



HAL
open science

Brief announcement: On the power of one bit in graph exploration without backtracking

Artur Menc, Dominik Pajak, Przemysław Uznaski

► To cite this version:

Artur Menc, Dominik Pajak, Przemysław Uznaski. Brief announcement: On the power of one bit in graph exploration without backtracking. DISC 2015, Toshimitsu Masuzawa; Koichi Wada, Oct 2015, Tokyo, Japan. hal-01207849

HAL Id: hal-01207849

<https://hal.science/hal-01207849>

Submitted on 1 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Brief announcement: On the power of one bit in graph exploration without backtracking.

Artur Menc¹, Dominik Pająk², and Przemysław Uznański^{3*}

¹ Wrocław University of Technology, Poland

² Computer Laboratory, University of Cambridge, UK

³ Helsinki Institute for Information Technology HIIT,
Department of Computer Science, Aalto University, Finland

1 Introduction

We consider a model of a deterministic $(\mathcal{M}_a, \mathcal{M}_w)$ -agent with \mathcal{M}_a bits of internal memory and \mathcal{M}_w bits at each node of some graph G that is able to visit vertices and traverse edges of G . The memory at each node is accessible only upon visiting to that node. The goal of the agent is to explore an anonymous and initially unknown G , i.e. visit all its n nodes. We assume that there is no global labeling of nodes but we assume local port-labeling i.e., edges outgoing from each node of degree d are uniquely labeled with numbers from 1 to d . The label of the port via which the agent entered to its current location is not a part of the input to our agent hence $\mathcal{M}_a + \mathcal{M}_w \geq \log_2 d$ as each deterministic algorithm needs at least d different inputs to produce d different outputs at each node.

2 Upper bounds

Exploration by a $(0, \log d)$ -agent is possible using ROTOR-ROUTER that works in pessimistic time $\Theta(mD)$ [1,4] in any graph with m edges and diameter D . In ROTOR-ROUTER the agent upon consecutive visits to node v is propagated to neighbors of v in a round-robin fashion. Previous work on ROTOR-ROUTER [1,4] showed that this procedure exhibits the following regular structure. In subsequent phases, it traverses Eulerian cycles of subgraphs of directed symmetric version of G , where i -th phase starts with the i -th traversal of the arc associated with port 1 outgoing from the starting vertex. In each phase, ROTOR-ROUTER follows the same trajectory as in previous phases, with the exception that each so-called border vertex v (a vertex whose set of outgoing arcs contains both traversed and not traversed arcs at the beginning of a given phase) is a root of a new exploration subphase. In an exploration subphase, some neighborhood of border vertex v is explored, including all unvisited neighbors of v , potentially creating new border vertices.

Algorithm overview. In our algorithm ONE-BIT, the agent carries one bit of information (two states: α -MODE and β -MODE), that is whether the last seen vertex was visited at most $\deg(v)$ times or not. Exploration subphase in one non-interrupted α -MODE resembles an exploration subphase from ROTOR-ROUTER,

* 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)

that is starting each time from some border vertex v , and ending in the same vertex v , while traversing only new arcs. Denote a sequence of vertices visited by $v_0 = v, v_1, \dots, v_k = v$. When returning to v for the $\deg(v)$ -th time, the algorithm detects that every neighbor of v was visited, and switches to β -MODE, where agent will follow the same sequence v_0, v_1, \dots, v_k traversing each arc for the second time. However, whenever in a β -MODE the agent detects that its current location is a new border vertex, let us call it v_i , then current β -MODE will be interrupted by a subsequent (recursive) phases of α -MODE and β -MODE both starting and ending in v_i . Only after those two phases (and possible recursive ones), the agent will fall back to original β -MODE traversal and continue to v_{i+1} .

The trajectory of ONE-BIT occurs as an emergent behavior of using round-robin propagation, as agent memory is too small to store anything relevant to the phases. It is important to note that even if locally the walk resembles ROTOR-ROUTER, the global structure of both walks are different (think of DFS vs. BFS).

Theorem 1. *An $(1, \mathcal{O}(\log d))$ -agent following algorithm ONE-BIT explores with return any graph in time $4m$, by traversing every edge twice in each direction. Moreover, after the exploration is completed, the memory state of the agent and of each node is equal to the initial state.*

3 Lower bounds

Exploration using a $(\infty, 0)$ -agent corresponds to a well known problem of Universal Traversal Sequences. Exploration time in this model is lower bounded by $\Omega(n^{1.51})$ on the path [3] and by $\Omega(n^4)$ in general graphs [2]. We show lower bound for $(0, \infty)$ -agent. The following theorem is an evidence that the ROTOR-ROUTER is the best strategy for oblivious agent with memory on nodes.

Theorem 2. *For any $(0, \infty)$ -agent the worst-case exploration time of graph G is at least $(n - 1)^2$ if G is a path and $\Omega(n^3)$ if G can be an arbitrary graph.*

4 Conclusions

We showed that in the model with unknown inport having only one type of memory leads to pessimistic exploration time $\Omega(n^3)$. On the other hand $(1, \mathcal{O}(\log d))$ -agent can explore in optimal time $4m$. Secondly we show that a single type of memory cannot guarantee that the agent will stop after completing the task while our ONE-BIT algorithm for $(1, \mathcal{O}(\log d))$ -agent has this property.

References

1. E. Bampas, L. Gąsieniec, N. Hanusse, D. Ilcinkas, R. Klasing, and A. Kosowski. Euler tour lock-in problem in the rotor-router model. In *DISC*, pages 423–435, 2009.
2. A. Borodin, W. L. Ruzzo, and M. Tompat. Lower bounds on the length of universal traversal sequences. *JCSS*, 45(2):180–203, 1992.
3. H. Dai and K. Flannery. Improved length lower bounds for reflecting sequences. In *Computing and Combinatorics*, volume 1090 of *LNCS*, pages 56–67, 1996.
4. V. Yanovski, I. A. Wagner, and A. M. Bruckstein. A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37(3):165–186, 2003.