Journal of Computer and System Sciences 81 (2015) 234–248

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Network verification via routing table queries ☆

Evangelos Bampas [a,1], Davide Bilò [b], Guido Drovandi [c], Luciano Gualà [d], Ralf Klasing [a,*], Guido Proietti [c,e]

[a] LaBRI, CNRS and University of Bordeaux, France
[b] Dipartimento di Teorie e Ricerche dei Sistemi Culturali, University of Sassari, Italy
[c] Istituto di Analisi dei Sistemi ed Informatica, CNR, 00185 Rome, Italy
[d] Dipartimento di Matematica, University of Tor Vergata, Rome, Italy
[e] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, University of L'Aquila, Italy

ARTICLE INFO

Article history:
Received 21 September 2011
Received in revised form 9 April 2014
Accepted 25 April 2014
Available online 20 June 2014

Keywords:
Network verification
Graph/network topology
Computational complexity
Approximation algorithms

ABSTRACT

The network verification problem is that of establishing the accuracy of a high-level description of its physical topology, by making as few measurements as possible on its nodes. This task can be formalized as an optimization problem that, given a graph and a query model specifying the information returned by a query at a node, asks for finding a minimum-size subset of nodes to be queried so as to univocally identify the graph. This problem has been studied with respect to different query models, assuming that a node had some global knowledge about the network. Here, we propose a new query model based on the local knowledge a node instead usually has. Quite naturally, we assume that a query at a given node returns the associated routing table, i.e., a set of entries which provides, for each destination node, a corresponding (set of) first-hop node(s) along an underlying shortest path.

1. Introduction

There is a growing interest in networks which are built and maintained by decentralized processes. In such a setting, it is natural to consider the problem of discovering the map of an unknown (in terms of edges) network, or to verify whether a given map is accurate, i.e., to check whether the edges of the map are exactly those of the underlying network. A common approach to discover or to verify a map is to make some local measurement on a selected subset of nodes that – once collected – can be used to derive information about the whole network (see for instance [6,9]). A measurement on a node is usually costly, so it is natural to try to make as few measurements as possible.

These two tasks – that of *discovering* a map and that of *verifying* a given map – have been formalized as optimization problems and have been studied in several papers. The idea is to model the network as a graph $G = (V, E)$, while

☆ A preliminary version of this work was presented at the 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO'11), June 26–29, 2011, Gdansk, Poland, Lecture Notes in Computer Science, vol. 6796, Springer, 2011, pp. 270–281. This work was partially supported by research projects PRIN-2008C255HN "COGENT" (COmputational and GamE-theoretic aspects of uncoordinated NeTworks) and PRIN-2010N5K7EB "ARS TechnoMedia" (Algorithms for Techno-Mediated Social Networks), funded by the Italian Ministry of Education, University, and Research. Part of this work was done while the second author was visiting LaBRI-Bordeaux.

* Corresponding author.

1 Partially supported by the ANR project DISPLEXITY (ANR-11-BS02-014). This study has been carried out in the frame of the "Investments for the future" Programme IdEx Bordeaux – CPU (ANR-10-IDEX-03-02).

http://dx.doi.org/10.1016/j.jcss.2014.06.003
0022-0000/© 2014 Elsevier Inc. All rights reserved.

a measurement at a given node can be seen as a unit-cost *query* returning some piece of information about G. In the *network discovery* problem, we know V but not E, and so we want to design an online algorithm that selects a minimum-size subset of nodes $Q \subseteq V$ to be queried that allows to precisely map the entire graph, i.e., to settle all the edges and all the non-edges of G. The quality of the algorithm is measured by its competitive ratio, i.e., the ratio between the number of queries made by the algorithm and the minimum number of queries which would be sufficient to discover the graph. On the other hand, the *network verification* problem, which is of interest for our paper, is the off-line version of the problem, and so we are given a graph $G = (V, E)$, and we want to compute a *minimum* number of queries sufficient to discover G (if E was unknown). This latter problem has an interesting application counterpart, since it models the activity of verifying the accuracy of a given map associated with an underlying real network (on which the queries are actually done).

In the literature, two main query models have been studied. In the *all-shortest-paths* query model, a query of a node q returns the subgraph of G consisting of the union of all shortest paths between q and every other node $v \in V$. A weaker notion of query is used in the *all-distances* query model, in which a query to a node q returns all the distances in G from q to every other node $v \in V$. Notice that both models inherently require global knowledge/information about the network, hence a central problem for these query models is whether/how the information can be obtained locally (without preprocessing of the network). In this paper, we propose a query model that uses only *local* knowledge/information about the network. Quite naturally, we assume that a query at a given node q returns the associated *routing table*, namely a set of entries which provides, for each destination node, a corresponding (set of) first-hop node(s) along an underlying shortest path. In the rest of the paper, this will be referred to as the *routing-table query model*.

Previous work. It turns out that the verification problem with the *all-shortest-paths* query model is equivalent to the problem of placing landmarks on a graph [17]. In this problem, we want to place landmarks on a subset of the nodes in such a way that every node is uniquely identified by the distance vector to the landmarks, and the minimum number of landmarks to be placed is called the *metric dimension* of a graph [14]. The problem has been shown to be NP-hard in [8]. An explicit reduction from 3-SAT is given in [17] which also provides an $O(\log n)$-approximation algorithm (n is the number of nodes) and an exact polynomial-time algorithm for trees. Subsequently, in [1], the authors prove that the problem is not $o(\log n)$ approximable, unless $\mathsf{P} = \mathsf{NP}$, showing thus that the algorithm in [17] is the best possible in an asymptotic sense. We finally mention that in [2] the authors studied the related problem of monitoring link failures (i.e., verifying only the graph edges) in a very similar query model.

As far as the *all-distances* query model is concerned, the verification problem has been studied in [1], where the NP-hardness is proved and an algorithm with $O(\log n)$-approximation guarantee is provided. Other results in [1] include exact polynomial-time algorithms for trees, cycles and hypercubes.

Concerning the network discovery problem, this received attention in several papers and in both query models. More precisely, for the *all-distances* query model, in [1] the authors have shown an $\Omega(\log n)$ lower bound on the competitive ratio of any *randomized* algorithm, and an $\Omega(\sqrt{n})$ lower bound on the competitive ratio of any *deterministic* online algorithm. Moreover, they also provided a randomized $O(\sqrt{n \log n})$ competitive algorithm. On the other hand, in the *all-shortest-paths* query model, in [1] the authors have shown a $3 - \epsilon$ lower bound on the competitive ratio of any *deterministic* online algorithm, for any $\epsilon > 0$, and they provided a randomized $O(\sqrt{n \log n})$ competitive algorithm. This was then improved in [20], where the authors provided an $O(\log^2 n)$-competitive Monte Carlo randomized algorithm. Still for the same query model, we finally mention that in [3] the authors studied how to discover several graph properties, while in [7] the authors focused on the *approximate* discovery of Erdös–Rényi random graphs.

Besides the two traditional query models we widely discussed above, it is worth mentioning that the network discovery and verification problems were also analyzed in the *edge-counting* query model, where a query at a set of vertices returns the number of edges in the corresponding induced subgraph. In this model, the information theoretic lower bound for the query complexity of discovering a graph is $\Omega(\frac{m \log n^2/m}{\log m})$, and in [4,18] a polynomial time algorithm was provided with a query complexity matching such a lower bound (which improved a previous result in [5]). Finally, concerning the graph verification, in [19] the authors gave a Monte Carlo randomized algorithm with error ϵ for using $O(\log 1/\epsilon)$ queries.

Our results. Throughout the paper, we focus on the verification problem w.r.t. the routing-table query model. We first show a lower bound of $\Omega(\log \log n)$ on the minimum number of queries needed to verify any graph with n nodes. This is in contrast to the previous two query models for which certain classes of graphs can be verified with a constant number of queries, like paths and cycles. Our proof also implies a lower bound of $\Omega(n)$ on the number of queries needed to verify a path or a cycle. So, one may wonder whether every graph needs a linear number of queries to be verified. We provide a negative answer to this question by exhibiting a class of graphs (of diameter 2) that can be verified with $O(\log n)$ queries. On the other hand, we show that this number of queries is asymptotically optimal, since we prove that for graphs of constant diameter $\Omega(\log n)$ queries are actually needed.

We then analyze the computational complexity of the problem. In this respect, although it remains open for general input graphs to establish whether the problem is in NPO, we are able to provide an $O(\log n)$-approximation algorithm to verify graphs of diameter 2. Moreover, we also show that this bound is asymptotically tight, unless $\mathsf{P} = \mathsf{NP}$. On the positive side, we provide exact linear-time algorithms to verify paths, trees and cycles of even length. In terms of number of queries, these algorithms show that for paths and cycles of even length, the size of the optimal query set is about a half and a third of n, respectively. On the other hand, for trees such a size eventually depends on the tree structure, but we show that it is at least $\lceil \frac{n}{3} \rceil$ (and we also show this is tight). Our result for trees is based on a characterization of a solution that can be used to reduce the problem to that of computing a minimum vertex cover of a certain class of graphs (for which a vertex