Pure Topological Mapping in Mobile Robotics

Dimitri Marinakis and Gregory Dudek

***Abstract*—In this paper, we investigate a pure form of the topological mapping problem in mobile robotics. We consider the mapping ability of a robot navigating a graph-like world in which it is able to assign a relative ordering to the edges, leaving a vertex with reference to the edge by which it arrived but is unable to associate a unique label with any vertex or edge. Our work extends and builds upon earlier approaches in this problem domain, which are based on construction of exploration tree of plausible world models. The main contributions of the paper are improved exploration strategies that reduce model ambiguity, a new method of search through consistent models in the exploration tree that maintains a bounded set of likely hypotheses based on the principle of Occam's Razor, the incorporation of arbitrary feature vectors into the problem formulation, and an investigation of various aspects of this problem through numerical simulations.**

***Index Terms*—Graph exploration, mobile robotics, topological mapping.**

I. Introduction

In THIS paper, we consider a *pure* form of the topological mapping problem from the perspective of a mobile robot equipped only with the ability to observe nonunique signatures from landmarks in the environment.

We represent the world as an undirected graph in which vertices represent discrete places and edges represent paths between them. We assume that the robot can consistently assign a cyclic ordering to the edges leaving a vertex with reference to the edge by which it arrived; however, it is unable to associate a unique label with any vertex or edge. Given this limited sensing capability and without the use of any markers or additional information, we will show that the construction of a topological map is nevertheless feasible.

Our approach applies the principle of Occam's Razor[1] to the selection of suitable topological representations of the environment. In particular, we use a ranking heuristic that gives preference to models of the environment that employ fewer vertices, have fewer edges leading to unexplored areas, and cluster statistically similar observations. Our ranking approach, when combined with exploration strategies designed to reduce ambiguity, is effective at producing a small set of suitable solutions for an explored region.

The topological mapping problem we consider in this paper can be considered a subclass of the simultaneous localization and mapping (SLAM) problem in robotics, which considers mapping an unknown environment using imperfect sensory data. Some important recent SLAM research includes the FastSLAM work of Montemerlo *et al.* [1], [2], work by Wolf and Sukhatme [3], work by Dellaert *et al.* [4], [5], and work by Blanco *et al.* [6], who incorporate topological representations of the environment into their approach.

One of the key problems in SLAM is that of *closing the loop* [7], i.e., determining whether a currently observed landmark or region corresponds to a previously visited location, or to a new portion of the world being explored. Examples of research focused on loop closing include the work by Newman and Ho [8], the work of Martinelli *et al.* [9], the work of Ranganathan *et al.* [10], and the work of Beevers and Huang [11].

The question we consider in this paper can be considered an extreme case of the loop-closing problem in SLAM in which the robot has almost no ability to characterize its surroundings or obtain meaningful odometry measurements. While most SLAM techniques are based on local, incremental localization, loop closing typically depends on global localization that takes into account the history of the robot's trajectory.

A. Robot Graph Exploration

The study of a robot equipped only with the sensing ability to assign a cyclic ordering to edges incident to a vertex in a graph-like world has been examined previously by Dudek *et al.* [12], [13], who presented a mapping strategy in which the robot constructs an *exploration tree* that enumerates *consistent* world hypotheses at each step of an exploration process. Here, a world hypothesis is called "consistent" if it can explain the observations gathered by the robot up to that step. The authors classified the potential errors that could be made during the construction of this tree into three classes: 1) errors of type OLD-LOOKS-NEW, in which the current vertex is assumed to be newly explored but was actually visited earlier; 2) errors of type MIS-CORRESPONDENCE, in which the current vertex is thought to be a certain previously visited vertex but is actually a different previously visited vertex; and 3) errors of type NEW-LOOKS-OLD, in which a vertex is assumed to have been previously visited but is actually new. These error types are called *correspondence* errors.

The authors discussed the fact that in the exploration tree, there will always exist a model that assumes that each vertex visited is a new location, i.e., multiple errors of the type OLD-LOOKS-NEW. Among the three types, this class of errors is

Manuscript received February 16, 2009; revised March 9, 2010 and September 5, 2010; accepted September 21, 2010. Date of publication October 28, 2010; date of current version December 8, 2010. This paper was recommended for publication by Associate Editor C. Stachniss and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

The authors were with the Centre for Intelligent Machines, McGill University, Montreal H3A 2K6, QC, Canada. They are now with the University of Victoria, Victoria V8N 6KB, BC, Canada (e-mail: dmarinak@cim.mcgill.ca; dudek@cim.mcgill.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2010.2081410

[1]Occam's Razor is the principle enunciated by William of Occam that the simplest explanation is the best.

unique, since the associated models can never be shown as inconsistent, given the absence of unique vertex signatures.

Dudek *et al.* [12], [13] proposed a heuristic to be used during the exploration process that prunes all models of size greater than

$$T = \gamma s + C \tag{1}$$

where s is the number of vertices in the current smallest model, and γ and C are constants. This heuristic effectively enforces a prior that puts a preference on smaller and more compact representations of the environment rather than larger and more complex descriptions. In this manner, branches of the hypothesis tree that are less suitable are discarded from future consideration. Experiments demonstrated the success of the heuristic on graphs of limited size. The technique was capable of both considerably reducing the number of hypotheses maintained and preserving the correct world model. We will refer to this threshold as the *D-size threshold* (DST) in the remainder of this paper. The approach we present in this paper is motivated by and builds upon the spirit of this threshold-based approach.

Related work on robot graph exploration such as that by Dudek *et al.* [14], [15], Rekleitis *et al.* [16], Dujmovic and Whitesides [17], and Deng *et al.* [18], [19] considered a version of the problem described earlier in which a robot with the same limited perceptual ability is capable of placing and recognizing one or more markers. The work by Bender *et al.* [20] considered a slightly broader, but related, problem in which the vertices of the graph being explored are entirely label free. Unlike the markerless version of the graph exploration problem, it has been shown that by using the supplementary global information, one can resolve potentially incorrect correspondences and, therefore, unambiguously map a finite world (given adequate exploration). Additional extensions to this research have considered the problem of exploring a graph-like world with multiple robots, e.g., Dudek *et al.* [21], Huang and Beevers [22], and Wang *et al.* [23].

B. Contributions and Paper Organization

In this paper, we revisit the problem in which a single, markerless robot explores a graph-like world. The main contributions of the paper are as follows:

1) the new exploration strategies that attempt to reduce correspondence errors, where possible;
2) the incorporation of arbitrary feature vectors into the problem formulation;
3) a new method to maintain consistent models in the exploration tree in which only a bounded set of likely hypotheses are stored based on the principle of Occam's Razor.

Simulations show that the new exploration strategies combined with our new hypothesis-management mechanism perform much better than earlier approaches.

This paper is based on, and expands on, work presented earlier in [24]. In addition to a more thorough description of the work in [24], we use a more realistic sensor model by incorporation of feature vectors and present several new experimental results.

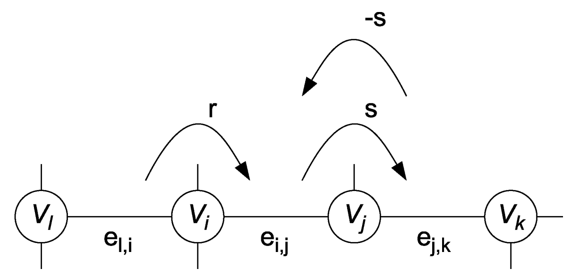


Fig. 1. Diagram showing relationship of visited vertices in the context of the transition function ψ. Some included edges are partially shown and not labeled.

In the remainder of this paper, we first define the specific topological mapping problem we study. Then, we introduce our methodologies and present an evaluation of their performance through simulations. Finally, we discuss some additional related work and discuss our results.

II. Problem Specification

We describe the problem of topological mapping in terms of the inference of an *undirected*, unweighted graph $G = \{V, E\}$ in which the edges leading from a vertex can be assigned a local ordering and where each vertex has a nonunique signature. In particular, we consider the specific case, where the signature is comprised of the degree of the node and a continuous valued feature vector drawn from a probability distribution function specific to each vertex.

The vertices of the graph correspond to distinguishable places in the world and the edges correspond to connect *bidirectional* paths. As the robot traverses the graph, it obtains an observation from each vertex and can apply a locally consistent edge ordering at each vertex. For example, it could identify the second edge clockwise from the edge on which it entered the vertex.

We refer to the edge by which the robot enters a vertex as the current *reference edge*. We then define the transition (or motion) function ψ as follows: $\psi(v_i, e_{l,i}, r) = v_j$, which means that leaving vertex v_i by the edge that is r edges (e.g., clockwise) from the reference edge $e_{l,i}$ takes us to vertex v_j. Since the graph is undirected, the robot can retrace its trajectory by recording its motions: If $\psi(v_i, e_{l,i}, r) = v_j$ and $\psi(v_j, e_{i,j}, s) = v_k$, then $\psi(v_j, e_{k,j}, -s) = v_i$ (see Fig. 1).

During each step t of the exploration process, the robot records the degree $\delta(v_t)$ of its current topological node v_t as well as the feature vector F_t observed from that node. The result is a list of observations $O = \{o_1, \ldots, o_T\}$, $o_t = \{\delta(v_t), F_t\}$, where $F_t = \{f_1, \ldots, f_M\}$ is a feature vector of length M.

As this exploration process continues, an exploration tree is constructed, the complete version of which contains a single world model for every consistent correspondence among all previously visited topological nodes. Each model is a graph, possibly with partial edges, with the current position of the robot specified. Each *level* of this exploration tree will be based on the information obtained from the traversal of a potentially unexplored edge. At any step t, each of the maintained hypotheses in the tree is consistent with the observational data collected up to that point. As discussed in [13], the number of models that is consistent with the observations depends on the type of graph explored but can experience explosive growth. This is especially true during the early part of the exploration in which not enough observations have been gathered to prove some models inconsistent. For the graphs we considered, we found that the size of the complete tree quickly becomes intractable for all but trivially small observation sequences. The goal of our paper is to manage the growth of the exploration tree so that only those world models that appear of relatively high likelihood are retained.

III. Exploration Strategies

A. Breadth-First Traversal

Here, for completeness, we briefly describe the original exploration strategy described in [13]. The strategy processes new edges in a first-in first-out manner, based on a breadth-first traversal (BFT) of the world as observed by the robot. For example, when beginning in a vertex with two edges, the robot will traverse the first edge, return to the original vertex, traverse the second edge, and then return again to the original vertex. It has now explored its world up to a radius of one edge traversal. Let us call this a "level one" exploration. In the next step of the BFT, the robot will explore its world up to a radius of two edge traversals. Starting from the original vertex, it will traverse the first edge again and then recursively do a "level one" exploration, starting with this new vertex. When complete, it will return to the original vertex, traverse edge two and do the same process again. Finally, it will return to the original vertex having completed the second level of exploration.

At each level of exploration, the BFT strategy will reach each of the ith neighbors of the vertex v_s, where the robot starts the exploration, and here, we define the ith neighbors of a vertex v as all vertices terminating traces of length i that originate from v. Note that a single vertex u may be present many times as an ith neighbor[2] of v, provided that $i > 1$. If d is the diameter of a finite graph, then the BFT algorithm is guaranteed to visit all the vertices after concluding a level d exploration. A limiting factor when applying the BFT exploration strategy is the size of the exploration tree. In the next sections, we will consider new exploration strategies, which are designed to help slow the growth of the exploration tree.

B. Breadth-First Ears Traversal

For our purposes, a good exploration strategy will limit, as much as possible, the number of world hypotheses that need to be considered. Of the three types of errors originally identified in [12], errors of the type MIS-CORRESPONDENCE and NEW-LOOKS-OLD can be used to show hypotheses inconsistent. Errors of the first type, i.e., OLD-LOOKS-NEW, in which the current location is assumed to be a new node, can only be diagnosed by considering the implausibility of the world model suggested. We can do no better than this, since there is no method of detection for errors of type OLD-LOOKS-NEW. The

[2]In the case of $i = 1$, this is only possible if multiple edges are allowed between the same vertices, i.e., we are exploring a multigraph.

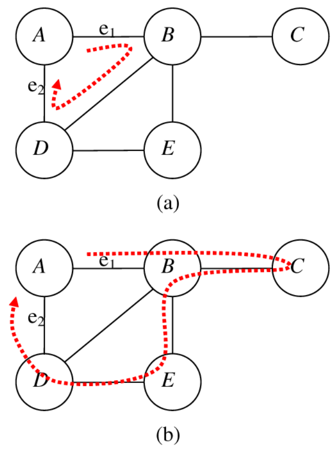


Fig. 2. Example of (a) counterclockwise and (b) clockwise ear starting from e_1 of vertex A.

strength of the original BFT exploration strategy is its guarantee of eventual coverage given a finite world. It appears, however, that strategies employing more passes through the potentially previously explored areas can help reveal correspondence errors of the second and third type better than BFT.

We now present a deterministic exploration strategy called breadth-first ears traversal (BFET) that, like BFT, is guaranteed to eventually visit all vertices (and edges) of a finite world. In the next section, we will describe a simple stochastic variant.

BFET incorporates within the original BFT algorithm, a subexploration strategy that attempts to traverse each *ear* leading from the current vertex v. In graph theory, any undirected, two-edge connected graph can be decomposed into a set of simple paths, which are called ears [25]. In our paper, however, we use the term "ear" in a slightly different manner, which reflects the fact that the edges leading from any vertex in our graph can be assigned a relative ordering. We define an ear as the closed cycle one obtains by leaving a vertex on a specific edge and selecting for traversal, at the following vertex, the edge that is next to the reference edge in a *consistent orientation* (clockwise or counterclockwise), until one returns to the original vertex. In other words, we consistently select $r = 1$ or -1 in the transition function ψ to trace out an ear. For example, leaving an edge and making only next "right turns" until one returns to the original vertex will trace out a counterclockwise ear. See Fig. 2 for more examples of this concept. The same definition of ear has been used earlier by Rekleitis *et al.* [26].

The BFET subexploration strategy works as follows. For each edge leading from the vertex being currently explored in the BFT strategy, take the following steps.

1) For an edge, e_1 leading from the vertex v, the robot explores the path p_1, beginning with e_1 in one direction (e.g., clockwise) for some number of steps (until, for example, a node with the same degree as v is encountered).
2) The robot then backtracks to vertex v and explores the path p_2 in the opposite direction (e.g., counterclockwise) for the same number of steps, beginning with the edge e_2 that is appropriately located with reference to e_1.

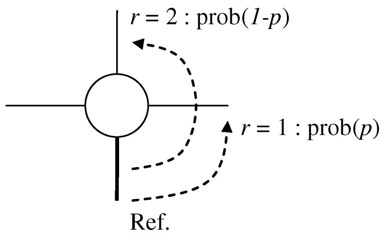


Fig. 3. Diagram showing pictorial example of how the LBE algorithm selects the next edge to traverse with respect to the reference edge when entering a vertex.

3) Steps 1 and 2 are repeated with larger and larger sets of steps taken in both directions until the degree trace for the path taken in two directions matches up, i.e., path p_1 visits its vertices in the reverse order of those in p_2.

This process is guaranteed to terminate a given finite graph, since there is a bound on both the longest ear in the graph and the number of ears to which any node can belong.

Upon completion of the subexploration strategy, there is at least the potential that the robot actually visited the same set of vertices twice in opposite order. Therefore, in the exploration tree, there must now exist a model of the world that reflects the fact that we have found a cycle that contains the node we are currently investigating.

The BFET strategy is guaranteed both to terminate and to cover the graph under exploration. Additionally, it is also better at discriminating among potential hypotheses than the BFT strategy, as we will demonstrate in experiments presented in Section V. This performance improvement comes, however, at an exploration cost that becomes prohibitive for larger graphs. We believe that the better performance of the BFET algorithm in small examples is due to its ability to eliminate inconsistent models by revisiting previously explored vertices in a cyclic manner.

In the next section, we present a stochastic exploration strategy, which attempts to exploit the apparent advantage of revisiting previously explored areas.

C. Loop-Based Exploration

In this section, we consider an exploration strategy we call loop-based exploration (LBE) that attempts to balance, in a stochastic manner, the exploration of new areas and the revisiting of previously explored areas. The intent of the revisiting portion of the strategy is to slow the growth of the hypothesis tree by showing the inconsistency of some of the currently maintained hypotheses. The approach has no coverage guarantees but shows good performance in practice.

LBE works as follows. If the robot is currently visiting a vertex of degree three or higher, then it selects, with a probability p, the first edge from the incoming reference edge for its next traversal. This corresponds to choose $r = 1$. Otherwise, it takes, with probability $(1 - p)$, the second edge $r = 2$ from the incoming reference edge (see Fig. 3). If the current vertex is of degree two, then it selects the edge that is not the reference edge, and if the vertex is of degree one, then it backtracks.

If a relatively large value of p is selected, this algorithm has the effect of visiting ears in the graph one at a time and having much the same effect on the exploration tree as the BFET algorithm for each ear examined. The larger the value of p, the more we repeat the exploration of a particular ear, but this comes at the cost of the average coverage time for the graph.

Although LBE cannot guarantee coverage of a finite graph, we will show that given a good choice for p, in practice, this strategy performs as well as or better than the more complex BFET strategy. LBE never fails to cover a connected graph in practice, and this appears to be assured for a value of $0 < p < 1$. This follows from a result of Koucky's [27],[3] which shows that a random walk in which one selects among at least two outbound edges with a nonzero probability *relative* to the inbound edge for vertices of degree larger than one is guaranteed to eventually cover the graph. The determining bounds for the expected cover time as a function of p should be possible for a class of graphs, such as those that are planar and would be related to the work of Jonasson and Schramn [28].

IV. Hypotheses Management

In this section, we consider the problem of controlling the potentially explosive growth of an exploration tree. We first present methods to assess the suitability of many different, but consistent, world models. We then present an algorithm to manage the storage of consistent hypotheses in an online fashion using the described assessment methods. Our approach uses heuristic evaluation functions that favor *simple* models that are consistent with the observed data while pruning more complex, although consistent, models.

A. Hypothesis Assessment Using Topology

This section considers assigning a relative ranking to candidate hypotheses based solely on their topology. To aid the description of our assessment heuristic, we first review the terms *pending gateway* and *closed* as defined by Savelli and Kuipers [29]. A pending gateway is an edge in a hypothesized graph model that is assumed by the model to lead to an unexplored area.[4] A closed-hypothesized graph model is one with no pending gateways.

We judge how "simple" a model is by considering both the number of vertices and the number of pending gateways. Specifically, we rank models first in inverse proportion to the number of vertices they contain and, for tie-breaking purposes, in inverse proportion to the number of pending gateways. The second component of this heuristic that considers the number of pending gateways is designed to reward models that are approaching a closed state, since we assume that ultimately the entire region will be explored.

To justify the inclusion of pending gateways in the heuristic function, consider the situation in which the robot has observed the node degrees: $(2, 3, 2, 3, 2, 3, 2, 3, \ldots)$, while following an

[3]See Theorem 15 in that paper and the associated subsequent commentary [27].

[4]We have used the term singly connected or *dangling* to described this type of edge in prior work [24].

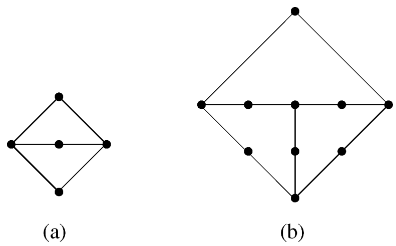


Fig. 4. Examples of closed graphs that could explain the sequence of observed vertex degrees: $2, 3, 2, 3, 2, 3, 2, 3, \ldots$.

arbitrary exploration strategy. We would reasonably surmise that our world is made up of vertices only of degree two and three and that each vertex of degree two has only neighbors of degree three, and likewise, each vertex of degree three has only neighbors of degree two. If we have done enough exploration to suggest that we should have achieved edge coverage of the entire environment, then we might suspect a world that looks like one of the ones depicted in Fig. 4. In this example, if we assume that the graph under exploration is not a multigraph, then our heuristic will assign the model shown in Fig. 4(a) the highest relative weight among all consistent graphs.

In most applications, there is probably some prior knowledge that can be exploited to give a rough idea of the size of the region being explored, and therefore, some guess of the probability of having achieved edge coverage of the area in question when using a given exploration strategy. Note that edge coverage is more informative than vertex coverage in this exploration process, since we are interested in the topology of the graph. We must traverse each edge in the true graph at least once in order for the true graph to be present in the hypothesis tree.

B. Hypothesis Assessment Using Feature Vectors

A robot equipped with the sensing and computational capability to reliably distinguish a vertex (or node) in the environment and assign a logical ordering to the edges incident on that vertex should also be able to extract a more distinguishing signature than just the degree of the node visited. We abstract this concept by assuming that a feature vector is also obtained from each node visited. For simplicity, we assume that the individual features in the vector have been normalized such that each has roughly the same range of values. Here, we consider how to use this feature vector in evaluation of the "simplicity" of a candidate hypothesis.

A particular hypothesis regarding the topology of the graph assumes each observation was generated by a specific vertex. Therefore, for the model h, we can group the feature vectors $F_1, \ldots, F_t$ obtained up to time t in the exploration process by the vertex that is assumed to have generated them. This gives us a set of vectors Γ_i for each node i in h.

We define a *feature vector agreement* metric θ that rewards a hypothesis in which feature vectors assumed to be generated by the same node are themselves similar but also penalizes a hypothesis that assigns similar feature vectors to different nodes. Our heuristic compares a variance-based statistic *within* the feature sets to one calculated *across* pairwise combinations of sets.

We define a dispersion measuring function λ as follows:

$$\lambda(\Gamma) = \begin{cases} \|\Sigma\|, & |\Gamma| > 1 \\ M, & |\Gamma| = 1 \end{cases}$$

where $\|\Sigma\|$ gives the norm of the covariance matrix for the set of feature vectors Γ, and M is the number of features in a single vector. The singleton value is selected to ensure a fair weighting for nodes that are assumed to have been visited only once.

Now, let $\lambda_i = \lambda(\Gamma_i)$ be a measure of the dispersion for the set of feature vectors associated with vertex i of hypothesis h. Likewise, let $\lambda_{i,j} = \lambda(\Gamma_i \cup \Gamma_j)$ be a measure of dispersion that is obtained for each pair of feature vectors sets in h when combined. There will be $(K^2 - K)/2$ pairs of feature vectors sets for h, where K is the number of vertices in h. Now, let Λ_i be defined for each vertex of h as follows:

$$\Lambda_i = \frac{\lambda_i}{\min(\lambda_{i,1} \ldots \lambda_{i,K})}$$

and define the *feature vector agreement* metric θ for hypothesis h as the max of all Λ values for h

$$\theta = \max(\Lambda_1 \ldots \Lambda_K). \tag{2}$$

Should the θ value for a hypothesis be near one or larger, then that is evidence that the model has incorrectly assigned the observations from two or more different nodes to the same node or else has incorrectly assigned observations from a single node to two or more nodes. However, a θ value near zero suggests that the model has assigned statistically similar observations to the same node and, additionally, has not split statistically similar observations across more than one node. For example, consider Fig. 5, which shows three clusterings of a set of two-dimensional (2-D) feature vectors that might occur for three different beliefs. Their relative θ values suggest that Fig. 5(a) is most suitable, Fig. 5(b) is marginally suitable, and Fig. 5(c) is the least suitable.

C. Hypotheses Management Algorithm

We now specify an algorithm for the management of hypotheses based on the assessment methods described in the previous sections. At each traversal of an edge during the exploration process, we first enumerate the new models that can be generated from each of the currently maintained world hypotheses, discard unsuitable models, and rank the remainder based on their relative weights using our heuristic assessment functions. The top N of these models are then selected for maintenance, and the rest are discarded.

Algorithm 1 gives the details of our hypotheses management algorithm (HMA). First, a topological assessment of each maintained hypothesis occurs. By making the weight on model size W_1 much larger than the weight placed on the number of pending edges W_2, we effectively use the second factor only for tie breaking purposes in the topological assessment of two models. Note that using this algorithm, models that are inconsistent with the latest observation will have no descendants and are pruned automatically.

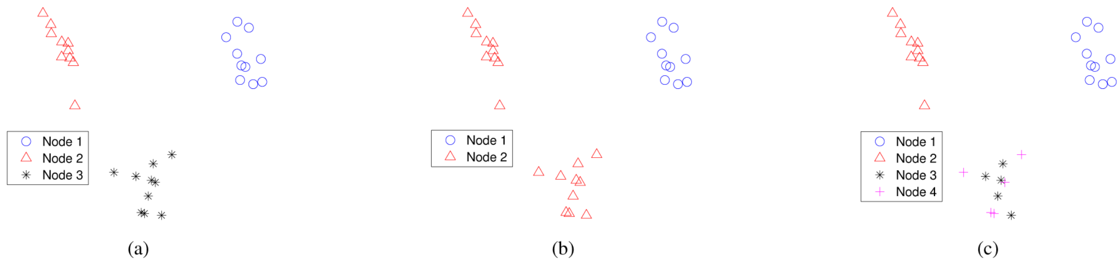


Fig. 5. Three different examples of observation to node assignments using the same set of 2-D feature vectors and the corresponding feature vector agreement metric as calculated according to (2). (a) $\theta = 0.08$. (b) $\theta = 0.69$. (c) $\theta = 1.96$.

Algorithm 1 Hypotheses Management Algorithm

INPUT: $O = (o_1 \ldots o_T)$ { ordered list of observations }
INPUT: N { number of models to maintain }
INPUT: $UseFeatures$ { true or false }
INPUT: ω { feature vector pruning threshold }
$H_0 \Leftarrow$ initial model based on o_0 { root of hypothesis tree}
for $t = 1$ to $t = T$ **do**
  $L \Leftarrow ()$ { initialize maintained models }
  for all $h \in H_{t-1}$ **do**
    $g \Leftarrow$ all possible descendants of h based on o_t
    $L \Leftarrow L \cup g$ { collect all descendants }
  end for
  for all $h \in L$ **do**
    $x \Leftarrow$ number of vertices in h
    $y \Leftarrow$ number of pending gateways in h
    $h_w \Leftarrow \frac{W_1}{x} + \frac{W_2}{y+1}$ { $W_1 >> W_2$ }
    if $UseFeatures$ **then**
      $\theta \Leftarrow$ feature vector agreement for h
      if $\theta > \omega$ **then**
        $h_w \Leftarrow 0$ { prune out this hypothesis }
      else
        $h_w \Leftarrow h_w * \frac{1}{\theta}$ { incorporate feature vector agreement }
      end if
    end if
  end for
  $M \Leftarrow h \in L$ AND $h_w = 0$
  $L \Leftarrow L \setminus M$ { prune zero weight hypotheses }
  sort L by h_w { order remaining hypotheses by weight }
  $H_t \Leftarrow$ the first $\min(N, |L|)$ ordered elements of L
end for
Return H_T: a ranked sub-set of models consistent with O

Should the parameter $UseFeatures$ be set to $true$, then the hypotheses are also judged based on the observed feature vectors. The *pruning threshold* ω is used to prune out unsuitable hypotheses and the feature agreement metric θ is incorporated into the assessment of candidate models.

Fig. 6 gives an example of a simple exploration tree and how the maintained world models would be managed according to the process we have specified in this section (with $UseFeatures = false$).

This approach allows online exploration, but risks throwing away the correct solution. Offline variants could run the same algorithm repeatedly on the same observational sequence but employing an iteratively larger value for N until a suitable solution was obtained. For example, one approach is to use the ranking heuristic algorithm described earlier beginning with $N = 1$. If the set of models H_T returned does not include at least one closed model, then the value assigned to N is doubled during the next iteration. If one or more closed models are present in H_T, then the method terminates with the models returned by the latest iteration as output.

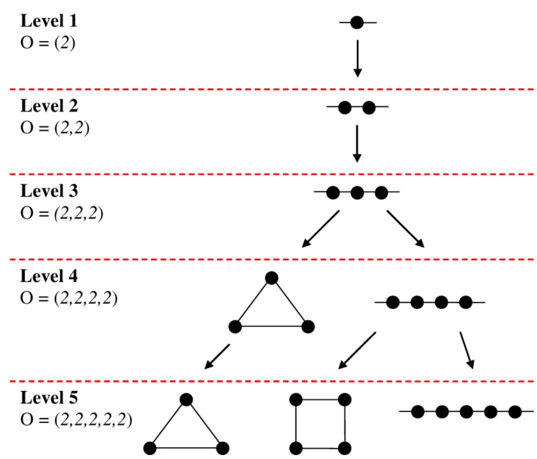
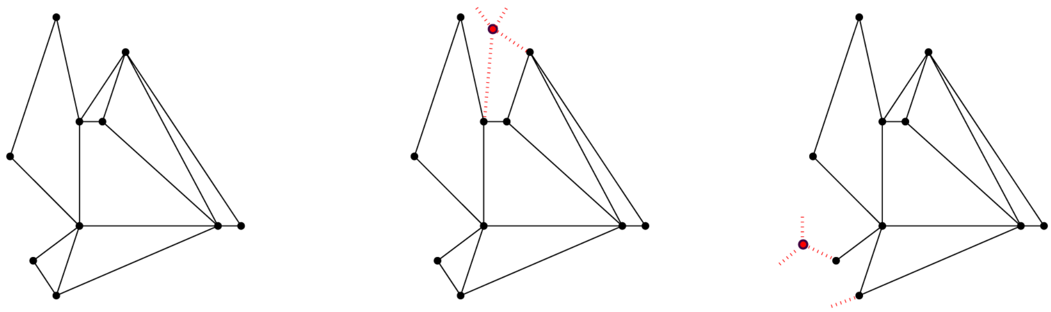


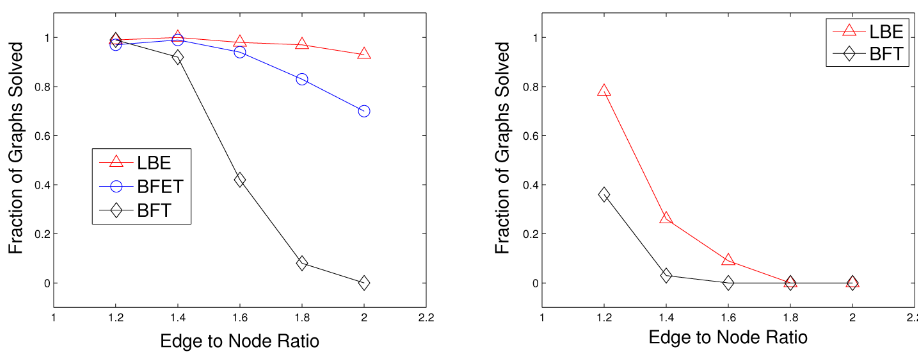
Fig. 6. Consider a robot following an exploration strategy that requires it to take an edge other than the reference edge for each traversal and which visits only nodes with the signature (degree) 2. This figure shows the full exploration tree with all models maintained for the first five observations. The models are ranked left to right for each level based on the heuristic discussed in Section IV-A. Up to Level 3 of the exploration tree, the model shown at each step is the only consistent world hypothesis, which can explain the observations. During steps four and five of the exploration process, which correspond to Levels 4 and 5 of the exploration tree, there are multiple models that are consistent with the data. During the fourth step, for example, we can either assume that the robot has revisited the first vertex from which it started or has discovered a new vertex. The first possibility corresponds to the higher ranking model, since it only requires three vertices and suggests that we have fully explored the world. The second possibility corresponds to a model with a lower ranking, since it requires four vertices and contains pending gateways. (We assume that the world cannot be a multigraph in this example.)

For the variant of our approach that does not consider feature vectors, it is generally sufficient to explore the graph only until edge coverage is achieved. Once edge coverage of the graph is achieved, the true hypothesis, if still retained in the set of consistent models H, is unlikely to be discarded. This is because the true hypothesis will be closed once the graph has been covered. In order for a closed model to drop in rank, there must be at least one, nonclosed, higher ranking model still retained in H that branches into a number of models that are themselves higher ranking than the closed model under question. Other than being shown as inconsistent, which cannot happen to the correct hypothesis, this is the only manner in which a closed model can risk being discarded.



(a) (b) (c)

Fig. 7. Example of the top three ranking world models, from (a)–(c), inferred by the algorithm with memory usage set to 20 models ($N = 20$) after running the BFET exploration strategy for 1000 steps on a ten-node graph with an edge to node ratio of 1.6. (Actual edge coverage was achieved at step 284.) The first ranked model is the correct one. Incorrect edges shown in dotted red.



(a) (b)

Fig. 8. Fraction of graphs for which the true solution was retained in the hypothesis space after the exploration strategy under consideration reached edge coverage of the graph. Results were obtained from 100 trials at each edge density for graphs of size. (a) Ten nodes and (b) 30 nodes. In this experiment, 100 hypotheses were maintained by the mapping algorithm ($N = 100$). For LBE, the parameter p was assigned a value of 0.99. (BFET results were unobtainable for the larger graphs because of its poor cover time.)

Our approach to hypothesis management is similar in spirit to the pruning heuristic based on the DST presented previously in [13] [see (1)]. The authors suggest limiting the growth of the exploration tree by pruning all models with more nodes than a threshold that is set based on the number of nodes in the current smallest model. Both the original pruning approach and the method we present here attempt to maintain simple solutions and discard more complex ones. The original approach, however, only slows the growth of the exploration tree, while the new approach places a bound on how many world models are maintained. Additionally, the new approach allows the incorporation of feature vectors into the assessment of suitable models.

V. Results

We examined our approach to topological mapping in this problem domain through a number of experiments. Our simulation tool takes as input an undirected graph to represent the world to explore and a set of parameters that dictate its behavior. These parameters include the exploration strategy employed by the robot, the number of observations to gather, the number of world hypotheses N to maintain, a Boolean that indicates if features should be used, and, if so, the pruning threshold ω. The simulator then determines if the robot, after its exploration, maintains in its world hypothesis space a graph that is isomorphically equivalent to the input graph (and its ranking in our hypothesis space). The graphs considered were randomly generated planar graphs produced by selection of a connected subgraph of the Delaunay triangulation of a set of random points. These experiments were run on a 2.2-GHz Intel Pentium 4 CPU with 1.00 GB of RAM using unoptimized MATLAB code.

A. Topological Assessment

In this section, we report results obtained using the HMA with only the topological method of model assessment, i.e., with the *UseFeatures* parameter set to *false*. For medium-sized sparse graphs, our heuristic approach to managing the size of the exploration tree was generally successful at retaining the correct solution by the time-edge coverage of the graph was achieved, provided an adequate number of hypotheses was maintained. This was true regardless of the exploration strategy used. Fig. 7 shows an example of a successful outcome on a ten-node graph. For each of the exploration strategies presented, the correct solution was found over 97% of the time in 100 trials of ten-node graphs with node-to-edge ratios of 1.2 with 100 maintained hypotheses (see Fig. 8). For edge to node densities of 1.4, the correct solution was found over 92% of the time in 100 trials of ten-node graphs and 100 maintained hypotheses (see Fig. 8).

TABLE I
RESULT OF PRUNING ALL MODELS USING THE DST WITH $\gamma = 1.05$ AND $C = 2$; PARAMETERS SUGGESTED IN [13]

Edge to Node Ratio	Average Memory Usage	Trials Solved using γs + C Pruning	Trials Solved using HMA
1.2	30.4	81	96
1.4	157.9	77	90
1.6	1564.5	82	87

Results obtained from 100 trials on random ten-node graphs for three different edge to node densities using the BFT exploration strategy until edge coverage. Memory usage refers to maximum number of models maintained at any one level of the exploration tree.

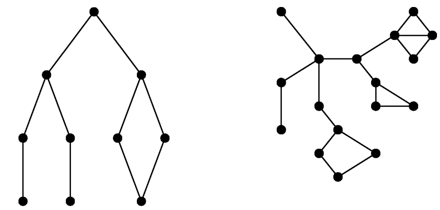


Fig. 9. Examples of graphs solved previously in (a) [12] and (b) [13]. Each of these graphs were solved by our approach using LBE ($p = 0.99$) in less than half a second with $N = 1$, i.e., only one model was maintained throughout the exploration process (which was the correct one).

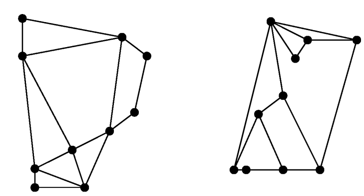


Fig. 10. Examples of graphs of varying difficulty. The LBE strategy was run 100 times with $N = 100$ and $p = 0.95$ on both graphs. For graph (a), a solution was found in 29 of the 100 trials, and for graph (b), a solution was found in all 100 trials.

Our HMA performs well in comparison with the original DST pruning strategy presented in [13]. Although effective at limiting the size of the exploration tree for simple graphs, the original DST approach occasionally prunes the correct solution and potentially requires an unlimited amount of memory. We found that our current hypothesis selection algorithm was more accurate on average than the DST approach when allowed the same memory usage. Table I shows the result of pruning all models using the DST. In this experiment, a graph was considered solved if the true solution was retained in the hypothesis space after the exploration was complete. For each trial, the pruning method was applied first and the memory usage measured. The HMA method was then run with the maximum memory usage (N) set to the value used by the pruning method on the same trial. Note that results from graphs of densities exceeding 1.6 could not be practically obtained using the DST pruning algorithm because of the memory usage required.

Fig. 9 illustrates our approach on graphs considered in previous work. By using the LBE exploration strategy and the HMA, we were able to solve each of these previously considered graphs while maintaining only one hypothesis for each step of the exploration process.

The difficulty of the topology inference problem increases on average with the density and size of the graph. The better performance of the new exploration strategies was apparent under the more difficult circumstances. Fig. 8 shows a comparison of the different exploration strategies over ten-node and 30-node graphs of various densities. The topology of the graph under exploration was also a factor in the difficulty. Fig. 10 shows an example of two graphs, both of the same size and with the same number of edges that vary in difficulty. The more difficult of the two graphs has nodes that are only of degree two or of degree four. This ambiguity among the vertex degrees causes difficulty for the inference process. The easier of the two graphs has several nodes of unique degree, which can be used to help identify and prune out inconsistent models.

The stochastic LBE exploration with a large enough value assigned to p performed as well or better than the BFET strategy. Fig. 11 shows the effect of varying the value assigned to p for various problem instances. A higher fraction of graphs can be solved with p values near one than with lower values, but higher p values result in longer and more variable cover times. It can be seen that for a given value of p, the problem becomes more challenging, and the average cover time increases predictably as a function of the density of the edges in the graph. We used $0.9 < p < 0.99$ in the experimental results reported in this section. The value selected was used as a parameter to alter the difficulty of the problem instance.

Generally, in our experiments, the correct graph was the first ranked model among those retained once edge coverage was achieved. For example, Table II gives the result of one experiment in which 10 000 trials resulted in 7027 solved problem instances of which 99.5% ranked the correct graph first. Additionally, in the experiment presented in Fig. 8, the LBE strategy retained the true graph in its hypothesis set in 478 of 500 trials. Of those correctly solved 478 trials, the correct graph was ranked first in 476 trials.

One parameter of interest when using the HMA is the number of models maintained. If not enough models are maintained throughout the exploration process (the value assigned to N), then the chance of discarding the true solution is increased (see Fig. 12). To solve more difficult problems, maintaining a larger set of models is required (see Fig. 13). The good results, however, can be obtained using the LBE strategy with just a handful of models for smaller graphs. By increasing the number of models maintained, it is possible to correctly infer quite large graphs (see Fig. 14).

The *distribution* of the vertex size of the hypotheses generated by the various exploration algorithms reveals that the newer strategies are better at discriminating among the smaller sized models, presumably by showing inconsistent errors of the MIS-CORRESPONDENCE and NEW-LOOKS-OLD types. For example, BFET quickly generates many hypotheses, a few of which are small and have stayed consistent through much exploration, and many which are in relation quite large and, therefore, less believable. In one experiment, we allowed each exploration strategy to run until its corresponding hypothesis tree (unpruned)

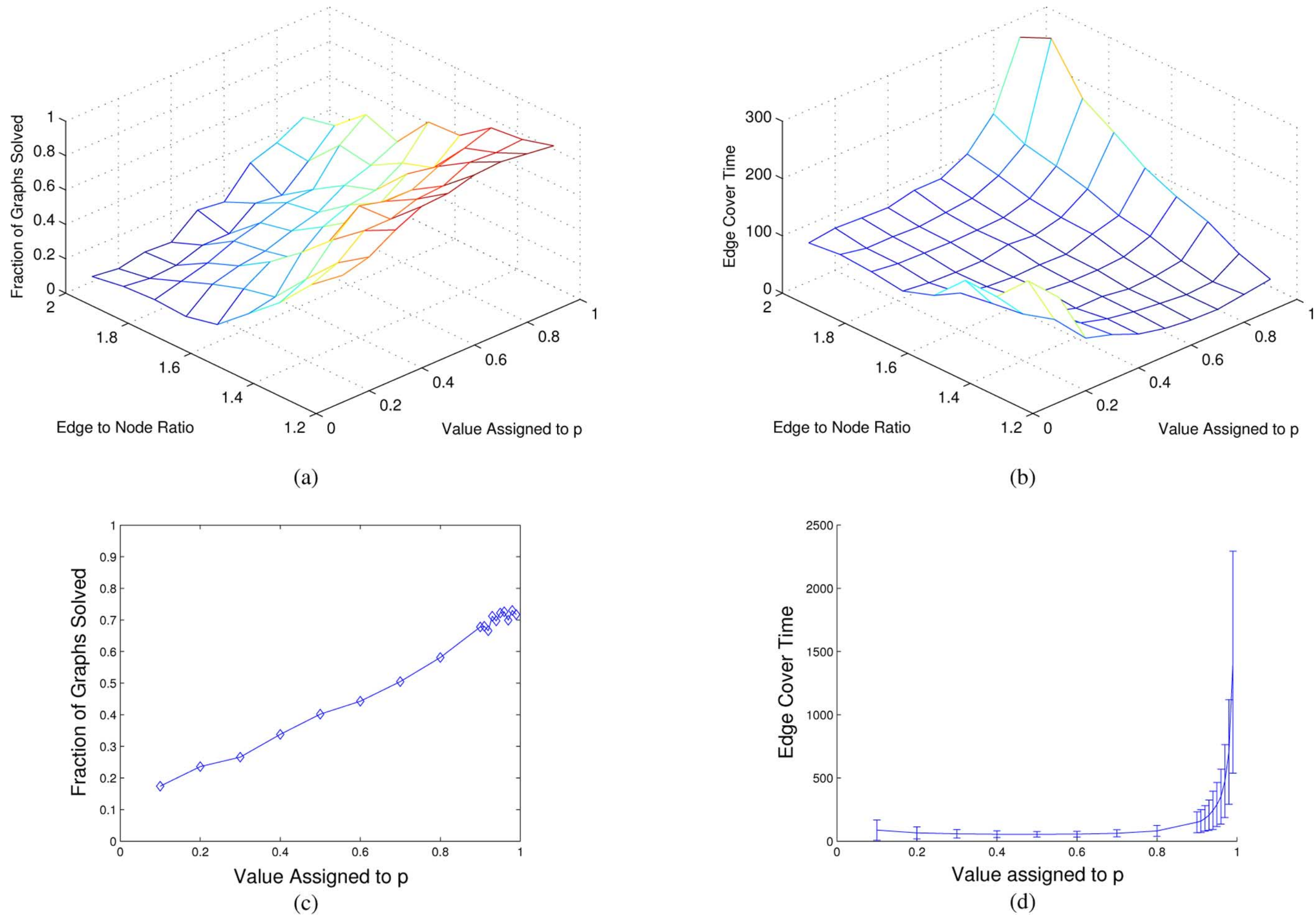


Fig. 11. Result of varying the value assigned to p parameter for the LBE on (a) and (c) fraction of graphs solved and (b) and (d) average edge cover time. Results obtained from 100 trials of the LBE algorithm with $N = 10$ on ten-node graphs.

TABLE II
RANKING RESULTS FOR THE TRUE GRAPH FOR TEN-NODE GRAPHS WITH AN EDGE TO NODE RATIO OF 1.6 USING LBE WITH $p = 10$ AND $N = 10$

Rank of True Graph	Number of Trials (out of 7027)
1	6992
2	29
3	4
4	1

Results are from 10 000 trials of which 7027 were solved.

grew to 1000 models. We then computed the normalized model size as the average of the number of vertices in the first 1000 hypotheses divided by vertex coverage obtained by the exploration strategy at that point (see Fig. 15). The statistic gives a sense of how much larger the average hypothesis is than the actual environment for the different exploration strategies (see Fig. 13 and Table III).

Although the BFET algorithm is guaranteed to cover a finite region, its cover time in practice was relatively poor (see Fig. 16). Unfortunately, this makes its use difficult for environments which are suspected to be large since the probability of coverage would be low even after considerable exploration. In the environments, we consider here, the LBE strategy does much better in practice, even with an aggressive value of p.

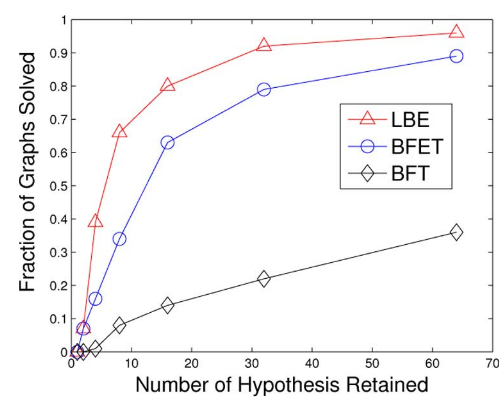


Fig. 12. Fraction of graphs solved for different numbers of hypotheses maintained by the algorithm (value of N). Results obtained from 100 trials of ten-node graphs with an edge to node ratio of 1.6. For LBE, the parameter p was assigned a value of 0.99. The exploration strategy under consideration was run until edge coverage of the graph.

Without the use of feature vectors, the computational time required to run the topology inference algorithm is largely a function of the number of hypotheses maintained (N) and the number of observations processed $|O|$. Since, in these experiments, the exploration strategy was followed until edge coverage

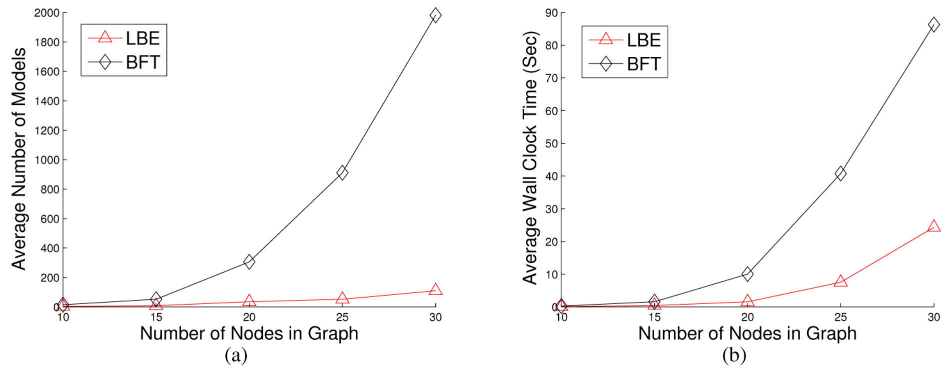


Fig. 13. Average number of (a) hypothesis required and (b) wall clock time in seconds required in order to solve graphs of various size. Results were averaged over 100 trials for each graph size using an edge to vertex density of 1.2. The value assigned to N was doubled until the true graph was present in the returned hypothesis set.

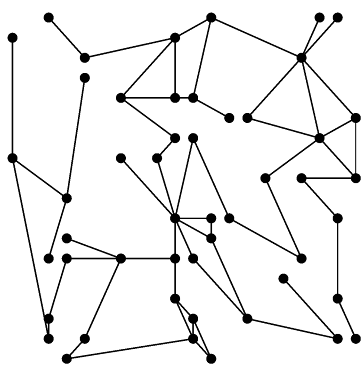


Fig. 14. Example of a 50-node graph with an edge to node ratio of 1.2 that was solved by our approach in less than an hour. The correct graph was maintained by the algorithm (with $N = 1000$) as the first ranking model from the point of edge coverage onward. LBE was used as the exploration strategy ($p = 0.99$) and achieved edge coverage at step 3918.

of the graph was achieved, the number of observations O processed is determined by the edge cover time of the exploration strategy used. Fig. 13 shows some average run time examples for the BFT and LBE strategies. Interestingly, it can be seen that in the case of the BFT strategy, the computational time is roughly proportional to the number of model used. For the LBE strategy, however, it appears that the relationship between the number of maintained models and running time is not linear. This is possibly due to the less predictable cover time of the LBE strategy.

Although the LBE strategy was not always the fastest approach for covering a given graph, it generally required less memory when processing and was able to solve difficult problem instances faster than the other strategies.

B. Incorporating Feature Vectors Into Hypothesis Assessment

In this section, we report on the effect of incorporating feature vectors into the HMA. In our simulations, we specify a feature vector of four dimensions for each vertex in the graph that is explored. The simulation tool is given a *separation metric* σ that determines how the feature vector values for the various

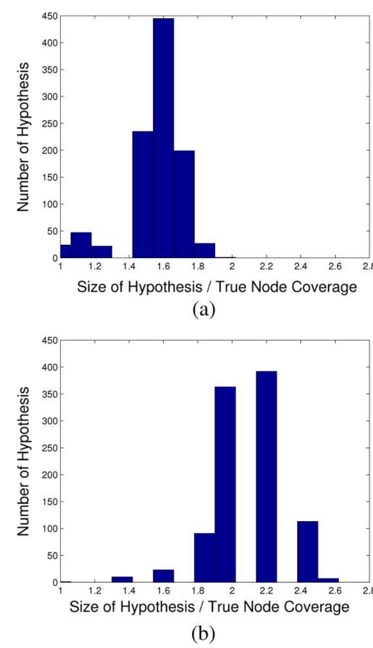


Fig. 15. Distribution of the first 1000 hypotheses generated for (a) the BFT exploration strategy and (b) the BFET exploration strategy. The result was obtained from a typical run of the algorithm on a ten-node graph with an edge-to-node density of 1.6. The horizontal axis indicates the number of vertices in a hypothesis normalized by the number of vertices reached by the exploration process at that point. BFT covered seven of the ten nodes in this time, while BFET covered only five.

TABLE III
MEAN AND STANDARD DEVIATION FOR COVERAGE AND MODEL SIZE NORMALIZED BY COVERAGE FOR THE FIRST 1000 HYPOTHESES GENERATED BY THE DIFFERENT EXPLORATION STRATEGIES

Strategy	Mean Node Coverage	Normalized Model Size
BFT	8.48 +/-(1.11)	1.22 +/-(0.19)
BFET	6.86 +/-(1.78)	1.67 +/-(0.30)
LBE ($p = 0.95$)	5.57 +/-(2.46)	2.15 +/-(0.57)
LBE ($p = 0.99$)	4.33 +/-(1.86)	2.56 +/-(0.79)

Results obtained from 100 trials on random ten-node graphs with an edge to node density of 1.6.

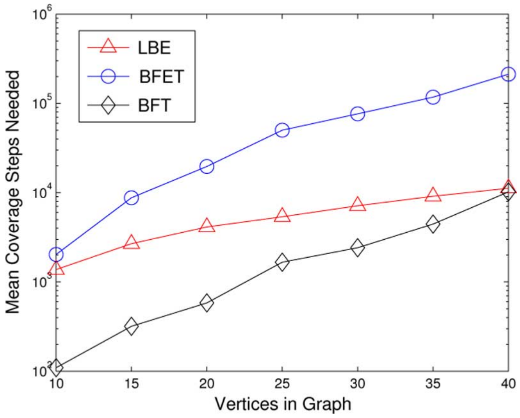


Fig. 16. Average number of steps required for edge coverage of the graph for the different exploration strategies. Note the log scale for the vertical axis. Average was taken over 100 trials using an edge density of 1.6. For LBE, the parameter p was assigned a value of 0.99.

vertices are dispersed. The vector for each vertex is drawn from a multivariate Gaussian. Each diagonal element of the covariance matrix for this multivariate is set to σ, and all other elements are set to zero. When the robot visits a specific vertex, an observation feature vector is generated by adding zero mean, normally distributed noise to the feature vector associated with that vertex. The diagonal elements of the covariance of the distribution used to generate the noise are set to one, and all other elements are set to zero.

Our experiments suggest that the use of feature vectors in the HMA is advantageous, provided there is an adequate degree of separation between the feature vectors associated with any two distinct vertices. Fig. 17 shows the result of using features vectors on various problem instances. It can be seen that when a low value is assigned to σ during the simulations, the origin of an observational feature vector becomes statistically ambiguous, and our current approach does not perform well. Under these conditions, there is a high level of noise in the θ value computed for a belief, which reduces the effectiveness of the hypotheses ranking mechanism (see Fig. 17). However, if the feature vectors are informative, their inclusion in the ranking mechanism significantly improves performance.

The prune threshold parameter ω must be set to a value higher than one in order to function properly. When a large value is assigned to the prune threshold ω, however, more computational effort is required on average to solve a problem instance [see Fig. 17(c)] .

Using feature vectors, it is possible to solve challenging problem instances. Table IV shows the result of using different separation metric values for 30-node graphs that could not be solved without the use of feature vectors. Likewise, using feature vectors with a separation metric of $\sigma = 4$, we were able to solve the graph shown in Fig. 14 with $N = 8$; this is as opposed to $N = 1000$ without the feature vector information. The computational effort require to compute the feature agreement metric θ, however, is significant. The wall clock time for this 50-node problem instance was about one third when features were used,

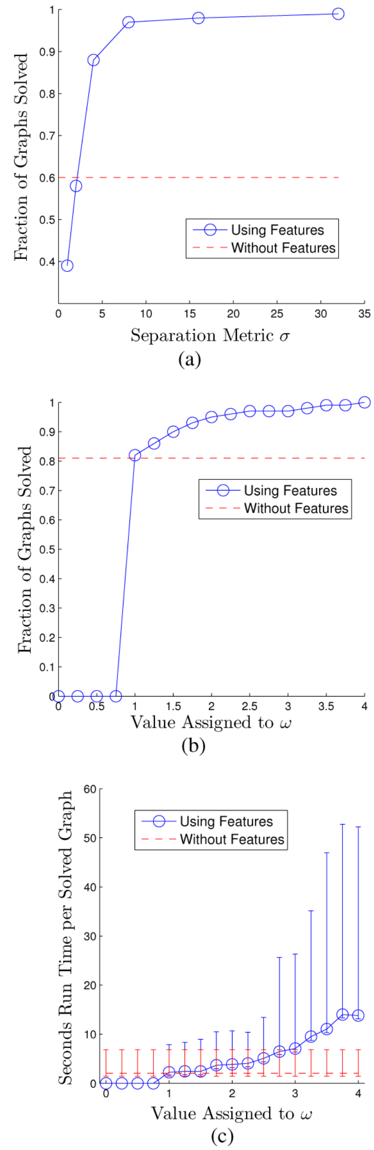


Fig. 17. Results obtained from 100 trials of ten-node graphs with an edge-to-node ratio of 1.6 using LBE with $p = 0.90$. (a) Fraction of graphs solved for different degrees of separation σ using $N = 8$ and $\omega = 2$. (b) Fraction of graphs solved for different prune thresholds ω using $N = 20$ and $\sigma = 8$. (c) Run times for *solved* problem instances as a function of ω using $N = 20$ and $\sigma = 8$; error bars show fifth and 95th percentiles.

even though less than 1% of hypotheses were maintained. The results reported in Table IV took 29 h to obtain.

VI. Background on Related Work

Topological mapping is a well-explored area in the mobile robotics community. A topological map provides: first, a useful representation of the environment that allows robot navigation without necessarily requiring the maintenance of the robot's pose in a global reference frame and, second, an abstraction provided by the topology information that can aid higher level planning and inference tasks. Early work in this area by Kuipers and Byun [30] constructed a topological network description of the environment by identifying, and then linking,

TABLE IV
FRACTION OF 30-NODE GRAPHS WITH 1.6 EDGE TO NODE RATIO SOLVED FOR DIFFERENT σ VALUES OUT OF 10 TRIALS; SAME EXPERIMENTAL SET UP AS SHOWN IN FIG. 8(B)

Separation Metric σ	Fraction of Trials Solved
4	0.2
8	0.5
16	1.0

Zero trials were solved in when feature vectors were not used.

distinctive places and paths based on the sensory input and control strategies of the robot. Later work by Dudek [31] describes an approach to build a hierarchy of representations of an unknown environment. At the lowest level, the hierarchy begins with geometric data obtained from the processing of sensory data gathered by the robot. The final abstraction is a topological map with attached semantic labels that could be used for higher level tasks. This approach is refined in later work by Simhon and Dudek [32] with the concept of *islands of reliability*, which link locally understood metric maps in a larger topological representation of the global environment.

The work by Shatkay and Kaelbling [33] addressed the topological mapping problem with statistical formulations and techniques. They model the robot's interaction with the world as a hidden Markov model and employ an extended Baum–Welch algorithm to recover its parameters. Their approach incorporates odometry data and abstracted sensory information collected by the robot. The outcome of these approaches is generally a graph, where vertices represent distinct locations or landmarks in the region and edges indicate navigability.

Our approach to topological mapping is similar in concept to work by Ranganathan and Dellaert [34]–[36] in which a robot builds up a model of the environment almost exclusively on odometry data. In addition to collect imperfect odometry data, their approach requires only that the robot have the ability to detect signature-free landmarks, i.e., the robot can detect a proximal landmark, but cannot differentiate between the different landmarks it encounters. The final outcome is a probabilistic distribution over potential topologies that describe the obtained observations. The authors refer to this result as a probabilistic topological map (PTM) and explore various inference techniques to generate the PTMs including Markov chain Monte Carlo (MCMC) [34], [36] and a Rao–Blackwellized particle filter [35]. The set of weighted partial world models, we maintain in our inference technique has some similarity to the concept of a PTM, as defined by these authors. In both our technique and theirs, a multihypothesis, topological representation is maintained. The distinguishing difference is that, while we only apply a ranking heuristic function, their work uses odometry measurements to assign relative probabilities to each of the potential world models. In our work, we do not presuppose the availability of odometry data.

Our work is also related to the work of Kuipers *et al.* [29], [37]–[39], who have also addressed the loop-closing problem in topological mapping by maintaining potential world models consistent with the observations gathered by the robot. The descriptions of the environment are modeled as undirected, embedded graphs in a manner similar to that first described in [14]. In the work by Kuipers *et al.*, planarity constraints [37] and prioritized circumscription policy [38] are considered when building a tree of topological maps that are consistent with the robot's observations. The planarity test discards potential world models that could be formed by matching up pairs of pending gateways by considering whether or not the pair belong to the same *face* in the graph under consideration. The prioritized circumscription policy is then used to assign a preference order to the topological maps that remain after applying the planarity test. Like our work, this approach gives priority to "simple" topologies when considering multiple maps consistent with the observations gathered. An active exploration by the robot is then used to discriminate among the remaining models.

The topological mapping approach described by Kuipers *et al.* is demonstrated on an environment of nine nodes. Although effective at this scale, its reliance on active exploration and the lack of a mechanism to restrict the number of hypothesis maintained during the exploration process suggests that the approach could have difficulties scaling up to larger environments. Although our approach could easily be extended to apply the planarity test, for the moment, we do not assume that the graphs we are exploring are planar. We assume only that we can assign a relative edge labeling with respect to the edge from which we entered the vertex. For example, our approach could work on a graphical representation of an urban road network in which there are underpasses and bridges. In further contrast with Kuipers' approach, we do not restrict our belief to only the "simplest" models, and we do not require that we must maintain all "simple" models in our belief space.

It should be noted that practical applications of topological mapping must provide a method for the robot to reliably identify a topological node (or landmark) in the world being explored. In work by Choset and Nagatani [40], sonar data is used to identify and position the robot on the Voronoi graph, the vertices of which correspond to topological nodes. In work by Kuipers and Beeson [39], [41], placed recognition is achieved through a multiprocess bootstrapping technique that includes sensory clustering and probabilistic inference. In additional, related work by Kuipers *et al.* [37], metric mapping methods are used to construct *local perceptual maps* that are then combined with a topological representation. The authors discuss identifying gateways (edges) within these locally mapped neighborhoods using various techniques, including using the medial axis of free space. Other approaches consider the extraction of features from vision or other sensory data, e.g., work by Se *et al.* [42], Sala *et al.* [43], and Giguere *et al.* [44]. In this paper, we have left, for the moment, the problem of identifying when the robot has reached a vertex and have focused on the topological mapping problem in its pure form.

VII. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have considered the topological mapping problem given a single mobile platform with limited sensory capabilities. We have shown that even in the case of highly ambiguous, nonunique "signatures," it is possible for a robot to infer a set of hypotheses for its environment that likely includes the true model. Our approach combines an exploration strategy that attempts to eliminate inconsistent models with a HMA that bounds the number of models maintained at each step based on the principle of Occam's razor.

In future work, we will consider the effect of dynamically adapting the exploration strategy to the region being explored and the current hypotheses maintained. For example, the next destination in the exploration process could be selected based on the information the site could potentially provide in terms of ruling out or selecting between current hypotheses. The distance that must be travelled by the robot in order to carry out the action could also be incorporated into the decision process when selecting the next region to explore. This type of dynamic exploration has been considered earlier, e.g., by Rao *et al.* [45], in related problem domains. For example, one could direct the robot toward any pending gateways that are present in the currently maintained models. A criteria for halting the exploration process would also be of use.

ACKNOWLEDGMENT

The authors would like to thank S. Whitesides for her comments.

REFERENCES

[1] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization mapping problem," in *Proc. AAAI National Conf. Artif. Intell.* Edmonton, AB, Canada: AAAI, 2002.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell*, Acapulco, Mexico: IJCAI, 2003.

[3] D. F. Wolf and G. S. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, Jul. 2005.

[4] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[5] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6. pp. 1365–1378, Dec. 2008.

[6] J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. Gonzalez, "Toward a unified bayesian approach to hybrid metric-topological slam," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 259–270, Apr. 2008.

[7] S. Thrun, "Robotic mapping: A survey," in *Proc. Exploring Artif. Intell. New Millennium*, 2003, pp. 1–35.

[8] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," presented at the IEEE Int. Conf. Robot. Autom., Barcelona, Spain, Apr. 2005.

[9] A. Martinelli, N. Tomatis, and R. Siegwart, "Some results on SLAM and the closing the loop problem," presented at the IEEE/RSJ Int. Conf. Intell. Robot. Syst., Edmonton, AB, Canada, 2005.

[10] A. Ranganathan, E. Menegatti, and F. Dellaert, "Bayesian inference in the space of topological maps," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 92–107, Feb. 2006.

[11] K. R. Beevers and W. H. Huang, "Loop closing in topological maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 4378–4383.

[12] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Using local information in a non-local way for mapping graph-like worlds," presented at the Int. Joint Conf. Artif. Intell., Chambery, France, Aug. 1993.

[13] G. Dudek, P. Freedman, and S. Hadjres, "Mapping in unknown graph-like worlds," *J. Robot. Syst.*, vol. 13, no. 8, pp. 539–559, Aug. 1996.

[14] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic exploration as graph construction," *IEEE Trans. Robot. Autom.*, vol. 7, no. 6, pp. 859–865, Dec. 1991.

[15] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Map validation and robot self-location in a graph-like world," *Robot. Autonomous Syst.*, vol. 22, no. 2, pp. 159–178, Nov. 1997.

[16] I. M. Rekleitis, V. Dujmović, and G. Dudek, "Efficient topological exploration," in *Proc. Int. Conf. Robot. Autom.*, Detroit, MI, May 1999, pp. 676–681.

[17] V. Dujmovic and S. Whitesides, "On validating planar worlds," in *Proc. 12th Annu. ACM-SIAM Symp. Discrete Algorithms*, Washington, DC, 2001, pp. 791–792.

[18] X. Deng and A. Mirzaian, "Competitive robot mapping with homogeneous markers," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 532–542, Aug. 1996.

[19] X. Deng, E. Milios, and A. Mirzaian, "Robot map verification of a graph world," *J. Comb. Optimization*, vol. 5, no. 4, pp. 383–395, Dec. 2001.

[20] M. A. Bender, A. Fernandez, D. Ron, A. Sahai, and S. Vadhan, "The power or a pebble: Exploring and mapping directed graphs," in *Proc. 30th Annu. ACM Symp. Theory Comput.*, Dallas, TX, 1998, pp. 269–278.

[21] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Topological exploration with multiple robots," presented at the 7th Int. Symp. Robot. Appl., Anchorage, AK, 1998.

[22] W. H. Huang and K. R. Beevers, "Topological map merging," *Int. J. Robot. Res.*, vol. 24, no. 8, pp. 601–613, Aug. 2005.

[23] H. Wang, M. Jenkin, and R. Dymond, "Enhancing exploration in graph-like worlds," in *Proc. Comput. Robot Vision*, Windsor, ON, Canada, 2008, pp. 53–60.

[24] G. Dudek and D. Marinakis, "Topological mapping with weak sensory data," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Vancouver, ON, Canada, Jul. 2007, pp. 1083–1088.

[25] Y. Maon, B. Schieber, and U. Vishkin, "Parallel ear decomposition search (EDS) and st-numbering in graphs," *Theor. Comput. Sci.*, vol. 47, pp. 277–298, 1986.

[26] I. M. Rekleitis, V. Dujmović, and G. Dudek, "Efficient topological exploration," in *Proc. Int. Conf. Robot. Autom.*, Detroit, MI, May 1999, pp. 676–681.

[27] M. Koucky, "Universal traversal sequences with backtracking," in *Proc. 16th Annu. IEEE Conf. Comput. Complexity*, 2001, pp. 21–27.

[28] J. Jonasson and O. Schramn, "On the cover time of planar graphs," *Electr. Commun. Probability*, vol. 5, pp. 85–90, 2000.

[29] F. Savelli and B. Kuipers, "Loop-closing and planarity in topological map-building," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 2, Sendai, Japan, Sep. 2004, pp. 1511–1517.

[30] B. Kuipers and Y. T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *J. Robot. Autonomous Syst.*, vol. 8, pp. 47–63, 1991.

[31] G. Dudek, "Environmental representation using multiple abstraction levels," *Proc. IEEE*, vol. 84, no. 11, pp. 1684–1704, Nov. 1996.

[32] S. Simhon and G. Dudek, "A global topological map formed by local metric maps," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, vol. 3, Victoria, BC, Canada, Oct. 1998, pp. 1708–1714.

[33] H. Shatkay and L. P. Kaelbling, "Learning topological maps with weak local odometric information," in *Proc. Int. Joint Conf. Artif. Intell.*, San Mateo, CA, 1997, pp. 920–929.

[34] A. Ranganathan and F. Dellaert, "Inference in the space of topological maps: An MCMC-based approach," in *Proc. Int. Conf. Intell. Robot. Syst.*, Sendai, Japan, 2004, pp. 1518–1523.

[35] A. Ranganathan and F. Dellaert, "A Rao-Blackwellized particle filter for topological mapping," in *Proc. Int. Conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 810–817.

[36] A. Ranganathan and F. Dellaert, "Data driven MCMC for appearance-based topological mapping," presented at the Robot.: Sci. Syst. Conf., Cambridge, MA, Jun. 2005.

[37] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metric and global topological maps in the hybrid spatial semantic hierarchy," in *Proc. IEEE Conf. Robot. Autom.*, 2004, pp. 4845–4851.

[38] E. Remolina and B. Kuipers, "Towards a general theory of topological maps," *Artif. Intell.*, vol. 152, no. 1, pp. 47–104, 2004.

[39] P. Beeson, "Creating and utilizing symbolic representations of spatial knowledge using mobile robots," Ph.D. dissertation, Univ. Texas at Austin, Austin, TX, Aug. 2008.

[40] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit

localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
[41] B. Kuipers and P. Beeson, "Bootstrap learning for place recognition," presented at the AAAI Nat. Conf. Artif. Intell., Edmonton, AB, Canada, 2002.
[42] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," presented at the IEEE Int. Conf. Robot. Autom., Seoul, Korea, May 2001.
[43] P. L. Sala, R. Sim, A. Shokoufadeh, and S. J. Dickinson, "Landmark selection for vision-based navigation," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 334–349, Apr. 2006.
[44] P. Giguere, G. Dudek, C. Prahacs, and S. Saunderson, "Environment identification for a running robot using inerial and actuator cues," presented at the Robot.: Sci. Syst. Conf., Philadelphia, PA, Aug. 2006.
[45] M. Rao, G. Dudek, and S. Whitesides, "Randomized algorithms for minimum distance localization," *Int. J. Robot. Res.*, vol. 26, pp. 917–933, 2007.

Dimitri Marinakis received the B.Sc. degree from the University of Victoria, Victoria, BC, Canada, and the Ph.D. degree from McGill University, Montreal, QC, Canada.

He was engaged in industry in the fields of embedded systems and wireless communications. He is currently with the Centre for Intelligent Machines, McGill University. His research interests include applying statistical and probabilistic methods to problems in sensor networks and mobile robotics.

Gregory Dudek received the Ph.D. degree from the University of Toronto, Toronto, ON, Canada.

He is currently the Director of the School of Computer Science, McGill University, Montreal, QC, Canada, and a Member and a James McGill Chair with the McGill Research Centre for Intelligent Machines, where he also directs the McGill Mobile Robotics Laboratory. His research interests include perception for mobile robotics, vision-based robotics, computational vision, and collaborative filtering.