Information Processing Letters 127 (2017) 17–20

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Time and space optimality of rotor-router graph exploration

Artur Menc [a], Dominik Pająk [a], Przemysław Uznański [b,*,1]

[a] Faculty of Fundamental Problems of Technology, Wrocław University of Technology, Poland
[b] Department of Computer Science, ETH Zürich, Switzerland

ARTICLE INFO

Article history:
Received 10 March 2016
Received in revised form 6 March 2017
Accepted 18 June 2017
Available online 23 June 2017
Communicated by M. Chrobak

Keywords:
Distributed computing
Graph exploration
Memory complexity

ABSTRACT

We consider the problem of exploration of an anonymous, port-labeled, undirected graph with n nodes, m edges and diameter D, by a single mobile agent. Initially the agent does not know the graph topology nor any of the global parameters. Moreover, the agent does not know the incoming port when entering a vertex. Each vertex is endowed with memory that can be read and modified by the agent upon its visit to that node. However the agent has no operational memory i.e., it cannot carry any state while traversing an edge. In such a model at least $\log_2 d$ bits are needed at each vertex of degree d for the agent to be able to traverse each graph edge. This number of bits is always sufficient to explore any graph in time $\mathcal{O}(mD)$ using algorithm Rotor-Router Yanovski et al. (2003) [1]. We show that even if the available node memory is unlimited then time $\Omega(mD)$ is required in some graphs, regardless of the algorithm. This shows that Rotor-Router is asymptotically optimal in the worst-case. Secondly we show that for the case of paths, Rotor-Router attains exactly optimal time.

1. Introduction

In this paper we consider the exploration problem of a port-labeled graph in the following setting. The exploration is performed by a single agent that has no memory (oblivious agent) and when it enters to a node it has no information about the port number through which it entered (no inport). Each node contains some number of bits of memory that can be read and modified by the agent upon its visit. Hence the whole navigation mechanism needs to be defined using only local information.

Rotor-Router is an algorithm in which each node maintains a pointer to one of its neighbors and a cyclic sequence of its neighbors. Upon each visit of the token to a node, the token is propagated along the pointer of its current node and the pointer of the node is advanced to the next position in the cyclic sequence. Studies of Rotor-Router show that it can be used as a graph-exploration algorithm and there are guarantees on exploration time even if the initial state of the pointers and the sequence at each node can be set by an adversary. In such a case, Rotor-Router can be also seen as a space-efficient algorithm since the only information that needs to be stored at a node is the current position of the pointer. Hence $\lceil \log_2 d \rceil$ bits of memory needs to be stored at each node with degree d to implement Rotor-Router. It is worth observing that with less memory the task becomes impossible as at each node with degree d, the algorithm needs at least d different inputs to traverse every outgoing edge, which is already necessary for exploring star graphs. This paper considers the following question: does there exist an algorithm for exploration with oblivious agent with no inport that is always faster than Rotor-Router if we allow

* Corresponding author.
E-mail addresses: arturmenc@gmail.com (A. Menc), pajakd@gmail.com (D. Pająk), przemyslaw.uznanski@inf.ethz.ch (P. Uznański).

[1] Part of this work was done while D. Pająk was visiting P. Uznański at Aix-Marseille Université. Partially supported by the Labex Archimède and by the ANR project MACARON (ANR-13-JS02-0002).

http://dx.doi.org/10.1016/j.ipl.2017.06.010
0020-0190/

more bits of memory at each node? We give a negative answer to this question by showing a family of graphs for which any such algorithm cannot be asymptotically faster than Rotor-Router even if we allow unbounded memory at each node.

1.1. Preliminaries

We assume anonymous graph $G = (V, E)$ with n nodes, m edges, diameter D and with no global labeling of the nodes nor the edges. In order to navigate in the graph, the agent needs to locally distinguish between the edges outgoing from its position, so we assume that all edges outgoing from a fixed vertex with degree d are distinctly labeled using numbers $\{1, 2, \dots, d\}$.

The agent is modeled as a memoryless token. Each node contains a label of $\mathcal{M}$ bits which can be read and modified by the agent upon its visit to that node. Such a model will be referred to as *an oblivious agent*.

Let us denote by $\mathcal{S} = \{0, 1, \dots, 2^{\mathcal{M}} - 1\}$ the set of node states. An oblivious agent is then defined as a function $f : \mathcal{S} \times \mathbb{N} \to \mathcal{S} \times \mathbb{N}$ whose input is a tuple $(s, d) \in \mathcal{S} \times \mathbb{N}$, where s is the state of the node currently occupied by the agent and d is the node degree. The output of function f is a tuple $(s', p) \in \mathcal{S} \times \{1, 2, \dots, d\}$, where s' is the new state of the currently occupied node and p is the port number through which the agent exits the current node in the current step. We say that the agent is located at a node v *at the beginning* of some step t, then traverses the chosen arc *during* step t and appears at the other end of the arc at the beginning of step $t + 1$. Initially, each node is in a starting state s^0.

Observe that the port label through which the agent entered to the current node is not part of the input, so the agent cannot easily backtrack its moves. We call this model feature *an unknown inport*.

At any step, the configuration of Rotor-Router is a pair $((\pi_v)_{v\in V}, r)$, where r is the current position of the agent. In the current step, the agent traverses the edge from port π_r, which is then advanced to the next port number modulo $\deg(r)$.

Graph exploration problem The goal of the agent is to visit all vertices of graph G. We assume that the initial agent position in the graph as well as the port-labeling of the edges can be chosen by an adversary. Initially the agent has no knowledge about the topology of G or even its size.

An oblivious agent that explores all unknown graphs, needs to traverse all edges and thus needs at least d different inputs at any vertex with degree d in order to traverse all of its outgoing edges. By considering stars with d leaves, this leads to the following lower bound on the total memory at a vertex.

Observation 1.1. *If an oblivious agent explores in the model with unknown inport all graphs of maximal degree* Δ, *then* $\mathcal{M} \geq \log_2 \Delta$.

1.2. Our results

In this paper we show two lower bounds. In the first we show that for any oblivious algorithm with no inport there exists a labeling of a path for which this algorithm needs at least $(n - 1)^2$ steps. This shows that unbounded memory at a node cannot decrease the exploration time of the path even by one step. On the other hand Rotor-Router needs only 1 bit and always explores a path in time at most $(n - 1)^2$. For general graphs we show that any oblivious agent in the model with no inport requires time $\Omega(mD)$ for some graphs with m edges and diameter D, regardless of the sizes of node memory. This shows that it is impossible to construct an algorithm for oblivious agent that would be asymptotically faster than the Rotor-Router in the worst-case even if unbounded memory at each node is available.

1.3. Related work

When exploring a graph using a Rotor-Router mechanism with arbitrary initialization, time $\Theta(mD)$ is always sufficient and required for specific pointer initialization and starting vertex, in arbitrary graph [1,2]. Since the Rotor-Router requires no special initialization, it can be implemented in a graph with $\lceil \log_2 d \rceil$-bits of memory at each node with degree d. An oblivious agent can simply exit the node v via port $w(v) + 1$, where $w(v)$ is the value on the whiteboard, and increment the value $w(v)$ modulo $\deg(v)$. Thus exploration in time $\mathcal{O}(mD)$ is possible by oblivious agents with $\lceil \log_2 d \rceil$-bit node memory, which is the smallest possible by Observation 1.1.

2. Lower bounds

In this section we prove a lower bound on the number of steps of graph exploration for any oblivious agent. First we need the characterization of behavior of oblivious agents in port-labeled graphs.

Definition 2.1. Let $\text{port}_d(i)$, for $d = 1, 2, \dots$, denote the outport chosen by the agent upon its i-th visit to a node v of degree d.

Lemma 2.2. *The behavior of any oblivious agent* $\mathcal{A}$ *in graph* G *with arbitrary size of node memory is fully characterized by the collection of functions* $\text{port}_d(i)$ *for* $d = 1, 2, \dots$.

Proof. Since the agent has no internal memory and does not know the label of the port through which it enters to a node, the only information the agent has is the degree of the current node and the state of the node memory. Thus the label of the next outport taken from a node v can only depend on the degree of v and on the labels of previously chosen outports from node v. This shows that $\text{port}_{\deg(v)}$ is well defined, and completes the proof. □

The following lemma characterizes the worst-case exploration time for any oblivious agent on a path.