This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Balliu, Alkida; Brandt, Sebastian; Efron, Yuval; Hirvonen, Juho; Maus, Yannic; Olivetti, Dennis; Suomela, Jukka

Brief Announcement: Classification of Distributed Binary Labeling Problems

Published in:
PODC 2020 - Proceedings of the 39th Symposium on Principles of Distributed Computing

DOI:
[10.1145/3382734.3405703](https://doi.org/10.1145/3382734.3405703)

Published: 31/07/2020

Document Version
Peer reviewed version

Please cite the original version:
Balliu, A., Brandt, S., Efron, Y., Hirvonen, J., Maus, Y., Olivetti, D., & Suomela, J. (2020). Brief Announcement: Classification of Distributed Binary Labeling Problems. In *PODC 2020 - Proceedings of the 39th Symposium on Principles of Distributed Computing* (pp. 349-351). ACM. <https://doi.org/10.1145/3382734.3405703>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Brief Announcement: Classification of Distributed Binary Labeling Problems*

Alkida Balliu
alkida.balliu@cs.uni-freiburg.de
University of Freiburg

Sebastian Brandt
brandts@ethz.ch
ETH Zurich

Yuval Efron
efronyuv@gmail.com
Technion

Juho Hirvonen
juho.hirvonen@aalto.fi
Aalto University

Yannic Maus
yannic.maus@cs.technion.ac.il
Technion

Dennis Olivetti
dennis.olivetti@cs.uni-freiburg.de
University of Freiburg

Jukka Suomela
jukka.suomela@aalto.fi
Aalto University

ABSTRACT

We present a complete classification of the deterministic distributed time complexity for a family of graph problems: *binary labeling problems* in trees in the usual LOCAL model of distributed computing. These are locally checkable problems that can be encoded with an alphabet of size two in the edge labeling formalism. Examples of binary labeling problems include sinkless orientation, sinkless and sourceless orientation, 2-vertex coloring, and perfect matching. We show that the complexity of any such problem is in one of the following classes: $O(1)$, $\Theta(\log n)$, $\Theta(n)$, or unsolvable. Furthermore, given the description of any binary labeling problem, we can easily determine in which of the four classes it is and what is an asymptotically optimal algorithm for solving it.

CCS CONCEPTS

• **Theory of computation** → **Distributed computing models; Complexity classes.**

KEYWORDS

LOCAL model, graph problems, locally checkable labeling problems, distributed computational complexity

ACM Reference Format:

Alkida Balliu, Sebastian Brandt, Yuval Efron, Juho Hirvonen, Yannic Maus, Dennis Olivetti, and Jukka Suomela. 2020. Brief Announcement: Classification of Distributed Binary Labeling Problems. In *ACM Symposium on Principles of Distributed Computing (PODC '20)*, August 3–7, 2020, Virtual Event, Italy. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3382734.3405703>

*The full version of the paper is available at <http://arxiv.org/abs/1911.13294>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PODC '20, August 3–7, 2020, Virtual Event, Italy

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7582-5/20/08...\$15.00
<https://doi.org/10.1145/3382734.3405703>

1 CONTRIBUTION

This work presents a complete classification of the deterministic distributed time complexity for a family of distributed graph problems: *binary labeling problems* in trees in the standard LOCAL model [15, 18]. These are a special case of widely-studied locally checkable labeling problems [16]. The defining property of a binary labeling problem is that it can be encoded with an *alphabet of size two* in the *edge labeling formalism*, which is a modern representation for locally checkable graph problems [2, 6, 17]; we will give the precise definition in Section 2. It is easy to see that there are binary labeling problems that fall in each of the following classes:

- Trivial problems, solvable in $O(1)$ rounds.
- Problems similar to sinkless orientation, solvable in $\Theta(\log n)$ rounds [7, 9, 13].
- Global problems, requiring $\Theta(n)$ rounds.
- Unsolvable problems.

We show that this is a *complete* list of all possible complexities. In particular, there are no binary labeling problems of complexities such as $\Theta(\log^* n)$ or $\Theta(\sqrt{n})$. For example, maximal matching is a problem very similar in spirit to binary labeling problems, it has a complexity $\Theta(\log^* n)$ in bounded-degree graphs [11, 15], and it can be encoded in the edge labeling formalism using an alphabet of size three [2]—our work shows that three labels are also necessary for all problems in this complexity class.

Moreover, using our results one can easily determine the complexity class of any given binary labeling problem. We give a simple, concise characterization of all binary labeling problems for classes $O(1)$, $\Theta(n)$, and unsolvable, and we show that all other problems belong to class $\Theta(\log n)$. Hence the deterministic distributed time complexity of a binary labeling problem is *decidable*, not only in theory but also *in practice*: given the description of any binary labeling problem, a human being or a computer can easily find out the distributed computational complexity of the problem, as well as an asymptotically optimal algorithm for solving the problem.

We continue with background and related work and then formally define binary labeling problems in Section 2. Section 2 also introduces notation to present our classification of all binary labeling problems precisely and concisely.

Our main result is presented in Table 1. Given any binary labeling problem Π , one can simply do mechanical pattern matching to find its complexity in this table.

2 BINARY LABELING PROBLEMS

Binary labeling problems in trees are a family of distributed graph problems that are a special case of widely-studied locally checkable labeling problems [16]. LCL problems have been traditionally specified by listing a *collection of permitted local neighborhoods* [16]. However, we will use the more recent *edge labeling* formalism [2, 6, 17], which is equally expressive in the case of trees, and it has the additional benefit that it makes it very convenient to apply the automatic round elimination technique [6, 17], which is very helpful to prove lower bounds. In the edge labeling formalism, problems are presented as follows:

- we have a bipartite graph and the nodes are colored with two colors, white and black,
- the task is to *label edges* with symbols from an alphabet Σ ,
- there are *both white and black constraints* that define the graph problem.

We emphasize that despite its bipartite appearance, the edge labeling formalism can easily be used to describe problems on general graphs by interpreting edges as “black nodes”. See the full version of the paper for details.

If we set $|\Sigma| = 2$ in the edge labeling formalism, we arrive at the definition of binary labeling problems. Formally, a *binary labeling problem* is a tuple $\Pi = (d, \delta, W, B)$, where

- $d \in \{2, 3, \dots\}$ is the *white degree*,
- $\delta \in \{2, 3, \dots\}$ is the *black degree*,
- $W \subseteq \{0, 1, \dots, d\}$ is the *white constraint*, and
- $B \subseteq \{0, 1, \dots, \delta\}$ is the *black constraint*.

An *instance* of problem Π is a pair (G, f) , where

- $G = (V, E)$ is a simple graph,
- $f: V \rightarrow \{\text{black}, \text{white}\}$ is a proper 2-coloring of G , i.e., $f(u) \neq f(v)$ for all $\{u, v\} \in E$.

In the distributed setting, we will assume that the color $f(v)$ is part of the local input of node $v \in V$.

Let $X \subseteq E$ be a subset of edges. For each node $v \in V$, its X -degree $\deg_X(v)$ is the number of edges in X that are incident to v . We say that $X \subseteq E$ is a *solution* to binary labeling problem Π if it satisfies the following constraints for all $v \in V$:

- If $f(v) = \text{white}$ and $\deg(v) = d$, then $\deg_X(v) \in W$.
- If $f(v) = \text{black}$ and $\deg(v) = \delta$, then $\deg_X(v) \in B$.

We use the term *relevant nodes* to refer to white nodes of degree d and black nodes of degree δ . The interpretation is that problem Π is interesting in regular neighborhoods in which all white nodes have degree d and all black nodes have degree δ ; any irregularities make the problem easier to solve, as all other nodes are unconstrained.

Example 2.1 (bipartite splitting). Let $d = \delta = 4$ and $W = B = \{1, 2, 3\}$. We can interpret a solution $X \subseteq E$ as a coloring: edges in X are colored red and all other edges are colored blue. Now Π is equivalent to the following graph problem on bipartite graphs: color all edges red or blue such that all degree-4 nodes are incident to at least one blue edge and at least one red edge.

Table 1: The complexity of a binary labeling problem given in the vector notation can be found by pattern matching.

Type	Problem family	White constraint	Black constraint	Deterministic complexity
I	I.a	100 ⁺	0** ⁺	unsolvable
	I.b	00 ⁺ 1	** ⁺ 0	
	I.c	0** ⁺	100 ⁺	
	I.d	** ⁺ 0	00 ⁺ 1	
II	II.a	000 ⁺	*** ⁺	
	II.b	*** ⁺	000 ⁺	
III	III.a	non-empty	111 ⁺	O(1)
	III.b	111 ⁺	non-empty	
IV	IV.a	1** ⁺	1** ⁺	
	IV.b	** ⁺ 1	** ⁺ 1	
V	V.a	10 ⁺ 1	010	$\Theta(n)$
	V.b	010	10 ⁺ 1	
VI	VI.a	0 ⁺ 1*	*10 ⁺	
	VI.b	*10 ⁺	0 ⁺ 1*	
VII	VII.a	all other cases		$\Theta(\log n)$

The full version contains several further examples of binary labeling problems and discusses the expressive power of the problem family.

Vector notation. In order to state our results in a concise and precise manner in Table 1, we interpret the set W as a bit vector $w_0 w_1 \dots w_d$ with $d + 1$ bits, so that $w_i = 1$ if $i \in W$ and $w_i = 0$ if $i \notin W$. Similarly, B can be interpreted as a bit vector with $\delta + 1$ bits.

Example 2.2. Using this notation, the problem of Example 2.1 can be represented by $W = 01110$ and $B = 01110$, or, in brief, $\Pi = (01110, 01110)$.

Note that when we use vector notation, vectors W and B fully determine problem Π ; therefore we do not need to specify d and δ separately and we can simply write $\Pi = (W, B)$. We will also use the following shorthand notation in Table 1: 1^+ signifies one or more 1s, 0^+ signifies one or more 0s, and $*$ signifies either 0 or 1. *Non-empty* means that the constraint is not 0^+ .

3 BACKGROUND AND RELATED WORK

The study of distributed graph algorithms has traditionally focused on specific graph problems—for example, investigating exactly what is the locality of finding a maximal independent set [2, 3]. However, in the recent years we have seen more focus on the development of a distributed complexity theory with which we can reason about entire *families of graph problems* [1, 4, 5, 7–10, 12, 14, 19, 20].

The key example is the family of *locally checkable labeling* problems (LCLs), introduced by [16]. Informally, a problem is locally checkable if the feasibility of a solution can be verified by looking at all constant-radius neighborhoods. For example, maximal independent sets are locally checkable, as we can verify both independence and maximality by looking at radius-1 neighborhoods.

In this line of research, among the most intriguing results are various *gap theorems*: For example, there are LCL problems solvable in $\Theta(\log^* n)$ rounds, while some LCL problems require $\Theta(\log n)$ rounds. However, between these two classes there is a gap: there are no LCL problems whose deterministic complexity in bounded-degree graphs is between $\omega(\log^* n)$ and $o(\log n)$ [9]. The existence of such a gap immediately suggests a follow-up question: given the description of an LCL problem, can we *decide* on which side of the gap it lies? And if so, can we automatically construct an asymptotically optimal algorithm for solving the problem?

As soon as we look at a family of graphs that contains e.g. 2-dimensional grids, questions related to the distributed complexity of a given LCL problem become undecidable [8, 16]. However, if we look at the case of paths and cycles, we can at least in principle write a computer program that determines the computational complexity of a given LCL problem [1, 8, 16], and some questions related to the complexity of LCLs in trees are also decidable [10]—unfortunately, we run into PSPACE-hardness already in the case of paths and cycles [1].

4 DISCUSSION

We conjecture that *all* questions about the distributed complexity of LCL problems in trees are decidable. Proving (or disproving) the conjecture is a major research program, but in this work we take one step towards proving the conjecture and we bring plenty of good news: we introduce a family of LCL problems, so-called *binary labeling problems*, and we show that we can completely characterize the deterministic distributed complexity of every binary labeling problem in trees. In particular, all questions about the deterministic distributed complexity of these problems are decidable not only in principle but also in practice—using our results, a human being or a computer can easily find an optimal algorithm for solving any given binary labeling problem.

Our work also sheds new light on the *automatic round elimination technique* [6, 17]. Previously, it was known that sinkless orientation is a nontrivial *fixed point* for round elimination [7]—such fixed points are very helpful for lower bound proofs, but little was known about the existence of other nontrivial fixed points. Our classification of binary labeling problems in this work led to the discovery of new nontrivial fixed points—this will hopefully pave the way for the development of a theoretical framework that enables us to understand when round elimination leads to fixed points and why.

While our focus in this work was on deterministic complexity, we also explored the randomized complexity of binary labeling problems. The main open question that we leave for future work is extending the characterization to randomized distributed algorithms: some binary labeling problems can be solved in $\Theta(\log \log n)$ rounds with randomized algorithms, but it is not yet known exactly which binary labeling problems belong to this class. Our work takes the first steps towards developing such a classification.

ACKNOWLEDGMENTS

We thank Jan Studený and anonymous reviewers for helpful comments on earlier versions of this work. This work was supported in

part by the Academy of Finland, Grant 314888 (Juho Hirvonen), and by the European Union's Horizon 2020 Research And Innovation Programme under grant agreement no. 755839 (Yuval Efron, Yannic Maus).

REFERENCES

- [1] Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Mikael Rabie, and Jukka Suomela. 2019. The Distributed Complexity of Locally Checkable Problems on Paths is Decidable. In *Proc. 38th ACM Symposium on Principles of Distributed Computing (PODC 2019)*. ACM Press, 262–271. <https://doi.org/10.1145/3293611.3331606> arXiv:1811.01672
- [2] Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikael Rabie, and Jukka Suomela. 2019. Lower bounds for maximal matchings and maximal independent sets. In *Proc. 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*. IEEE, 481–497. <https://doi.org/10.1109/FOCS.2019.00037> arXiv:1901.02441
- [3] Alkida Balliu, Sebastian Brandt, and Dennis Olivetti. 2020. Distributed Lower Bounds for Ruling Sets. arXiv:2004.08282 <http://arxiv.org/abs/2004.08282>
- [4] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. 2018. Almost global problems in the LOCAL model. In *Proc. 32nd International Symposium on Distributed Computing (DISC 2018) (Leibniz International Proceedings in Informatics (LIPIcs))*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 9:1–9:16. <https://doi.org/10.4230/LIPIcs.DISC.2018.9>
- [5] Alkida Balliu, Juho Hirvonen, Janne H Korhonen, Tuomo Lempiäinen, Dennis Olivetti, and Jukka Suomela. 2018. New classes of distributed time complexity. In *Proc. 50th ACM Symposium on Theory of Computing (STOC 2018)*. ACM Press, 1307–1318. <https://doi.org/10.1145/3188745.3188860>
- [6] Sebastian Brandt. 2019. An Automatic Speedup Theorem for Distributed Problems. In *Proc. 38th ACM Symposium on Principles of Distributed Computing (PODC 2019)*. ACM Press, 379–388. <https://doi.org/10.1145/3293611.3331611> arXiv:1902.09958
- [7] Sebastian Brandt, Orr Fischer, Juho Hirvonen, Barbara Keller, Tuomo Lempiäinen, Joel Rybicki, Jukka Suomela, and Jara Uitto. 2016. A lower bound for the distributed Lovász local lemma. In *Proc. 48th ACM Symposium on Theory of Computing (STOC 2016)*. ACM Press, 479–488. <https://doi.org/10.1145/2897518.2897570>
- [8] Sebastian Brandt, Juho Hirvonen, Janne H Korhonen, Tuomo Lempiäinen, Patric R J Östergård, Christopher Purcell, Joel Rybicki, Jukka Suomela, and Przemysław Uznański. 2017. LCL problems on grids. In *Proc. 36th ACM Symposium on Principles of Distributed Computing (PODC 2017)*. ACM Press, 101–110. <https://doi.org/10.1145/3087801.3087833>
- [9] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. 2016. An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model. In *Proc. 57th IEEE Symposium on Foundations of Computer Science (FOCS 2016)*. IEEE, 615–624. <https://doi.org/10.1109/FOCS.2016.72>
- [10] Yi-Jun Chang and Seth Pettie. 2019. A Time Hierarchy Theorem for the LOCAL Model. *SIAM J. Comput.* 48, 1 (2019), 33–69. <https://doi.org/10.1137/17M1157957>
- [11] Richard Cole and Uzi Vishkin. 1986. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control* 70, 1 (1986), 32–53. [https://doi.org/10.1016/S0019-9958\(86\)80023-7](https://doi.org/10.1016/S0019-9958(86)80023-7)
- [12] Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. 2017. On the complexity of local distributed graph problems. In *Proc. 49th ACM SIGACT Symposium on Theory of Computing (STOC 2017)*. ACM Press, 784–797. <https://doi.org/10.1145/3055399.3055471>
- [13] Mohsen Ghaffari and Hsin-Hao Su. 2017. Distributed Degree Splitting, Edge Coloring, and Orientations. In *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*. Society for Industrial and Applied Mathematics, 2505–2523. <https://doi.org/10.1137/1.9781611974782.166>
- [14] Janne H Korhonen and Jukka Suomela. 2018. Towards a complexity theory for the congested clique. In *Proc. 30th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2018)*. ACM Press, 163–172. <https://doi.org/10.1145/3210377.3210391>
- [15] Nathan Linial. 1992. Locality in Distributed Graph Algorithms. *SIAM J. Comput.* 21, 1 (1992), 193–201. <https://doi.org/10.1137/0221015>
- [16] Moni Naor and Larry Stockmeyer. 1995. What Can be Computed Locally? *SIAM J. Comput.* 24, 6 (1995), 1259–1277. <https://doi.org/10.1137/S0097539793254571>
- [17] Dennis Olivetti. 2020. Round Eliminator: a tool for automatic speedup simulation. <https://github.com/olidennis/round-eliminator>
- [18] David Peleg. 2000. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898719772>
- [19] Will Rosenbaum and Jukka Suomela. 2020. Seeing Far vs. Seeing Wide: Volume Complexity of Local Graph Problems. In *Proc. 39th ACM Symposium on Principles of Distributed Computing (PODC 2020)*. ACM Press.
- [20] Václav Rozhoň and Mohsen Ghaffari. 2020. Polylogarithmic-Time Deterministic Network Decomposition and Distributed Derandomization. In *Proc. 52nd Annual ACM Symposium on Theory of Computing (STOC 2020)*. arXiv:1907.10937 <http://arxiv.org/abs/1907.10937>