Sebastian, Brandt; Keller, Barbara; Rybicki, Joel; Suomela, Jukka; Uitto, Jara

Brief Announcement: Efficient Load-Balancing through Distributed Token Dropping

Published in:
34th International Symposium on Distributed Computing (DISC 2020)

DOI:
10.4230/LIPIcs.DISC.2020.40

Published: 01/01/2020

Document Version
Publisher's PDF, also known as Version of record

Brief Announcement: Efficient Load-Balancing Through Distributed Token Dropping

Sebastian Brandt
ETH Zürich, Switzerland
brandts@ethz.ch

Barbara Keller
Aalto University, Finland
barbara.keller@aalto.fi

Joel Rybicki
IST Austria, Klosterneuburg, Austria
joel.rybicki@ist.ac.at

Jukka Suomela
Aalto University, Finland
jukka.suomela@aalto.fi

Jara Uitto
Aalto University, Finland
jara.uitto@aalto.fi

Abstract

We introduce a new graph problem, the *token dropping game*, and we show how to solve it efficiently in a distributed setting. We use the token dropping game as a tool to design an efficient distributed algorithm for the *stable orientation* problem, which is a special case of the more general *locally optimal semi-matching* problem. The prior work by Czygrinow et al. (DISC 2012) finds a locally optimal semi-matching in $O(\Delta^5)$ rounds in graphs of maximum degree Δ, which directly implies an algorithm with the same runtime for stable orientations. We improve the runtime to $O(\Delta^4)$ for stable orientations and prove a lower bound of $\Omega(\Delta)$ rounds.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases distributed algorithms, graph problems, semi-matching

Digital Object Identifier 10.4230/LIPIcs.DISC.2020.40

Related Version The full version of the paper is available at https://arxiv.org/abs/2005.07761.

1 Introduction

In this work, we study efficient distributed algorithms for assignment problems. The task is to assign each customer to one adjacent server, and the customers prefer servers with a low load, i.e., few other customers. We are interested in finding a *stable assignment*, that is, an assignment in which no customer has an incentive to unilaterally switch servers. The stable assignment problem that we study here is also known as a *locally optimal semi-matching*, and it was studied in the distributed setting by Czygrinow et al. [4].

Stable Orientations. We start by studying a restricted version of the problem, *stable orientation*, which is the special case in which all customers can choose between two possible servers – we will then see how the same ideas generalize also to the stable assignment problem.

We model this setting as the following graph problem: The task is to orient all edges. We say that an oriented edge $e = (u, v)$ is *happy* if $\text{indegree}(v) \leq \text{indegree}(u) + 1$, that is, turning the orientation of e from (u, v) to (v, u) would not lower the indegree of the head of edge e. An orientation is *stable* if all edges are happy, i.e., no customer has an incentive to change his or her server.

The intuition here is that if an edge $e = \{u, v\}$ is oriented from u to v, then customer e is using server v. The *load* of a server is the total number of customers using it, i.e., its indegree. Customers would like to use servers with a low load to maximize the quality of service. From this viewpoint, a stable orientation corresponds to a game-theoretic equilibrium.

Model of Computing. We study the stable orientation problem and its generalizations in the standard LOCAL [7] model of distributed computing.

State of the Art. By prior work [4], it is known that the problem can be solved in $O(\Delta^5)$ communication rounds in graphs of maximum degree Δ; note that the running time is independent of the number of nodes.

2 Contributions

The main technical contribution is that we improve the distributed round complexity of stable orientations from $O(\Delta^5)$ to $O(\Delta^4)$. We also prove a lower bound of $\Omega(\Delta)$ for any algorithm that finds a stable orientation.

New techniques. On a high level, we use the following new techniques to save time in comparison with the prior algorithm:

- In the prior work, one starts with an arbitrary orientation. This potentially creates a large amount of unhappiness and resolving it takes a lot of time.
- In our work, we iteratively orient edges more carefully, so that there is always at most one unit of excess load per node.
- We show that the carefully oriented edges have a nice structure and model this structure by a new problem called the *token dropping game*. We show how to solve this problem efficiently, that is, in $O(\Delta^3)$ rounds.
- We play token dropping with the excess load in order to resolve unhappiness. After $O(\Delta)$ such iterations, all edges are happily oriented.

Token Dropping Game. The key new idea that we use to solve the orientation problem is to introduce a new graph problem that we call the *token dropping game*. The input consists of a graph in which the nodes are organized in *layers*, numbered from 0 to L. Some of the nodes hold a token; a node can hold at most one token. The rules are simple:

> Any token can move downwards from layer ℓ to layer $\ell - 1$ along any edge to any node that does not currently hold a token. Each edge can be used at most once.

Put otherwise, once an edge has been used to move a token, it is deleted. The task is to find some possible sequence of token movements such that we reach a configuration in which no token can be moved anymore, i.e., the only goal of this single-player game is to get stuck.

Our algorithm for this problem follows a simple proposal strategy. In every round, a node that has no token, but is adjacent to a token in a higher layer, send a proposal to the node with a token. A node that has a token accepts one proposal, and passes its token to the proposing node. The main technical challenge is to show that this process terminates quickly.

Using Token Dropping to Find Stable Orientations. We show that the token dropping problem can be solved in $O(L \cdot \Delta^2)$ rounds with a distributed algorithm; we also prove a lower bound of $\Omega(L + \Delta)$ rounds. We show that any algorithm that solves token dropping in $T(L, \Delta)$ rounds can be used to find a stable orientation in $O(\Delta \cdot T(\Delta, \Delta))$ rounds. Plugging in our algorithm for token dropping, we obtain an algorithm for finding a stable orientation in $O(\Delta^4)$ rounds, a factor-Δ improvement over the previous algorithm by [4]. In the full version we also investigate generalizations and relaxations of stable orientations problem.

Discussion. This work is part of the ongoing effort of understanding the distributed computational complexity of *locally verifiable problems*. In brief, these are problems in which a solution is globally correct if it looks good in all constant-radius neighborhoods. Stable orientations are by definition locally verifiable: if all edges are happy, the orientation is stable, and the happiness of an edge only depends on the other edges adjacent to it.

Typically, the complexity of locally verifiable problems is studied as a function of two parameters, the number of nodes n and the maximum degree Δ. In essence, these capture two complementary notions of scalability: how does the complexity of finding a solution increase when the input graph gets larger vs. when the input graph gets denser.

To study dependency on Δ, it is helpful to identify natural examples of graph problems that can be solved in $T(\Delta)$ rounds for some function T, *independently of* n. Key examples of problems that can be solved in $T(\Delta)$ rounds include maximal matching on bipartite graphs [2,6], maximal fractional matching [1,5], and weak coloring in odd-degree graphs [3,8]. However, all of these problems have a complexity at most linear in Δ. Stable orientation is perhaps one of the simplest locally verifiable graph problems that is known to be solvable in $T(\Delta)$ rounds, but for which the current upper bound is *superlinear* in Δ.

In this work we take the first steps towards deriving tight bounds on the round complexity of the stable orientation problem, with the long-term goal of showing that it indeed requires $\omega(\Delta)$ rounds.

References

1. Matti Åstrand and Jukka Suomela. Fast distributed approximation algorithms for vertex cover and set cover in anonymous networks. In *SPAA*, 2010. doi:10.1145/1810479.1810533.
2. Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower Bounds for Maximal Matchings and Maximal Independent Sets. In *FOCS*, 2019. doi:10.1109/FOCS.2019.00037.
3. Sebastian Brandt. An Automatic Speedup Theorem for Distributed Problems. In *PODC*, 2019. doi:10.1145/3293611.3331611.
4. Andrzej Czygrinow, Michal Hanćkowiak, Edyta Szymańska, and Wojciech Wawrzyniak. Distributed 2-Approximation Algorithm for the Semi-matching Problem. In *DISC*, 2012. doi:10.1007/978-3-642-33651-5_15.
5. Mika Göös, Juho Hirvonen, and Jukka Suomela. Linear-in-Δ lower bounds in the LOCAL model. *Distributed Computing*, 30(5), 2017. doi:10.1007/s00446-015-0245-8.
6. Michal Hanckowiak, Michal Karonski, and Alessandro Panconesi. On the Distributed Complexity of Computing Maximal Matchings. In *SODA*, 1998.
7. Nathan Linial. Locality in Distributed Graph Algorithms. *SIAM J. Comput.*, 21(1), 1992. doi:10.1137/0221015.
8. Moni Naor and Larry Stockmeyer. What Can be Computed Locally? *SIAM J. Comput.*, 24(6), 1995. doi:10.1137/S0097539793254571.