CRUISABLE GRAPHS

RAPHAËL JUNGERS AND VINCENT D. BLONDEL

Abstract. Motivated by practical questions for sensor networks, we introduce in this paper the notion of cruisable graphs and study some of their properties. An edge-colored directed graph is cruisable if an agent that moves along its directed edges is able to determine his position after a sufficiently long observation of the traversed edge colors. We provide a complete characterization of cruisable graphs and show how cruisability can be checked in polynomial time. We then consider the question of assigning colors to the edges of a directed graph so as to make it cruisable. In particular, we prove that finding the minimal number of colors needed to make a graph cruisable is NP-complete. Finally, we describe a number of related questions and open problems.

1. Introduction

Consider an agent moving from node to node in a directed graph whose edges are colored. The agent knows perfectly the colored graph but doesn't know his position in the graph. From the sequence of colors he observes he wants to deduce his position. We say that an edge-colored directed graph[1] is *cruisable* if there is some observation time length after which, whatever the color sequence observed, the agent is able to determine his position in the graph and is able to do so for all subsequent times. Of course, if all edges are of different colors, or if edges with different end-nodes are of different colors, then an agent is always able to determine his position after just one observation. So the interesting situation is when there are fewer colors than there are nodes.

Consider for instance the two graphs on Figure 1 (the difference between the two graphs is the direction of the edge between the nodes 1 and 3). The edges are colored with two "colors": solid (S) and dashed (D). We claim that the graph (a) is cruisable but that (b) is not. In graph (a), if the observed color sequence is DDS then the agent must be at node 1, if the sequence is SDD he must be at node 3, and similarly for all other observation sequences of length three. For this graph it follows from the results presented in this paper that the observation of color sequences of length three always suffices to determine the exact position of the agent in the graph. Consider now the graph (b) and assume that the observed sequence is SSDSS; after these observations are made, the agent may either be at node 1, or at node 3. There are two paths that produce the color sequence SSDSS and these paths have different end-nodes. Sequences of increasing length and with the same property can be constructed and so graph (b) is not cruisable.

There are of course very natural conditions for a graph to be cruisable. A first condition is that no two edges of identical colors may leave the same node (strictly speaking this condition applies only to

Date: July 2006.

The research reported here was performed while the authors were at MIT, Cambridge. It was partially supported by Communauté francaise de Belgique - Actions de Recherche Concertées, by the HYCON Network of Excellence (contract number FP6-IST-511368), by the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office, and by the DoD AFOSR URI for "Architectures for Secure and Robust Distributed Infrastructures", F49620-01-1-0365 (led by Stanford University). The scientific responsibility rests with its authors. Raphaël Jungers is a FNRS fellow (Belgian Fund for Scientific Research).

[1]The property of being cruisable is a property of directed graphs that have their edges colored. For simplicity, we will talk in the sequel about cruisable *graphs* rather than cruisable *edge-colored directed graphs*.

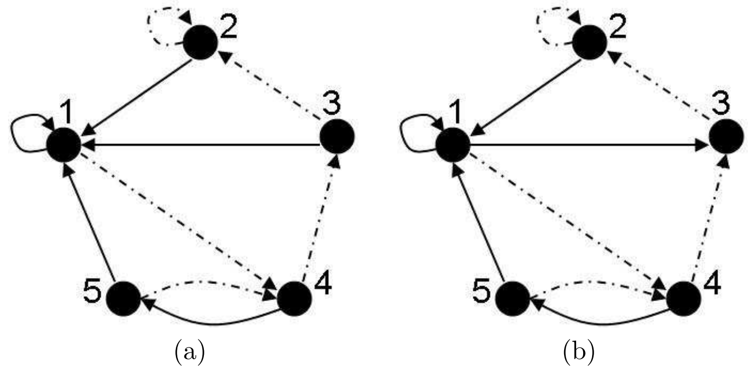
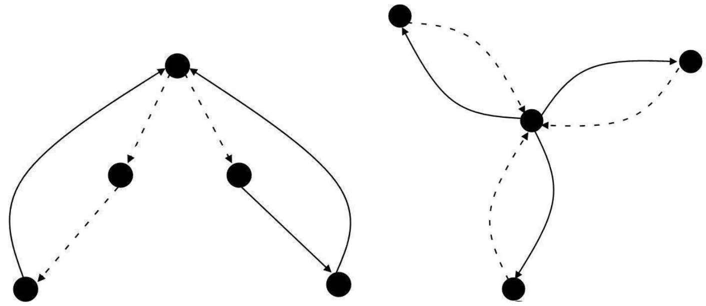


FIGURE 1. An agent is moving along directed edges. Edges are "colored" solid (S) or dashed (D). The graph (a) is cruisable because the observation of the last three colors suffices to determine the position of the agent in the graph. The graph (b) is not cruisable; the sequence SSDSSD. . .SSD can be obtained from a path ending at node 2 or at node 4.

the nodes that can be reached by paths of arbitrary long length, see below). Indeed, if a node has two outgoing edges with identical colors then an agent leaving that node and observing that color wouldn't be able to determine his next position in the graph. So this is clearly a necessary condition. Another condition is that the graph may not have two distinct cycles with identical color sequences, as in graph (b) of Figure 1 (cycles $2-1-3-2$ and $4-5-1-4$). If such cycles are present in the graph, then an agent observing that repeated particular color sequence is not able to determine if he is moving on one or the other cycle. So, these two conditions are clearly necessary conditions for a graph to be cruisable. In our first theorem we prove that these combined conditions are in fact also sufficient.

In a cruisable graph an agent is able to determine his position in the graph for all observations of length *larger* than some T. Some graphs are not cruisable in this way but in a weaker sense. Consider for example the graph (a) of Figure 2. The graph is made of two colored cycles of length three that intersect at a common (top) node. There are two dashed edges that leave the top node and so, whenever the agent passes through that node his next observation will be D and his next position in the graph will be uncertain. This graph is therefore not cruisable. We will nevertheless say that it is *partly cruisable* because, even though the agent is not able to determine his position at all time, he is able to do so infinitely often. Indeed, after observing the sequence SD in this graph (and this sequence occurs in every observation sequence of length 3), the next observation is either S or D. If it is S, the agent is at the bottom right node; if it is D, he is at the bottom left node. In both cases the next observed color is S and the agent is then at the top node. So the agent is able to determine his position infinitely often (more specifically, 2/3 of the time). In a cruisable graph, an agent is able to determine his positions at all times beyond a certain limit; in a partly cruisable graph the agent is able to determine his position infinitely often (for formal definitions, see below). Notice that in the graph (a) of Figure 2, the agent is always able to *a posteriori* reconstruct its entire trajectory, except maybe for its last position. This is however not the case of all partly cruisable graphs. Consider for example the star graph (b) of Figure 2. The only possible color sequences obtained in this graph are SDSDS. . . or DSDSDS. . . . When the last observed color is D, the agent is at the central node. When the last observed color is S, the agent only knows that he is at one of the extreme nodes. Hence this graph is partly cruisable but in this case, contrary to what we have with graph (a), the trajectory cannot be reconstructed a posteriori.

In this paper, we prove a number of results related to cruisable and partly cruisable graphs. We first prove that the conditions described above for cruisability are indeed sufficient and we derive analogous



(a) (b)

Figure 2. Both graphs are partly cruisable but none of them is cruisable. In graph (a), complete trajectories can always be reconstructed a posteriori; which is not possible in graph (b).

necessary and sufficient conditions for a graph to be partly cruisable. From these results it follows that in a cruisable graph an agent can determine his position in the graph after an observation of length *at most* n^2 where n is the number of nodes. This quadratic increase cannot be avoided: we provide a family of graphs for which $\Theta(n^2)$ observations are necessary. Based on some of these properties we also provide a polynomial-time algorithm for checking whether or not a graph is (partly) cruisable. We then consider the question of assigning colors to the edges of a directed graph so as to make it cruisable. We prove that the problem of finding the minimal number of colors that are necessary to make a graph cruisable is a problem that is NP-hard.

The concept of cruisable graph is related to a number of concepts in graph and automata theory, networks and Markov models. Perhaps the most natural and direct connection is with the notion of synchronizing automata. In a cruisable graph we ask the agent to be able to determine his position after some finite time and require this to be possible *whatever path* he chooses in the graph. On the other hand, an automata is said to be *synchronizing* if there is *some choice* of color sequence for which the agent is able to determine its position in the graph after having chosen a path that produces that color sequence (these color sequences are then said to be "synchronizing"). For this definition to be unambiguous, the notion of synchronizing automata only applies to automata for which there is exactly one outgoing edge of every color leaving from every node. Thus, in cruisable graphs, the agent strolls in the graphs and wonders where he is, whereas in synchronizing automata the agent chooses a particular sequence of colors so as to be able to determine where he is. Both Cruisable and synchronizable graphs can be characterized in terms of paths in their power graph. For an automata to be synchronizing, there must be a path from the "complete set" node to some "singleton" node in the corresponding power automata; for a graph to be cruisable, all paths of the power graph need to lead to "singleton" nodes.

Synchronizing automata have been the subject of intense research for more than forty years and have led to a number of interesting results and conjectures. A celebrated conjecture due to Černý states that if an automaton is synchronizing, then it admits a synchronizing color sequence of length less than $(n-1)^2$, where n is the number of nodes [2]. Weaker bounds have been obtained and the conjecture has been proved for some particular cases [3, 5, 8, 10, 13], but the original conjecture is still open. The question of determining if an automata is synchronizing is an *analysis* question. The corresponding *design* question is the so-called *road coloring problem*. In the road coloring problem one is given a directed graph whose nodes all have the same outgoing degree Δ and one is asked to color the edges with Δ colors so as to make the automaton synchronizing. It is easy to see that for this to be possible the graph has to be aperiodic. According to the celebrated – and still unsolved – road coloring conjecture introduced by Adler and Weiss [1], this condition is also sufficient.

One cannot possibly fail to notice the analogy between these questions and those raised in this paper for cruisable graphs. The road coloring problem is in our context the problem of chosing edge colors that make a given graph cruisable and Černý's conjecture is the analog to the (much simpler) result proved in this paper that in a cruisable graph the position in the graph can be determined after at most n^2 observation.

Unlike what is done in the literature on synchronizing automata, we do not restrict nodes of cruisable graphs to have identical out-degrees. From this point of view our approach is closer to questions raised in the context of sensor networks, in which agents are moving in a network, and release only limited information on their positions. It is in this context that Crespi et al. [4] have recently introduced the notion of trackable networks. An edge-colored directed graph is said to be *trackable* if the maximum number of trajectories compatible with some color sequence of length k grows subexponentially with k. So, in the context of trackability, one cares about the number of trajectories, but not about the position of the agent in the graph. It has recently been proved that the problem of determining if a network is trackable is a problem that can be solved in polynomial time [9] (see also [4]). It is clear that if a network is cruisable then it is trackable; but the converse is not true in general. For example the graph on Figure 2 (a) is trackable but not cruisable. Notice also that, as exemplified by the graph on Figure 2 (b) it is not necessary to be trackable for being partly cruisable.

Finally, the results presented in this paper can also be interpreted in the context of *Hidden Markov Processes*, or HMP [6, 14]. More precisely, the graphs we consider can be seen as *finite alphabet HMPs* (also called *aggregated Markov processes*), except for the fact that no values are given for the transition probabilities. In our context, we allow transitions between states to be possible or not, but we do not consider probabilities different from 0 or 1. Also, we assign to every transition an allowed set of colors, without considering their probabilities. Our results therefore apply to these particular HMPs. In particular, we provide sufficient conditions for a finite alphabet HMP to allow exact state identifiability for sufficiently long observations, whatever values are given to the different probabilities.

References

[1] R. L. Adler and B. Weiss. Similarity of automorphisms of the torus. *Memoirs of the American Mathematical Society*, 98, 1970.

[2] J. Černý. Poznámka k. homogénnym experimentom s konecnými automatmi. *Matematicko-fysikalny Casopis SAV*, 14:208–215, 1964.

[3] J. Černý, A. Pirická, and B. Rosenauerova. On directable automata. *Kybernetica*, 7:289–298, 1971.

[4] V. Crespi, G. V. Cybenko, and G. Jiang. The Theory of Trackability with Applications to Sensor Networks. Technical Report TR2005-555, Dartmouth College, Computer Science, Hanover, NH, August 2005.

[5] L. Dubuc. Les automates circulaires biaisés verifient la conjecture de Černý. *Information Theory and Applications*, 30:495–505, 1996.

[6] Y. Ephraim and N. Merhav. Hidden markov processes. *IEEE Transactions on Information Theory*, 48(6):1518–1569, 2002.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[8] W. Göhring. Minimal initializing word: A contribution to černý's conjecture. *Journal of Automata, Languages and Combinatorics*, 4:209–226, 1997.

[9] R. Jungers, V. Protasov, and V. D. Blondel. Efficient algorithms for deciding the type of growth of products of integer matrices. 2006. Submitted to publication.

[10] J. Kari. Synchronizing finite automata on eulerian digraphs. *Theoretical Computer Science*, 295(1-3):223–232, 2003.

[11] Z. Kohavi. *Switching and finite automata theory*. McGraw Hill, New York, NY, USA, 1970.

[12] E. F. Moore. Gedanken-experiments on sequential machines. *Annals of mathematics studies*, 34:129153, 1956.

[13] J.-E. Pin. Sur un cas particulier de la conjecture de černý. In *5th ICALP*, number 62 in LNCS, pages 345–352, Berlin, 1978. Springer.

[14] L. R. Rabiner. A tutorial on hidden markov models and selected apllications in speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Kaufmann, San Mateo, CA, 1990.

[15] P. H. Starke. *Abstrakte Automaten*. V.E.B. Deutscher Verlag der Wissenschaften, Berlin, Germany, 1969.
[16] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.
[17] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transaction on Information Theory*, pages 260–269, 1967.

Division of Applied Mathematics, Université catholique de Louvain, 4 avenue Georges Lemaitre, B-1348 Louvain-la-Neuve, Belgium
E-mail address: blondel@inma.ucl.ac.be

Division of Applied Mathematics, Université catholique de Louvain, 4 avenue Georges Lemaitre, B-1348 Louvain-la-Neuve, Belgium
E-mail address: jungers@inma.ucl.ac.be