Sinkless orientation is hard also in the supported LOCAL model

Janne H. Korhonen · janne.korhonen@ist.ac.at
IST Austria

Ami Paz · ami.paz@univie.ac.at
Faculty of Computer Science, Universität Wien, Austria

Joel Rybicki · joel.rybicki@ist.ac.at
IST Austria

Stefan Schmid · stefan_schmid@univie.ac.at
Faculty of Computer Science, Universität Wien, Austria &
Fraunhofer Institute for Secure Information Technology (SIT), Germany

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

arXiv:2108.02655v1 [cs.DC] 5 Aug 2021

Abstract. We show that any algorithm that solves the sinkless orientation problem in the supported LOCAL model requires $\Omega(\log n)$ rounds, and this is tight. The supported LOCAL is at least as strong as the usual LOCAL model, and as a corollary this also gives a new, short and elementary proof that shows that the round complexity of the sinkless orientation problem in the deterministic LOCAL model is $\Omega(\log n)$.

1 Introduction

Sinkless orientations. In the *sinkless orientation problem*, the task is to orient the edges of a graph such that all nodes of degree at least 3 have out-degree at least 1. The problem is always solvable, and easy to solve in the centralized setting, but a lot more challenging to solve efficiently in distributed or parallel settings.

The sinkless orientation problem is the canonical example of a problem that has round complexity $\Theta(\log \log n)$ rounds in the randomized LOCAL model but $\Theta(\log n)$ rounds in the deterministic LOCAL model. It is a rare example of a *locally checkable problem* in which randomness helps exponentially, and also an example of a locally checkable problem with an *intermediate* complexity—not solvable in $O(\log^* n)$ rounds but solvable in sub-diameter time.

Supported LOCAL model. While the complexity landscape in the usual LOCAL model is nowadays well understood [1, 3–5, 7, 8, 10, 11], and also some *weaker* models of distributed computing have been already explored [2,9,14], it has been wide open how the landscape changes when we switch to *stronger* models of computing. In this work we focus on the *supported* LOCAL model, which is strictly stronger than the LOCAL model.

In the supported LOCAL model [12, 15], the communication network $G = (V, E)$ and the unique identifiers are all known to all nodes, and the input is a subgraph H of G. That is, each node v receives as input *the entire structure* of the communication network G, including all the unique identifiers, and a list of its incident edges in H; we refer to the latter edges as *input edges*. Otherwise, the computation proceeds as in the standard LOCAL model, using all edges of G for communication. In our case, we would like to find a sinkless orientation in the input graph H.

The availability of the underlying globally-known communication graph G (a.k.a. the *support*) helps a lot with many problems. For example, all locally checkable problems with complexity $O(\log^* n)$ admit constant-time algorithms in the supported model—in essence, the support can be used to *break symmetry for free* [12]. Also if we had the promise that the support G is a *tree*, then the sinkless orientation problem would become trivial: we can orient all edges of G towards a leaf, and this orientation is also a valid orientation for any subgraph H. However, in this work we show that this trick only works in trees—we show that if, for example, G is a 5-regular graph, then the support is essentially useless.

The supported LOCAL model was originally introduced in the context of *software-defined networks* (SDNs). The underlying idea is that the communication graph G represents the unchanging physical network, and the input graph represents the logical state of the network to which the control plane (here, distributed algorithm) needs to respond to; see reference [15] for more details. However, supported LOCAL have proven to be useful as a purely theoretical model for lower bounds (this work, Reference [12], and the very recent work of Haeupler et al. [13]).

Our contributions. We prove that the round complexity of the sinkless orientation problem in the deterministic supported LOCAL model is $\Omega(\log n)$ rounds. By prior work, we also know that this is tight: the problem is solvable in $O(\log n)$ rounds (with or without support). Furthermore, the same problem can be solved in the randomized supported LOCAL model in $O(\log \log n)$ rounds.

In particular, we learn that in the supported LOCAL model there are locally checkable problems in which randomness helps exponentially. As a corollary, we cannot use the support to efficiently derandomize algorithms.

The classic proof for the LOCAL model. As a by-product, our work gives a new, short and elementary proof that shows that the round complexity of the sinkless orientation problem in the deterministic LOCAL model is $\Omega(\log n)$.

The standard proof is somewhat long and complicated. It builds on the *round elimination* technique [1, 7, 8], but round elimination has so far been unable to handle unique identifiers. Hence in prior work one has always taken a detour: first prove an $\Omega(\log \log n)$ lower bound in the randomized model without unique identifiers [8], and then apply the *deterministic gap result* of Chang et al. [10] to derive a deterministic $\Omega(\log n)$ lower bound.

Besides being complicated, the standard proof seems to be also fundamentally incompatible with the supported LOCAL model. Hence until now it has remained open what is the computational complexity of the sinkless orientation in the supported LOCAL model, and more generally whether there are any problems in the supported model where randomness helps exponentially.

Our new proof for the supported LOCAL model. By switching to the supported LOCAL model, we can give a direct proof without any detours through randomness and gap results. We directly show with elementary arguments that the complexity of sinkless orientation is $\Omega(\log n)$, both in the usual LOCAL model and also in the supported LOCAL model.

The underlying idea is, in essence, the same as the *ID graph technique* from the very recent work by Brandt et al. [9]. The ID graph in their work plays a role similar to the support in our work. However, the details differ, as the goals are different. Brandt et al. aimed at proving a lower bound for randomized local computation algorithms, while our aim is at proving a lower bound for deterministic supported LOCAL algorithms.

The key technical difference is that ID graphs [9] need to have a large chromatic number, while our proof goes through even if the support is a bipartite graph. On the other hand, we need to do more work in the base case when we argue that 0-round algorithms do not exist.

2 Sinkless orientation lower bound

Roadmap. For technical convenience, we prove the result in a stronger *bipartite* version of the model. The lower bound in this setting then implies lower bounds for (non-bipartite) LOCAL and supported LOCAL, by observing that algorithms from a weaker model can be translated to the stronger models with no overhead in round complexity.

The overall structure of our lower bound proof is as follows. We fix a bipartite 5-regular graph G with girth $\Omega(\log n)$, and an assignment of unique identifiers on G. We then show that in bipartite supported LOCAL, any algorithm that solves sinkless orientation, even with the promise that the support graph is G, requires $\Omega(\log n)$ rounds.

The proof has two main steps. First, we give a round elimination lemma showing that any T-round sinkless orientation algorithm on H can be converted into a $(T-1)$-round algorithm, as long as T sufficiently less than the girth of G. By iterating this lemma, we can thus turn a T-round algorithm into a 0-round algorithm. Second, we show that no such 0-round algorithm can exist, implying that any algorithm requires $\Omega(\log n)$ rounds.

2.1 Setup

Bipartite model. In bipartite supported LOCAL, we are given a promise that the support graph G is bipartite, and a 2-coloring is given to the nodes as an input; we refer to the two colors as black and white. In the bipartite model, we consider either the black or white nodes to be *active*, and the other color to be *passive*. All nodes of the graph run an algorithm as per supported LOCAL model; upon termination of the algorithm, the active nodes produce an output, and the passive nodes output nothing. The outputs of the active nodes must form a globally valid solution; in particular, in sinkless orientation, the outputs of the active nodes already orients all edges, and both active and passive nodes must not be sinks.

Sinkless orientation in bipartite model. We encode sinkless orientation in the bipartite supported LOCAL model as follows. Each active node outputs, for each incident input edge, one label from the alphabet $\Sigma = \{\mathsf{O}, \mathsf{I}\}$. The edge-output O indicates that the edge is outgoing from the active node, and the edge-output I indicates it is incoming to the active node. An output is correct if for each active node of degree at least 3, there is at least one output O on an incident input edge, and for each passive node of degree at least 3, there is at least one output I on an incident input edge. Note that the labels O, I represent orientation w.r.t. the active node, and we require each active node to have at least one O label (indicating an outgoing edge), and each passive node to have at least one I label (indicating an edge incoming to an active neighbor, thus outgoing from the passive node we consider). Hence, any solution on general graphs can immediately be translated to a solution in the bipartite model.

In more detail, consider a sinkless orientation algorithm A running in T rounds in (supported) LOCAL model, with some reasonable output encoding. To turn this into a bipartite (supported) LOCAL algorithm, one first runs algorithm A in the bipartite model—this requires no modifications, as computation in the bipartite model is done exactly as in the original. After A has terminated, (1) the passive nodes discard the output of A and output nothing, and (2) the active nodes inspect the output of A, and output I for each incident edge directed towards them, and O for each edge directed away from them in the output of A. Since A is a sinkless orientation algorithm, these outputs also guarantee that each passive node has one edge with output I incident to it. In particular, it follows that lower bounds for bipartite algorithms are also lower bounds for the standard models.

2.2 Step one: Round elimination

Lemma 1. *Let G be a fixed 5-regular bipartite graph with girth g, and fixed unique identifiers and 2-coloring of the nodes. Let $0 < T < g/2$, and assume there is a T-round algorithm $\mathcal{A}_T$ that solves sinkless orientation on G. Then there is a $(T-1)$-round algorithm $\mathcal{A}_{T-1}$ that solves sinkless orientation on G.*

Proof. The proof proceeds by the standard round elimination strategy. Let us assume without loss of generality that black nodes are active in $\mathcal{A}_T$. For a non-negative integer t and any node $v \in V$, let us denote by $B(v,t)$ the nodes within distance t from the node v in the graph G.

We construct an algorithm $\mathcal{A}_{T-1}$ where white nodes are active. In algorithm $\mathcal{A}_{T-1}$, each white node $u \in V$ performs the following steps:

(1) Node u gathers the inputs in its $(T-1)$-radius neighborhood $B(u, T-1)$.

(2) For each neighbor v of u, the node u enumerates all possible input graphs H' on $B(v,T)$ that are compatible with the actual input graph H on $B(u, T-1)$. For each such H', u simulates $\mathcal{A}_T$ to compute what output v would output on the edge $\{u, v\}$ under input H'. Let $S(u,v)$ denote the set of all possible outputs obtained for edge $\{u,v\}$ this way.

(3) If $S(u,v) = \{\mathsf{I}\}$ then u outputs O on $\{u,v\}$, and otherwise it outputs I on it.

We now prove $\mathcal{A}_{T-1}$ produces a valid solution for sinkless orientation.

Consider a white node u, and its two neighbors v, v' in H. Since $T < g/2$, we have $B(v,T) \cap B(v',T) = B(u, T-1)$, and thus the inputs in $B(v,T) \setminus B(u,T-1)$ do not affect the output of v' in $\mathcal{A}_T$, and likewise the inputs in $B(v',T) \setminus B(u,T-1)$ do not affect the output of v in $\mathcal{A}_T$. Thus, any combination of $\mathsf{L}_v \in S(u,v)$ and $\mathsf{L}_{v'} \in S(u,v')$ may occur as an output: for any such L_v and $\mathsf{L}_{v'}$, there is an input graph such that v in $\mathcal{A}_T$ outputs L_v for the edge $\{u,v\}$, and v' outputs $\mathsf{L}_{v'}$ for the edge $\{u,v'\}$.

Let u be a white node of degree at least 3 in H with neighbors $N(u)$ in H. By the above argument, for any choice of one $\mathsf{L}_v \in S(u,v)$ for each neighbor $v \in N(u)$ of u, there is an input graph on which $\mathcal{A}_T$ outputs L_v for the edge $\{u,v\}$. If each $S(u,v)$ contains O, there would be an input graph on which $\mathcal{A}_T$ outputs O for all incident input edges of u, rendering it incorrect. Hence, at least one neighbor v' of u satisfies $S(u,v') = \{\mathsf{I}\}$, and in $\mathcal{A}_{T-1}$ where u is active, u outputs O on the edge $\{u,v'\}$.

On the other hand, consider black node v of degree at least 3 with neighbors $N(v)$ in H. On the true input H node v in $\mathcal{A}_T$ will output O on an incident edge $\{v,u\}$, for some $u \in N(v)$. In $\mathcal{A}_{T-1}$, the node u will consider the input H on $B(v,T)$ (among other inputs), so we have $\mathsf{O} \in S(u,v)$. Thus, in $\mathcal{A}_{T-1}$ the node u will output I on $\{v,u\}$, and v has an incident edge labeled I as desired. ◻

2.3 Step two: There exists no 0-round algorithm

Lemma 2. *Let G be a fixed 5-regular bipartite graph with girth g, and assume unique identifiers and 2-coloring on G are fixed. There is no algorithm solving sinkless orientation in bipartite supported* LOCAL *in 0 rounds on G.*

Proof. Assume for contradiction that there is a 0-round algorithm $\mathcal{A}_0$ with black nodes as active. Label each edge e of G by the set of all outputs $\mathcal{A}_0$ can output for e when e is part of the input. For any black node v, there must be at least three edges labeled with either $\{\mathsf{O}\}$ or $\{\mathsf{O}, \mathsf{I}\}$, as otherwise, for some input v would have exactly three incident input edges on which it would output I.

Since every edge is incident to exactly one black node, at most 2/5 of the edges are labeled $\{\mathsf{I}\}$. Hence, there is a white node u such that u is incident to at least three edges $\{u, v_1\}$, $\{u, v_2\}$

and $\{u, v_3\}$ labeled with either $\{\mathsf{O}\}$ or $\{\mathsf{O}, \mathsf{I}\}$. Now consider an input where these three edges are the only input edges incident to u. Since the output of each node v_i depends only on its incident input edges, we can select for each v_i an input where v_i outputs O for edge $\{u, v_i\}$. Moreover, since $\mathcal{A}_0$ is a 0-round algorithm and nodes v_1, v_2 and v_3 are not neighbors, we can do this for all of them simultaneously. Thus, there exists an input where $\mathcal{A}_0$ outputs O on all incident input edges of the passive node u, a contradiction. □

2.4 Putting things together

Theorem 3. *Any deterministic algorithm solving sinkless orientation in the supported* LOCAL *model requires* $\Omega(\log n)$ *rounds.*

Proof. Let G be a bipartite 5-regular graph with girth $g = \Omega(\log n)$. Observe that we can obtain one e.g. by taking the bipartite double cover of any 5-regular graph of girth $\Omega(\log n)$, which are known to exist (see e.g., [6, Ch. 3]).

Assume that there is a supported LOCAL algorithm $\mathcal{A}_T$ that solves sinkless orientation in $T < g/2$ rounds on communication graph G. This implies that there is a bipartite supported LOCAL algorithm for sinkless orientation on G running in time T. By repeated application of Lemma 1, there is a sequence of bipartite supported LOCAL algorithms

$$\mathcal{A}_T, \mathcal{A}_{T-1}, \ldots, \mathcal{A}_1, \mathcal{A}_0\,,$$

where algorithm $\mathcal{A}_i$ solves sinkless orientation in i rounds.

In particular, $\mathcal{A}_0$ solves sinkless orientation in 0 rounds. By Lemma 2, this is impossible, so algorithm $\mathcal{A}_T$ cannot exist. □

Corollary 4. *Any deterministic algorithm solving sinkless orientation in the* LOCAL *model requires* $\Omega(\log n)$ *rounds.*

Proof. Any LOCAL algorithm $\mathcal{A}$ that runs in $T(n)$ rounds can be simulated in supported LOCAL in $T(n)$ rounds by ignoring non-input edges: simply run $\mathcal{A}$ on the input graph H. Thus, the claim follows immediately from Theorem 3. □

Acknowledgements

This research is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 805223 ScaleML), the Austrian Science Fund (FWF) and netIDEE SCIENCE project P 33775-N, and the Austrian Science Fund (FWF) project ADVISE, I 4800-N, 2020-2023.

References

[1] Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower bounds for maximal matchings and maximal independent sets. In *Proc. 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*, 2019.

[2] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, Jan Studený, Jukka Suomela, and Aleksandr Tereshchenko. Locally checkable problems in rooted trees. In *Proc. 40th ACM Symposium on Principles of Distributed Computing (PODC 2021)*, 2021.

[3] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. Almost global problems in the LOCAL model. In *Proc. 32nd International Symposium on Distributed Computing (DISC 2018)*, 2018. doi:10.4230/LIPIcs.DISC.2018.9.

[4] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. How much does randomness help with locally checkable problems? In *Proc. 39th ACM Symposium on Principles of Distributed Computing (PODC 2020)*, 2020. doi:10.1145/3382734.3405715.

[5] Alkida Balliu, Juho Hirvonen, Janne H Korhonen, Tuomo Lempiäinen, Dennis Olivetti, and Jukka Suomela. New classes of distributed time complexity. In *Proc. 50th ACM Symposium on Theory of Computing (STOC 2018)*, 2018. doi:10.1145/3188745.3188860.

[6] Béla Bollobás. *Extremal graph theory.* Courier Corporation, 2004.

[7] Sebastian Brandt. An Automatic Speedup Theorem for Distributed Problems. In *Proc. 38th ACM Symposium on Principles of Distributed Computing (PODC 2019)*, 2019. doi:10.1145/3293611.3331611.

[8] Sebastian Brandt, Orr Fischer, Juho Hirvonen, Barbara Keller, Tuomo Lempiäinen, Joel Rybicki, Jukka Suomela, and Jara Uitto. A lower bound for the distributed Lovász local lemma. In *Proc. 48th ACM Symposium on Theory of Computing (STOC 2016)*, 2016. doi:10.1145/2897518.2897570.

[9] Sebastian Brandt, Christoph Grunau, and Václav Rozhoň. The randomized local computation complexity of the Lovász local lemma. In *Proc. 40th ACM Symposium on Principles of Distributed Computing (PODC 2021)*, 2021.

[10] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model. In *Proc. 57th IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, 2016. doi:10.1109/FOCS.2016.72.

[11] Yi-Jun Chang and Seth Pettie. A Time Hierarchy Theorem for the LOCAL Model. *SIAM Journal on Computing*, 48(1):33–69, 2019. doi:10.1137/17M1157957.

[12] Klaus-Tycho Foerster, Juho Hirvonen, Stefan Schmid, and Jukka Suomela. On the Power of Preprocessing in Decentralized Network Optimization. In *Proc. IEEE Conference on Computer Communications (INFOCOM 2019)*, 2019. doi:10.1109/INFOCOM.2019.8737382.

[13] Bernhard Haeupler, David Wajc, and Goran Zuzic. Universally-optimal distributed algorithms for known topologies. In *STOC*, pages 1166–1179. ACM, 2021.

[14] Will Rosenbaum and Jukka Suomela. Seeing Far vs. Seeing Wide: Volume Complexity of Local Graph Problems. In *Proc. 39th ACM Symposium on Principles of Distributed Computing (PODC 2020)*, 2020. doi:10.1145/3382734.3405721.

[15] Stefan Schmid and Jukka Suomela. Exploiting locality in distributed SDN control. In *Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 2013)*, pages 121–126. ACM Press, 2013. doi:10.1145/2491185.2491198.