Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Lower and upper competitive bounds for online directed graph exploration ☆

Klaus-Tycho Foerster *, Roger Wattenhofer

ETH Zürich, Gloriastrasse 35, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 31 March 2015
Received in revised form 8 November 2015
Accepted 11 November 2015
Available online xxxx

Keywords:
Graph exploration
Online algorithms
Competitive analysis
Directed graphs

ABSTRACT

We study the problem of exploring all nodes of an unknown directed graph. A searcher has to construct a tour that visits all nodes, but only has information about the parts of the graph it already visited. Analogously to the traveling salesman problem, the goal is to minimize the cost of such a tour. In this article, we present upper and lower bounds for the competitive ratio of both the deterministic and the randomized online version of exploring all nodes of directed graphs. Our bounds are sharp or sharp up to a small constant, depending on the specific model. As it turns out, restricting the diameter, the incoming/outgoing degree, or randomly choosing a starting point does not improve lower bounds beyond a small constant factor. Even supplying the searcher in a planar euclidean graph with the nodes' coordinates does not help. Essentially, exploring a directed graph has a multiplicative overhead linear in the number of nodes. Furthermore, if one wants to search for a specific node in unweighted directed graphs, a greedy algorithm with quadratic multiplicative overhead can only be improved by a small constant factor as well.

1. Introduction

The hotel concierge promised that this tourist attraction is easy to find, just a short drive in your car, and she was right. However, how do you now get back to your hotel, in this cursed city full of one-way streets? After finally being back at your hotel, totally exhausted, you have a hunch that one-way streets render navigation more difficult, but is it true?!

In this article we quantitatively analyze navigation problems in unknown directed graphs from a worst-case perspective. We present a whole flurry of tight upper and lower bounds, showing that directed graphs exhibit a penalty in the order of the number of nodes of the graph, even with coordinates.

Navigation problems in directed graphs are not restricted to the playful introductory example of one-way streets. Staying in the car context, if we are for instance interested in minimizing gasoline cost, any hill-side city becomes directed, as driving downhill is virtually free, whereas driving uphill may incur a high cost. As such, when applying a cost measure, edges of a graph must often be represented by two directed edges with an appropriate cost.

The most important applications for investigating navigation in directed graphs are however beyond street networks. In computer networks, for instance, directed graphs have for instance been studied in the context data aggregation [49], routing [57], web crawlers [53], or traversing social networks [58]. Brass et al. [13] compared the exploration of directed

☆ A preliminary extended abstract appeared in Proceedings of the 16th International Conference on Principles of Distributed Systems (OPODIS), Springer, 2012 [37].

* Corresponding author. Tel.: +41 44 63 24776.
E-mail addresses: foklaus@ethz.ch (K.-T. Foerster), wattenhofer@ethz.ch (R. Wattenhofer).

http://dx.doi.org/10.1016/j.tcs.2015.11.017
0304-3975/

graphs to exploring the state space of a finite automaton, where the states are nodes and the transitions are edges. Deng and Papadimitriou [21] proposed the exploration of directed graphs as a model for learning, for example for a newborn: current states can be detected by sensor information (like eyes or ears) and possible actions leading to other states are known, but it is not known what the situation will be at a not yet explored state. And last but not least, exploring an unknown graph is considered one of the fundamental problems in robotics [17,33]. Because of all these applications, directed graph exploration will be the main focus in this article. In addition, we look at other navigation problems, such as searching for a node, which turn out to be related to exploration.

1.1. Model

For ease of notation, in the remainder of our paper a graph $G = (V, E)$ is directed and strongly connected with $|V| = n \geq 6$ nodes and $|E| = m$ edges. We denote an edge from $u \in V$ to $v \in V$ with uv. All nodes $v \in V$ have unique IDs and all edges $uv \in E$ have non-negative weights of weight(uv).

A walk is a concatenation of incident edges $uv, vw, \ldots$, with a path being a walk that visits no node twice. The weight or cost of a walk or a path is the sum of the weights of each edge traversal. The distance from a node u to a node v is the weight of a shortest path from u to v, i.e. of a path with the least cost.

Similarly, the hop-cost of a walk or a path is the number of edges traversed ("hops"), with the hop-distance from a node u to a node v being the hop-cost of a path from u to v with the least hop-cost. The radius of a node u is the largest hop-distance to any other node in V, with the diameter of a graph G being the largest radius of any node $u \in V$.

A searcher that explores a graph via some deterministic or randomized algorithm has unlimited computational power and memory, and may only traverse edges from tail to head. Upon arriving at a node v, the following information is made available: all outgoing incident edges including their weight, plus the IDs (cf. [46,52]) of the corresponding nodes at the head of these edges. We call a graph explored, if a searcher starting from some node s has visited all nodes and returned to s.

We only consider the common model of strongly connected directed graphs [1,20,21,33,50], since a searcher else might get stuck right away (Section 8). In Section 9, we also include geometric coordinates. Graph exploration is an online problem since only partial information about the graph is available [8,12]. For other exploration models, e.g., unique edge names, or information about incoming edges, we refer to Section 8 as well.

The cost of such an online exploration tour T is measured by the total sum of the weight of the traversed edges, denoted as cost(T). It is allowed (and might be necessary) to visit nodes multiple times, but if we traverse an edge again it costs the same as for the first time.

The competitive ratio c_r of a tour T is measured by the cost of the (online) tour divided by the cost of an (offline[1]) tour of minimum cost OPT, i.e. $c_r(T) = \frac{\text{cost}(T)}{\text{cost}(OPT)}$. Should OPT have a cost of 0 while cost(T) has a positive value, then we set $c_r(T) = \infty$. We say a deterministic algorithm A has a competitive ratio of $r(n)$ w.r.t. some graph class $\mathcal{G}$, if for every graph $G \in \mathcal{G}$ the computed tour T by A has a competitive ratio of at most $r(n)$, i.e., $r(n) \geq c_r(T)$ for all $G \in \mathcal{G}$. Should A induce a competitive ratio of ∞ for some $G \in \mathcal{G}$, then $r(n)$ is set to ∞ as well. If we do not denote a specific graph class $\mathcal{G}$, then we assume $\mathcal{G}$ to be the class of all strongly connected directed graphs.

We also study graph exploration by randomized algorithms,[2] where we study the *expected* costs instead of the worst case. Thus, the competitive ratio of a tour is now calculated by $\mathbb{E}\left[\frac{\text{cost}(T)}{\text{cost}(OPT)}\right]$. Let us consider a small introductory example, where the cost of an optimal tour is 10: If the randomized algorithm chooses a tour with a cost of 80 with a 25% chance and a tour with a cost of 40 with a 75% chance, then the competitive ratio is $(20 + 30)/10 = 5$. The competitive ratio of a randomized algorithm is analogously $r(n)$ if $r(n) \geq \mathbb{E}\left[\frac{\text{cost}(T)}{\text{cost}(OPT)}\right]$ for all $G \in \mathcal{G}$.

1.2. Results

In this article we give the first matching lower and upper bounds for the competitive exploration of an unknown directed graph. Our results are sharp for both the general weighted and unweighted cases. For randomized exploration, our results only have a gap of less than four. We prove similar results for various commonly used graph classes, like planar or complete graphs or bounding different parameters like degree or diameter. We also discuss changes in the model, like randomly choosing a starting position or more powerful searchers. We are able to show that in all these cases, the exploration of unknown directed graphs has a multiplicative overhead of $\Theta(n)$. Perhaps surprisingly, even allowing to see the nodes' coordinates in planar euclidean graphs will not help a searcher beyond a constant factor.

In a similar fashion, searching for a single node has $\Theta(n^2)$ overhead if all edges have unit weight (see Section 6). Furthermore, we look at the impact of randomly choosing a starting point. It turns out that even the best possible starting node can decrease any lower bound only by a factor of at most four.

[1] Offline in the sense that all information about the graph is available to the searcher.

[2] We note that every deterministic algorithm can also be used as a randomized algorithm. Furthermore, while an algorithm designer could choose to, e.g., pick new edges uniformly at random for exploration, any other strategy is possible as well, e.g., pick the next node to explore with a probability inversely proportional to the cost of reaching that node.

Please cite this article in press as: K.-T. Foerster, R. Wattenhofer, Lower and upper competitive bounds for online directed graph exploration, Theoret. Comput. Sci. (2015), http://dx.doi.org/10.1016/j.tcs.2015.11.017