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

*Published: 31/07/2020*

*Document Version*
Peer reviewed version

*Please cite the original version:*
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
eyfronyuv@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.

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.

---

We use the term relevant nodes. We emphasize that despite its bipartite appearance, the edge labeling formalism 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.

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 \( X \) 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>
<thead>
<tr>
<th>Type</th>
<th>Problem family</th>
<th>White constraint</th>
<th>Black constraint</th>
<th>Deterministic complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I.a 100*</td>
<td>0**</td>
<td>unsolvable</td>
<td></td>
</tr>
<tr>
<td>I.b 00' 1</td>
<td>**0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.c 0** 100*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.d **0 00' 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>II.a 000*</td>
<td>***</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II.b *** 000*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>III.a non-empty</td>
<td>111*</td>
<td>O(1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>III.b 111*</td>
<td>non-empty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>IV.a 1**1</td>
<td>1**</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IV.b **11</td>
<td>**1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>V.a 10'1</td>
<td>010</td>
<td>Θ(n)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V.b 010</td>
<td>10'1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VI</td>
<td>VI.a 0'1*</td>
<td><em>10</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VI.b +10*</td>
<td>0'1*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td>VII.a all other cases</td>
<td>Θ(log n)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 \( w_0w_1 \ldots w_d \) 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 \( \Pi \); therefore we do not need to specify \( d \) and \( \delta \) 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 \( 1 \)s, \( 0^* \) signifies one or more \( 0 \)s, 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


