

---

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; Kuhn, Fabian; Nowicki, Krzysztof; Olivetti, Dennis;  
Rotenberg, Eva; Suomela, Jukka

## **Brief Announcement: Local Advice and Local Decompression**

*Published in:*

PODC 2024 - Proceedings of the 2024 ACM Symposium on Principles of Distributed Computing

*DOI:*

[10.1145/3662158.3662805](https://doi.org/10.1145/3662158.3662805)

Published: 17/06/2024

*Document Version*

Publisher's PDF, also known as Version of record

*Published under the following license:*

CC BY

*Please cite the original version:*

Balliu, A., Brandt, S., Kuhn, F., Nowicki, K., Olivetti, D., Rotenberg, E., & Suomela, J. (2024). Brief Announcement: Local Advice and Local Decompression. In *PODC 2024 - Proceedings of the 2024 ACM Symposium on Principles of Distributed Computing* (pp. 117-120). ACM.  
<https://doi.org/10.1145/3662158.3662805>

---

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: Local Advice and Local Decompression

Alkida Balliu  
alkida.balliu@gssi.it  
Gran Sasso Science Institute (GSSI)  
L'Aquila, Italy

Sebastian Brandt  
brandt@cispa.de  
Helmholtz Center for Information  
Security (CISPA)  
Saarbrücken, Germany

Fabian Kuhn  
kuhn@cs.uni-freiburg.de  
University of Freiburg  
Freiburg, Germany

Krzysztof Nowicki  
krzysiek@pathway.com  
Pathway.com  
Wrocław, Poland

Dennis Olivetti  
dennis.olivetti@gssi.it  
Gran Sasso Science Institute (GSSI)  
L'Aquila, Italy

Eva Rotenberg  
erot@dtu.dk  
Technical University of Denmark  
Kongens Lyngby, Denmark

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

## ABSTRACT

In this work we study *local computation with advice*: the goal is to solve a graph problem  $\Pi$  with a distributed algorithm in  $f(\Delta)$  communication rounds, for some function  $f$  that only depends on the maximum degree  $\Delta$  of the graph, and the key question is how many bits of advice per node are needed. Our main results are:

- (1) Any *locally checkable labeling problem* (LCL) can be solved in graphs with *sub-exponential growth* with only 1 bit of advice per node. Moreover, we can make the set of nodes that carry advice bits arbitrarily sparse.
- (2) The assumption of sub-exponential growth is necessary: assuming the *Exponential-Time Hypothesis* (ETH), there are LCLs that cannot be solved in general with any constant number of bits per node.
- (3) In any graph we can find an *almost-balanced orientation* (indegrees and outdegrees differ by at most one) with 1 bit of advice per node, and again we can make the advice arbitrarily sparse.
- (4) As a corollary, we can also *compress an arbitrary subset of edges* so that a node of degree  $d$  stores only  $d/2 + 2$  bits, and we can *decompress* it locally, in  $f(\Delta)$  rounds.
- (5) In any graph of maximum degree  $\Delta$ , we can find a  $\Delta$ -coloring (if it exists) with 1 bit of advice per node, and again, we can make the advice arbitrarily sparse.
- (6) In any 3-colorable graph, we can find a 3-coloring with 1 bit of advice per node. Here, it remains open whether we can make the advice arbitrarily sparse.

## CCS CONCEPTS

• **Theory of computation** → **Distributed computing models; Distributed algorithms.**

## KEYWORDS

distributed advice, distributed decompression, locality

### ACM Reference Format:

Alkida Balliu, Sebastian Brandt, Fabian Kuhn, Krzysztof Nowicki, Dennis Olivetti, Eva Rotenberg, and Jukka Suomela. 2024. Brief Announcement: Local Advice and Local Decompression. In *ACM Symposium on Principles of Distributed Computing (PODC '24)*, June 17–21, 2024, Nantes, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3662158.3662805>

## 1 INTRODUCTION

Our work explores *what can and cannot be computed locally with the help of advice*. Our main focus is understanding advice in the context of classic local graph problems, such as vertex coloring.

While computation with different forms of advice has been explored in a wide range of distributed settings [10, 11, 16–20, 24, 26–28, 35–37, 39], there is hardly any prior work on solving classic local graph problems. A rare example of prior work is [14] from 2007, which studied the question of how much advice is necessary to break Linial's [33] lower bound for coloring cycles.

We initiate a systematic study of exactly how much advice is needed in the context of a wide range of graph problems. Let us first formalize the setting we study:

A graph problem  $\Pi$  can be *solved with  $\beta$  bits of advice* if there exists a  $T(\Delta)$ -round distributed algorithm  $\mathcal{A}$ , such that for any graph  $G$  that admits a solution to  $\Pi$ , there is an assignment of  $\beta$ -bit labels on vertices, such that the output of  $\mathcal{A}$  on the labeled graph is a solution to  $\Pi$ .

We will work in the usual LOCAL model of distributed computing. In an  $n$ -node graph, the nodes are labeled with unique identifiers from  $\{1, 2, \dots, \text{poly}(n)\}$ . We emphasize that the advice may depend



This work is licensed under a Creative Commons Attribution International 4.0 License.  
*PODC '24, June 17–21, 2024, Nantes, France*  
© 2024 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0668-4/24/06  
<https://doi.org/10.1145/3662158.3662805>

on the assignment of identifiers, and algorithm  $\mathcal{A}$  can freely make use of both the advice and the identifiers.

Now we seek to understand this question:

What is the smallest  $\beta$  such that  $\Pi$  can be solved with  $\beta$  bits of advice?

Note that if  $\Pi$  is, for example, the 3-coloring problem, it is trivial to solve with  $\beta = 2$  bits of advice per node, as we can directly encode the solution. The key question is how much better we can do.

Our work shows that for many problems the key threshold is not whether we can achieve, say, 1 bit of advice per node, but whether we can make the advice *arbitrarily sparse*, that is, make the ratio between 1s and 0s assigned to the nodes of the graph to be an arbitrarily small constant. This is particularly useful, as it enables us to *compose* multiple sparse advice schemes, so that it suffices to use just one bit per node in total. Hence, a large part of this work addresses the following question:

Which problems admit arbitrarily sparse advice?

We show that some problems can be solved with arbitrarily sparse advice. On the other hand, we also show that assuming the Exponential-Time Hypothesis, for any constant  $c$ , there exist problems that cannot be solved with  $c$  bits of advice. Finally, there are some problems like 3-coloring that can be solved with 1 bit of advice, but where it is not clear whether it can be solved with arbitrarily sparse advice. We discuss all of those points in what follows; we refer to the full version of this work [1] for more details.

## 2 LCLS IN BOUNDED-GROWTH GRAPHS

Locally checkable labeling problems (LCL), first introduced by Naor and Stockmeyer [38] in the 1990s, are one of the most extensively studied families of problems in the theory of distributed graph algorithms [2–5, 7–9, 13, 21, 23, 41]. These are graph problems that can be specified by giving a *finite* set of valid local neighborhoods. Many key problems such as vertex coloring, edge coloring, maximal independent set, maximal matching, sinkless orientation, and many other splitting and orientation problems are examples of LCLs, at least when restricted to bounded-degree graphs.

We show how to solve *any* LCL problem with just one bit of advice in graphs with a sub-exponential growth (the number of nodes in a radius- $r$  neighborhood is sub-exponential in  $r$ ):

Any LCL problem can be solved with 1 bit of advice per node in graphs with sub-exponential growth.

We also show that the encoding can be made arbitrarily sparse.

One prominent application of this result is its connection with *distributed proofs* [12, 29–32], and in particular with *locally checkable proofs* [25]. Consider any LCL  $\Pi$ . Assume that our task is to prepare a distributed proof that shows that in a graph  $G$  there exists a feasible solution of  $\Pi$  (for example, if  $\Pi$  is the task of 10-coloring, then the task is to certify that the chromatic number of  $G$  is at most 10). Now if  $G$  has sub-exponential growth, we can use our result to prepare a 1-bit advice that enables the algorithm to find a

solution of  $\Pi$ . Our advice is the proof: to verify it, we simply try to recover a solution with the help of the advice, and then check that the output is feasible in all local neighborhoods. We obtain the following corollary:

Any LCL problem admits a locally checkable proof with 1 bit per node in graphs with sub-exponential growth.

## 3 LCLS IN GENERAL GRAPHS

At this point a natural question is whether the assumption about bounded growth is necessary. Could we solve all LCL problems in all graphs with 1 bit of advice? We show that the answer is likely to be no:

Fix any  $\beta$ . If all LCL problems can be solved locally with at most  $\beta$  bits of advice, then the Exponential-Time Hypothesis (ETH) is false.

The intuition here is that if some LCL problem  $\Pi$  can be solved with, say, 1 bit of advice per node with some local algorithm  $\mathcal{A}$ , then we could solve it with a centralized sequential algorithm as follows: check all  $2^n$  possible assignments of advice, apply  $\mathcal{A}$  to decode the advice, and see if the solution is feasible. The total running time (from the centralized sequential perspective) would be  $2^n \cdot n \cdot s(n)$ , where  $s(n)$  is the time we need to simulate  $\mathcal{A}$  at one node. Then we need to show that assuming the Exponential-Time Hypothesis, this is too fast for some LCL problem  $\Pi$ . However, the key obstacle is that it may be computationally expensive to simulate  $\mathcal{A}$ , as it might perform arbitrarily complicated calculations that depend on the numerical values of the unique identifiers.

Hence, we need to show that  $\mathcal{A}$  can be made cheap to simulate. The key ingredient is the following technical result, which we prove using a Ramsey-type argument that is inspired by the proof of Naor and Stockmeyer [38]:

Assume that problem  $\Pi$  can be solved with  $\beta$  bits of advice per node, using some algorithm  $\mathcal{A}$ . Then the same problem can be also solved with  $\beta$  bits of advice using an *order-invariant* algorithm  $\mathcal{A}'$ , whose output does not depend on the numerical values of the identifiers but only on their relative order.

Now  $\mathcal{A}'$  can be represented as a finite lookup table; hence the simulation of  $\mathcal{A}'$  is cheap, and we can make a formal connection to the Exponential-Time Hypothesis.

## 4 BALANCED ORIENTATIONS

We now move on to a specific graph problem: we study the task of finding balanced and almost-balanced orientations. The goal is to orient the edges so that for each node indegree and outdegree differ by at most 1. This is a hard problem to solve in a distributed setting, while slightly more relaxed versions of the problem admit efficient (but not constant-time) algorithms [22].

Here it is good to note that if we could place our advice on *edges*, then trivially one bit of advice per edge would suffice (simply use the single bit to encode whether the edge is oriented from lower to

higher identifier). However, we are here placing advice on *nodes*, and encoding the orientation of each incident edge would require a number of bits proportional to the maximum degree. Surprisingly, we can do it, in any graph:

We can find almost-balanced orientations with 1 bit of advice per node.

Again, we can make the advice arbitrarily sparse.

## 5 DISTRIBUTED DECOMPRESSION

Equipped with the advice schema for solving almost-balanced orientations, we can now make a formal connection to what we call *distributed decompression*. Here the task is to encode some graph labeling so that it can be decompressed locally (in  $T(\Delta)$  rounds).

Local decompression is closely linked with local computation with advice. If we can compress some solution to  $\Pi$  with only  $\beta$  bits per node, and decompress it locally, then we can also solve  $\Pi$  with  $\beta$  bits of advice per node. Furthermore, if  $\Pi$  is a problem such that for any graph there is only one feasible solution, then the two notions coincide.

We will now show yet another connection between local decompression and local computation with advice. Consider the task of compressing an *arbitrary subset of edges*  $X \subseteq E$ . In a trivial encoding, we label each node  $v$  of degree  $d$  with a  $d$ -bit string that indicates which of the incident edges are present in  $X$ . On the other hand, we need a total of  $|E|$  bits in order to distinguish all subsets of the edge-set  $E$ . For  $d$ -regular graphs, this means we need at least  $d/2$  bits per node to recover an arbitrary subset of edges.

It turns out that once we can solve almost-balanced orientations, we can also compress a subset of edges efficiently. We simply use 1 bit of advice per node to encode an almost-balanced orientation. Now a node of degree  $d$  has outdegree  $\delta \leq \lceil d/2 \rceil$ , and it can simply store a  $\delta$ -bit vector that indicates which of its outgoing edges are in  $X$ . Overall, we will need  $\lceil d/2 \rceil + 1$ , i.e.  $\leq d/2 + 2$ , bits per node:

We can encode an arbitrary set of edges  $X \subseteq E$  so that a node of degree  $d$  only needs to store  $\lceil d/2 \rceil + 1$  bits, and we can decompress  $X$  locally, in  $T(\Delta)$  rounds.

## 6 VERTEX $\Delta$ -COLORING

Next we study the problem of finding a  $\Delta$ -coloring in graphs of maximum degree  $\Delta$ . Our main result there is:

In any graph of maximum degree  $\Delta$ , we can find a  $\Delta$ -coloring (if it exists) with 1 bit of advice per node.

Again, we can make the advice arbitrarily sparse.

Our schema for encoding  $\Delta$ -colorings consists of three steps. First, we compute a vertex coloring with  $O(\Delta^2)$  colors, with the help of advice. Then we reduce the number of colors down to  $\Delta + 1$ , using the algorithm by [6, 15, 34]. Finally, we follow the key idea of the algorithm by Panconesi and Srinivasan [40] to turn  $(\Delta + 1)$ -coloring into a  $\Delta$ -coloring, and again we will need some advice to make this part efficient.

## 7 VERTEX 3-COLORING

So far we have seen primarily results of two flavors: many problems can be solved with 1 bit of advice so that we can make the advice arbitrarily sparse, while there are also some problems that require arbitrarily many bits of advice.

We now turn our attention to a problem that seems to lie right at the boundary of what can be done with only 1 bit per node: vertex 3-coloring in any 3-colorable graph. Note that this is a problem that is hard to solve without advice not only in the distributed setting (it is a global problem) but also in the centralized setting (it is an NP-hard problem).

In the centralized setting, 1 bit of advice per node makes the problem easy: we can simply use the bit to indicate which nodes are of color 3. Then the rest of the graph has to be bipartite, and we can simply find a proper 2-coloring in polynomial time.

In the distributed setting, the trivial solution does not work: 2-coloring in bipartite graphs is still a global problem. Nevertheless, we show that 3-coloring is still doable with 1 bit of advice:

In any 3-colorable graph, we can find a 3-coloring with 1 bit of advice per node.

Our encoding essentially uses one bit to encode one of the color classes, but we adjust the encoding slightly so that throughout the graph there are *local hints* that help us to also choose the right parity for the region that we need to 2-color.

Here, our encoding genuinely needs one bit per node (it just barely suffices); we cannot make our advice arbitrarily sparse.

## 8 COMPOSABILITY FRAMEWORK

A key technique that we use in many of our algorithms is the framework of *composable schemas*. While the definition is a bit technical, it has two key properties:

- (1) As the name suggests, composable schemas can be easily composed, in the following sense: once we have (1) a composable schema for solving  $\Pi_1$  and (2) a composable schema for solving  $\Pi_2$  assuming an oracle for  $\Pi_1$ , we can also compose them and obtain (3) a composable schema that solves  $\Pi_2$  without the oracle. This way we can solve problems in a modular fashion, in essence using schemas as “subroutines.”
- (2) A composable schema can be then encoded with only 1 bit of advice per node, and we can make the advice arbitrarily sparse.

For example, our algorithms for finding almost-balanced orientations and  $\Delta$ -coloring with advice are based on the framework of composable schemas. We refer to the full version of this work [1] for more details.

## ACKNOWLEDGMENTS

This work was initiated at Dagstuhl Seminar 23071: “From Big Data Theory to Big Data Practice”. We would like to thank Sameep Dahal and Augusto Modanese for insightful discussions, and the anonymous reviewers for their helpful comments on previous versions of this work. This work was supported by the IRFD grant 2020-2023 (9131-00044B) and the VILLUM Foundation grant (VIL37507).

## REFERENCES

- [1] Alkida Balliu, Sebastian Brandt, Fabian Kuhn, Krzysztof Nowicki, Dennis Olivetti, Eva Rotenberg, and Jukka Suomela. 2024. Local Advice and Local Decompression. arXiv:2405.04519 [cs.DC]
- [2] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. 2020. How much does randomness help with locally checkable problems?. In *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, Yuval Emek and Christian Cachin (Eds.). ACM, 299–308. <https://doi.org/10.1145/3382734.3405715>
- [3] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. 2021. Almost global problems in the LOCAL model. *Distributed Computing* 34, 4 (2021), 259–281. <https://doi.org/10.1007/S00446-020-00375-2>
- [4] Alkida Balliu, Keren Censor-Hillel, Yannic Maus, Dennis Olivetti, and Jukka Suomela. 2021. Locally Checkable Labelings with Small Messages. In *35th International Symposium on Distributed Computing, DISC 2021, October 4-8, 2021, Freiburg, Germany (Virtual Conference) (LIPIcs, Vol. 209)*, Seth Gilbert (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 8:1–8:18. <https://doi.org/10.4230/LIPIcs.DISC.2021.8>
- [5] Alkida Balliu, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Dennis Olivetti, and Jukka Suomela. 2018. New classes of distributed time complexity. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, Ilias Diakonikolas, David Kempe, and Monika Henzinger (Eds.). ACM, 1307–1318. <https://doi.org/10.1145/3188745.3188860>
- [6] Leonid Barenboim, Michael Elkin, and Uri Goldenberg. 2022. Locally-iterative Distributed  $(\Delta + 1)$ -coloring and Applications. *J. ACM* 69, 1 (2022), 5:1–5:26. <https://doi.org/10.1145/3486625>
- [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 *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, Daniel Wichs and Yishay Mansour (Eds.). ACM, 479–488. <https://doi.org/10.1145/2897518.2897570>
- [8] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. 2019. An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model. *SIAM J. Comput.* 48, 1 (2019), 122–143. <https://doi.org/10.1137/17M1117537>
- [9] 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>
- [10] Dariusz Dereniowski and Andrzej Pelc. 2012. Drawing maps with advice. *J. Parallel and Distrib. Comput.* 72, 2 (2012), 132–143. <https://doi.org/10.1016/J.JPDC.2011.10.004>
- [11] Stefan Dobrev, Rastislav Kráľovic, and Euripides Markou. 2012. Online Graph Exploration with Advice. In *Structural Information and Communication Complexity - 19th International Colloquium, SIROCCO 2012, Reykjavik, Iceland, June 30-July 2, 2012, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 7355)*, Guy Even and Magnús M. Halldórsson (Eds.). Springer, 267–278. [https://doi.org/10.1007/978-3-642-31104-8\\_23](https://doi.org/10.1007/978-3-642-31104-8_23)
- [12] Laurent Feuilloley and Pierre Fraigniaud. 2017. Survey of Distributed Decision. <https://doi.org/10.48550/arXiv.1606.04434> arXiv:1606.04434 [cs.DC]
- [13] Manuela Fischer and Mohsen Ghaffari. 2017. Sublogarithmic Distributed Algorithms for Lovász Local Lemma, and the Complexity Hierarchy. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria (LIPIcs, Vol. 91)*, Andréa W. Richa (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 18:1–18:16. <https://doi.org/10.4230/LIPIcs.DISC.2017.18>
- [14] Pierre Fraigniaud, Cyril Gavoille, David Ilcinkas, and Andrzej Pelc. 2009. Distributed computing with advice: information sensitivity of graph coloring. *Distributed Computing* 21, 6 (2009), 395–403. <https://doi.org/10.1007/S00446-008-0076-Y>
- [15] Pierre Fraigniaud, Marc Heinrich, and Adrian Kosowski. 2016. Local Conflict Coloring. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, Irit Dinur (Ed.). IEEE Computer Society, 625–634. <https://doi.org/10.1109/FOCS.2016.73>
- [16] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. 2008. Tree exploration with advice. *Information and Computation* 206, 11 (2008), 1276–1287. <https://doi.org/10.1016/J.IC.2008.07.005>
- [17] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. 2010. Communication algorithms with advice. *J. Comput. System Sci.* 76, 3-4 (2010), 222–232. <https://doi.org/10.1016/J.JCSS.2009.07.002>
- [18] Pierre Fraigniaud, Amos Korman, and Emmanuelle Lebhar. 2007. Local MST computation with short advice. In *SPAA 2007: Proceedings of the 19th Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Diego, California, USA, June 9-11, 2007*, Phillip B. Gibbons and Christian Scheideler (Eds.). ACM, 154–160. <https://doi.org/10.1145/1248377.1248402>
- [19] Emanuele G. Fusco and Andrzej Pelc. 2011. Trade-offs between the Size of Advice and Broadcasting Time in Trees. *Algorithmica* 60, 4 (2011), 719–734. <https://doi.org/10.1007/S00453-009-9361-9>
- [20] Emanuele G. Fusco, Andrzej Pelc, and Rossella Petreschi. 2016. Topology recognition with advice. *Information and Computation* 247 (2016), 254–265. <https://doi.org/10.1016/J.IC.2016.01.005>
- [21] Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. 2018. On Derandomizing Local Distributed Algorithms. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, Mikkel Thorup (Ed.). IEEE Computer Society, 662–673. <https://doi.org/10.1109/FOCS.2018.00069>
- [22] Mohsen Ghaffari, Juho Hirvonen, Fabian Kuhn, Yannic Maus, Jukka Suomela, and Jara Uitto. 2020. Improved distributed degree splitting and edge coloring. *Distributed Computing* 33, 3-4 (2020), 293–310. <https://doi.org/10.1007/S00446-018-00346-8>
- [23] Mohsen Ghaffari and Hsin-Hao Su. 2017. Distributed Degree Splitting, Edge Coloring, and Orientations. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, Philip N. Klein (Ed.). SIAM, 2505–2523. <https://doi.org/10.1137/1.9781611974782.166>
- [24] Christian Glacet, Avery Miller, and Andrzej Pelc. 2017. Time vs. Information Tradeoffs for Leader Election in Anonymous Trees. *ACM Transactions on Algorithms* 13, 3 (2017), 31:1–31:41. <https://doi.org/10.1145/3039870>
- [25] Mika Göös and Jukka Suomela. 2016. Locally Checkable Proofs in Distributed Computing. *Theory of Computing* 12, 1 (2016), 1–33. <https://doi.org/10.4086/TOC.2016.V012A019>
- [26] Barun Gorain and Andrzej Pelc. 2019. Deterministic Graph Exploration with Advice. *ACM Transactions on Algorithms* 15, 1 (2019), 8:1–8:17. <https://doi.org/10.1145/3280823>
- [27] David Ilcinkas, Dariusz R. Kowalski, and Andrzej Pelc. 2010. Fast radio broadcasting with advice. *Theoretical Computer Science* 411, 14-15 (2010), 1544–1557. <https://doi.org/10.1016/J.TCS.2010.01.004>
- [28] Dennis Komm, Rastislav Kráľovic, Richard Kráľovic, and Jasmin Smula. 2015. Treasure Hunt with Advice. In *Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings (Lecture Notes in Computer Science, Vol. 9439)*, Christian Scheideler (Ed.). Springer, 328–341. [https://doi.org/10.1007/978-3-319-25258-2\\_23](https://doi.org/10.1007/978-3-319-25258-2_23)
- [29] Amos Korman and Shay Kutten. 2006. On Distributed Verification. In *Distributed Computing and Networking, 8th International Conference, ICDCN 2006, Guwahati, India, December 27-30, 2006 (Lecture Notes in Computer Science, Vol. 4308)*, Soma Chaudhuri, Samir R. Das, Himadri S. Paul, and Srikanta Tirathapura (Eds.). Springer, 100–114. [https://doi.org/10.1007/11947950\\_12](https://doi.org/10.1007/11947950_12)
- [30] Amos Korman and Shay Kutten. 2007. Distributed verification of minimum spanning trees. *Distributed Computing* 20, 4 (2007), 253–266. <https://doi.org/10.1007/S00446-007-0025-1>
- [31] Amos Korman, Shay Kutten, and David Peleg. 2010. Proof labeling schemes. *Distributed Computing* 22, 4 (2010), 215–233. <https://doi.org/10.1007/S00446-010-0095-3>
- [32] Amos Korman, David Peleg, and Yoav Rodeh. 2010. Constructing Labeling Schemes through Universal Matrices. *Algorithmica* 57, 4 (2010), 641–652. <https://doi.org/10.1007/S00453-008-9226-7>
- [33] Nathan Linial. 1992. Locality in Distributed Graph Algorithms. *SIAM J. Comput.* 21, 1 (1992), 193–201. <https://doi.org/10.1137/0221015>
- [34] Yannic Maus and Tigran Tonoyan. 2022. Linial for lists. *Distributed Computing* 35, 6 (2022), 533–546. <https://doi.org/10.1007/S00446-022-00424-Y>
- [35] Avery Miller and Andrzej Pelc. 2015. Fast rendezvous with advice. *Theoretical Computer Science* 608 (2015), 190–198. <https://doi.org/10.1016/J.TCS.2015.09.025>
- [36] Avery Miller and Andrzej Pelc. 2015. Tradeoffs between cost and information for rendezvous and treasure hunt. *J. Parallel and Distrib. Comput.* 83 (2015), 159–167. <https://doi.org/10.1016/J.JPDC.2015.06.004>
- [37] Avery Miller and Andrzej Pelc. 2016. Election vs. Selection: How Much Advice is Needed to Find the Largest Node in a Graph?. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016*, Christian Scheideler and Seth Gilbert (Eds.). ACM, 377–386. <https://doi.org/10.1145/2935764.2935772>
- [38] Moni Naor and Larry J. Stockmeyer. 1995. What Can be Computed Locally? *SIAM J. Comput.* 24, 6 (1995), 1259–1277. <https://doi.org/10.1137/S0097539793254571>
- [39] Nicolas Nisse and David Soguet. 2009. Graph searching with advice. *Theoretical Computer Science* 410, 14 (2009), 1307–1318. <https://doi.org/10.1016/J.TCS.2008.08.020>
- [40] Alessandro Panconesi and Aravind Srinivasan. 1992. Improved Distributed Algorithms for Coloring and Network Decomposition Problems. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis (Eds.). ACM, 581–592. <https://doi.org/10.1145/129712.129769>
- [41] Václav Rozhn and Mohsen Ghaffari. 2020. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (Eds.). ACM, 350–363. <https://doi.org/10.1145/3357713.3384298>