What a Fool Believes: Characterizing Plausible Environments from Weak Sensing Histories

Aral Köse[1,2], Vadim Weinstein[1], and Steven M. LaValle[1*]

[1] Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland
[2] Electrical & Electronics Engineering Department, Boğaziçi University, Istanbul, Turkey

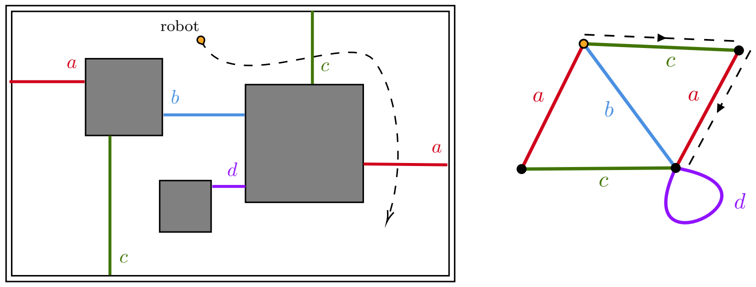
Abstract. Learning environment representations through robot interaction is an active research topic both in practice and in theory. A weaker question is that of indistinguishability. If two environments are indistinguishable but different then at least one of them cannot be properly learned. We first prove a characterization of indistinguishability in the setting of deterministic edge-labeled symmetric multigraphs. We show then that the results have a parallel in a continuous setting as well. We then show with counterexamples that the non-deterministic case is more challenging. Finally, we address the graph learning problem in which the robot is allowed to place one recognizable pebble in the environment. We provide an algorithm which is based on taking graph quotients iteratively and prove that the algorithm always produces an isomorphic copy of the underlying graph.

Keywords: graph exploration, localization and mapping, covering spaces, bisimulation, combinatorial filtering, sensor fusion, formal methods

1 Introduction

Imagine trying to fool a robot by secretly altering its environment so that its internal behavior is unchanged. Accordingly, we study problems in which a robot's information space reduces to a walk on an edge-labeled undirected multigraph. While at a vertex the robot senses all the available "ports": the labels of all edges emanating from that vertex. The robot's action is then one of those labels, which makes the robot traverse an edge with that label. If there are multiple such edges, then the edge is selected at random. A deterministic graph is one where an edge label uniquely determines a neighboring vertex of a given vertex. We ask the following questions. Under what conditions can the robot learn the graph up to isomorphism? If not, what is the equivalence relation on graphs that corresponds to indistinguishability from the robot's perspective? We prove a characterization for indistinguishability for deterministic graphs using covering graphs and give

* This work was supported by a European Research Council Advanced Grant (ERC AdG, ILLUSIVE: Foundations of Perception Engineering, 101020977) and Academy of Finland (BANG! 363637). (e-mail: firstname.lastname@oulu.fi).



(a) (b)

Fig. 1: (a) A point robot moves in a 2D environment with obstacles. If it crosses a detection region (sensor beam), then it receives a label, a, b, or c. (b) The corresponding edge-labeled multigraph, in which each undirected edge implies that motion is possible in either direction between a pair of vertices.

counterexamples for non-deterministic ones. We show that placing one pebble in a deterministic graph is sufficient to reconstruct its isomorphism type.

These questions can be derived from a variety of settings in which a robot moves through 2D or 3D worlds and receives limited sensor feedback from simple detection sensors [6, 12, 16]. Each sensor merely reports whether the robot has entered or crossed its detection region. Figure 1 shows an example. More complex examples may include a mixture of 1D or 2D detection regions, or a 3D world with 2D and 3D detection regions.

We assume that the underlying graph is completely unknown as the robot is placed arbitrarily into the environment. We also assume that after it arrives in a vertex, it can return along the same edge, if desired; this ability is lost after it transitions to another vertex. However, the robot may carry uniquely distinguishable markers called *pebbles*. It can drop one to the occupied vertex, or pick it up if there already is one.

The problem of exploring (visiting each vertex) and learning graphs (reconstrucing the graph) has been studied in more general contexts. Such as in distributed computing, where covering graph related notions for indistinguishability of graphs by a network of anonymous processors has been proposed [1,19]. These results are closely related to our Corollaries 7 and 11. A question that arose from this work was whether two graphs that had a common universal cover also had a finite common covering, which was confirmed to be true in [11]. Whereas these works focused on message-passing networks, we motivate and prove our indistinguishability results for the characterization of possible graphs explored by a robot.

In robotics, the pebble requirements of undirected [4, 5] and directed [2], graph learning has been studied with the ubiquitous assumption of a *local port labeling* where the edges incident to a vertex are labeled locally. For example, one edge can be labeled differently from the perspective of the two vertices incident, so the robot cannot *backtrack*—go back a sequence of edges it has just traversed. Precisely with this assumption, [5] gives a validation-based learning algorithm with one pebble. This also can be done with the assumption that an upper bound on the number of vertices of the unknown graph is known [2]. Otherwise, $O(\log\log n)$ pebbles are required [2, 4]. Mazes have also been studied and are generally simpler [3].

The organization of the paper is as follows: In Section 2 we formalize the notions related to graphs, define *bisimulation equivalence* and show its equivalence to indistinguishability of graphs by a robot. In Section 3 we elaborate on the larger question of topology recognition in the continuous setting. In Section 4 we treat quotients over deterministic graphs and show that a specific bisimulation relation, induced the placement of *pebbles*, is unique. In Section 5, we discuss computational aspects of quotient based algorithms and give a proof of convergence.

2 Graphs and Bisimulation

We will work with labeled undirected (bidirected) connected multigraphs with backtracking information which we will call just "graphs":

Definition 1 A *graph* is a quadruple $\mathcal{S} = (S, E, g, b)$ where S is a countable (possibly finite) set of vertices, $E \subset S \times S \times \mathbb{N}$ is a set of edges, $g\colon E \to \Lambda$ is a labeling of the edges, and $b\colon E \to E$ an edge involution (backtracking). The third edge component, a natural number, allows for multiple edges between the same ordered pairs. A graph satisfies the following conditions:

(1) The function b is a bijection and an involution (it is its own inverse), and has no fixed points, i.e. $b(e) \neq e$ for all $e \in E$,
(2) For all edges $(s_1, s_2, n), (s_3, s_4, m) \in E$, if $b(s_1, s_2, n) = (s_3, s_4, m)$, then $s_1 = s_4$ and $s_2 = s_3$
(3) For all $e \in E$, $g(e) = g(b(e))$,
(4) (Connected) For all $s, s' \in S$ there are $s_0, \ldots, s_n$ such that $s = s_0$, $s' = s_n$ and for all $k < n$ there is m with $(s_k, s_{k+1}, m) \in E$.

Given $s_1, s_2 \in S$, the notation $s_1 \xrightarrow{\lambda} s_2$ means that there is n such that $(s_1, s_2, n) \in E$ and $g(s_1, s_2, n) = \lambda$. A graph is *deterministic*, if for all $e, e' \in E(s)$, $e \neq e'$ implies $g(e) \neq g(e')$ where $E(s)$ is the set of edges emanating from s,

$$E(s) = \{(s_0, s_1, n) \in E \mid s_0 = s\}.$$

The *degree* of $s \in S$, denoted $\deg(s)$, is the cardinality of $E(s)$. Note that a self-loop is counted twice due to the technical conditions (1), (2)—hence a deterministic graph cannot have self loops. The *port set* associated with s is the multiset of labels of the edges emanating from s, $P(s) = (g(e))_{e \in E(s)}$. The function P is called the *port mapping*. An *initialized graph* is a quintuple (S, E, g, b, s_0) where $s_0 \in S$ and (S, E, g, b) is a graph, which we sometimes denote by $(\mathcal{S}, s_0)$.

Definition 2 A *path* (of length k) in a graph (S, E, g, b) is a sequence

$$(s_0, n_0, s_1 \ldots, n_{k-1}, s_k) \in (S \times \mathbb{N})^k \times S$$

for some k such that for all $j < k$, $(s_j, s_{j+1}, n_j) \in E$.

The difference of the above definition to the one stated implicitly in Def.1(4) is that here we specify the edges along which the path goes in case there are multiple edges between the same vertices. The length of the path is the number of edges in it. The path (s) of length 0 for any $s \in S$ is called *trivial*. The length of a path p is denoted $\ell(p)$. The path is a *cycle* if $s_1 = s_k$.

A *never-backtracking path* is a path $(s_0, n_0, \ldots, n_{k-1}, s_k)$ such that for all $j < k-1$, $b(s_j, s_{j+1}, n_j) \neq (s_{j+1}, s_{j+2}, n_{j+1})$. A path is *proper* if it is non-trivial and never-backtracking. A *proper cycle* is a proper path which also a cycle. A path $(s_0, n_0, \ldots, n_{k-1}, s_k)$ is an *one-step extension* of a path $(t_0, m_0, \ldots, m_{l-1}, t_l)$, if $l \leq k$ and for all $j < l$ we have $t_j = s_j$, $m_j = n_j$, and $t_l = s_l$. It is an *r-step extension*, if $k - l = r$.

Suppose a robot is walking on a graph $\mathcal{S} = (S, E, g, b)$. We assume at each vertex $s \in S$, the robot can sense the port mapping $P(s)$. It can select one of the labels in $P(s)$ after which it will traverse one of the edges which has that label (but it cannot choose which one if there are many.) It can also choose to backtrack its last move, i.e. to go along the edge $b(e)$ right after it has traversed e. We define the notion of a *bisimulation* on a graph to describe the indistinguishability of graphs explored by this robot.

Definition 3 A *bisimulation* between two graphs $\mathcal{S}_1 = (S_1, E_1, g_2, b_1)$ and $\mathcal{S}_2 = (S_2, E_2, g_2, b_2)$ is a relation $R \subset S_1 \times S_2$ satisfying the following conditions:

(1) (Zig) For all $(s_1, s_2) \in R$ and $s \in S_1$, if $s_1 \xrightarrow{\lambda} s$, then there exists $s' \in S_2$ such that $(s, s') \in R$ and $s_2 \xrightarrow{\lambda} s'$,
(2) (Zag) For all $(s_1, s_2) \in R$ and $s' \in S_2$, if $s_2 \xrightarrow{\lambda} s'$, then there exists $s \in S_1$ such that $(s, s') \in R$ and $s_1 \xrightarrow{\lambda} s$,
(3) There is $(s_1, s_2) \in R$.

If the graphs are initialized at $s_1 \in S_1$ and $s_2 \in S_2$ respectively, then, in addition, we require $(s_1, s_2) \in R$ (which implies (3).) A bisimulation is *graded* if it preserves the port data:

(4) For all $(s_1, s_2) \in R$, $P(s_1) = P(s_2)$.

If $\mathcal{S}_1 = \mathcal{S}_2$, the bisimulation is called an *auto-bisimulation*. If there is a graded bisimulation between two (initialized) graphs, we call them *indistinguishable*.

A relation $R \subset S_1 \times S_2$ is *total*, if for all $s_1 \in S_1$ there is $s_2 \in S_2$ with $(s_1, s_2) \in R$ and for all $s_2 \in S_2$ there is $s_1 \in S_1$ with $(s_1, s_2) \in R$. We note that a bisimulation between two graphs is always total (Lemma 21 in the Appendix.) Accordingly, the graphs $\mathcal{S}_1 = (S_1, E_1, g_1, b_1)$, $\mathcal{S}_2 = (S_2, E_2, g_2, b_2)$ are indistinguishable if and only if for all $s_1 \in S_1$, there exists $s_2 \in S_2$ such that the initialized graphs $(\mathcal{S}_1, s_1)$ and $(\mathcal{S}_2, s_2)$ are indistinguishable, and vice versa with the indices swapped.

Lemma 4 *Indistinguishability is an equivalence relation on the set of graphs.*

Proof. The identity relation on S witnesses reflexivity. Symmetricity follows from the symmetricity of the definition of bisimulation. We will show transitivity. If R is a graded bisimulation between $\mathcal{S}_1$ and $\mathcal{S}_2$ and P is one between $\mathcal{S}_2$ and $\mathcal{S}_3$, then

$$R * P = \{(s_1, s_3) \in S_1 \times S_3 \mid \exists s_2 \in S_2((s_1, s_2) \in R \wedge (s_2, s_3) \in P)\}$$

is a graded bisimulation between $\mathcal{S}_1$ and $\mathcal{S}_3$. ◻

Definition 5 Suppose $\mathcal{S}_1 = (S_1, E_1, g_1, b_1)$ and $\mathcal{S}_2 = (S_2, E_2, g_2, b_2)$ are graphs. The pair of maps (p, q) where $p \colon S_1 \to S_2$ and $q \colon E_1 \to E_2$ is a *(label-preserving) covering map* from $\mathcal{S}_1$ to $\mathcal{S}_2$, if the following conditions are satisfied:

(1) (Surjectivity) Both maps p and q are onto,
(2) (Label preservation) For all $e \in E_1$, $g_1(e) = g_2(q(e))$,
(3) (Edge preservation) For all $(s, s') \in S_1 \times S_1$ and all $n \in \mathbb{N}$, if $(s, s', n) \in E_1$, then for some m, $q(s, s', n) = (p(s), p(s'), m)$,
(4) (Local Isomorphism) For all $s \in S_1$, $q \restriction E(s)$ is one-to-one.

If the graphs are initialized at $s_1 \in S_1$ and $s_2 \in S_2$ respectively, then we also require $p(s_1) = s_2$. If it also satisfies

(5) (Injectivity) Both maps p and q are one-to-one,

then the pair of maps is called an *isomorphism*.

The graph $\mathcal{S}_1$ is a *covering* of (*isomorphic* to) $\mathcal{S}_2$ if there exists a covering map (isomorphism) $(p, q) \colon \mathcal{S}_1 \to \mathcal{S}_2$. We note that (analogous to how *graph homomorphisms* are classicaly written) the mapping q is redundant and can be uniquely determined from the map p when the graphs are deterministic. A covering map induces a bisimulation:

Lemma 6 *Suppose $\mathcal{S}_1 = (S_1, E_1, g_1, b_1)$ and $\mathcal{S}_2 = (S_2, E_2, g_2, b_2)$ are graphs and (p, q) is a covering map from $\mathcal{S}_1$ to $\mathcal{S}_2$. Then they are indistinguishable.*

Proof. Let $R = \{(s, p(s)) \mid s \in S_1\}$. We will check Def.3(1),(2). Suppose $(s_1, s_2) \in R$ and $s \in S_1$ is such that $s_1 \xrightarrow{\lambda} s$. For 3(1) (Zig), suppose $(s_1, s, n) \in E_1$ is an edge with label λ. Then by definition of R, $s_2 = p(s_1)$. Let $s' = p(s)$. By Def.5(3), s' is connected to s_2 by the edge $q(s_1, s, n) = (p(s_1), p(s), m)$ for some m and the label of that edge, by Def.5(2), is the same that of (s_1, s, n), i.e. equal to λ. To check Def.3(2) (Zag), suppose that $(s_1, s_2) \in R$ and $s' \in S_2$ is such that $s_2 \xrightarrow{\lambda} s'$. Then by Def.5(4) (Local Isomorphism), q is bijective from $E(s_1)$ to $E(s_2)$, so there is a unique $e \in q^{-1}(s') \cap E(s_1)$. Let s be the other endpoint of e (the one that is not s_1). By an analogous argument as above, we have $s_1 \xrightarrow{\lambda} s$ as required. By Def.5(4), this relation is graded. ◻

The graphs $\mathcal{S}_1$ and $\mathcal{S}_2$ have a *common covering* if there exists graph $\mathcal{S}'$ and the pair of covering maps (p_i, q_i) from $\mathcal{S}'$ to $\mathcal{S}_i$, $i = 1, 2$. Applying Lemma 4 to the above result, we establish a slight generalization.

Corollary 7 *If two graphs $\mathcal{S}_1 = (S_1, E_1, g_1, b_1)$ and $\mathcal{S}_2 = (S_2, E_2, g_2, b_2)$ have a common covering, then they are indistinguishable.*

We now investigate the converse of Corollary 7: Does indistinguishable graphs always share a common cover? The remainder of this section will establish this to be the case for deterministic graphs (Theorem 10) and provide a counterexample for non-deterministic graphs (Example 12.) We note that it is clear from Definition 3(4) that if $\mathcal{S}_1$ is deterministic and $\mathcal{S}_1$, $\mathcal{S}_2$ are indistinguishable, then $\mathcal{S}_2$ is deterministic.

Definition 8 Let $\mathcal{S} = (S, E, g, b)$ be a graph. The *unraveling* of $\mathcal{S}$ *(rooted) at* $s_0 \in S$ is the graph $T_{s_0}(\mathcal{S}) = (\bar{S}, \bar{E}, \bar{g}, \bar{b})$ defined as follows:

- $\bar{S}$*:* The set of all never-backtracking paths $\bar{s} = (t_0, n_0, \ldots, n_{k-1}, t_k)$ in $\mathcal{S}$ that start at $t_0 = s_0$, including the trivial path (s_0).
- $\bar{E}$*:* $(\bar{s}, \bar{s}', n) \in \bar{E}$ (which implies $(\bar{s}', \bar{s}, m) \in \bar{E}$ for some $m \in \mathbb{N}$) if and only if $n = 0$ and either $\bar{s}'$ is a one-step extension of $\bar{s}$, or vice versa.
- $\bar{g}$*:* For any edge $(\bar{s}, \bar{s}', 0) \in \bar{E}$, the label is $\bar{g}(\bar{s}, \bar{s}', 0) = g(e')$, where e' is the last edge of the longer of the two paths.
- $\bar{b}$*:* The edge involution is defined by $\bar{b}(\bar{s}, \bar{s}', 0) = (\bar{s}', \bar{s}, 0)$ for all $(\bar{s}, \bar{s}', 0) \in \bar{E}$.

A *tree* is a graph which has no proper cycles. It is straightforward to see that the unraveling of a tree is itself, and that the unraveling of a graph with proper cycles is a tree with infinitely many vertices (see Lemma 22 in the Appendix). Therefore the unraveling is always a tree. A covering $\bar{\mathcal{S}}$ of $\mathcal{S}$ is *universal* if for any covering $\mathcal{S}'$ of $\mathcal{S}$, $\bar{\mathcal{S}}$ is also a covering of $\mathcal{S}'$. The construction of the universal covering is what we call the *unraveling* of Definition 8. The following result is standard:

Fact 9 *[11] The unraveling of a graph is its unique universal covering graph.*

Then the unraveling of a graph is the unique covering which is a tree (see Figure 3 for an example). When the graphs are deterministic the unraveling describes the unique histories of the robot, the equivalence of which we will show is implied from indistinguishability in the following theorem.

Theorem 10 *Two deterministic graphs are indistinguishable if and only if they have isomorphic unravelings. The unravelings can be chosen to be rooted at the initial points of the corresponding graphs if the graphs are initialized.*

Proof. Suppose there is a graded bisimulation R between the deterministic graphs $\mathcal{S}_1 = (S_1, E_1, g_1, b_1)$ and $\mathcal{S}_2 = (S_2, E_2, g_2, b_2)$. By definition there exists $s_1 \in S_1$

and $s_2 \in S_2$ such that $(s_1, s_2) \in R$. We will take $\mathcal{S}_1$ and $\mathcal{S}_2$ to be initialized at s_1 and s_2 respectively. We want to show that $T_{s_1}(\mathcal{S}_1) = (\bar{S}_1, \bar{E}_1, \bar{g}_1, \bar{b}_1)$ is isomorphic to $T_{s_2}(\mathcal{S}_2) = (\bar{S}_2, \bar{E}_2, \bar{g}_2, \bar{b}_2)$. Let $\bar{s} = (t_0, n_0, \dots, n_{k-1}, t_k)$ be a never-backtracking path in $\mathcal{S}_1$ starting at $t_0 = s_1$ which represents an arbitrary element of $T_{s_1}(\mathcal{S}_1)$. We will define $p : \bar{S}_1 \to \bar{S}_2$ by induction on k. If $k = 0$ and $\bar{s} = (s_1)$, we let $p(\bar{s}) = (s_2)$. Suppose that p has been defined for paths of length $k = j$ and we will define p for paths $\bar{s}$ of length $k = j + 1$. Let $\bar{s} \restriction j$ be the initial segment of $\bar{s}$ of length j. Then $p(\bar{s} \restriction j)$ is defined to be the path $\bar{s}'$ in $T_{s_2}(\mathcal{S}_2)$. Let t'_j be the last end point of $\bar{s}'$ and let t_{j+1} be the last end point of $\bar{s}$. Then $(t_j, t_{j+1}, n) \in E_1$ for some $n \in \mathbb{N}$ and there is a $t'_{j+1} \in S_2$ and $m \in \mathbb{N}$ such that $g((t'_j, t'_{j+1}, m)) = g((t_j, t_{j+1}, n))$ from Def.3(1)(Zig), and $t'_{j+1} \in S_2$ and $m \in \mathbb{N}$ are unique because $\mathcal{S}_2$ is deterministic. We let $p(\bar{s})$ be the concatenation of $p(\bar{s} \restriction j)$ with (m, t'_{j+1}). Injectivity follows from determinism of S_2: If two non-backtracking paths in $\bar{S}_1$ first differ at step j, then their images under p also differ at step j. Surjectivity follows symmetrically Def.3(2)(Zag) and from the fact that S_1 is deterministic. We now define $q : \bar{E}_1 \to \bar{E}_2$ with $q(e) = (p(\bar{s}_1), p(\bar{s}_2))$, $e = (\bar{s}_1, \bar{s}_2, 0) \in \bar{E}$ and it is easy to confirm the requirements of Definition 5. This proves the direction left to right. For right to left, suppose that two graphs have isomorphic unravelings. By Fact 9, the unraveling is a covering graph of both. By Lemma 6 the unraveling is indistinguishable to both. By Lemma 4 the graphs themselves are indistinguishable. □

Applying Fact 9 and Theorem 10, we obtain the following.

Corollary 11 *Let graphs $\mathcal{S}_1 = (S_1, E_1, g_1, b_1), \mathcal{S}_2 = (S_2, E_2, g_2, b_2)$ be deterministic. The following are equivalent:*

(1) *$\mathcal{S}_1$ and $\mathcal{S}_2$ are indistinguishable,*
(2) *$T_{s_1}(\mathcal{S}_1)$ and $T_{s_2}(\mathcal{S}_2)$ are isomorphic for any $s_1 \in S_1$, $s_2 \in S_2$,*
(3) *$\mathcal{S}_1$ and $\mathcal{S}_2$ have a common covering graph.*

Proof. (1)↔(2) Follows from Theorem 10. (2)↔(3) Follows from Fact 9. □

The converse of Corollary 7 is seen to be true for deterministic graphs in Theorem 10. However, it isn't true for non-deterministic graphs:

Example 12 The relation $R = \{(0, a), (1, b), (2, c), (3, b)(4, d), (4, e)\}$ is a graded bisimulation between the graphs in Figure 2, hence they are indistinguishable. However, they don't share a common covering. Denote these graphs by $\mathcal{S}_1$ (left) and $\mathcal{S}_2$ (right). The purple and the red edges signify two different edge labels, but that is not important for this example. To see that these graphs do not have a common covering, suppose for a contradiction that $\mathcal{S}$ is one and that $(p_k, q_k) \colon \mathcal{S} \to \mathcal{S}_k$ are covering maps for $k \in \{1, 2\}$. Since 0 and a are the only vertices of degree 4, by Def.5(4) there must be $s_0 \in S$ with $p_1(s_0) = 0$ and $p_2(s_0) = a$ and the degree of s_0 is 4. All outgoing edges from s_0 must be "red" by Def.5(2). Let $1', 2', 3', 4'$ be the vertices at the other end of these four edges in $\mathcal{S}$. Then, following q_1, their degrees must be $3, 3, 3$, and 1 (in some order) but following q_2, they must be $3, 3, 1$, and 1, a contradiction.

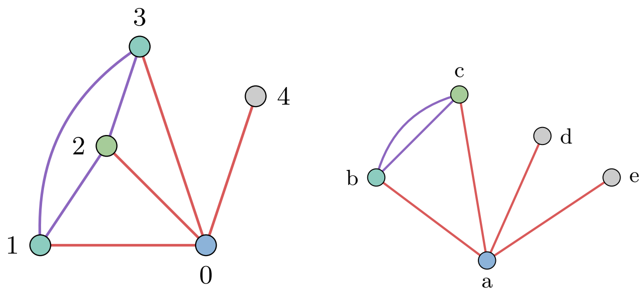


Fig. 2: Two non-deterministic graphs which are indistinguishable but have no common cover. Edge colors denote their label.

3 Connection to the Continuous Setting

We will show, relying on results from [18], that similar results can be obtained in a topological setting where the notion of covering is the standard topological one as opposed to graph-theoretic as in Definition 5. The paper [18] investigates a related problem to the one treated in the previous section from a topological and continuous perspective. In both approaches a robot finds itself in an environment where it can sense and move. The questions we ask are:

1. Under which conditions can the robot distinguish between two environments?
2. Under which conditions can the robot reconstruct the space in which it is up to a certain equivalence relation?

To generalize the setup of the previous section to a topological setup, replace the graph (S, E, g, b) with a tuple (X, p, h). Here S is replaced by a continuum or a more general topological space X. The edge-relation is implicitly replaced by the topology on X. The set of labels is replaced by the (compact) topological space U which we fix throughout this section. Denote by $\mathcal{X}$ the set of continuous paths in X and by $\mathcal{U}$ the set of measurable (continuous) paths in U. The function $p\colon X \times \mathcal{U} \to \mathcal{X}$ is called a path action and encodes similar information as the labeling function g. The function $h\colon X \to Y$ is a sensor mapping. In the previous section we assumed that the agent senses the port mapping $P(s)$ at a state $s \in S$. Here, we can analogously assume either that $h(x)$ is a local topological invariant at $x \in X$ or a homeomorphism type of a fixed neighborhood of x or an invariant of the tangent space at x (if X is a smooth manifold) or something else. It can also be, analogously to the mapping P, the homeomorphism type of the set $\{\mathbf{u} \in \mathcal{U} \mid p(x, \mathbf{u}) \neq x\}$, i.e. the set of those actions that take the robot away from x.

A covering map from a topological space X' to a topological space X is a continuous onto map which is a local homeomorphism.

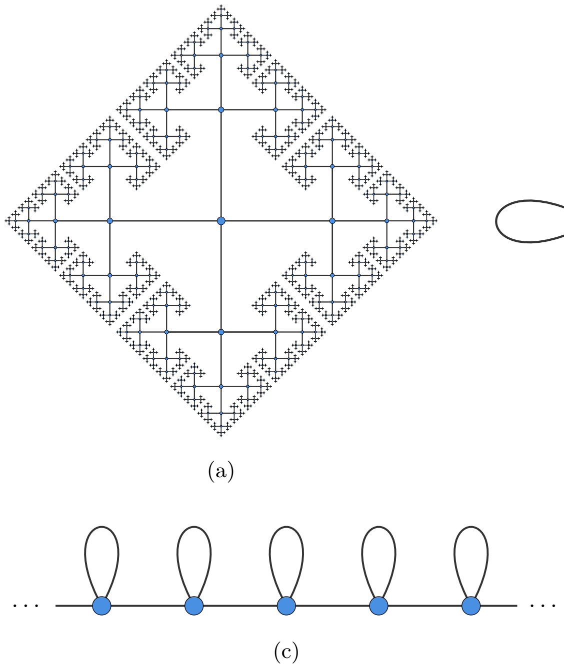


Fig. 3: The graph (b) is a covering graph of (c) and (d), whereas (a) is the universal covering graph of (b), (c), and (d). The robot cannot distinguish (a), (b), (c), and (d). Figures are reproduced from [7].

Definition 13 A *covering map* from an environment (X', p', h') to (X, p, h) is a map $f\colon X' \to X$ which is a covering map topologically and also commutes with p, p' as well as with h, h'. To be more precise, if $f\colon X' \to X$ is a continuous map, it induces a map between the path spaces $\bar{\pi}\colon \mathcal{X}' \to \mathcal{X}$ by $\bar{\pi}(x)(t) = \pi(x(t))$. Then the commutation condition means that $h \circ f = h'$ and $p(f(x'), \mathbf{u}) = \bar{f}(p'(x', \mathbf{u}))$ for all $(x', \mathbf{u}) \in X \times \mathcal{U}$.

Two environments (X_1, p_1, h_1) and (X_2, p_2, h_2) have a *common covering* if there exists the environment (X', p', h') and the covering maps $f_i : X' \to X_i$, $i = 1, 2$. Indistinguishability is defined analogously to our Definition 3, where for any input function $u_1 \in \mathcal{U}$ there exists $u_2 \in \mathcal{U}$ such that the output equality holds $h(p(x_0, u_1)) = h'(p'(x_0, u_2))$, or vice versa with the indices swapped. The following theorem was established in [18], analogous to Corollary 7.

Theorem 14 *If (X, p, h) and (X', p', h') have a common covering, then they are indistinguishable.*

In the topological setting the converse of Theorem 14 does not hold—there are spaces which are indistinguishable but do not have a common covering space:

Example 15 Let $X' = \mathbb{R}^2$, $h' \colon \mathbb{R}^2 \to \mathbb{R}$, $h'(x, y) = x$, and let U be the unit disk $U = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ (moving in all directions at a speed at most 1). Assume for the sake of this example that we limit ourselves to differentiable paths in $\mathcal{U}$. Let p' be the path action such that

$$\frac{d}{dt} p'((x, y), \mathbf{u}) = \mathbf{u}.$$

Let $X = \mathbb{R}$, $h \colon \mathbb{R} \to \mathbb{R}$, $h(x) = x$, and U is the same as above. Let p be defined by

$$p(x, \mathbf{u}) = p'((x, 0), \mathrm{pr} \circ \mathbf{u})$$

where $\mathrm{pr} \colon \mathbb{R}^2 \to \mathbb{R}$ is the projection $\mathrm{pr}(x, y) = x$. The environments (X', h', p') and (X, h, p) are indistinguishable, but X' is not a covering space of X.

In this example the projection map $\mathbb{R}^2 \to \mathbb{R}$ plays the role of a covering map, but it is not a local homeomorphism. It is conceivable that such a model corresponds to a real-life situation. For example a vacuum-cleaning robot is probably ambivalent towards the third, vertical, dimension and cannot distinguish between a room with high ceiling from a room with a low ceiling because all of its sensorimotor apparatus is confined to the two dimensions of the floor. Yet, such an example seems quite *ad hoc*.

Indeed, with some regularity assumptions on (p, h) and the manifolds they are valued on, there are necessary and sufficient characterization results [15], [10] for the existence and uniqueness of the *minimal realization* (X', p', h') of the family of input-output maps $h(p(x_0, \cdot))$, obtained by varying $x_0 \in \mathcal{X}$. Simply, the argument used is to show that the family of input-output maps of the *quotient system* obtained by the equivalence relation $x_1 \sim x_2$ iff $h(p(x_1, \cdot)) = h(p(x_2, \cdot))$ on $\mathcal{X}$ is equivalent to the input-output maps of the original system. Details about how these quotients are obtained is out of the scope of this paper. But in this case we can thus say that the minimal realizations of two environments are isomorphic, and in particular share a common covering space if they are indistinguishable.

4 Bisimulation Quotients of Deterministic Graphs

Our next question is, how much can the robot know about the underlying graph and under which conditions? Above we showed that two deterministic graphs are indistinguishable by a robot if and only if they have isomorphic universal coverings. But these graphs might still be non-isomorphic—a graph with cycles cannot be isomorphic to its unraveling, such as the graph of Figure 4 but it is indistinguishable from it by Corollary 11.

Thus, the robot cannot, in general, reconstruct the isomorphism type of the graph on which it walks. But what if the robot can place a *pebble*, a marker

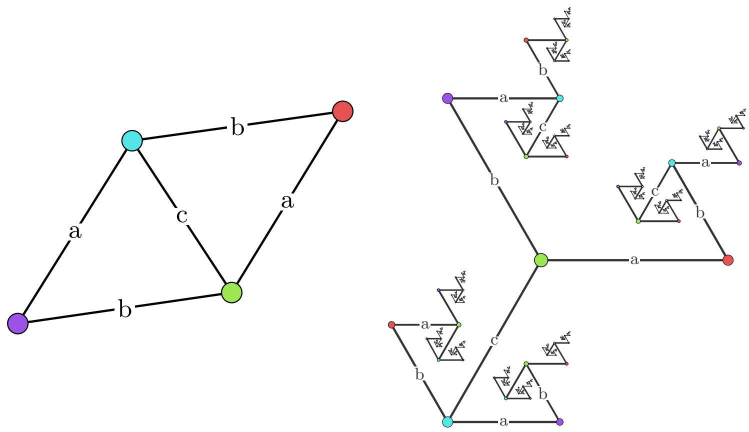


Fig. 4: A deterministic graph and its unraveling. If the robot places a pebble at the green vertex, it can learn the quotient of the unraveling with the unique bisimulation relation where the set of green vertices are an equivalence class.

which, when placed, uniquely identifies a vertex if the robot visits it in the future? When the robot places a pebble and later encounters it, it knows that it is the same pebble indeed. So having a pebble in one vertex corresponds to a unique sensory reading which the robot knows does not repeat anywhere else. Correspondingly, the vertices of the unraveling should be identified such that the vertices corresponding to each pebbled vertex is a unique equivalence class.

Definition 16 Let $\mathcal{S} = (S, E, g, b)$ be a graph, $R \subseteq S \times S$ be an equivalence relation. The *quotient* graph $\mathcal{S}/R \coloneqq (S/R, E/R, g/R, b/R)$ is defined as follows. The state set S/R is the set of R-equivalence classes. For any two classes $U, V \in S/R$ and label $\lambda \in \Lambda$, there is an edge $(U, V, \cdot) \in E/R$ with multiplicity n and label λ where $n = \max_{u \in U} \left(\sum_{v \in V} k^{\lambda}_{u,v}\right)$, where $k^{\lambda}_{u,v}$ is the multiplicity of the edge $(u, v, \cdot) \in E$ with label λ.

After the quotient, not every equivalence relation R preserves bisimulation equivalence to the original graph. On one extreme, we can have $(s_1, s_2) \in R$ for all $s_1, s_2 \in S$ which gives a single vertex quotient graph with all self loops. We will show that it is sufficient for an equivalence relation to be a graded bisimulation for the quotient to preserve bisimulation.

Proposition 17 *Let $\mathcal{S} = (S, E, g, b)$ be a graph and $R \subseteq S \times S$ be an equivalence relation. If R is a graded auto-bisimulation of $\mathcal{S}$, then $\mathcal{S}$ and $\mathcal{S}/R$ are indistinguishable.*

Proof. Assume R is a graded auto-bisimulation. Let $R' = \{(s, [s]) \mid s \in S\}$. We show that R' is a graded bisimulation. R' is total as R is. Let U and V be R-equivalence classes. Then for all $u \in U$ and label λ, the sum $N = \sum_{u \in V} k^{\lambda}_{u,v}$

equals the multiplicity of the label λ in $P(u)$. Since R is graded, by Def.3(4), this number does not depend on the choice of $u \in U$. Consequently, the multiplicity of the edge between U and V with label λ equals N. Thus, $P(U) = P(u)$ for all $u \in U$ and so $P(s) = P([s])$ for all $s \in S$.

Let us now prove conditions Def.3(1) and (2). For 3(1)(Zig), let $(s, [s]) \in R'$ and $s \xrightarrow{\lambda} s'$. Let $V = [s']$. The existence of the edge $s \to s'$ implies $\sum_{v \in V} k^{\lambda}_{s,v} \geq 1$. Thus, the multiplicity in the quotient is at least 1, meaning $[s] \xrightarrow{\lambda} [s']$. The pair $(s', [s'])$ is in R' by definition. (Zag) Let $(s, [s]) \in R'$ and $[s] \xrightarrow{\lambda} V$ for some R-equivalence class V. By Definition 16, the edge multiplicity in the quotient is non-zero, which implies $\max_{u \in [s]} \sum_{v \in V} k^{\lambda}_{u,v} \geq 1$. Therefore, there exists a representative $u \in [s]$ and a state $v \in V$ such that $u \xrightarrow{\lambda} v$. Since $(s, u) \in R$ and R is a bisimulation, there exists s' such that $s \xrightarrow{\lambda} s'$ and $(s', v) \in R$. Thus $s' \in [v] = V$, so $(s', V) \in R'$. □

In general, the equivalence relation obtained by identifying pebbled vertices is not a bisimulation relation. To apply Proposition 17, we must consider bisimulation relations which contain it. We denote the finite or infinite sequences with elements in Λ as Λ^*. We denote $s_1 \xrightarrow{\lambda_1} \cdots \xrightarrow{\lambda_n} s_n$ by $s_1 \xrightarrow{w} s_2$ for $w = \{\lambda_i\}_{i=1}^{n} \in \Lambda^*$.

Lemma 18 *Let $\mathcal{S} = (S, E, g, b)$ be a deterministic graph. Let $\mathcal{S}' = (S', E', g', b')$ be a covering of $\mathcal{S}$ via (p, q), and $R \subseteq \{(s'_1, s'_2) \in S' \times S' \mid p(s'_1) = p(s'_2)\}$ be symmetric. The unique smallest auto-bisimulation relation on $\mathcal{S}'$ containing R is the relation B_R defined as:*

$$(u', v') \in B_R \iff \exists (s'_1, s'_2) \in R,\ \exists w \in \Lambda^* \text{ s.t. } s'_1 \xrightarrow{w} u' \text{ and } s'_2 \xrightarrow{w} v' . \quad (1)$$

If $R = \{(s'_1, s'_2) \in S' \times S' \mid s'_1, s'_2 \in p^{-1}(y),\ y \in Y\}$ for some $Y \subseteq S$, then B_R is the unique auto-bisimulation relation containing R with the property that for all $(u, v) \in B_R$ and for all $y \in Y$, we have $p(u) = y$ if and only if $p(v) = y$.

Proof. Let $\mathcal{B}$ be the set of all auto-bisimulation relations on $\mathcal{S}'$ containing R. We show $B_R \in \mathcal{B}$. First, for $w = \varepsilon$ denoting the empty word, we have $u' \xrightarrow{\varepsilon} u'$ and $v' \xrightarrow{\varepsilon} v'$ for any pair $(u', v') \in R$, hence $R \subseteq B_R$. (Zig) Let $(u'_1, v'_1) \in B_R$ witnessed by history $w \in \Lambda^*$ and origins $(s'_1, s'_2) \in R$. Let $u'_1 \xrightarrow{\lambda} u'_2$. Since the $\mathcal{S}$ is deterministic, u'_2 is the unique destination of the sequence $w\lambda$ starting from y'_1. Since $(s'_1, s'_2) \in R$, we know $p(s'_1) = p(s'_2)$. Thus, s'_1 and s'_2 cover the same vertex in $\mathcal{S}$ and admit identical transition sequences. Therefore, $s'_2 \xrightarrow{w\lambda} v'_2$ exists. Consequently, (u'_1, v'_1) is reached by history $w\lambda$ from the pair $(y'_1, y'_2) \in R$, so $(u'_2, v'_2) \in B_R$. (Zag) holds symmetrically.
(B_R is minimal) Let $B \in \mathcal{B}$ be a bisimulation relation with $R \subseteq B$. If $(u, v) \in B_R$, they are reached by some history w from a pair $(s'_1, s'_2) \in R$. Since $R \subseteq B$ and B is a bisimulation, the equivalence must be preserved along w. The endpoints are unique as $\mathcal{S}$ is deterministic, forcing $(u, v) \in B$. Thus $B_R \subseteq B$, and $B_R \subseteq \bigcap_{B \in \mathcal{B}} B$ as B was arbitrary. As $B_R \in \mathcal{B}$, we conclude $B_R = \bigcap_{B \in \mathcal{B}} B$.
(B_R is unique) Assume $R = \{(s'_1, s'_2) \in S' \times S' \mid s'_1, s'_2 \in p^{-1}(y), y \in Y\}$ for some $Y \subseteq S$. For any $(u', v') \in B_R$ originating from $(s'_1, s'_2) \in R$ via w, the covering

property $p(s_1') = p(s_2')$ and deterministic projection imply $p(u') = p(v')$. Thus for any $y \in Y$, $u' \in p^{-1}(y)$ if and only if $v' \in p^{-1}(y)$. Let $B \supseteq R$ be any bisimulation on $\mathcal{S}$ satisfying this condition. Minimality implies $B_R \subseteq B$. Conversely, for any $(u', v') \in B$ and for any $y \in Y$, connectedness imply a path w exists from u' to some $s_1' \in p^{-1}(y)$. Bisimulation ensures $\exists s_2'\ v' \xrightarrow{w} s_2'$ with $(s_1', s_2') \in B$. By assumption on B, $s_2' \in p^{-1}(y)$, implying $(s_1', s_2') \in R$. The symmetry (undirected edges) of the graph ensures the reversibility of w, hence (u', v') is reachable from $(s_1', s_2') \in R$, so $(u', v') \in B_R$. Thus $B = B_R$. □

Remark 19 *In previous work, B_R was called the* minimal sufficient refinement *of R [14, 17].*

Let (π_1, π_2) be a covering map from $T_{s_0}(\mathcal{S})$ to $\mathcal{S}$.

Theorem 20 *Let $\mathcal{S} = (S, E, g, b)$ be a deterministic graph, $s_0 \in S$ and R an equivalence relation on S such that $\pi_1^{-1}(s_1)$ is an equivalence class for some $s_1 \in S$. Then $T_{s_0}(\mathcal{S})/B_R$ is isomorphic to $\mathcal{S}$ for B_R of (1).*

Proof. The unique graded bisimulation B_R (1) established in Lemma 18 satisfies $(u, v) \in B_R$ if and only if $\pi_1(u) = \pi_1(v)$. It follows that B_R is the kernel of map π_1, and that $T_{s_0}(\mathcal{S})/B_R$ is isomorphic to $\mathcal{S}$. □

5 Learning Deterministic Graphs

We now address the problem of learning a deterministic graph. It has been shown in [8] that a finite automaton cannot explore an arbitrary graph using only one pebble while it was proved in [3] that two pebbles are enough. Our algorithms, however, use unbounded memory because they are supposed to construct an isomorphic copy of the graph they are in. This is a strictly harder task than exploration which only demands that each vertex is visited (and sometimes it is made more challenging by demanding that the algorithm halts after the task has been completed). Learning the isomorphism type requires exploring the graph and it also requires unbounded memory. Therefore our result that the isomorphism type can be learned using only one pebble is not a contradiction to [8].

Let $\mathcal{S} = (S, E, g, b, s_0)$ be a deterministic graph. For this case, as established in Theorem 10, the robot's history is canonically represented by the unraveling tree $T_{s_0}(\mathcal{S})$. For any graph containing cycles, $T_{s_0}(\mathcal{S})$ is infinite. By Theorem 20, the learning problem is equivalent to identifying the relation R identifying all vertices in $T_{s_0}(\mathcal{S})$ where the pebble is detected. While the full relation R required to quotient $T_{s_0}(\mathcal{S})$ to $\mathcal{S}$ is infinite, for $\mathcal{S}$ with finitely many vertices, Lemma 18 can be augmented (Lemma 23 in the appendix) to ensure the unique bisimulation relation which retains the lifts of the pebbled vertices is generated by a finite relation R.

In essence, the efficiency of a quotient based algorithm is determined by the efficiency of the traversal strategy it employs on the unknown graph. Indeed, a traversal strategy which occasionally gets stuck on the loops of the graph would result in worse performance than one which does not. Before defining a specific traversal strategy actuating the robot, we establish the correctness of and complexity bounds applicable to the general class of algorithms that iteratively construct a hypothesis graph via quotienting.

Correctness Consider any exploration strategy that progressively expands a hypothesis graph $\mathcal{H}_k$ and applies the quotient operation whenever the pebble is detected. At every step, the quotient of the unraveling with respect to the B_R is indistinguishable from the true environment $\mathcal{S}$ by Proposition 17.

For the learning to be complete, the exploration strategy must ensure that every edge in the physical graph is eventually traversed. Since the environment is finite and connected, and the quotient respects the deterministic transitions, the sequence of hypotheses $\mathcal{H}_k$ must converge to a graph isomorphic to $\mathcal{S}$ once all cycles forming the *generators of the fundamental group* of the graph are traversed and identified. Equivalently, let $T_S \subset \mathcal{S}$ be a *spanning tree* (a tree subgraph, which contains every vertex of $\mathcal{S}$). The robot must separately traverse (and identify) paths which go any edge in $\mathcal{S}$ which is not in T_S.

Formally, we can prove that in finite time the algorithm reaches the quotient $T_{s_0}(\mathcal{S})/B_{\pi_1}$ where $B_{\pi_1} = \{(t_1, t_2) \mid \pi_1(t_1) = \pi_1(t_2)\}$ is the same as B_R of Theorem 20. To prove this, we can assume that the algorithm produces pairs (T_0, E_0), $(T_1, E_1), \dots$ where T_k is a finite subtree of $T_{s_0}(\mathcal{S})$ and E_k is an equivalence relation on T_k and $\mathcal{H}_k = \bar{T}_k/\bar{E}_k$ where $(\bar{T}_k, \bar{E}_k)$ is the *closure* of (T_k, E_k). The closure is defined so that it corresponds to the "wrapping" of a potentially infinite part of T "around" the part of the graph that has already been successfully explored and where the equivalence has been partially established correctly. The details of this formal treatment and proof are in the Appendix.

Computational Complexity The time complexity of this class of algorithms is dominated by the cost the computation of quotient graphs. Let $n = |S|$ be the number of states and $m = |E|$ be the number of edges in graph $\mathcal{S}$.

The crucial operation is the quotient computation. When a pebble is detected, the algorithm must propagate bisimulation equivalence on the relation R identifying the pebbles throughout the graph, and must merge equivalent vertices. This partition refinement process is analogous to Hopcroft's algorithm [9] for minimizing finite automata which runs in $O(\bar{m} \log \bar{n})$ time, in which $\bar{n}$ and $\bar{m}$ are the numbers of vertices and edges of the graph to be quotiented. Denote the k-step finite subtree of the unraveling as $T^k_{s_0}(\mathcal{S})$. We are looking for $T^k_{s_0}(\mathcal{S})$ with smallest $k \in \mathbb{N}$ s.t. $T^k_{s_0}(\mathcal{S})/B_R$ is isomorphic to $\mathcal{S}$. The proof of validity we give requires $k \geq 2N$, where N is the *pumping number* (see the Appendix). The degree of the graph is bounded by $m - n + 1$. We set $k = 2n + 2 \geq 2N$. It follows that $\bar{n}, \bar{m} \leq (2n+2)(m-n+1)$, and hence the time complexity of the quotient is $O(mn \log mn)$. From our proof of validity, we obtain this time complexity, which might not be optimal. We will leave the further improvements for future work.

Mechanical Complexity The robot cannot teleport between vertices in the frontier of its exploration. Any exploration algorithm requires the robot to backtrack through the known structure to reach the next pending vertex.

In the worst case, the robot traverses a path of length $O(n)$ to discover each of the m edges. Consequently, the total number of edge traversals invoked by the robot is bounded by $O(mn)$. While this matches the polynomial time bounds for single-pebble exploration in [5], our approach offers a distinct advantage: The algorithm never moves the pebble. Furthermore, the quotient step reduces the effective size of the graph after every loop closure, potentially reducing the path lengths required for subsequent backtracking steps. This bound can be potentially improved with more pebbles: If the robot were to place a pebble at each vertex it visits, traversal can be done in $O(n)$ [13].

We present a sample algorithm that iteratively constructs a hypothesis graph $\mathcal{H}_k$ by computing the quotient of the explored unraveling with respect to the constraints imposed by a single pebble.

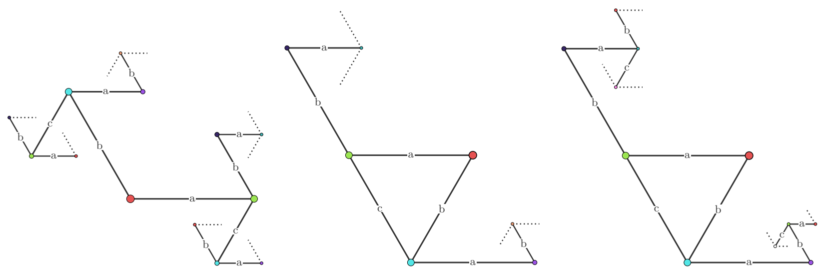


Fig. 5: The BFS algorithm applied to the environment in Figure 4 with pebble placed at the red vertex. The algorithm first searches to depth 3 obtaining the graph on the left, and applies the quotient to obtain the middle graph. To obtain the base graph in Figure 4, the algorithm searches to depth 4 to obtain the graph on the right, and applies the quotient once more.

5.1 A Sample Algorithm

We present a simple Breadth-First Search (BFS) approach. We expand the frontier of a hypothesis graph $\mathcal{H}_k$ layer-by-layer, applying the quotient operation *online*. Denote the k-step finite subtree of the unraveling as $T^k_{s_0}(\mathcal{S})$. The procedure is defined as follows:

1. **Initialization:** The robot starts at the pebbled vertex s_0. The initial hypothesis is $\mathcal{H}_0 = (\{s_0\}, \emptyset)$.
2. **Expansion:** At step k, perform a BFS expansion on the physical graph from locations corresponding to the frontier $F_k \subset V(\mathcal{H}_k)$ of vertices with

unexplored edges. For every observed transition $u \xrightarrow{\lambda} v$ in $\mathcal{S}$, add an edge and a new vertex to $\mathcal{H}_k$, resulting in the expansion $T^k_{s_0}(\mathcal{S})$.
3. **Quotient Update:** Check $T^k_{s_0}(\mathcal{S})$ for the pebble. Let P_k be the set of pebbled vertices. If $|P_k| > 1$, then with $R = P_k \times P_k$ compute the new hypothesis by taking the quotient
$$\mathcal{H}_{k+1} = T^k_{s_0}(\mathcal{S})/B_R. \tag{2}$$
4. **Termination:** Repeat steps 2–3 until the frontier is empty.

It is clear that the specific algorithm we proposed is rather inefficient and heuristics can improve the polynomial runtime. For example, the mechanical complexity can be improved in expectation by employing routines which move the pebble to the frontier, and by the applying the quotient immediately after the appearance of a pebbled vertex on the frontier vertices. Additionally, the computational complexity of the quotient operation can be improved by applying the quotient $\mathcal{H}_k$ to the hypothesis graph instead of $T^k_{s_0}(\mathcal{S})$.

6 Conclusion

We studied indistinguishability of graphs explored by a robot and have characterized the class of indistinguishable deterministic graphs by a covering graph condition for a robot without pebbles. One of the directions of this characterization holds for the continuous category of topological environments. We proved the robot can learn deterministic graphs up to graph isomorphism and showed the validity of a class of quotient based learning algorithms for a robot with one pebble. As opposed to [8] which shows that this is not possible for finite automata, similar to [5], our algorithms use unbounded memory to construct the isomorphic representation.

Several open questions remain. First, our characterization of indistinguishability does not fully extend to nondeterministic graphs, where the uniqueness of the universal covering fails. Future work should investigate "weak coverings" to resolve the equivalence relation for nondeterministic settings. Second, given our model of the robot, it is clear that the robot with one pebble cannot *deterministically learn* (provide a certificate when it has finished learning) on nondeterministic graphs. For example, any k-regular graph with one edge label and more than two vertices cannot distinguished. A graph is said to be *nondeterministically learned* if for any $1 > \delta > 0$, there is $k \in \mathbb{N}$ such that the reconstruction is isomorphic to the graph with probability $1 - \delta$ for a random walk of length k. To give a bounds on the number of minimum pebbles required to deterministically and nondeterministically learn graphs is an interesting direction for future research.

References

1. Angluin, D.: Local and global properties in networks of processors (extended abstract). In: Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing. p. 82–93. STOC '80, Association for Computing Machinery, New York, NY, USA (1980). https://doi.org/10.1145/800141.804655, https://doi.org/10.1145/800141.804655
2. Bender, M.A., Fernández, A., Ron, D., Sahai, A., Vadhan, S.: The power of a pebble: Exploring and mapping directed graphs. Information and Computation **176**(1), 1–21 (2002). https://doi.org/https://doi.org/10.1006/inco.2001.3081, https://www.sciencedirect.com/science/article/pii/S0890540101930810
3. Blum, M., Kozen, D.: On the power of the compass (or, why mazes are easier to search than graphs). 19th Annual Symposium on Foundations of Computer Science (sfcs 1978) pp. 132–142 (1978), https://api.semanticscholar.org/CorpusID:11806648
4. Disser, Y., Hackfeld, J., Klimm, M.: Tight bounds for undirected graph exploration with pebbles and multiple agents. J. ACM **66**(6) (Oct 2019). https://doi.org/10.1145/3356883, https://doi.org/10.1145/3356883
5. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: Robotic exploration as graph construction. IEEE Transactions on Robotics and Automation **7**, 859 – 865 (01 1992). https://doi.org/10.1109/70.105395
6. Guibas, L.: Sensing, tracking and reasoning with relations. Signal Processing Magazine, IEEE **19**, 73 – 85 (04 2002)
7. Hatcher, A.: Algebraic topology. Cambridge University Press (CUP) (2002)
8. Hoffmann, F.: One pebble does not suffice to search plane labyrinths. In: Gécseg, F. (ed.) Fundamentals of Computation Theory. pp. 433–444. Springer Berlin Heidelberg, Berlin, Heidelberg (1981)
9. Hopcroft, J.: An n log n algorithm for minimizing states in a finite automaton. In: Kohavi, Z., Paz, A. (eds.) Theory of Machines and Computations, pp. 189–196. Academic Press (1971). https://doi.org/https://doi.org/10.1016/B978-0-12-417750-5.50022-1, https://www.sciencedirect.com/science/article/pii/B9780124177505500221
10. Jakubczyk, B.: Existence and uniqueness of realizations of nonlinear systems. SIAM Journal on Control and Optimization **18**(4), 455–471 (1980). https://doi.org/10.1137/0318034, https://doi.org/10.1137/0318034
11. Leighton, F.T.: Finite common coverings of graphs. Journal of Combinatorial Theory, Series B **33**(3), 231–238 (1982). https://doi.org/https://doi.org/10.1016/0095-8956(82)90042-9, https://www.sciencedirect.com/science/article/pii/0095895682900429
12. O'Kane, J.M., Shell, D.A.: Automatic reduction of combinatorial filters. In: Proc. IEEE International Conference on Robotics and Automation (2013)
13. Panaite, P., Pelc, A.: Exploring unknown undirected graphs. Journal of Algorithms **33**(2), 281–295 (1999). https://doi.org/https://doi.org/10.1006/jagm.1999.1043, https://www.sciencedirect.com/science/article/pii/S019667749991043X
14. Sakcak, B., Weinstein, V., LaValle, S.M.: The limits of learning and planning: Minimal sufficient information transition systems. In: 2022 International Workshop on the Algorithmic Foundations of Robotics (WAFR) (2022)
15. Sussmann, H.J.: Existence and uniqueness of minimal realizations of nonlinear systems. Mathematical systems theory **10**(1), 263–284 (1976). https://doi.org/10.1007/BF01683278, https://doi.org/10.1007/BF01683278

16. Tovar, B., Cohen, F., Bobadilla, L., Czarnowski, J., LaValle, S.M.: Combinatorial filters: Sensor beams, obstacles, and possible paths. ACM Trans. Sen. Netw. **10**(3) (May 2014). https://doi.org/10.1145/2594767, https://doi.org/10.1145/2594767
17. Weinstein, V., Sakcak, B., LaValle, S.M.: An enactivist-inspired mathematical model of cognition. Frontiers in Neurorobotics **16** (2022)
18. Weinstein, V.K., LaValle, S.M.: Equivalent environments and covering spaces for robots (2024), https://arxiv.org/abs/2402.18323
19. Yamashita, M., Kameda, T.: Computing on anonymous networks. i. characterizing the solvable cases. IEEE Transactions on Parallel and Distributed Systems **7**(1), 69–89 (1996). https://doi.org/10.1109/71.481599