Observability Properties of Colored Graphs

Mark Chilenski, George Cybenko *Fellow, IEEE,* Isaac Dekine, Piyush Kumar, and Gil Raz

Abstract—A colored graph is a directed graph in which either nodes or edges have been assigned colors that are not necessarily unique. Observability problems in such graphs are concerned with whether an agent observing the colors of edges or nodes traversed on a path in the graph can determine which node they are at currently or which nodes they have visited earlier in the path traversal. Previous research efforts have identified several different notions of observability as well as the associated properties of colored graphs for which those types of observability properties hold. This paper unifies the prior work into a common framework with several new analytic results about relationships between those notions and associated graph properties. The new framework provides an intuitive way to reason about the attainable path reconstruction accuracy as a function of lag and time spent observing, and identifies simple modifications that improve the observability properties of a given graph. This intuition is borne out in a series of numerical experiments. This work has implications for problems that can be described in terms of an agent traversing a colored graph, including the reconstruction of hidden states in a hidden Markov model (HMM).

Index Terms—Graph theory, graph labeling, Markov processes, weak models, tracking.

1 Introduction

Consider an agent traversing a directed graph whose nodes (or edges) are assigned one or more colors. The agent seeks to localize itself within the graph based on the colors observed. In such problems, the nodes can represent states, the edges can represent allowed state transitions, and the colors can represent the discrete symbols that can be emitted by a given state (in the case of node-colored graphs) or a given state transition (in the case of edge-colored graphs).

For example, in the cybersecurity domain, the nodes may be the basic blocks of a software program's control flow graph (CFG), the edges may be the allowed control flow transitions (dictated by jump, call, and return instructions in the program), and the colors may be the specific signals emitted in a side-channel (such as electromagnetic emissions) [1], [2]. An additional example is the application of non-probabilistic weak models to tracking targets in sensor networks [3]. Although our formulations and framework do not require probabilistic models for state transitions or color emissions, the formulations can still be useful for understanding observability properties of stochastic models such as hidden Markov models (HMMs) [4].

We consider various formulations of inferring the nodes and edges visited by an agent when only the sequence of colors emitted can be observed directly. What can be inferred about nodes and edges is collectively called an observability property. Previous work has identified several classes of colored graphs, each having different implications for the tractability and accuracy of this inference problem [5], [3], [6]. The present work unites these concepts in a comprehensive framework based on the presence or absence of particular pathologies in the graph and its coloring. We also provide a unified structure to reason about the effects of specific pathologies and to design mitigations that improve the observability properties of a given graph.

The readers should note that we are not studying the more challenging problem of inferring the graph and its coloring from sequences of color observations which is a different computational problem [7]. In particular, we are assuming that the graph and its coloring are known *a priori* and not being learned by the observer.

The rest of the paper is structured as follows: Section 2 describes our notation and reviews the previous work on colored graph models and observability classes. Section 3 presents the colored graph pathologies and discusses their implications and possible mitigations. Section 4 presents the relationships between colored graph observability classes. Section 5 illustrates the implications of the various pathologies using simulations. Finally, Section 6 summarizes the contributions of this paper and discusses potential directions for future work.

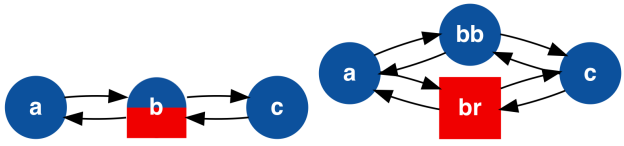
2 Background and Previous Work

2.1 Definitions: Colored Graphs, Weak Models, and Hidden Markov Models

A *node-colored directed graph* $G = (V, E, L, \Phi)$ consists of a set of nodes V, a set of edges E consisting of ordered pairs of nodes, a set of possible colors Φ, and a mapping $L : V \to 2^{\Phi}$ which indicates which subset of Φ can be emitted by a given node. This is identical to the definition of a *weak model* given in [3]. (Note that some authors also include the set of nodes which the system can start at as part of their definition of a weak model [8], [5].)

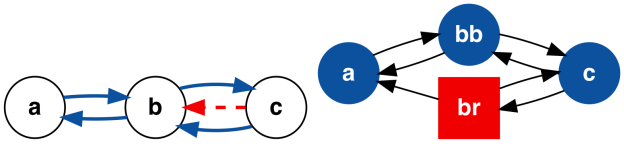
If each edge $(i, j) \in E$ is endowed with a transition probability $P_{ij} = P(X_{t+1} = j | X_t = i)$ (where X_t is the node visited at time t) and each node $i \in V$ is endowed with a set of emission probabilities $B_{i\alpha} = P(Y_t = \alpha | X_t = i)$ (where

- *M. Chilenski, I. Dekine, P. Kumar, and G. Raz are with Systems & Technology Research.*
- *G. Cybenko is with Dartmouth College.*
- *Preprint of paper to be submitted to* IEEE Transactions on Network Science and Engineering.



(a) Multi-colored (b) Single-colored

Fig. 1: Transformation of multi-colored graph to single-colored graph. Node b can emit either ● or ■, so it is split into nodes bb and br, respectively. (This paper will use both color and shape to distinguish node "colors.")



(a) Edge-multi-colored (b) Node-colored

Fig. 2: Transformation of edge-multi-colored graph to node-colored graph. Node b has incident edges which are both blue/solid and red/dashed, so it is split into blue/circle node bb and red/square node br.

Y_t is the color emitted at time t and $\alpha \in \Phi$), the model is a *hidden Markov model* (HMM) with discrete symbols [4]. We are concerned here with structural properties of such systems that depend only on whether certain state transitions and emissions are possible or not. Consequently, our results are about whether certain inferences about colored graphs are true or not true, as opposed to what the probabilities or likelihoods of inferences are.

A node-colored graph is *multi-colored* if there exists a $v \in V$ such that $|L(v)| > 1$. In the context of tracking an agent traversing the graph, the implication is that *one* of the possible colors $c \in L(v)$ will be emitted when the agent visits node v. It is often useful to reduce such a graph to the equivalent single-colored graph. This can be accomplished by replacing every multi-colored node with multiple nodes, one for each color, then duplicating the appropriate edges. This is illustrated in Figure 1.

An *edge-colored directed graph* is defined as above, but instead associates the mapping to colors with the edges: $L : E \to 2^{\Phi}$. As with the node-colored case, it is possible to have an *edge-multi-colored graph* if there exists an $e \in E$ such that $|L(e)| > 1$. Edge-colored graphs can describe higher-order dependencies (e.g., they represent a system where the color emitted depends on both the current node and the previous node), so it is often useful to reduce an edge-colored graph to an equivalent node-colored graph. This can be accomplished by replacing every node which has incident edges of more than one color by multiple nodes, one for each color, then assigning each node the color of its incident edges. This is illustrated in Figure 2.

Therefore, every node-multi-colored graph and every edge-colored graph, whether multi-colored or not, can be reduced to an equivalent node-colored graph for which each node emits only one color. Moreover, the reduction results in modest growth of the graph. Specifically, if there are n nodes in the graph and the multi-colored node or edge with the most colors has C colors, the resulting simply colored node graph will have no more than Cn unicolored nodes.

Consequently, in the remainder of this paper, we consider, without loss of generality, node colored graphs for which each node can emit only one color.

2.2 Observability Classes

Suppose that an agent is traversing a given node-colored graph, yielding a sequence of observed colors $Y_{1:t} = Y_1, \ldots, Y_t$ from which we seek to infer something about the underlying state or node sequence $X_{1:t} = X_1, \ldots, X_t$ that generated those observed colors. A "hypothesis" in the context of such an observability problem is any sequence $X_{1:t} = X_1, \ldots, X_t$ of nodes that can be visited when traversing a directed path that emits the observed colors $Y_{1:t} = Y_1, \ldots, Y_t$.

Applications often distinguish between real-time tracking where we try to find X_t given $Y_{1:t}$ and *a posteriori* reconstruction where we try to infer some part (even all) of the sequence of nodes $X_{1:t}$ given $Y_{1:t}$. In addition, different versions of the problem can make different assumptions about whether the start state X_1 is know or not. In general, there is no guarantee that any of the nodes can be unambiguously identified, even *a posteriori*.

Previous work has identified a number of classes of colored graphs for which guarantees of varying strength can be made, however:

- In a *trackable* graph the number of hypotheses consistent with an observation sequence grows polynomially in the length of the observation sequence [3]. In a graph which is not trackable, the number of hypotheses grows exponentially. It is known that the number of hypotheses can grow either polynomially or exponentially, with no intermediate growth rates possible [3].
- In a *unifilar* graph the current node X_t is unambiguously determined given the previous node X_{t-1} and the current color Y_t [5]. Furthermore, each node emits exactly one color and each color can be emitted by at most one of the starting nodes. (Thus, the start node is determined unambiguously by the initial observed color.) The implication of these constraints is that there is a one-to-one correspondence between color sequences and node sequences.
- In a *partly a posteriori observable* graph it is possible, given a sufficiently long observation sequence, to unambiguously determine the state at at least one point in the past [6].
- In a *partly observable* graph there is an upper bound, K, on how much time there is between opportunities for X_t to be unambiguously determined given the observation sequence $Y_{t_0:t}$, where $t_0 \leq t < t_0 + K$ [6].
- In an *observable* graph, the node X_t can be unambiguously determined given the observation sequence $Y_{1:t}$, provided $t > T$ (where T is a deterministic burn-in period determined by the structure of the graph) [6].

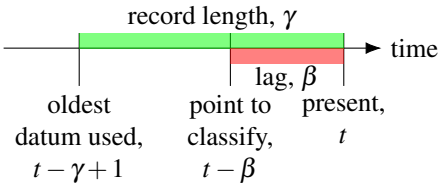


Fig. 3: Illustration of (α, β, γ)-currency. The record length γ may be set either by the time at which observations started (in which case it grows at each time step) or by the finite number of observations stored in the tracking system's memory (in which case it is constant). The oldest datum used is at $t - \gamma + 1$ because we are considering the discrete-time case and the number of observations is γ.

While the previous work characterized these classes using a variety of approaches and definitions [3], [6], [5], Section 4 shows that they can all be expressed in a common framework.

2.3 Currency of Estimates

Previous work on indexing systems (e.g., search engines) has characterized the quality of estimates in terms of (α, β)-currency [9], [10], [11]. Specifically, a quantity which was previously observed at time t_0 is said to be β-current at time t if it has not changed between time t_0 and time $t - \beta$. The quantity is said to be (α, β)-current if it is β-current with probability α.

Appropriately interpreted for the new domain, (α, β)-currency provides a useful metric for characterizing the properties of the various observability classes. Specifically, when tracking an agent traversing a colored graph, our estimate is said to be (α, β)-current at time t if we can correctly identify the node $X_{t-\beta}$ with probability α. Note that this definition does not make reference to the "time of last observation," t_0, because there are no times at which the node is observed directly. Instead, what matters is how long we have been observing the color sequence: the record length, γ. Therefore, we say that an estimate is (α, β, γ)-current if we can correctly identify the node $X_{t-\beta}$ with probability α using the observed colors $Y_{t-\gamma+1}, \ldots, Y_t$. This is illustrated in Figure 3. As an example, an observable graph is $(1, 0, T + 1)$-current. Because of the coloring constraint on the set of starting nodes, a unifilar graph is $(1, 0, 1)$-current. Other classes are more complicated; attainable values of α, β, and γ depend on the specific structure of the graph.

3 Colored Graph Pathologies

3.1 Description of the Pathologies

Jungers and Blondel present polynomial-time algorithms to check whether a graph is observable or partly *a posteriori* observable by checking for the following properties [6]:

1) Presence/absence of nodes which have out-neighbors of the same color.
2) Presence/absence of separated cycles having the same sequence of colors. Two cycles $\pi_1, \pi_2 : \mathbb{Z} \to V$ indexed by i and permitting the same sequence of colors (i.e., $L(\pi_1(i)) \cap L(\pi_2(i)) \neq \emptyset \ \forall i$) are said to be *separated* if $\pi_1(i) \neq \pi_2(i)$ for all steps i. (It is possible, however, to have $\pi_1(i) = \pi_2(j)$ for $i \neq j$; see Figure 4.)

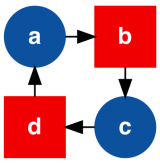


Fig. 4: Graph with *separated* cycles which share nodes. The cycles $\pi_1 = (a, b, c, d, a, \ldots)$ and $\pi_2 = (c, d, a, b, c, \ldots)$ have the same sequence of colors and involve the same nodes, but there is no i such that $\pi_1(i) = \pi_2(i)$.

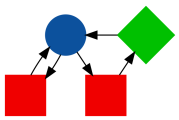
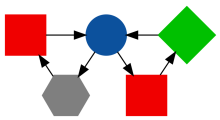
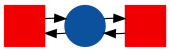
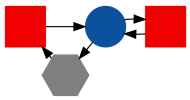
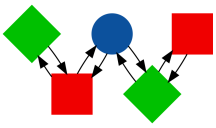
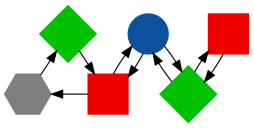
Note, however, that same-colored out-neighbors (item 1) can manifest in one of two ways, depending on whether or not they form two or more *intersecting* cycles having the same sequence of colors. (Two cycles $\pi_1(i), \pi_2(i)$ having the same sequence of colors are said to be *intersecting* if there is at least one i such that $\pi_1(i) = \pi_2(i)$.) This suggests that the three pathologies listed in Table 1 are useful for characterizing the various observability classes. In fact, in Section 4 we show that these three pathologies are sufficient to describe observable, partly *a posteriori* observable, trackable, and a looser class of unifilar graphs (i.e., without the constraint on the set of starting nodes). Partly observable graphs do not fit quite as cleanly into this framework as the others, but some cases can be characterized by the absence of a specific type of same-colored out-neighbor described in Section 4.5.

3.2 Effects of the Pathologies

In loose terms, the effects of each pathology are:

- *Same-colored out-neighbors* cause tracking to be lost once they are encountered. But, it will often be possible to reconstruct which branch was taken *a posteriori*. For example, in the top row of Table 1, once ■ is seen, we do not know which node the agent is at. But, the next observation will either be ● (in which case we know the previous step took the left-hand branch) or ◆ (in which case we know the previous step took the right-hand branch). Therefore, the net effect on (α, β, γ)-currency is to increase the lag β necessary to obtain a given accuracy α. Provided that $\gamma > \beta$, the record length γ will have no effect on the ability to determine which branch was taken.
- *Intersecting cycles with the same coloring* is a special case of same-colored out-neighbors which causes tracking to be lost in a way which can prevent even *a posteriori* reconstruction of the visited nodes. For example, in the middle row of Table 1, the observations will always be an alternating sequence of ■ and ●, but it will never be possible to determine which of the ■ nodes was visited. Therefore, the net effect on (α, β, γ)-currency is to decrease the accuracy α which can be obtained for any lag β or record length γ.
- *Separated cycles with the same coloring* increase the "burn-in" time for which colors must be observed

TABLE 1: Colored Graph Pathologies, Examples, Effects, and Mitigations

Name	Example	Effect	Mitigation
Same-colored out-neighbors		Lose track, may be able to reconstruct *a posteriori*, increases β for given α	
Intersecting cycles with same coloring		Lose track, may never be able to reconstruct, decreases maximum attainable α	
Separated cycles with same coloring		Increases "burn-in" time before nodes can be reconstructed unambiguously, increases required γ for a given α	

before a node (past or present) can be identified unambiguously. For example, consider a sequence of observations from the graph in the bottom row of Table 1 consisting of alternating ◆ and ■. Until ● is observed, it is not possible to know which side of the graph the agent is on. But, as soon as one of the sequences (■, ●) or (◆, ●) is observed, we know not only where the agent is, but where it was at all previous times.

When there is no risk of confusion, the words "with the same coloring" will be omitted when referring to intersecting and separated cycles.

3.3 Mitigating the Pathologies

Enumerating the pathologies is useful not just to understand their implications, but also to create ways of mitigating their deleterious effects on tracking performance. A simple way to modify a colored graph's observability class without significantly changing the functionality of the underlying system is to add uniquely-colored but otherwise non-functional "indicator nodes" at strategic locations. These are indicated by the grey hexagons (⬣) in the last column of Table 1. This approach was demonstrated experimentally in [1], and further simulated examples are given in Section 5.

The basic idea is to insert an indicator node either just before a same-colored out-neighbor or in a cycle with the same coloring in order to remove the pathology. Because the addition of an indicator node causes a delay in the traversal of the graph, it is desirable to put the indicator nodes in parts of the graph which are less-frequently visited. In an HMM, one can determine the long-run frequency of each transition in order to decide where to place indicator nodes.

In principle it is also possible to change the observability class through the deletion of nodes and/or edges, instead of the insertion of nodes into existing edges discussed above. This could correspond to restructuring the system to avoid certain ambiguous behaviors. But, because this will clearly affect the functionality of the system more than the addition of indicator nodes, we do not consider these cases any further here.

4 Pathology-Based Taxonomy of Colored Graph Observability Classes

A taxonomy of colored graph classes based on the presence/absence of the graph pathologies is given in Figure 5. Simple example graphs from each region of the Venn diagram are given in Figure 6. The full reasoning for each class/region is given in the following subsections.

4.1 Region I: General Colored Graphs

The outer part of the Venn diagram is the universe of all possible colored graphs, which may have all of the pathologies represented, and for which no performance guarantees can be made.

4.2 Region II: Partly *a Posteriori* Observable

A graph is partly *a posteriori* observable if there are no separated cycles with the same color sequence [6].

To check if a colored graph $G = (V, E, L, \Phi)$ possesses this property, construct the auxiliary graph G^2 whose nodes are of the form (v_1, v_2), where $v_1, v_2 \in V$ and $v_1 \neq v_2$. The auxiliary graph contains an edge $((v_1, v_2), (v_1', v_2'))$ if $(v_1, v_1') \in E$, $(v_2, v_2') \in E$, and v_1' and v_2' have the same color (i.e., $L(v_1') \cap L(v_2') \neq \emptyset$). If G^2 is acyclic, then G contains no separated cycles with the same color sequence [6].

4.3 Region III: Trackable

Crespi et al. characterize trackability by considering the node sequences consistent with all possible color sequences [3]. A graph is trackable if and only if, for each possible color sequence, there is at most one path from each node v at time t_1 back to itself at time t_2 which is consistent with the color sequence (this is the "unique path property"). But, if two different paths begin and end at the same node and have the same color sequence, then they form a pair of intersecting cycles with the same coloring. Therefore, the absence of such cycles is a necessary and sufficient condition for a graph to be trackable.

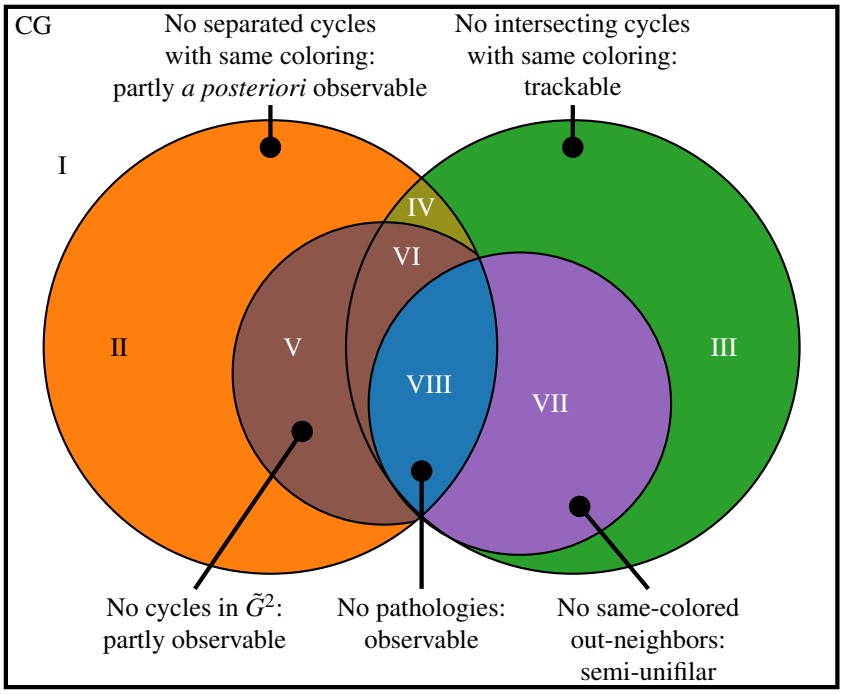


Fig. 5: Venn diagram depicting the taxonomy of colored graphs. Roman numerals in each portion of the diagram are used to refer to the different portions throughout the text. The "CG" in the upper left (region I) stands for "colored graph" – the universe of all possible colored graphs.

4.4 Region IV: Partly *a Posteriori* Observable and Trackable

As shown by the example in Figure 6d, it is possible for a graph to lack both separated and intersecting cycles with the same coloring but to not belong to any of the more restrictive classes. Namely, the absence of both separated and intersecting cycles with the same color sequence is not a sufficient condition for a graph to be partly observable. (Nor is it a necessary condition, see Figure 6e.)

4.5 Regions V and VI: Partly Observable

Partial observability is characterized by another auxiliary graph, $\tilde{G}^2$ [6]. To construct $\tilde{G}^2$, add the edge $((v_1, v_2), (v_1', v_2'))$ to G^2 if the following three conditions are met:

1) G has edge (v_1, v_1') or (v_2, v_1')
2) G has edge (v_1, v_2') or (v_2, v_2')
3) v_1' and v_2' have the same color (i.e., $L(v_1') \cap L(v_2') \neq \emptyset$)

A graph is partly observable if $\tilde{G}^2$ is acyclic. Because $\tilde{G}^2$ is a supergraph of G^2, it has at least as many cycles as G^2 and hence partly observable is a subset of partly *a posteriori* observable.

Graphs which are partly *a posteriori* observable but not partly observable (i.e., G^2 is acyclic but $\tilde{G}^2$ is not) appear to be characterized by a specific class of same-colored out-neighbors similar to the example in Figure 6d: there is a cycle connected to a path which permits the same sequence of colors as the cycle, but the path ends at a different node and hence does not form an intersecting cycle. The net effect of this configuration is to permit color sequences of arbitrary length with ambiguous endpoints, thereby violating the conditions for partial observability. Specifically, in Figure 6d, we cannot know in real-time when a sequence of the form (●, ■, ●, ■, ...) has transitioned from the left-hand branch to the right-hand branch. But, once we see the sequence (●, ■) twice in a row, we know that the *previous* two nodes were in the left-hand branch, but remain uncertain about which branch the agent is currently on. Therefore, the net effect of this pathology is simply to increase the lag β for correct identification of nodes. Because the effect on (α, β, γ)-currency is identical to the more general class of same-colored out-neighbors which do not form intersecting cycles, we have chosen to not include this as a specific pathology in Table 1, but simply include partly observable as a subset in our taxonomy.

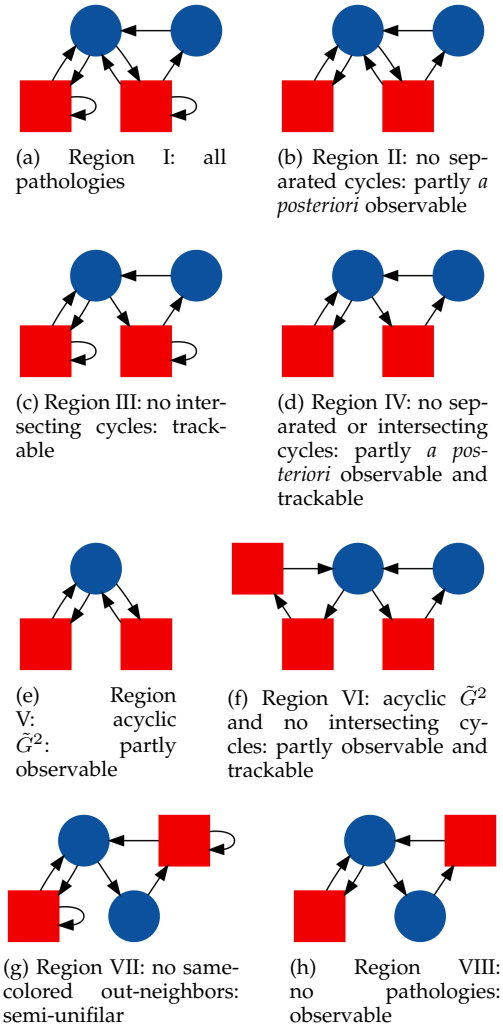
As noted in [6], partly observable graphs may or may not be trackable. We have designated these cases regions V and VI, respectively.

4.6 Region VII: Semi-Unifilar

A graph $G = (V, E, L, \Phi)$ is unifilar if the following three conditions are met [5]:

1) Each node $v \in V$ emits exactly one color: $|L(v)| = 1$.
2) For each node $v \in V$ and each color $c \in \Phi$, there is at most one out-neighbor of v which can emit c.
3) For each color $c \in \Phi$, the agent can start at at most one node which emits c.

Condition 1 is not necessary for the favorable tracking properties of unifilar graphs described in Section 2.2. (It is used in [5] to establish bounds on the rate of growth of the set of possible color sequences from weak models.)



(a) Region I: all pathologies

(b) Region II: no separated cycles: partly *a posteriori* observable

(c) Region III: no intersecting cycles: trackable

(d) Region IV: no separated or intersecting cycles: partly *a posteriori* observable and trackable

(e) Region V: acyclic $\tilde{G}^2$: partly observable

(f) Region VI: acyclic $\tilde{G}^2$ and no intersecting cycles: partly observable and trackable

(g) Region VII: no same-colored out-neighbors: semi-unifilar

(h) Region VIII: no pathologies: observable

Fig. 6: Example graphs for each region of Figure 5.

Condition 2 simply states that there are no same-colored out-neighbors. Our taxonomy does not consider the set of nodes which the agent is permitted to start at, so we do not consider condition 3. Therefore, we call a graph *semi-unifilar* if it satisfies condition 2 (no same-colored out-neighbors).

Semi-unifilar graphs have the property that, once a node is identified unambiguously, all nodes from then on will also be identified unambiguously. But, semi-unifilar graphs do not have any guarantees that you will be able to perform this initial localization. For example, the graph in Figure 4 is semi-unifilar but, because of its symmetry, it will never be possible to unambiguously identify any nodes without additional information beyond the observed color sequence.

Because intersecting cycles with the same coloring are a specific case of same-colored out-neighbors, semi-unifilar is clearly a subset of trackable. Contradicting the assertion that unifilar graphs are a particular case of observable graphs in Section 1 of [6], we note that unifilar (and semi-unifilar) graphs can have separated cycles with the same coloring, and are therefore in fact a superset of observable graphs. An example of a semi-unifilar graph which is not observable is given in Figure 6g.

4.7 Region VIII: Observable

A graph is observable if it lacks both separated cycles with the same coloring and same-colored out-neighbors [6]. In other words, observable graphs are pathology-free.

5 Numerical Experiments: Changing Graph Class With Indicator Nodes

In order to illustrate the effects of the pathologies and mitigations, we have conducted a series of numerical experiments using the graph shown in Figure 7a. This graph has all three pathologies present, and hence is expected to have poor tracking performance.

5.1 Improving Tracking Performance: Trackable and Semi-Unifilar

The base case shown in Figure 7a has a pair of intersecting cycles of the form (●, ■, ■, ■, ■). These will prevent reconstruction of which nodes were visited: every time the sequence (●, ■) is observed, the size of the hypothesis set doubles, consistent with the exponential growth expected for an untrackable graph.

To illustrate this, we simulated 10 000 draws of 50 steps each from an HMM defined by the graph shown in Figure 7a. The probabilities of transitions out of each node were set to be equal, and the nodes were taken to be single-colored. To capture the steady-state behavior, we set the initial state distribution of the HMM to be equal to the equilibrium distribution. We then used the Viterbi algorithm to reconstruct the node sequence from the color sequence with various lags β and record lengths γ.

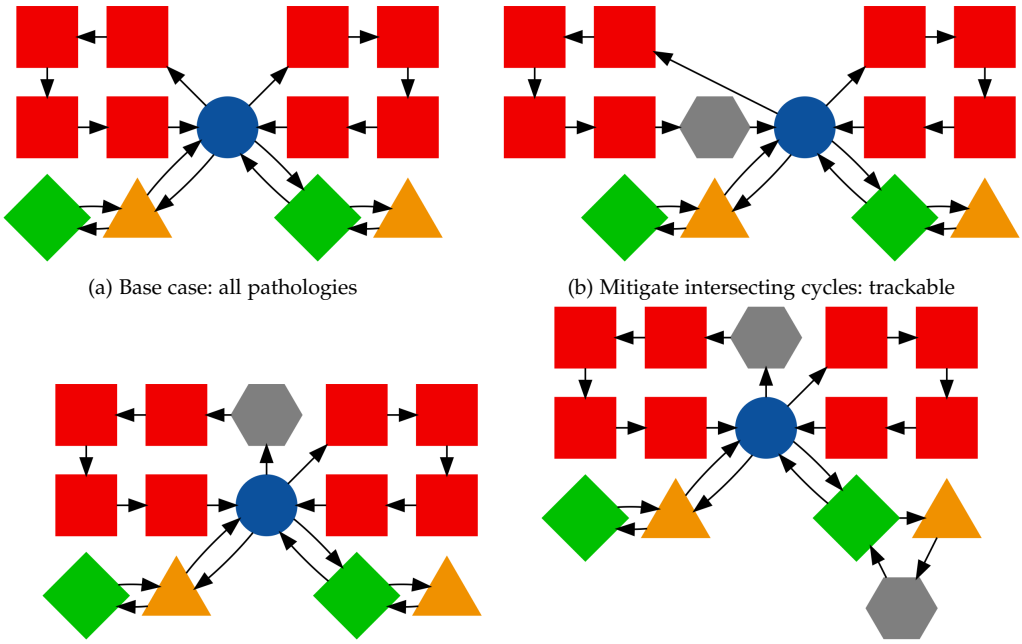
The accuracy $\alpha = P(\hat{X}_{t-\beta} = X_{t-\beta})$ (where $\hat{X}_t$ and X_t are the predicted and true nodes at time t, respectively) is shown in Figure 8a. The median accuracy for the base graph is 78%, and drops to 50% for very short record lengths. The steady-state (i.e., high-γ) behavior is shown in Figure 9. The tracking accuracy is equally poor at all lags: longer record lengths and/or lags do not help reduce the effects of intersecting cycles with the same coloring.

Now consider the addition of an indicator node to mitigate the intersecting cycles, as shown in Figure 7b. The indicator node was placed just *before* the central node to preserve the same-colored out-neighbors: the modified graph is trackable, but not semi-unifilar. Tracking accuracy for the modified graph (Figure 8b) is 100% for $\beta \geq 4$, but the same-colored out-neighbors cause it to drop to around 79% for $\beta = 0$.

Next, consider moving the indicator node so it also mitigates the same-colored out-neighbors, as shown in Figure 7c. Tracking accuracy for the modified graph (Figure 8c) is 100% for $\beta < \gamma - 4$, but the accuracy drops for the first four time steps recorded because there is no way to determine which branch an initial sequence of ■ came from.

5.2 Reducing Burn-In Time: Observable

The graphs considered above contain a pair of separated cycles of the form (◆, ▲) which are expected to increase the burn-in time before states can be identified unambiguously. The stationary distribution for the examples above has most



(a) Base case: all pathologies

(b) Mitigate intersecting cycles: trackable

(c) Mitigate same-colored out-neighbors: semi-unifilar

(d) Mitigate separated cycles: observable

Fig. 7: Successive mitigations of colored graph pathologies.

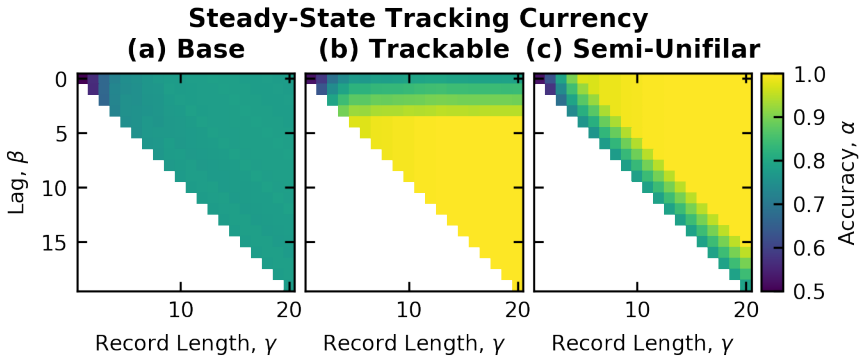


Fig. 8: Accuracy $\alpha = P(\hat{X}_{t-\beta} = X_{t-\beta})$ as a function of lag β and record length γ for the (a) basic, (b) trackable, and (c) semi-unifilar graphs. The white regions correspond to $\beta \geq \gamma$, for which the behavior is undefined.

of its mass in the eight ■ nodes, which ends up masking this effect. In order to characterize the effect of separated cycles, we generated another set of 10 000 realizations, each with 50 time steps. For this experiment the initial state distribution was set to be uniform over the lower four nodes in Figure 7a (i.e., the ones which form the separated cycles with the same coloring).

We start with the semi-unifilar graph in Figure 7c because the effects of the other pathologies have already been shown above. Tracking accuracy for this case is shown in Figure 10a. The accuracy starts out poor for all lags β, but gradually improves as the record length γ increases. The lack of dependence on the lag β is expected: the graph is semi-unifilar, so once ● is observed for the first time unambiguous real-time tracking is guaranteed. Furthermore, because the observations cannot start at any of the ■ nodes, all of the initial states are reconstructed once one of the two sequences (◆, ●) or (▲, ●) is observed.

The real-time ($\beta = 0$) tracking performance is shown in Figure 11. The tracking accuracy for the semi-unifilar graph asymptotically approaches 100%. The time constant of this approach can be controlled by varying the probabilities of transitioning to the ● node; the same equal probabilities used above were used here.

Now consider the addition of an indicator node into the lower right (◆, ▲) cycle, as shown in Figure 7d. The modified graph is observable. Consistent with this, 100% tracking accuracy is obtained after a fixed burn-in period of two time steps (see Figure 10b). For the equal transition probabilities used here, this corresponds to approximately 5× faster burn-in compared to the semi-unifilar case.

6 Conclusions and Future Work

This paper has presented a new pathology-based taxonomy of colored graph observability classes which unifies the

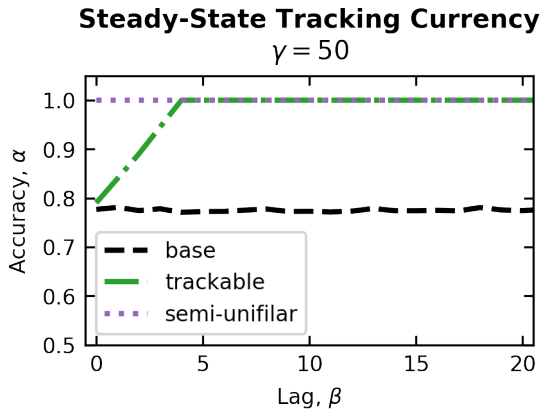


Fig. 9: Steady-state (record length $\gamma = 50$) tracking accuracy for the basic (black dashed), trackable (green dot-dash) and semi-unifilar (purple dotted) graphs. This is essentially a vertical slice of the data shown in Figure 8. The base graph starts around 77% accuracy and never improves even for very long lags. The trackable graph has 100% accuracy for lags $\beta \geq 4$, but drops to the base 77% level for shorter lags because of the same-colored out-neighbors of the form (●, ■). The semi-unifilar graph has 100% accuracy for the lags shown, including real-time tracking ($\beta = 0$).

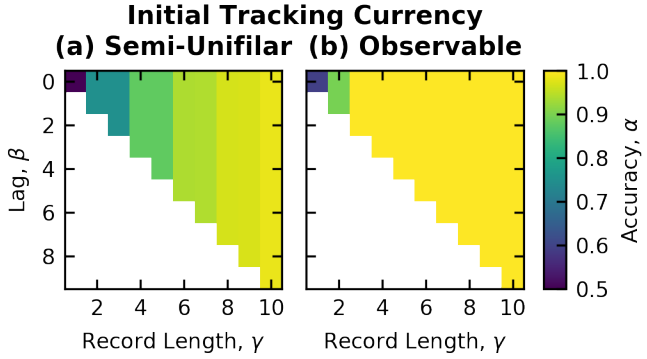


Fig. 10: Accuracy $\alpha = P(\hat{X}_{t-\beta} = X_{t-\beta})$ as a function of lag β and record length γ for the (a) semi-unifilar and (b) observable graphs when observations start while the agent is at either a ◆ or ▲ node. The white regions correspond to $\beta \geq \gamma$, for which the behavior is undefined.

results of [5], [3], [6] into a common framework. The three colored graph pathologies identified provide an intuitive picture of the differences between the various observability classes, and the expanded concept of (α, β, γ)-currency provides a principled way of reasoning about the effects of the various pathologies. Numerical experiments have shown the ability to change the observability class of a graph through the addition of indicator nodes, providing a more complete view of this topic than the initial experimental results in [1]. The formulation of the taxonomy has intentionally avoided questions of transition/emission probabilities and initial state distributions so that the results are as general as possible, and hence can be applied to any situation where a hidden state sequence is to be reconstructed from noisy/potentially ambiguous observations.

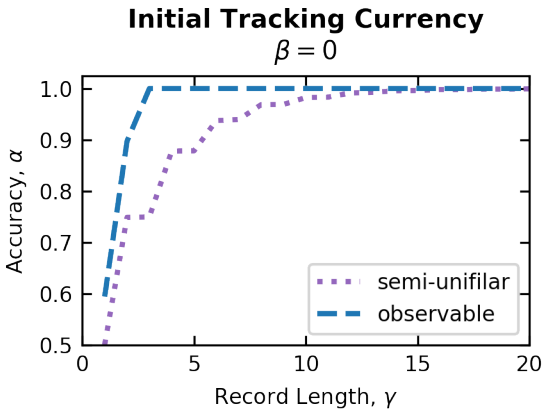


Fig. 11: Real-time (lag $\beta = 0$) tracking accuracy for the semi-unifilar (dotted purple) and observable (dashed blue) graphs. This is essentially a horizontal slice of the data shown in Figure 10. The semi-unifilar graph asymptotically approaches 100% accuracy, while the observable graph obtains 100% accuracy for $\gamma > 2$.

In terms of possible future research directions, consider the following observations. There is a large well-known literature on problems of graph colorability. Loosely speaking, those problems ask how many colors are required to color nodes in a graph so that no two adjacent nodes have the same color [12]. The classic problem in this area is of course the four color problem for planar graphs.

The "chromatic number" of a graph is the smallest number of colors that make the graph colorable in the above sense. Determining the chromatic number is known to be NP-complete, therefore the most efficient algorithms currently known are of exponential complexity [13].

By analogy, we can imagine defining a "p-observability number" of a directed graph as being the smallest number of colors required to make the graph have the p-observability property, where such a p-observability property is one of the observability properties discussed in this paper. Moreover, just as the chromatic polynomial, $P_G(k)$, is a polynomial with the property that the number of legal classical colorings of graph G using k colors is precisely $P_G(k)$ (so that the chromatic number of G is the smallest k for which $P_G(k) > 0$), we imagine there might be a "p-observability polynomial" with the similar property for the "p-observability" of a graph. Such a "p-observability polynomial" would inform us about how difficult or easy realizing the "p-observability property" would be for a specific graph.

Acknowledgements

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)

The authors wish to thank the Defense Advanced Research Projects Agency (DARPA) for funding this work as part of the LADS program under contract number FA8650-16-C-7622. In particular, we thank Dr. Angelos Keromytis, the DARPA program manager of LADS, for his encouragement and support throughout the program.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

REFERENCES

[1] M. Chilenski, G. Cybenko, I. Dekine, P. Kumar, and G. Raz, "Control flow graph modifications for improved RF-based processor tracking performance," in *Cyber Sensing 2018*, ser. Proc. SPIE, vol. 10630, 2018, p. 106300I.

[2] G. Cybenko and G. M. Raz, "Large-scale analogue measurements and analysis for cyber-security," *Data Science For Cyber-security*, vol. 3, p. 227, 2018.

[3] V. Crespi, G. Cybenko, and G. Jiang, "The theory of trackability with applications to sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 3, p. 16, 2008.

[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] Y. Sheng and G. V. Cybenko, "Distance measures for nonparametric weak process models," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*. IEEE, 2005.

[6] R. M. Jungers and V. D. Blondel, "Observable graphs," *Discrete Applied Mathematics*, vol. 159, no. 10, pp. 981–989, 2011.

[7] M. Kearns and L. Valiant, "Cryptographic limitations on learning boolean formulae and finite automata," *Journal of the ACM (JACM)*, vol. 41, no. 1, pp. 67–95, 1994.

[8] G. Jiang, "Weak process models for robust process detection," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, ser. Proc. SPIE, vol. 5403, 2004.

[9] B. E. Brewington and G. Cybenko, "How dynamic is the Web?" *Computer Networks*, vol. 33, no. 1-6, pp. 257–276, 2000.

[10] ——, "Keeping up with the changing web," *Computer*, vol. 33, no. 5, pp. 52–58, 2000.

[11] B. E. Brewington, "Observation of changing information sources," Ph.D. dissertation, Dartmouth College, June 2000.

[12] T. R. Jensen and B. Toft, *Graph Coloring Problems*. John Wiley & Sons, 2011, vol. 39.

[13] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Computer Science*, vol. 1, no. 3, pp. 237–267, 1976.

Mark Chilenski received the BS degree in aeronautical and astronautical engineering from the University of Washington in 2010 and the PhD degree in nuclear science and engineering from the Massachusetts Institute of Technology in 2016. He is a senior scientist at Systems & Technology Research LLC. His research interests include machine learning, Bayesian inference, and cybersecurity.

George Cybenko received his B.Sc. and Ph.D. degrees in Mathematics from the University of Toronto and Princeton. He is currently the Dorothy and Walter Gramm Professor of Engineering at Dartmouth. His research interests include cyber security, advanced machine learning algorithms and information deception.

Isaac Dekine received the BS and MS degrees in electrical and computer engineering from Carnegie Mellon University in 2006. He is a senior engineer at Systems & Technology Research LLC. His research interests include RF system design, signal processing, and cybersecurity.

Piyush Kumar received the Master of Science degree in Physics from the Indian Institute of Technology Kharagpur in 2001, MS degree in Physics from the University of Chicago in 2004 and the PhD degree in Physics from the University of Michigan Ann Arbor in 2007. He is a lead scientist at Systems & Technology Research LLC. His research interests include applications of probabilistic methods to problems in physics and engineering, machine learning, and graph theory.

Gil Raz received a bachelor's degree in electrical engineering from the Technion – Israel Institute of Technology in 1988 and a PhD in electrical engineering (minor in mathematics) from the University of Wisconsin – Madison in 1998. He is a chief scientist at Systems & Technology Research LLC. His research interests include applied mathematics and statistics for solving problems in multiple application areas.