Journal of Computer and System Sciences 82 (2016) 802–816

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Bounds on the cover time of parallel rotor walks ☆

Dariusz Dereniowski [a], Adrian Kosowski [b], Dominik Pająk [c,*], Przemysław Uznański [d]

[a] Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gdańsk, Poland
[b] GANG Project, Inria Paris and IRIF, Paris, France
[c] Department of Computer Science, Faculty of Fundamental Problems of Technology, Wrocław University of Technology, Wrocław, Poland
[d] Department of Computer Science, ETH Zürich, Switzerland

ARTICLE INFO

Article history:
Received 18 December 2014
Received in revised form 11 January 2016
Accepted 29 January 2016
Available online 9 March 2016

Keywords:
Distributed graph exploration
Rotor-router
Cover time
Collaborative robots
Parallel random walks
Derandomization

ABSTRACT

The rotor-router mechanism was introduced as a deterministic alternative to the random walk in undirected graphs. In this model, a set of k identical walkers is deployed in parallel, starting from a chosen subset of nodes, and moving around the graph in synchronous steps. During the process, each node successively propagates walkers visiting it along its outgoing arcs in round-robin fashion, according to a fixed ordering. We consider the cover time of such a system, i.e., the number of steps after which each node has been visited by at least one walk, regardless of the initialization of the walks. We show that for any graph with m edges and diameter D, this cover time is at most $\Theta(mD/\log k)$ and at least $\Theta(mD/k)$, which corresponds to a speedup of between $\Theta(\log k)$ and $\Theta(k)$ with respect to the cover time of a single walk.

1. Introduction

In graph exploration problems, a walker or group of walkers (agents) is placed on a node of a graph and moves between adjacent nodes, with the goal of visiting all the nodes of the graph. The study of graph exploration is closely linked to central problems of theoretical computer science, such as the question of deciding if two nodes of the graph belong to the same connected component (*st-connectivity*). For example, fast approaches to connectivity testing in little memory rely on the deployment of multiple random walks [6,13]. In these algorithms, the initial locations of the walkers are chosen according to a specific probability distribution.

More recently, multiple walks have been studied in a worst-case scenario where the k agents are placed on some set of starting nodes and deployed in parallel, in synchronous steps. The considered parameter is the *cover time* of the process, i.e., the number of steps until each node of the graph has been visited by at least one walker. Alon et al. [2], Efremenko and Reingold [11], and Elsässer and Sauerwald [12] have studied the notion of the *speedup* of the random walk for an undirected graph G, defined as the ratio between the cover time of a k-agent walk in G for worst-case initial positions of agents and that of a single-agent walk in G starting from a worst-case initial position, as a function of k. A characterization of the

☆ Research partially supported by ANR project DISPLEXITY and by NCN under contract DEC-2011/02/A/ST6/00201. Dariusz Dereniowski was partially supported by a scholarship for outstanding young researchers funded by the Polish Ministry of Science and Higher Education. Some of the results of this paper appeared in the extended abstract [9], published in the Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS 2014).

* Corresponding author.
E-mail address: pajakd@gmail.com (D. Pająk).

http://dx.doi.org/10.1016/j.jcss.2016.01.004
0022-0000/

speedup has been achieved for many graph classes with special properties, such as small mixing time compared to cover time. However, a central question poised in [2] still remains open: what are the minimum and maximum values of speedup of the random walk in arbitrary graphs? The smallest known value of speedup is $\Theta(\log k)$, attained e.g. for the cycle, while the largest known value is $\Theta(k)$, attained for many graph classes, such as expanders, cliques, and stars.

In this work, we consider a deterministic model of walks on graphs, known as the *rotor-router*. The rotor-router model, introduced by Priezzhev et al. [22], provides a mechanism for the environment to control the movement of the agent deterministically, mimicking the properties of exploration as the random walk. In the rotor-router, the agent has no operational memory and the whole routing mechanism is provided within the environment. The edges outgoing from each node v are arranged in a fixed cyclic order known as a *port ordering*, which does not change during the exploration. Each node v maintains a *pointer* which indicates the edge to be traversed by the agent during its next visit to v. If the agent has not visited node v yet, then the pointer points to an arbitrary edge adjacent to v. The next time when the agent enters node v, it is directed along the edge indicated by the pointer, which is then advanced to the next edge in the cyclic order of the edges adjacent to v. In this paper we also consider a class of processes called *fair strategies* which are a generalization of the rotor-router model. In a fair strategy an agent entering a node can choose an arbitrary outgoing arc among arcs with the minimum number of traversals.

For a single agent, the (deterministic) cover time of the rotor-router and the (expected) cover time of the random walk prove to be surprisingly convergent for many graph classes. In general, it is known that for any n-node graph of m edges and diameter D, the cover time of the rotor-router in a worst-case initialization is precisely $\Theta(mD)$ [26,3]. By comparison, the random walk satisfies an upper bound of $O(mD \log n)$ on the cover time, though this bound is far from tight for many graph classes. Different locally fair exploration strategies were considered in [7]: Oldest-First and Least-Used-First. In the Oldest-First strategy, an agent visiting a node chooses an edge that has not been traversed for the longest time. In the Least-Used-First strategy it chooses an edge with the smallest number of traversals. In both strategies ties can be broken in an arbitrary way. Cooper et al. [7] showed that in undirected graphs any Oldest-First strategy achieves cover time of $O(mD)$ whereas exploration using Least-Used-First strategy can lead to exponential cover time. Note that in directed symmetric graphs, the Oldest-First strategy is equivalent to the rotor-router and the Least-Used-First is more general and is equivalent to the class of fair strategies. Yanovski et al. [26] observed, based on the analysis of Koenig and Simmons [17], that any fair strategy has a cover time of $O(mD)$.

The behavior of the rotor-router model with multiple agents appears to be much more complicated. Since the parallel walkers interact with the pointers of a single rotor-router system, they cannot be considered independent (in contrast to the case of parallel random walks). In the first work on the topic, Yanovski et al. [26] showed that adding a new agent to a rotor-router system with k agents cannot increase the cover time, and showed experimental evidence suggesting that a speedup does indeed occur. Klasing et al. [16] have provided the first evidence of speedup, showing that for the special case when G is a cycle, a k-agent system explores an n-node cycle $\Theta(\log k)$ times more quickly than a single agent system.

In this work we completely resolve the question of the possible range of speedups of the parallel rotor-router model in a graph, showing that its value is between $\Theta(\log k)$ and $\Theta(k)$, for any graph. Both of these bounds are tight. Thus, the proven range of speedup for the rotor-router corresponds precisely to the conjectured range of speedup for the random walk. We also show that the speedup of at least $\Theta(\log k)$ is achieved for any fair strategy.

1.1. Related work

The rotor-router model Studies of the rotor-router started with works of Wagner et al. [25] who showed that in this model, starting from an arbitrary configuration (arbitrary cyclic orders of edges, arbitrary initial values of the port pointers and an arbitrary starting node) the agent covers all m edges of an n-node graph within $O(nm)$ steps. Bhatt et al. [5] showed later that within $O(nm)$ steps the agent not only covers all edges but enters (establishes) an Eulerian cycle. More precisely, after the initial stabilization period of $O(nm)$ steps, the agent keeps repeating the same Eulerian cycle of the directed symmetric version $\vec{G}$ of graph G (see Section 3 for a definition). Subsequently, Yanovski et al. [26] and Bampas et al. [3] showed that the Eulerian cycle is in the worst case entered within $\Theta(mD)$ steps in a graph of diameter D. Considerations of specific graph classes were performed in [14]. Robustness properties of the rotor-router were further studied in [4], where it has been considered the time required for the rotor-router to stabilize to a (new) Eulerian cycle after an edge is added or removed from the graph. Regarding the terminology, we note that the rotor-router model has also been referred to as the *Propp machine* [3] or *Edge Ant Walk algorithm* [25,26], and has also been described in [5] in terms of traversing a maze and marking edges with pebbles. Studies of the multi-agent rotor-router were performed by Yanovski et al. [26] and Klasing et al. [16], and its speedup was considered for both worst-case and best-case scenarios.

A variant of the multi-agent rotor-router mechanism has been extensively studied in a different setting, in the context of balancing the workload in a network. In such a scenario, the walks in the graph are performed by entities referred to as *tokens*. Cooper and Spencer [8] studied d-dimensional grid graphs and showed a constant bound on the discrepancy, defined as the difference between the number of tokens at a given node v in the rotor-router model and the expected number of tokens at v in the random-walk model. Subsequently, Doerr and Friedrich [10] analyzed in more detail the distribution of tokens in the rotor-router mechanism on the 2-dimensional grid. Akbari and Berenbrink [1] showed an upper bound of $O(\log^{3/2} n)$ on the discrepancy for hypercubes and a bound of $O(1)$ for a constant-dimensional torus.