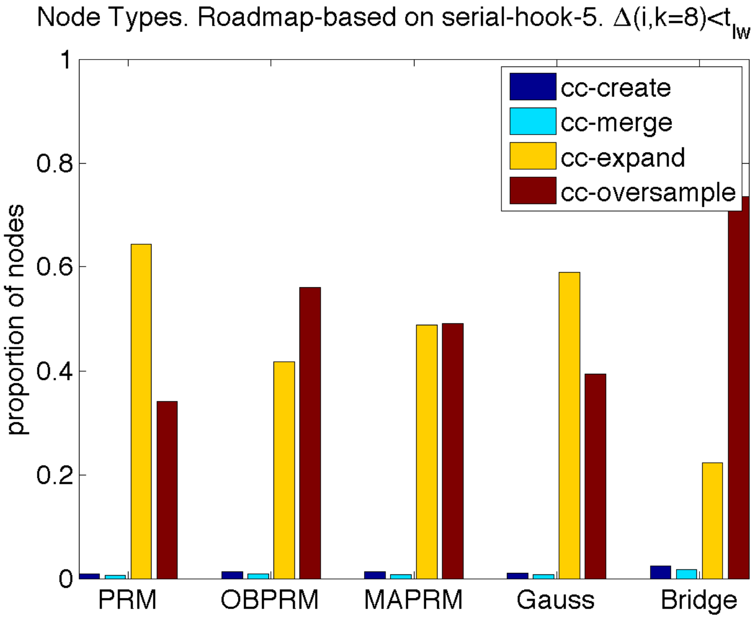


Fig. 46. Average population distribution of node types produced by different planners on the *serial-hook-5* problem at the beginning of the *learning decay* stage.

TABLE XIII

PLANNERS AT START OF *Learning Decay* ON *serial-hook-5*

		Modeling	**Diameters**		**Witness**
Problem	**Nodes**	**Time [s]**	***max***	***sum***	**%**
Basic-PRM	1430	479	324	550	30
OBPRM	1750	437	471	578	80
MAPRM	1478	606	478	591	60
Gauss-PRM	1473	585	356	545	30
Bridge-Test	1600	2748	529	711	60

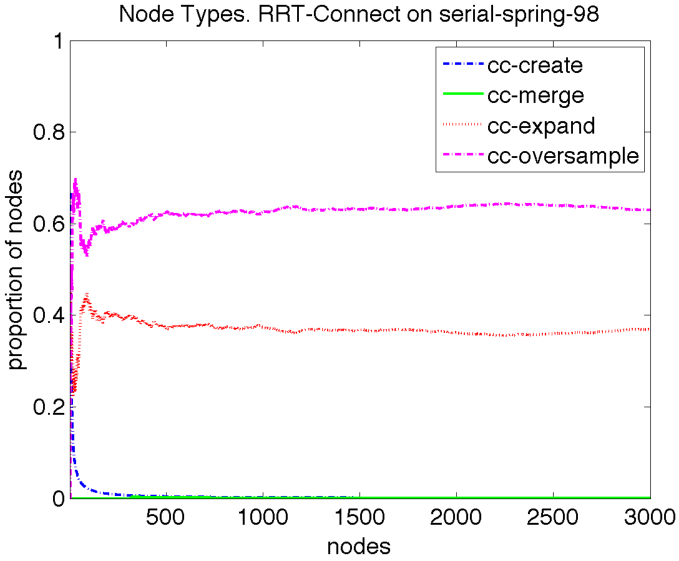
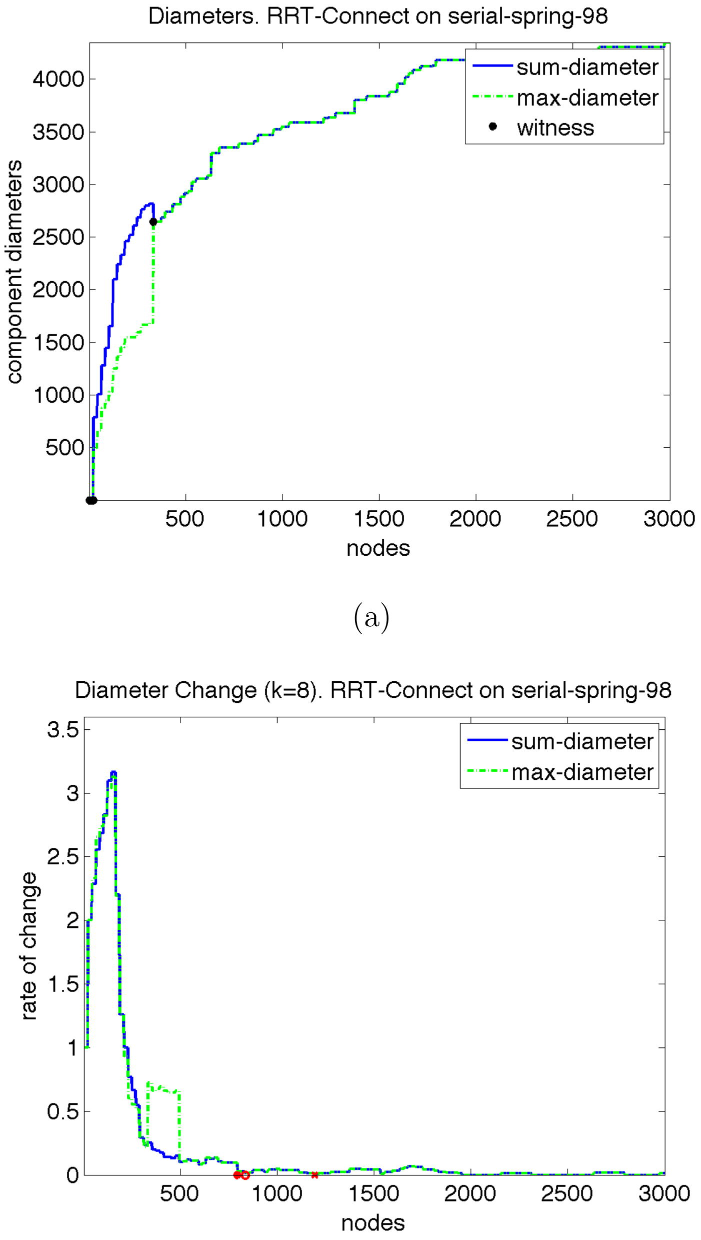


Fig. 47. Population distribution of node types produced by *RRT*-Connect when modeling the *serial-spring-98* problem.

Global-level metrics (Figure 48) confirm that the *learning decay* stage is starting after a few hundreds of nodes. The trees join together at about 400 nodes, shown by the diameter measures joining together as well. Although the witness query can be solved at this point in time, the planner is still able to keep learning about the problem. The points shown in Figure 48(b) at about 800 nodes and 1200 nodes show the moment when the rate of change of both diameters fall below thresholds $t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$. These points are when we would stop planning or switch strategies.

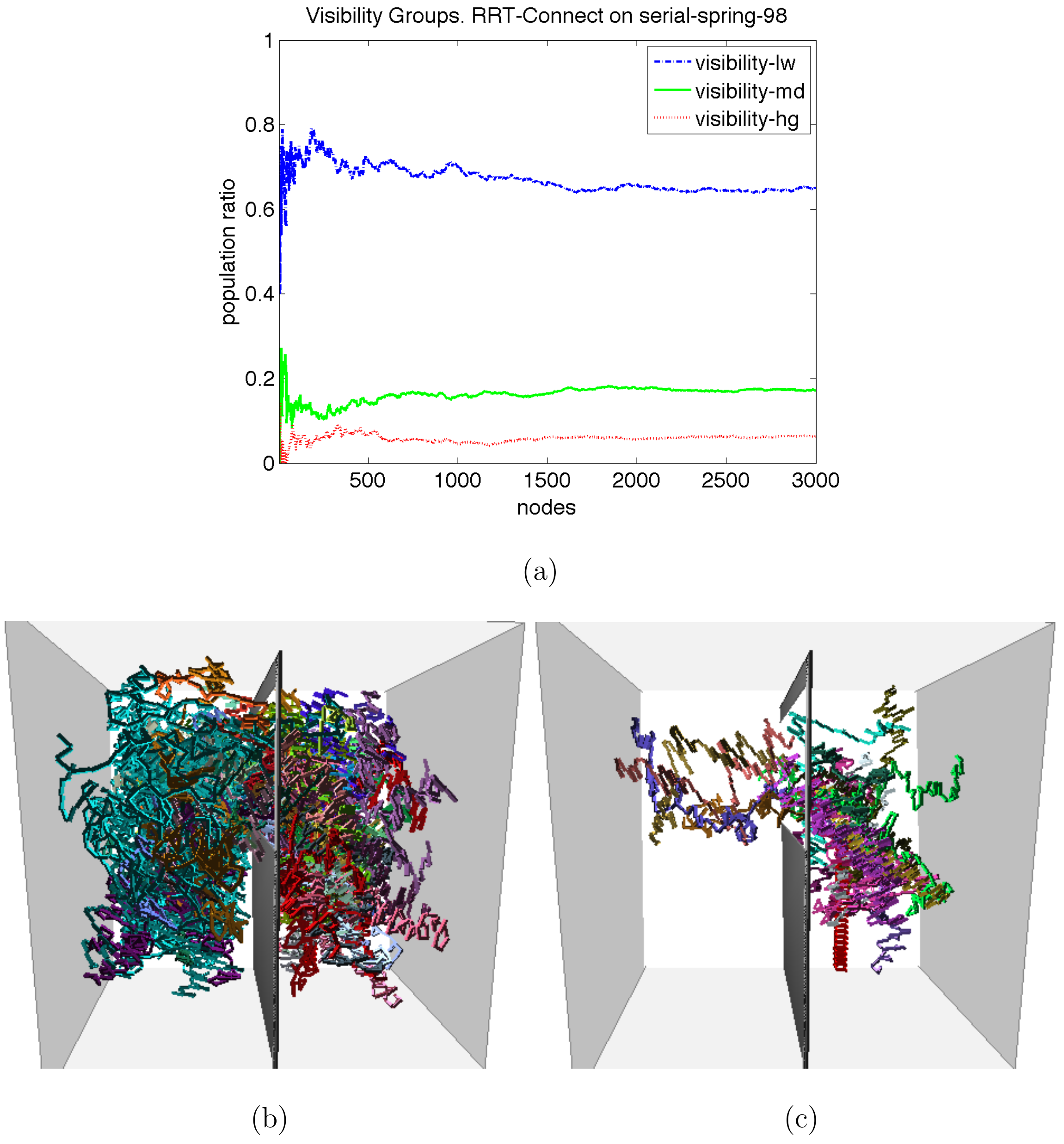
Region-Level metrics (Figure 49) show that the planner finds many low-visibility regions and very few high-visibility regions.



(a)

(b)

Fig. 48. Evolution of global-level metrics for one instance of *RRT*-Connect on the *serial-spring-98* problem. (a) *max-diameter* and *sum-diameter*. (b) changes in *max-diameter* and *sum-diameter* (also shown, the moment when the rate of change of both diameters fall below thresholds $t_{hg} = 0.1$, $t_{md} < 0.05$, and $t_{lw} < 0.02$).



(a)

(b)

(c)

Fig. 49. Regions found in the *serial-spring-98* problem with *RRT*-Connect (at about 400 nodes when the expanding trees join together). (a) Population distribution of visibility regions. (d) Low-visibility region at 400 nodes. (c) High-visibility region at 400 nodes.

CHAPTER IX

CONCLUSIONS

Motion planning is an intractable problem that can be approached through methods that build an approximate model of potential motions through random sampling. This technique can be applied to many types of robots as long as appropriate functions to evaluate potential robot configurations and motions can be defined. This flexibility allows us to address applications for planning and studying motions in many areas such as robotics, biology and chemistry, manufacturing and mechanical design, animation of characters, and video games.

Although there have been intensive efforts to take advantage of geometric features of the obstacles and the robot that are likely to favor a better exploration of the areas of the C-Space that are more challenging, these efforts have not yet produced a planner that is best suited for every case. In fact, we do not know how to choose among the many planners available for each particular instance of the motion planning problem. This has motivated recent efforts in adaptive planning to evaluate features discovered during the exploration of the C-Space in order to dynamically adapt the planning strategy.

This research contributes to this problem by proposing metrics that allow us to evaluate the features discovered by sampling-based motion planners that correlate with three important properties of the C-Space: coverage, connectivity, and topology. In order evaluate different aspects of these properties we discussed the validity, visibility, covered region, connectability and homotopy in C-Space. A direct measurement of these features is tantamount to the unfeasible computation of the C-Space. Instead, the metrics discussed here measure features that correlate with the ability of the planner to sustain its learning about the C-Space properties mentioned. In our discussion we assumed models where samples and connections are tested with binary validity tests. Nevertheless, we briefly discussed potential ways to address other types of models such as those that use potential evaluations of motions.

In the description of the metrics and their applications we applied roadmap-based and incrementally-exploring planners to several instances of the motion planning problem. The roadmap-based planners discussed were *Basic-PRM*, *OBPRM*, *Gauss-PRM*, *Bridge-Test*, and *MAPRM*. The incrementally-exploring planners discussed were *RRT*-Expand, *RRT*-Connect, *RPP*, and *EST*. The motion planning instances have a variety of densities and difficulty.

The metrics operate at multiple levels: node level, global level, and region level. At the node level, they allow us to measure the contributions of nodes and connections to the coverage and connectivity of the model and the local features around them. At the global level, they allow us to measure the transformations on the global structure of the model that lead to improvements in topology, and therefore in coverage and connectivity. At the region level, they allow us to identify regions composed by groups of nodes that share similar node and global level metrics. The joint use of these metrics enables us to understand better sampling-based planners to improve the process.

At the node level, we defined different types of nodes: *cc-create*, *cc-merge*, *cc-expand*, and *cc-oversample*. The first three types represent significant improvements at this level, while the later does not. We defined an affordable approximation to estimate node types in both roadmap-based and incrementally-exploring planners. Also, we defined a mechanism to estimate the visibility around growth sites whose only additional cost is to maintain the number of growth attempts and successful growths for each node. We learned that all planners show a very low standard deviation for multiple runs after just a few nodes in the model. This indicates that each planner has a consistent sampling mechanism for each problem.

At the global level, we estimated the structural changes that the model is undergoing and the homotopy classes, or pathways, represented in it by measuring an approximation of the diameter of components. The overhead spent in these metrics is very low and the accuracy of the approximation is sufficient for our needs. We also defined a mechanism to quantify the rate of change in global structure and to detect when this rate slows down.

At the region level, we grouped samples into regions using basic clustering strategies based on their node-level metrics. These regions have many applications, for example to identify the samples that have high constraints and have higher chances to be inside narrow passages.

These metrics only rely on indirect measurements that do not depend on the dimensionality of the problem in contrast with reachability-based evaluations that directly compare the connectivity and coverage of the resulting motion models with the underlying C-Space [19]. They can be used in addition to time evaluations as in [3, 18, 32].

We identified three stages of the learning process that planners go through: *quick learning*, *model enhancement*, and *learning decay*. By detecting transitions between the stages we can decide when to stop sampling or change strategies. Throughout these stages, roadmap-based planners had clear distinctions in node types, and incrementally-exploring planners' node types were more similar. Roadmap-based planners also showed a faster stabilization in global-level metrics than the incrementally-exploring planners which have a more gradual stabilization due to their orderly expansion.

We showed we can identify different types of regions in the C-Space in order to monitor the evolution of the sampling distribution and to spatially adapt planning based on the complexity of the regions. We used the global-level metrics to stop planning or to change strategies when the rate of change indicates that the *learning decay* stage has started.

We discussed a strategy to make comparative evaluations of planners based on our multi-level metrics that gives insight into qualitative features of the problem. We compared the speed of coverage change in incrementally-exploring planners in some problems where *RRT*-Expand covers the space faster than *EST* and *RRT*-Connect was faster than *RPP*. We also compared the node types achieved by roadmap-based planners at the start of the *learning decay* stage. In two cases, *OBPRM* had more *cc-expand* nodes than the other planners, but in another case it had fewer such nodes than *Basic-PRM*, *Gauss-PRM*, and *MAPRM*. It is clear that performance varies with the features of the problem.

We studied a high-*DOF* problem to show how the metrics allow us to characterize the planning process in these complex cases just as in the low-*DOF* problems.

The main potential applications of these metrics are in adaptive planning to make better decisions to match planners to problems and in adapting the exploration of the C-Space to the information obtained while sampling.

REFERENCES

[1] M. Akinc, K. E. Bekris, B. Y. Chen, A. M. Ladd, E. Plaku, and L. E. Kavraki, "Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps," in *The International Symposium on Robotics Research (ISRR)*, Sienna, Italy, October 2003.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo, "OBPRM: An obstacle-based PRM for 3D workspaces," in *Robotics: The Algorithmic Perspective.* Natick, MA: A.K. Peters, 1998, pp. 155–168, proc. Third Workshop on Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.

[3] ——, "Choosing good distance metrics and local planners for probabilistic roadmap methods," *IEEE Trans. Robot. Automat.*, vol. 16, no. 4, pp. 442–447, August 2000.

[4] N. M. Amato and G. Song, "Using motion planning to study protein folding pathways," *J. Comput. Biol.*, vol. 9, no. 2, pp. 149–168, 2002, special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2001.

[5] J. Barraquand and J. C. Latombe, "Robot motion planning: A distributed representation approach," *Int. J. Robot. Res.*, vol. 10, no. 6, pp. 628–649, 1991.

[6] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, USA, 2000, pp. 529–536.

[7] ——, "Ligand binding with OBPRM and haptic user input: Enhancing automatic motion planning with virtual touch," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, 2001, pp. 954–959.

[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 Int. Conf. Intel. Rob. Syst. (IROS)*, vol. 2, Tokyo, Japan, 1993, pp. 1373–1380.

[9] R. Bohlin and L. E. Kavraki, “Path planning using Lazy PRM,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, USA, 2000, pp. 521–528.

[10] V. Boor, M. H. Overmars, and A. F. van der Stappen, “The Gaussian sampling strategy for probabilistic roadmap planners,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, Detroit, USA, May 1999, pp. 1018–1023.

[11] B. Burns and O. Brock, “Information theoretic construction of probabilistic roadmaps,” in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, Las Vegas, USA, Oct. 2003, pp. 650–655.

[12] ——, “Sampling-based motion planning using predictive models,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 3120–3125.

[13] J. F. Canny, *The Complexity of Robot Motion Planning.* Cambridge, MA, USA: MIT Press, 1988.

[14] H. Chang and T. Y. Li, “Assembly maintainability study with motion planning,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Nagoya, Japan, 1995, pp. 1012–1019.

[15] D. Coppersmith and S. Winograd, “Matrix multiplication via arithmetic progressions,” in *Proc. 9th Annu. ACM Sympos. Theory Comput.*, New York, NY, USA, May 1987, pp. 1–6.

[16] D. G. Corneil, F. F. Dragan, and E. Köhler, “On the power of bfs to determine a graph’s diameter.” *Networks*, vol. 42, no. 4, pp. 209–222, 2003.

[17] T. K. Dey, H. Edelsbrunner, and S. Guha, “Computational topology,” in *Advances in Discrete and Computational Geometry - Proc. 1996 AMS-IMS-SIAM*

Joint Summer Research Conf. Discrete and Computational Geometry: Ten Years Later, ser. Contemporary Mathematics, B. Chazelle, J. E. Goodman, and R. Pollack, Eds., no. 223. Providence, RI, USA: American Mathematical Society, 1999, pp. 109–143.

[18] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Nice, France, December 2002.

[19] ——, "Reachablility analysis of sampling based planners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 406–412.

[20] S. Gottschalk, M. Lin, and D. Manocha, "OBB-tree: A hierarchical structure for rapid interference detection," in *Proc. ACM SIGGRAPH*, New Orleans, LA, Aug. 1996, pp. 171–180.

[21] T. Horsch, F. Schwarz, and H. Tolle, "Motion planning for many degrees of freedom – random reflections at c-space obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Diego, CA, USA, May 1994, pp. 3318–3323.

[22] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "Bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Taipei, Taiwan, Dec. 2003, pp. 4420–4426.

[23] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Hanover, NH, USA, Mar. 2000, pp. SA1–SA18.

[24] ——, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, March 2002.

[25] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configu-

ration spaces," *Int. J. Comput. Geom. & Appl.*, pp. 2719–2726, 1997.

[26] D. Hsu, G. Sánchez-Ante, and Z. Sun, "Hybrid PRM sampling with a cost-sensitive adaptive strategy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 3885–3891.

[27] L. Jaillet, A. Yershova, S. M. Lavalle, and T. Simeon, "Adaptive tuning of the sampling domain for dynamic-domain RRTs," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, Edmonton, Alberta, Canada, Aug. 2005.

[28] L. E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan, "Randomized query processing in robot path planning," in *Proc. ACM Symp. Theory of Computing (STOC)*, Las Vegas, NV, USA, May 1995, pp. 353–362.

[29] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, August 1996.

[30] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, CA, USA, Apr. 2000, pp. 995–1001.

[31] J.-P. Laumond and T. Siméon, "Notes on visibility roadmaps and path planning," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Hanover, NH, USA, Mar. 2000.

[32] S. M. LaValle and M. S. Branicky, "On the relationship between classical grid search and probabilistic roadmaps," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Nice, France, 2002.

[33] S. M. Lavalle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 673–692, 2004.

[34] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Detroit, MA, USA, May 1999, pp. 473–479.

[35] ——, "Rapidly-Exploring Random Trees: Progress and Prospects," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Hanover, NH, USA, Mar. 2000, pp. SA45–SA59.

[36] ——, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[37] J.-M. Lien, S. L. Thomas, and N. M. Amato, "A general framework for sampling on the medial axis of the free space," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Taipei, Taiwan, Sept. 2003, pp. 4439–4444.

[38] J.-M. Lien, O. B. Bayazit, R.-T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, USA, Apr. 2004, pp. 4159–4164.

[39] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 3413–3418.

[40] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, October 1979.

[41] E. Mazer, J. M. Ahuactzin, and P. Bessiere, "The Ariadne's clew algorithm," *Journal of Artificial Robotics Research (JAIR)*, vol. 9, pp. 295–316, 1998.

[42] M. Morales, R. Pearce, and N. M. Amato, "Analysis of the evolution of C-Space models built through incremental exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Rome, Italy, Apr. 2007, pp. 1029–1034.

[43] M. Morales, S. Rodriguez, and N. M. Amato, "Improving the connectivity of PRM roadmaps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 3, Taipei, Taiwan, 2003, pp. 4427–4432.

[44] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato, "A machine learning approach for feature-sensitive motion planning," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Utrecht/Zeist, The Netherlands, July 2004, pp. 361–376.

[45] M. A. Morales A., R. Pearce, and N. M. Amato, "Metrics for analyzing the evolution of C-Space models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Orlando, FL, USA, May 2006, pp. 1268–1273.

[46] M. A. Morales A., L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato, "C-space subdivision and integration in feature-sensitive motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 3114–3119.

[47] R. Motwani and P. Raghavan, *Randomized Algorithms.* New York: Cambridge University Press, 1995.

[48] C. L. Nielsen and L. E. Kavraki, "A two level fuzzy PRM for manipulation planning," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Takamatsu, Japan, Nov. 2000, pp. 1716–1722.

[49] C. Nissoux, T. Simeon, and J.-P. Laumond, "Visibility based probabilistic roadmaps," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, Kyongju, Korea, Oct. 1999, pp. 1316–1321.

[50] M. Overmars, "A random approach to path planning," Computer Science, Utrecht University, The Netherlands, Tech. Rep. RUU-CS-92-32, 1992.

[51] M. Overmars and P. Svestka, "A probabilistic learning approach to motion planning," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*,

Stanford, CA, USA, Feb. 1994, pp. 19–37.

[52] E. Plaku and L. Kavraki, “Quantitative analysis of nearest-neighbors search in high-dimensional sampling-based motion planning,” in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, New York, NY, USA, July 2006, to appear.

[53] J. R. Quinlan, “Induction of decision trees,” in *Machine Learning*, vol. 1, 1986, pp. 81–106.

[54] ——, *C4.5: Programs for Machine Learning.* Sydney: Morgan Kaufmann, 1993.

[55] J. H. Reif, “Complexity of the mover’s problem and generalizations,” in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, San Juan, Puerto Rico, October 1979, pp. 421–427.

[56] S. Rodriguez, S. Thomas, R. Pearce, and N. M. Amato, “Resampl: A region-sensitive adaptive motion planner,” in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, New York, NY, USA, July 2006, to appear.

[57] J. T. Schwartz and M. Sharir, “On the “piano movers” problem II: General techniques for computing topological properties of real algebraic manifolds,” *Adv. Appl. Math.*, vol. 4, pp. 298–351, 1983.

[58] R. Seidel, “On the all-pairs-shortest-path problem,” in *Proc. 24th Annu. ACM Sympos. Theory Comput.*, Victoria, Canada, May 1992, pp. 745–749.

[59] A. Singh, J. Latombe, and D. Brutlag, “A motion planning approach to flexible ligand binding,” in *7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, Heidelberg, Germany, Aug. 1999, pp. 252–261.

[60] G. Song, S. L. Miller, and N. M. Amato, “Customizing PRM roadmaps at query time,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, May 2001, pp. 1500–1505.

[61] X. Tang, B. Kirkpatrick, S. Thomas, G. Song, and N. M. Amato, “Using motion planning to study RNA folding kinetics,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, San Diego, CA, USA, Mar. 2004, pp. 252–261.

[62] S. Thomas, M. Morales, X. Tang, and N. M. Amato, “Biasing samplers to improve performance,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Rome, Italy, Apr. 2007, pp. 1625–1630.

[63] S. Thomas, X. Tang, L. Tapia, and N. M. Amato, “Simulating protein motions with rigidity analysis,” in *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, Venice Lido, Italy, Apr. 2006, pp. 394–409.

[64] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, “MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, Detroit, MA, USA, 1999, pp. 1024–1031.

[65] D. Xie, M. Morales, R. Pearce, S. Thomas, J.-M. Lien, and N. M. Amato, “Incremental map generation (IMG),” in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, New York, NY, USA, July 2006, to appear.

VITA

Marco Antonio Morales Aguirre received his Bachelor of Science in computer engineering from the Universidad Nacional Autónoma de México in 1996. He received his Master of Science in electrical engineering also from the Universidad Nacional Autónoma de México in 1998. He was awarded a Fulbright/García-Robles scholarship to continue his graduate education in the computer science program of Texas A&M University where he received his Doctor of Philosophy degree in 2007. His research interests include motion planning and supporting areas, such as machine learning and computational geometry, and its applications to robotics, bio-informatics, and computational neuroscience. He is also interested on artificial intelligence, optimization, computational sciences, architectures, and parallel computing.

Mr. Morales may be reached at Instituto Tecnológico Autónomo de México, Río Hondo No. 1, Tizapán San Angel, México D.F. 01080, México. His email address is mmorales@acm.org.