A Lower Bound for the Distributed Lovász Local Lemma

Sebastian Brandt
ETH Zurich,
Switzerland

Orr Fischer
School of Computer Science,
Tel Aviv University,
Israel

Juho Hirvonen
HIIT, Department of Computer
Science, Aalto University,
Finland

Barbara Keller
ETH Zurich,
Switzerland

Tuomo Lempiäinen
HIIT, Department of Computer
Science, Aalto University,
Finland

Joel Rybicki
HIIT, Department of Computer
Science, Aalto University,
Finland

Jukka Suomela
HIIT, Department of Computer
Science, Aalto University,
Finland

Jara Uitto
Bitsplitters GmbH,
Switzerland

ABSTRACT

We show that any randomised Monte Carlo distributed algorithm for the Lovász local lemma requires $\Omega(\log \log n)$ communication rounds, assuming that it finds a correct assignment with high probability. Our result holds even in the special case of $d \in O(1)$, where d is the maximum degree of the dependency graph. By prior work, there are distributed algorithms for the Lovász local lemma with a running time of $O(\log n)$ rounds in bounded-degree graphs, and the best lower bound before our work was $\Omega(\log^* n)$ rounds [Chung et al. 2014].

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.3 [**Mathematics of Computing**]: Probability and Statistics—*probabilistic algorithms (including Monte Carlo)*

General Terms

Theory, Algorithms

Keywords

Lovász local lemma, distributed complexity, lower bounds, locality, graph colouring, sinkless orientations

1. INTRODUCTION

In this work, we give a lower bound for the constructive Lovász local lemma (LLL) in the context of distributed algorithms. We study the running time as a function of n (the number of events), and prove a lower bound that holds even if d (the maximum degree of the dependency graph) is bounded by a constant. By prior work, there are distributed algorithms for LLL with a running time of $O(\log n)$ communication rounds in this case and $o(\log n)$ rounds for restricted variants [11], and it is known that any distributed algorithm for LLL requires $\Omega(\log^* n)$ rounds [11, 28, 35]. We prove a new lower bound of $\Omega(\log \log n)$ rounds.

1.1 Distributed Lovász Local Lemma

Recall the following symmetric version of LLL:

THEOREM 1. *Let $\mathcal{E} = \{E_1, \ldots, E_n\}$ be a finite set of events such that each E_i is independent of all but d other events. If $\Pr[E_i] \le p$ and $ep(d+1) \le 1$, then there is a positive probability that none of the events occur.*

We consider *distributed* algorithmic variants of LLL. The basic framework is as follows. Let $\mathcal{X} = \{X_1, \ldots, X_m\}$ be a set of mutually independent random variables and assume that each E_i depends only on variables in $\mathcal{X}$; denote by $\mathrm{vbl}(E_i) \subseteq \mathcal{X}$ the subset of variables that event E_i depends on. Form the *dependency graph* $G_{\mathcal{E}} = (\mathcal{E}, \mathcal{D})$, where $\mathcal{D} = \{\{E_i, E_j\} : \mathrm{vbl}(E_i) \cap \mathrm{vbl}(E_j) \neq \emptyset\}$. Now consider a distributed system in which the communication network is identical to the graph $G_{\mathcal{E}}$: each node of the system is associated with a bad event $E \in \mathcal{E}$, and two nodes are adjacent if and only if their associated events depend on at least one common variable. The task is for each node to find an assignment to its variables $\mathrm{vbl}(E)$ such that adjacent nodes agree on the values of their common variables and all the events in $\mathcal{E}$ are avoided.

We use the standard LOCAL model of distributed computing [28, 37]. Initially each node is only aware of its own part of the input and the number of nodes, but the nodes can exchange messages to learn more about the structure of the problem instance. Eventually, each node has to stop and output its own part of the variable assignment. Communication takes place in synchronous communication rounds, and the running time is defined to be equal to the number of communication rounds. Following the common practice, we say that an event occurs with high probability if it occurs with probability at least $1 - 1/n^c$, where c is an arbitrarily large constant. We consider randomised Monte Carlo algorithms,

Table 1: Upper and lower bounds for the distributed LLL.

LLL criterion	Running time	References
$ep(d+1) < 1$	$O(\log n \log d)$	[11, 19]
$epd^2 < 1$	$O(\log n)$	[11]
$pf(d) < 1$, where $f(d)$ exponential	$O(\log n / \log\log n)$	[11]
$pf(d) \leq 1$ for any f	$\Omega(\log^* n)$	[11]
$pf(d) \leq 1$ for f satisfying $f(4) \leq 16$	$\Omega(\log\log n)$	this work

in which all bad events are avoided with high probability and the running time is deterministic (more precisely, some function of n).

1.2 Main Result and Key Techniques

We prove the following lower bound for LLL algorithms in the LOCAL model: any randomised Monte Carlo algorithm that produces a correct solution with high probability requires $\Omega(\log\log n)$ communication rounds, even if we restrict our input instances to d-regular graphs with $d \in O(1)$. This is a substantial improvement over the lower bound of $\Omega(\log^* n)$ from prior work [11, 28, 35].

To derive the lower bound, we introduce two new graph problems that are closely related to each other: *sinkless orientation* and *sinkless colouring* (see Section 2). Then we proceed as follows:

(1) We show that any Monte Carlo distributed algorithm for LLL implies a Monte Carlo distributed algorithm for sinkless orientation in 3-regular graphs, with asymptotically the same running time (see Section 3).

(2) We show that any Monte Carlo distributed algorithm for sinkless orientation in 3-regular graphs has a running time of $\Omega(\log\log n)$.

For the second step, we study the sinkless orientation problem in high-girth graphs. The key ingredient is a *mutual speedup lemma* (see Section 4) that holds in graphs of girth larger than $2t+1$:

(1) If we can find a sinkless colouring in t rounds, we can find a sinkless orientation in t rounds.

(2) If we can find a sinkless orientation in t rounds, we can find a sinkless colouring in $t-1$ rounds.

By iterating the mutual speedup lemma, we can then obtain an algorithm for finding a sinkless orientation in high-girth graphs with a running time of 0 rounds, which is absurd. The mutual speedup lemma amplifies the failure probability, but not too much—if the original algorithm works with high probability, we can still reach the contradiction after $o(\log\log n)$ iterations.

As a by-product, we also obtain a lower bound for d-colouring d-regular high-girth graphs: any proper node colouring with d colours is also a sinkless colouring (while the converse is not true).

Our lower-bound proof does not make use of the full power of LLL—it also holds if we replace the usual assumption of $ep(d+1) \leq 1$ with, e.g., the classical formulation [17] of Erdős and Lovász where the assumption is $4pd \leq 1$. In addition, our bound holds for the symmetric version of LLL, and therefore, it trivially applies to the asymmetric LLL [3] as well.

1.3 Prior Work on LLL

The celebrated Lovász local lemma was first introduced in 1975 [17] and has since then found applications in proving the existence of various combinatorial structures [3, 30, 32]. However, the original proof was non-constructive, and thus, did not yield an efficient (centralised) algorithm for *finding* such a structure.

Beck [7] showed that constructive versions of the local lemma do exist, albeit with weaker guarantees: there exists a deterministic algorithm that finds a satisfying assignment to a certain variant of LLL in polynomial time. This breakthrough result stimulated a long line of research in devising new *algorithmic* versions of the local lemma with more general conditions and better performance [1, 9, 13, 23, 31, 33, 34, 40]. The algorithmic LLL has found numerous applications e.g. in the context of colouring, scheduling, and satisfiability problems [11–13, 16, 26, 32, 33, 39].

A key breakthrough was the result by Moser and Tardos [34]: they showed that even a very general form of the local lemma has a constructive counterpart; a natural resampling algorithm finds a satisfying assignment efficiently. Moser and Tardos also gave a parallel variant of this algorithm which can easily be implemented in a distributed setting as well. Indeed, already Alon [1] observed that LLL admits parallelism by showing how to parallelise Beck's original approach [7]. Subsequently, many papers have also considered how to attain efficient parallel and distributed algorithms for LLL [9, 11, 22, 34]; Table 1 summarises known upper and lower bounds for the distributed LLL.

The algorithmic framework of Moser and Tardos [34] is based on an iterative random sampling method. The idea is to start with a random assignment and while a violated constraint exists, the algorithm then iteratively resamples variables in some violated constraint. Resampling is continued until no more violated constraints exist. The algorithm is easy to parallelise by noting that one can resample variables in *independent* constraints, that is, in constraints that do not share variables. Now it suffices to pick a maximal independent set in the subgraph of the dependency graph induced by the violated constraints and resample variables related to these constraints. Moser and Tardos use Luby's algorithm [29] to find a maximal independent set in $O(\log n)$ rounds in each resampling iteration. In total, this algorithm requires $O(\log n)$ resampling iterations thus leading to a total running time of $O(\log^2 n)$ rounds in the distributed setting.

One approach for speeding up this basic algorithm is to use faster algorithms for computing the independent sets. For example, in constant-degree graphs, a maximal independent set can be found in $\Theta(\log^* n)$ rounds [28]. More generally in low-degree graphs, maximal independent sets can be found in $O(d + \log^* n)$ rounds [5] and $O(\log d \cdot \sqrt{\log n})$ rounds [6], thus making it possible to solve LLL in $o(\log^2 n)$ rounds when the degrees are small. However, this approach has an inherent barrier—the KMW lower bounds for the complexity of finding maximal matchings [24, 25].

As pointed out by Moser and Tardos [34], it is not necessary to find a *maximal* independent set, but a large independent set suffices. Following this idea, Chung et al. [11] gave a distributed algorithm where they instead compute so-called *weakly maximal* independent sets, where the probability that a node is not in the produced independent set S or neighbouring a node in set S is bounded by $1/\operatorname{poly}(d)$. They showed that this can be done in $O(\log^2 d)$ rounds, thus the dependency on n in the total running time of the LLL algorithm is only $O(\log n)$. Recently, Ghaffari improved this further by showing that weakly maximal independent sets can be computed in $O(\log d)$ rounds [19].

If one is interested in weaker forms of LLL, Chung et al. [11] also provide faster algorithms running in $O(\log n/\log\log n)$ rounds for the LLL criterion $pf(d) < 1$, where $f(d)$ is an exponential function.

While there are numerous positive results, only a few lower bounds for LLL are known. Moser and Tardos point out that in their resampling approach, $\Omega(\log_{1/p} n)$ expected iterations of resampling are needed. Recently, Haeupler and Harris [22] conjectured that parallel resampling algorithms need $\Omega(\log^2 n)$ time.

In the distributed setting, Chung et al. [11] show an unconditional lower bound showing that essentially any distributed LLL algorithm takes $\Omega(\log^* n)$ rounds. This bound follows from the fact that LLL can be used to properly colour a ring using only a constant number of colours, which is known to take $\Omega(\log^* n)$ rounds [28, 35].

1.4 Prior Work on Other Lower Bounds

Overall, not that many unconditional, nontrivial lower bounds for the LOCAL model are known [4, 41]. Many of the prior lower bound techniques fall in one of the following classes: either they are limited to lower bounds of the form $\Omega(\log^* n)$ [14, 20, 27, 28, 35, 36], or they are relevant only in graph families with non-constant degrees [18, 21, 24, 25]. In this work, we prove lower bounds of the form $\Omega(\log\log n)$ for the case of a maximum degree $\Delta \in O(1)$.

From the perspective of the results, perhaps closest to our work is Linial's [28] lower bound for colouring d-regular trees with $O(\sqrt{d})$ colours. However, Linial's technique is limited to $O(d/\log d)$-colouring, while our work implies a lower bound for d-colouring (Corollary 1).

From the perspective of the techniques, our mutual speedup lemma bears some resemblance to another result by Linial, the lower bound for 3-colouring cycles [28]. Linial's proof can be interpreted as a speedup result: if we can colour a cycle in t rounds, we can also colour it in $t-1$ rounds, provided that we tolerate a larger number of colours. In our proof, the number of colours remains constant (but the failure probability is amplified).

2. PRELIMINARIES

Let $G = (V, E)$ be a simple graph. An *orientation* σ of a graph G assigns a direction

$$\sigma(\{u, v\}) \in \{u \to v, u \leftarrow v\}$$

for each edge $\{u, v\} \in E$. For convenience, we write $(u, v) \in \sigma(E)$ to denote an edge $\{u, v\} \in E$ oriented $u \to v$ by σ. For all $v \in V$ we define $\operatorname{indeg}(v, \sigma) = |\{u : (u, v) \in \sigma(E)\}|$ as the number of incoming edges, $\operatorname{outdeg}(v, \sigma)$ as the number of outgoing edges, and $\deg(v) = \operatorname{indeg}(v, \sigma) + \operatorname{outdeg}(v, \sigma)$ as the degree of v. Graph G is d-regular if for all $v \in V$ we have $\deg(v) = d$.

A node v with $\operatorname{indeg}(v, \sigma) = \deg(v)$ is called a *sink*. We call an orientation σ *sinkless* if no node is a sink, that is, every node v has $\operatorname{outdeg}(v, \sigma) > 0$.

2.1 Colourings

In the following, for any integer $k > 0$ we write $[k] = \{0, 1, \dots, k-1\}$. A function $\varphi\colon V \to [\chi]$ is a *proper node χ-colouring* if for all $\{u, v\} \in E$ we have $\varphi(u) \neq \varphi(v)$. We say that $\psi\colon E \to [\chi]$ is a *proper edge χ-colouring* if we have $\psi(e) \neq \psi(e')$ for all $e, e' \in E$ with $e \neq e'$ and $e \cap e' \neq \emptyset$. That is, any two adjacent edges have a different colour.

Given a properly edge χ-coloured graph $G = (V, E, \psi)$, we call $\varphi\colon V \to [\chi]$ a *sinkless colouring* of G if for all edges $e = \{u, v\} \in E$ it holds that

$$\varphi(u) = \psi(e) \implies \varphi(v) \neq \psi(e).$$

Put otherwise, φ is a sinkless colouring if it does not contain a *forbidden configuration*, where $\varphi(u) = \varphi(v) = \psi(e)$ for some edge $e = \{u, v\} \in E$; see Figure 1 for an example. Note that a sinkless colouring is not necessarily a proper node colouring. The name "sinkless colouring" refers to its close relation to sinkless orientations (see Section 2.4).

2.2 Model of Computation

In this work, we consider the LOCAL model of distributed computing [28, 37]. In this framework, we have a simple connected undirected graph $G = (V, E)$ that serves both as a communication network and as the problem instance. Each node $v \in V$ is a computational unit, edges denote direct communication links between nodes, and all nodes in the system execute the same algorithm A.

Initially each node v knows the total number of nodes n, the maximum degree of the graph Δ, and possibly a task-specific local input. Computation proceeds in synchronous rounds. In each round, every node performs the following three steps:

(1) send a message to each neighbour,
(2) receive a message from each neighbour,
(3) perform local computation.

After the final round each node announces its own local output, that is, its own part in the solution. We do not bound the local computation performed by nodes each round in any way or the size of the messages sent. In particular, nodes can send *infinitely* long messages to their neighbours in a single round. The running time of an algorithm is defined to be the number of communication rounds needed for all nodes to announce their local output.

In the case of *randomised* algorithms, we assume that each node can toss a countably infinite number of random coins, or equivalently, is provided with a real number $x(v)$ taken uniformly at random from the interval $[0, 1]$. Note that $x(v)$ provides a globally unique identifier with probability 1 and it can be used to obtain a globally unique $O(\log n)$-bit identifier with high probability.

We emphasise that while some of our assumptions may not be realistic, they only make the lower-bound result stronger.

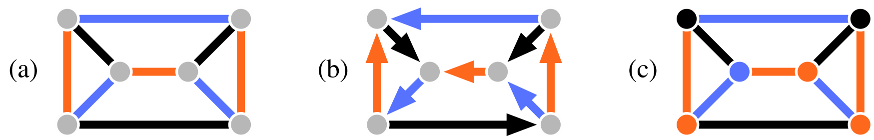

Figure 1: (a) A 3-regular edge 3-coloured graph. (b) A sinkless orientation. (c) A sinkless colouring. Note that the solutions (b) and (c) are closely related: if the colour of a node is c in Figure (c), then its incident edge of colour c is one of its out-edges in Figure (b).

2.3 Local Neighbourhoods

We denote the radius-t neighbourhood of a node u by $N^t(u) = \{v \in V : \mathrm{dist}(u, v) \leq t\}$, where $\mathrm{dist}(u, v)$ is the length of the shortest path between u and v. Note that in t rounds, any node u can only gather information from t hops away, and hence, has to decide its output based solely on $N^t(u)$. Thus, any distributed t-time algorithm can be considered as a function that maps the radius-t neighbourhoods to output values.

It is often convenient to consider the *edges* instead of the nodes as active entities, that is, every edge outputs e.g. its own orientation. Hence, we analogously define the radius-t neighbourhood of an edge $\{u, v\}$ as $N^t(\{u, v\}) = N^t(u) \cap N^t(v)$. Now for any algorithm that runs in time t, the output of an edge $\{u, v\} \in E$ can only depend on $N^t(\{u, v\})$.

2.4 Distributed Sinkless Orientation and Sinkless Colouring

In order to prove our lower bound, we consider the tasks of finding a sinkless orientation and a sinkless colouring in d-regular graphs with distributed algorithms. In particular, we will show that solving these problems is hard even in the case where we are given an edge d-colouring. In the following, let $G = (V, E, \psi)$ denote our input graph, where ψ is a proper edge d-colouring; see Figure 1 for illustrations.

Problem 1. Sinkless colouring. Given an edge d-coloured d-regular graph $G = (V, E, \psi)$, find a sinkless colouring φ. That is, compute a colouring φ such that for no edge $e = \{u, v\} \in E$ we have $\varphi(u) = \varphi(v) = \psi(e)$.

In the sinkless colouring problem, each node $v \in V$ only outputs its own colour $\varphi(v)$ in the computed colouring.

Problem 2. Sinkless orientation. Given an edge d-coloured d-regular graph $G = (V, E, \psi)$, find a sinkless orientation σ. That is, compute an orientation σ such that $\mathrm{outdeg}(v, \sigma) > 0$ for all $v \in V$.

Note that in the sinkless orientation problem, the output relates to *edges* instead of nodes. Therefore, we require that for an edge $e = \{u, v\}$ both endpoints u and v agree on the orientation (i.e., either $u \to v$ or $u \leftarrow v$) and output the same value $\sigma(e)$ with probability 1. Put otherwise, u and v have to break symmetry in order not to have the nodes trying to orient the edge in a conflicting manner, e.g. outwards from themselves. In the case of randomised algorithms, the random input value $x(v) \in [0, 1]$ breaks symmetry with probability 1; the event $x(v) = x(u)$ occurs with probability 0, so we simply ignore this case for the remainder of this paper.

Finally, note that the problems of sinkless colouring and sinkless orientation are closely related (see Figure 1). Given a sinkless colouring φ, node u can orient the edge $\{u, v\}$ with colour $\varphi(u)$ towards v and inform v of this in one communication round; edges that are still unoriented can be oriented arbitrarily. This produces a sinkless orientation. On the other hand, given a sinkless orientation σ we can compute a sinkless colouring φ as follows: node u outputs the smallest colour $\psi(e)$, where $e = \{u, v\}$ is an outgoing edge, that is, $\sigma(e) = u \to v$. Any edge $e = \{u, v\}$ can be an outgoing edge for at most one of the nodes u and v, and thus at most one of them outputs the colour $\psi(e)$. Hence we have the following trivial observations:

(1) If we can find a sinkless orientation in t rounds, we can find a sinkless colouring in t rounds.

(2) If we can find a sinkless colouring in t rounds, we can find a sinkless orientation in $t + 1$ rounds.

The mutual speedup lemma (see Section 4) shows that we can save 1 communication round in both steps (1) and (2), at least in high-girth graphs.

2.5 Distributed Lovász Local Lemma

Let $\mathcal{X}$ be the set of random variables and $\mathcal{E} = \{E_1, \ldots, E_n\}$ be the set of events as in Theorem 1 in Section 1.1. Denote by $\mathrm{vbl}(E_k) \subseteq \mathcal{X}$ the subset of variables that event $E_k \in \mathcal{E}$ depends on and the dependency graph by $G_\mathcal{E} = (\mathcal{E}, \mathcal{D})$, where $\mathcal{D} = \{\{E_i, E_j\} : \mathrm{vbl}(E_i) \cap \mathrm{vbl}(E_j) \neq \emptyset\}$.

Problem 3. Distributed Lovász local lemma. Let the dependency graph $G_\mathcal{E} = (\mathcal{E}, \mathcal{D})$ be the communication graph, where each node v corresponds to an event $E_v \in \mathcal{E}$ and knows the set $\mathrm{vbl}(E_v)$. The task is to have each node output an assignment a_v of the variables $\mathrm{vbl}(E_v)$ such that

(1) for any $\{E_u, E_v\} \in \mathcal{D}$ and $X \in \mathrm{vbl}(E_u) \cap \mathrm{vbl}(E_v)$ it holds that $a_u(X) = a_v(X)$, and

(2) the event E_v does not occur under assignment a_v.

To make the lower-bound results as widely applicable as possible, we consider the *explicit, finite version* of the LLL problem: the random variables are discrete variables with a finite range, and for each event E_v, node v has access to an explicit specification of all combinations of the variables $\mathrm{vbl}(E_v)$ for which E_v occurs. In particular, we do not need to assume that the events are black boxes.

3. FROM LLL TO SINKLESS ORIENTATION

In this section, we reduce the sinkless orientation problem to the distributed Lovász local lemma. More specifically, we show the following:

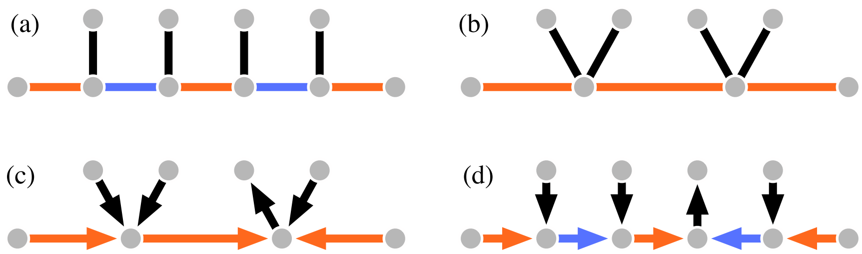

Figure 2: (a) Part of a high-girth 3-regular edge 3-coloured graph. (b) A 4-regular graph obtained by contracting all blue edges—note that the graph is not properly edge coloured. (c) With distributed LLL we can find a sinkless orientation in the 4-regular graph. (d) We can now orient each blue edge greedily to obtain a sinkless orientation of the original graph.

THEOREM 2. *Let $f: \mathbb{N} \to \mathbb{R}$ be such that $f(4) \leq 16$. Let A be a Monte Carlo distributed algorithm for LLL such that A finds an assignment avoiding all the bad events under the LLL criterion $pf(d) \leq 1$ in time T for some $T: \mathbb{N} \to \mathbb{N}$. Then there is a Monte Carlo distributed algorithm B that finds a sinkless orientation in 3-regular graphs of girth at least 5 in time $O(T)$.*

The case for 4-regular graphs would be almost immediate. The interesting case will be 3-regular graphs, for which we will show a reduction to the 4-regular case. Note that if we used the LLL criterion of Shearer [38], we could use LLL to find sinkless orientations directly in 3-regular graphs, but this would yield a weaker lower bound.

PROOF OF THEOREM 2. The approach that we take is illustrated in Figure 2. Let G be a 3-regular edge 3-coloured graph of girth at least 5. We first contract all edges of colour 2 (blue edges in the illustration) to obtain a 4-regular graph G'. As G had a sufficiently high girth, graph G' will be simple.

Note that the contraction only changes the distances by a constant factor. Hence we can easily simulate any distributed algorithm on the 4-regular graph G' with a constant multiplicative overhead: each communication round in G' corresponds to at most 3 communication rounds in G.

In particular, we can apply the LLL algorithm A to find a sinkless orientation of graph G'. From the LLL perspective, each edge $e = \{u, v\}$ corresponds to a variable X_e that ranges over $\{u \to v, u \leftarrow v\}$, and each node v corresponds to the bad event E_v that all incident edges are oriented towards v. Conveniently, the dependency graph $G_{\mathcal{E}} = (\mathcal{E}, \mathcal{D})$ is isomorphic to graph G'. If the mutually independent random variables X_e are sampled uniformly at random, we have $\Pr[E_v] = 1/16$ for each $v \in V$, and hence we can apply algorithm A to find a sinkless orientation in G'.

Recall that each node of G' corresponds to an edge of colour 2 in G. Hence from the perspective of graph G, we have now oriented all edges of colours 0 and 1 such that for each edge $e = \{u, v\}$ of colour 2, at least one endpoint of e has a positive outdegree; let this endpoint be v. Now if we orient e from u to v, both endpoints of e will have a positive outdegree. $\square$

4. THE MUTUAL SPEEDUP LEMMA

In this section we show that if we can find a sinkless colouring in t rounds with failure probability p, then it is possible to find a sinkless orientation in t rounds with failure probability roughly $p^{1/3}$ assuming the graph has a high girth. Furthermore, if we can find a sinkless orientation in t rounds with failure probability q, then it is possible to find a sinkless colouring in $t-1$ rounds with failure probability roughly $q^{1/4}$.

To be precise, we define a *sinkless colouring algorithm with failure probability* p as follows: the algorithm will always output some colouring, but the colouring is not necessarily sinkless; for any edge e, the probability that e has a forbidden configuration is bounded by p. Similarly, we define a *sinkless orientation algorithm with failure probability* q as follows: the algorithm will always output some orientation, but the orientation is not necessarily sinkless; for any node u, the probability that u is a sink is bounded by q.

We will assume throughout this section that the input graph is a 3-regular edge 3-coloured graph with a girth larger than $2t+1$. Thus, for each edge e, its radius-$(t+1)$ neighbourhood is a tree. In addition, we say that the radius-t neighbourhood $N^t(u)$ of node u is *fixed* when we fix the random input values (i.e., coin flips) for the nodes in $N^t(u)$. Similarly, the radius-t neighbourhood $N^t(e) = N^t(u) \cap N^t(v)$ of an edge $e = \{u, v\}$ is *fixed* if all the random values in $N^t(e)$ are fixed. We denote the probability of an event $\mathcal{E}$ conditioned on a fixed radius-t neighbourhood of u by $\Pr[\mathcal{E} \mid N^t(u)]$, and respectively, for an edge e by $\Pr[\mathcal{E} \mid N^t(e)]$.

4.1 From Sinkless Colouring to Sinkless Orientation

In this section, we assume we are given a randomised sinkless *colouring* algorithm B (with some failure probability) that runs in t rounds. We use this algorithm to construct a randomised sinkless *orientation* algorithm B' that also runs in t rounds. For brevity, we write $B(u)$ for the colour that u outputs according to B and $B'(e)$ for the orientation B' outputs for edge e.

The intuition behind the construction is as follows. In B', an edge $e = \{u, v\}$ of colour $\psi(e)$ is oriented $u \to v$ if, based on the local neighbourhood of e, both of the following hold:

- we have $B(u) = \psi(e)$ with a "large" probability,
- we have $B(v) = \psi(e)$ with a "small" probability.

The distinction between "large" and "small" is based on a threshold value. For each node u, the output $B(u)$ is equal to the colour of at least one incident edge $e = \{u, v\}$ with a "large" probability; therefore node u will have an incident edge oriented towards v and therefore u will not be a sink—

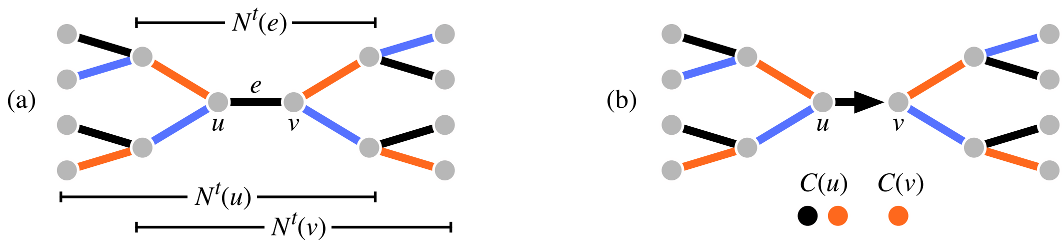

Figure 3: From sinkless colouring to sinkless orientation. Here we are given a sinkless colouring algorithm B with a running time of $t = 2$, and our goal is to construct a sinkless orientation algorithm B' with the same running time. (a) In algorithm B, the colour of node u is determined by the random bits in $N^t(u)$. However, when algorithm B' chooses the orientation of the black edge e it will only look at the random bits in $N^t(e) \subsetneq N^t(u)$. We say that black is a *candidate colour* of u if, based on the information in $N^t(e)$ alone, the probability of node u outputting black in algorithm B is at least K. (b) If black is one of the candidate colours of u, and it is not one of the candidate colours of v, algorithm B' will orient the edge $u \to v$.

unless v also has a large probability of having the same colour as e. But the probability of the latter happening is bounded via the failure probability of B since u, v and e having the same colour is a forbidden configuration.

Consider any node $u \in V$. Algorithm B' consists of three steps. First, node u gathers its radius-t neighbourhood $N^t(u)$ in t communication rounds. Note that $N^t(u)$ contains both the topology (which is locally a tree) *and* the random input values $x(v)$ for all $v \in N^t(u)$. Second, node u computes the set $C(u)$ of *candidate colours* defined as

$$C(u) = \{\psi(e) : \Pr[B(u) = \psi(e) \mid N^t(e)] \geq K \text{ and } e = \{u, v\}\},$$

where K is a parameter we will fix later (see Figure 3 for an illustration). Then, for all of its incident edges $e = \{u, v\}$, node u calculates the probability of node v outputting the colour $\psi(e)$ when executing algorithm B given the radius-t neighbourhood $N^t(e) = N^t(u) \cap N^t(v)$ of edge e. Thereby, u can determine whether $\psi(e) \in C(v)$.

Finally, we decide the orientation of each edge $e = \{u, v\}$ as follows. In the case $\psi(e) \in C(u) \cap C(v)$ or $\psi(e) \notin C(u) \cup C(v)$, choose the orientation $B'(e)$ of edge e arbitrarily and break any ties using the random coin flips of u and v. Otherwise, edge e is oriented according to the following rule:

$$B'(e) = \begin{cases} u \to v & \text{if } \psi(e) \in C(u) \text{ and } \psi(e) \notin C(v), \\ u \leftarrow v & \text{if } \psi(e) \notin C(u) \text{ and } \psi(e) \in C(v). \end{cases}$$

We will now analyse algorithm B'. For each colour c, let $e = \{u, v\}$ be an edge incident to u such that $\psi(e) = c$ and define

$$A_c(u) = N^{t-1}(v) \setminus N^{t-1}(u).$$

Definition 1. Lucky random bits. Given a fixed radius-$(t-1)$ neighbourhood $N^{t-1}(u)$ of u, we say that the random coin flips in $A_c(u)$ are *lucky* if

$$\Pr[B(u) = c \mid N^{t-1}(u) \cup A_c(u)] \geq K$$

holds, and otherwise that the coin flips are *unlucky*.

Observe that $c \in C(u)$ if and only if the random coin flips in $A_c(u)$ are lucky, since $N^{t-1}(u) \cup A_c(u) = N^t(e)$. Let $\mathcal{E}_c$ be the event that the random coin flips in $A_c(u)$ are unlucky, that is, the event that $\Pr[B(u) = c \mid N^{t-1}(u) \cup A_c(u)] < K$ holds.

Lemma 1. *Given any fixed neighbourhood $N^{t-1}(u)$ of node u, the set $C(u)$ is empty with probability at most $3K$.*

Proof. Let $\mathcal{E} = \bigcap \mathcal{E}_c$ be the event that the random values in each $A_c(u)$ are unlucky given $N^{t-1}(u)$. This is the case if and only if $C(u) = \emptyset$, which implies that

$$\Pr[C(u) = \emptyset \mid N^{t-1}(u)] = \Pr[\mathcal{E}],$$

where the right-hand side can be written as

$$\Pr[\mathcal{E}] = \sum_c \Pr[\mathcal{E} \text{ and } B(u) = c].$$

Observe that since $\mathcal{E} \subseteq \mathcal{E}_c$ for any colour c, we have that

$$\begin{aligned} \Pr[\mathcal{E} \text{ and } B(u) = c] &= \Pr[\mathcal{E} \text{ and } B(u) = c \mid \mathcal{E}_c] \cdot \Pr[\mathcal{E}_c] \\ &\leq \Pr[B(u) = c \mid \mathcal{E}_c] \cdot \Pr[\mathcal{E}_c] \\ &\leq \Pr[B(u) = c \mid \mathcal{E}_c]. \end{aligned}$$

Since by definition the coin flips in $A_c(u)$ are unlucky in the event $\mathcal{E}_c$, we get that $\Pr[B(u) = c \mid \mathcal{E}_c] < K$. Thus combining the above, we have that

$$\Pr[C(u) = \emptyset \mid N^{t-1}(u)] = \sum_c \Pr[\mathcal{E} \text{ and } B(u) = c] < \sum_c K.$$

Since we have three colours, the claim follows. □

Definition 2. Nice edge neighbourhoods. For an edge $e = \{u, v\}$, we call its fixed neighbourhood $N^t(e)$ *nice* if

$$\Pr[B(u) = \psi(e) = B(v) \mid N^t(e)] < K^2.$$

That is, after fixing the random coin flips in $N^t(e)$, the algorithm outputs $\psi(e)$ at both u and v with probability less than K^2. Otherwise, we call $N^t(e)$ a *bad* neighbourhood.

Lemma 2. *Let $e = \{u, v\}$ be an edge with no cycles in its radius-$(t+1)$ neighbourhood. If the fixed neighbourhood $N^t(e)$ is nice, then $\psi(e) \notin C(u) \cap C(v)$.*

Proof. Let $N^t(e)$ be fixed and nice. Assume for contradiction that $\psi(e) \in C(u) \cap C(v)$. By definition of the candidate colour set, for both $w \in e$ we have $\psi(e) \in C(w)$ if

$$\Pr[B(w) = \psi(e) \mid N^t(e)] \geq K.$$

As the output $B(u)$ of node u is determined by the coin flips in $N^t(u)$ and the coin flips in $N^t(e) = N^t(u) \cap N^t(v)$ are fixed, we now have that $B(u)$ only depends on the coin flips in $N^t(u) \setminus N^t(v)$. Similarly, the output $B(v)$ of v only depends on the coin flips in $N^t(v) \setminus N^t(u)$. Therefore, the events $B(u) = \psi(e)$ and $B(v) = \psi(e)$ are independent, as $N^{t+1}(e)$ contains no cycles, and we get

$$\Pr[B(u) = \psi(e) = B(v) \mid N^t(e)] \geq K^2,$$

which contradicts the assumption that $N^t(e)$ was nice. □

Now it is easy to check that if a node u has at least one candidate colour and all its incident edges have nice neighbourhoods, then u will not be a sink according to B'.

LEMMA 3. *Suppose $N^t(u)$ is fixed and the neighbourhoods $N^t(e)$ are nice for all edges $e = \{u, w\}$ incident to u. If $C(u) \neq \emptyset$, then $B'(e') = u \to v$ for some $e' = \{u, v\}$.*

PROOF. Since $C(u) \neq \emptyset$, there is some $\psi(e) \in C(u)$. Moreover as $N^t(e)$ is nice, Lemma 2 implies that $\psi(e) \notin C(u) \cap C(v)$, and thus, $\psi(e) \notin C(v)$. By definition of B', we have $B'(e) = u \to v$. □

Now we have all the pieces to show the first part of the mutual speedup lemma.

LEMMA 4. *Suppose B is a sinkless colouring algorithm that runs in t rounds such that for any edge $e = \{u, v\}$ the probability of outputting a forbidden configuration $B(u) = \psi(e) = B(v)$ is at most p. Then there exists a sinkless orientation algorithm B' that runs in t rounds such that for any node u the probability of being a sink is at most $6p^{1/3}$.*

PROOF. Let B' be as given earlier and consider a node u. By Lemma 3, algorithm B' can produce a sink at node u only if $C(u) = \emptyset$ or one of the edges incident to u has a bad (i.e., not nice) neighbourhood. Let

$$S = \max_{e \ni u} \Pr[N^t(e) \text{ is bad}]$$

be the maximum probability that a particular edge has a bad neighbourhood; the probability of having a bad neighbourhood need not be the same for edges of different colours. By the union bound, the probability that $N^t(e)$ is bad for *some* edge $e = \{u, v\}$ is at most $3S$. By Lemma 1, the probability that $C(u) = \emptyset$ is at most $3K$. Thus, applying the union bound once again, we get that

$$\begin{aligned}
&\Pr[\text{node } u \text{ is a sink}] \\
&\leq \sum_{e=\{u,v\}} \Pr[N^t(e) \text{ is bad}] + \Pr[C(u) = \emptyset] \\
&\leq 3S + 3K.
\end{aligned}$$

Now let us consider the probability that an edge $e = \{u, v\}$ has a forbidden configuration, where e is an edge that attains $\Pr[N^t(e) \text{ is bad}] = S$. Recall that the probability of $B(u) = \psi(e) = B(v)$ is at most p, and thus,

$$\begin{aligned}
p &\geq \Pr[B(u) = \psi(e) = B(v)] \\
&\geq \Pr[N^t(e) \text{ is bad}] \cdot \Pr[B(u) = \psi(e) = B(v) \mid N^t(e) \text{ is bad}] \\
&\geq SK^2
\end{aligned}$$

by Definition 2. By setting $K = p^{1/3}$ we get that

$$p \geq SK^2 = Sp^{2/3} \iff p^{1/3} \geq S$$

and we have $3S + 3K \leq 6p^{1/3}$, which proves our claim. □

4.2 From Sinkless Orientation Back to Sinkless Colouring

We now show how to construct a randomised sinkless *colouring* algorithm B'' that runs in time $t - 1$ given a sinkless *orientation* algorithm B' that runs in time t. The approach is analogous to the one in the previous section. The high level idea is that any node u first checks which of its incident edges are likely to be pointed outwards by B', and then it can choose the colour of one of these edges to output a sinkless colouring with a large probability.

Unlike before, each node will gather only its radius-$(t-1)$ neighbourhood in $t-1$ rounds. Again, let L be a threshold we fix later. Define the candidate colour set $C'(u)$ as

$$C'(u) = \{\psi(e) : \Pr[B'(e) = u \leftarrow v \mid N^{t-1}(u)] \leq L\},$$

that is, the set of colours which are pointed towards u with probability at most L. The node u will then output the smallest candidate colour or an arbitrarily chosen colour if there are no candidates, or formally,

$$B''(u) = \begin{cases} \min C'(u) & \text{if } C'(u) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Our goal now is to show that this produces a sinkless colouring with a large probability. To do this, we show that the probabilities of the following two events are large: (1) the candidate set being non-empty and (2) $\psi(e) \notin C(u) \cap C(v)$ for any edge $e = \{u, v\}$.

Analogously to Section 4.1, we define the notions of lucky and unlucky bits as well as nice and bad neighbourhoods.

Definition 3. Lucky random bits. For any $e = \{u, v\}$, let $A_u(e) = N^{t-1}(u) \setminus N^{t-1}(v)$. We say that the random coin flips in $A_u(e)$ are *lucky* if

$$\Pr[B'(e) = u \leftarrow v \mid N^{t-1}(e) \cup A_u(e)] \leq L.$$

Otherwise, the coin flips in $A_u(e)$ are *unlucky*.

LEMMA 5. *Given any fixed neighbourhood $N^{t-1}(e)$ of edge e, we have $\Pr[\psi(e) \in C'(u) \cap C'(v) \mid N^{t-1}(e)] \leq 2L$.*

PROOF. Fix the random coin flips in $N^{t-1}(e)$. Let $\mathcal{E}_u$ be the event that the coin flips in $A_u(e)$ are lucky and let $\mathcal{E} = \mathcal{E}_u \cap \mathcal{E}_v$ be the event that coin flips in both $A_u(e)$ and $A_v(e)$ are lucky. Observe that $\psi(e) \in C'(u)$ if and only if the coin flips in $A_u(e)$ are lucky. Therefore,

$$\begin{aligned}
\Pr[\mathcal{E}] &= \Pr[\psi(e) \in C'(u) \cap C'(v) \mid N^{t-1}(e)] \\
&= \Pr[\mathcal{E} \text{ and } B'(e) = u \to v] \\
&\quad + \Pr[\mathcal{E} \text{ and } B'(e) = u \leftarrow v].
\end{aligned}$$

Since $\mathcal{E} \subseteq \mathcal{E}_u$, it follows that

$$\begin{aligned}
&\Pr[\mathcal{E} \text{ and } B'(e) = u \leftarrow v] \\
&= \Pr[\mathcal{E} \text{ and } B'(e) = u \leftarrow v \mid \mathcal{E}_u] \cdot \Pr[\mathcal{E}_u] \\
&\leq \Pr[B'(e) = u \leftarrow v \mid \mathcal{E}_u] \leq L
\end{aligned}$$

by Definition 3 as the coin flips in $A_u(e)$ are lucky in the event $\mathcal{E}_u$. Symmetrically, we also get the bound

$$\Pr[\mathcal{E} \text{ and } B'(e) = u \to v] \leq L.$$

Combining the above observations we get that

$$\Pr[\mathcal{E}] = \Pr[\psi(e) \in C'(u) \cap C'(v) \mid N^{t-1}(e)] \leq 2L,$$

and the claim follows. □

Definition 4. Nice node neighbourhoods. Let $N^{t-1}(u)$ be fixed. We say that the neighbourhood $N^{t-1}(u)$ is *nice* if the probability that u is a sink when executing B' is at most L^3, that is, if

$$\Pr[B'(e) = u \leftarrow v \text{ for all } e = \{u,v\} \mid N^{t-1}(u)] \leq L^3$$

holds. Otherwise, we call $N^{t-1}(u)$ a *bad* neighbourhood.

LEMMA 6. *Assume that the fixed neighbourhood $N^{t-1}(u)$ is nice. Then $C'(u) \neq \emptyset$.*

PROOF. Fix the coin flips in $N^{t-1}(u)$ and assume $N^{t-1}(u)$ is nice. For the sake of contradiction, suppose $C'(u) = \emptyset$. Now by definition of $C'(u)$ we have

$$\Pr[B'(e) = u \leftarrow v \mid N^{t-1}(u)] > L$$

for each edge $e = \{u,v\}$. Since the coin flips in $N^{t-1}(u)$ are fixed, the output $B'(e)$ only depends on the coin flips in $N^{t-1}(v) \setminus N^{t-1}(u)$. Since the girth is larger than $2t$, for each $e = \{u,v\}$ and $e' = \{u,v'\}$, where $v \neq v'$, the coin flips in $N^{t-1}(v) \setminus N^{t-1}(u)$ and $N^{t-1}(v') \setminus N^{t-1}(u)$ are independent. Therefore, the events $B'(e) = u \leftarrow v$ and $B'(e') = u \leftarrow v'$ are independent. This implies that

$$\begin{aligned} &\Pr[C'(u) = \emptyset \mid N^{t-1}(u)] \\ &= \prod_{e=\{u,v\}} \Pr[B'(e) = u \leftarrow v \mid N^{t-1}(u)] > L^3, \end{aligned}$$

contradicting the assumption that $N^{t-1}(u)$ is nice. $\square$

LEMMA 7. *Suppose B' is a sinkless* orientation *algorithm that runs in time t such that the probability that any node u is a sink is at most ℓ. Then there exists a sinkless* colouring *algorithm B'' that runs in time $t-1$ such that the probability for any edge $e = \{u,v\}$ having a forbidden configuration $B''(u) = \psi(e) = B''(v)$ is less than $4\ell^{1/4}$.*

PROOF. Let B'' as defined earlier and consider an edge $e = \{u,v\}$. If algorithm B'' outputs a forbidden configuration $B''(u) = \psi(e) = B''(v)$, then either $C'(u) \cup C'(v) = \emptyset$ or $\psi(e) \in C'(u) \cap C'(v)$ holds. We will now bound the probability of both events.

Observe that before fixing any random bits, the probability of having a bad radius-$(t-1)$ neighbourhood is the same for all nodes, as all radius-$(t-1)$ node neighbourhoods are identical. Let $S = \Pr[N^{t-1}(u) \text{ is bad}]$ be this probability. By union bound and Lemma 6 we get that

$$\begin{aligned} &\Pr[C'(u) \cup C'(v) = \emptyset] \\ &\leq \Pr[C'(u) = \emptyset] + \Pr[C'(v) = \emptyset] \\ &\leq \Pr[N^{t-1}(u) \text{ is bad}] + \Pr[N^{t-1}(v) \text{ is bad}] \leq 2S. \end{aligned}$$

From Lemma 5 we get that

$$\Pr[\psi(e) \in C'(u) \cap C'(v)] \leq 2L.$$

Using the union bound and the above, we get that the probability of a forbidden configuration is

$$\Pr[B''(u) = \psi(e) = B''(v)] \leq 2S + 2L.$$

To prove the claim, observe that from Definition 4 and the assumption that B' produces a sink at u with probability at most ℓ, it follows that

$$\begin{aligned} \ell &\geq \Pr[u \text{ is a sink}] \\ &\geq \Pr[u \text{ is a sink} \mid N^{t-1}(u) \text{ is bad}] \cdot \Pr[N^{t-1}(u) \text{ is bad}] \\ &> SL^3. \end{aligned}$$

Therefore, $\ell > SL^3$. By setting $L = \ell^{1/4}$ we get that $S < \ell^{1/4}$ implying $2S + 2L < 4\ell^{1/4}$. $\square$

4.3 The Speedup Lemma

LEMMA 8. *Suppose B is a sinkless colouring algorithm that runs in time t such that for any edge e the probability that B produces a forbidden configuration at e is at most p. Then there is a sinkless colouring algorithm B'' that runs in $t-1$ rounds such that it produces a forbidden configuration at any edge with probability less than $4 \cdot 6^{1/4} \cdot p^{1/12}$.*

PROOF. Follows from Lemmas 4 and 7. $\square$

5. LOWER BOUNDS

LEMMA 9. *Fix $d \geq 3$. There exists an infinite family of d-regular graphs $\mathcal{G}$ such that the edges of every $G \in \mathcal{G}$ can be coloured with d colours and the girth of G is $\Omega(\log n)$.*

PROOF. Let $\mathcal{H}$ be an infinite family of d-regular graphs with girth $\Theta(\log n)$; see e.g. [8, Ch. 3] how to obtain one. For any graph $H = (U,F) \in \mathcal{H}$ consider its bipartite double cover $G = (V,E)$, where $V = \{u_i : u \in U, i \in \{0,1\}\}$ and $E = \{\{u_i, v_{1-i}\} : \{u,v\} \in F, i \in \{0,1\}\}$. Note that G is also d-regular and has girth of $\Theta(\log n)$. By König's line colouring theorem the edges of G can be coloured with d colours [15, Ch. 5.3]. $\square$

THEOREM 3. *There does not exist a Monte Carlo distributed algorithm solving the sinkless colouring problem in d-regular graphs, where $d \geq 3$, with high probability in $o(\log \log n)$ rounds.*

PROOF. For the sake of contradiction, suppose A is an algorithm that solves sinkless colouring in $f_c(n) \in o(\log\log n)$ rounds with probability at least $1 - 1/n^c$ for an arbitrarily large constant c. Now fix a sufficiently large 3-regular graph G of n nodes given by Lemma 9.

Let $t = f_c(n)$ and for $i \in \{0, \ldots, t\}$ let A_i be the algorithm attained after i iterated applications of Lemma 8. Let p_i be the probability that A_i produces a forbidden configuration at any given edge e. By assumption $p_0 \leq 1/n^c$ and from Lemma 8 it follows that $p_{i+1} \leq z p_i^{1/12}$, where $z = 4 \cdot 6^{1/4}$. In particular, the probability that algorithm A_t running in 0 rounds produces a forbidden configuration at edge e is

$$p_t \leq z^s p_0^{1/12^t} \leq z^s q(n,c),$$

where

$$s = \sum_{i=0}^{t} 1/12^i < 2 \quad \text{and} \quad q(n,c) = n^{-c/(12^t)}.$$

By applying the union bound we get that for any node u executing A_t, the probability that one of its incident edges has a forbidden configuration is at most

$$\sum_{e : u \in e} z^2 q(n,c) < 3z^2 q(n,c).$$

Since $f_c(n) \in o(\log\log n)$, picking a sufficiently large n yields $t = f_c(n) \leq (\log\log n)/4$ and $1/12^t \geq 1/\log n$. It follows that $q(n,c) \leq 1/2^c$ and by setting $c = \log(30z^2) \in O(1)$ we obtain

$$\begin{aligned} &\Pr[\text{node } u \text{ is incident to a forbidden configuration}] \\ &\leq 3z^2 \cdot 1/2^c \leq 1/10. \end{aligned}$$

Finally, observe that in 0 rounds, all nodes choose their output independently of each other (and using the same algorithm). Since each node needs to output a colour, at least one colour c out of the three colours is picked with probability at least 1/3. Now the probability that node u has an edge of colour c with a forbidden configuration is at least $1/3^2 > 1/10$, which is a contradiction. $\square$

As observed in Section 2.4 we can obtain a sinkless colouring from a sinkless orientation without communication. This implies the following result.

COROLLARY 1. *There does not exist any Monte Carlo distributed algorithm solving the sinkless orientation problem in d-regular graphs, where $d \geq 3$, with high probability in $o(\log \log n)$ rounds.*

Together with Theorem 2 we get our main result. Note that in the following theorem we can plug in, for example, either of the commonly used LLL criteria: $ep(d+1) \leq 1$ or $4pd \leq 1$.

COROLLARY 2. *Let $f : \mathbb{N} \to \mathbb{R}$ be such that $f(4) \leq 16$. Let A be a Monte Carlo distributed algorithm for LLL that finds an assignment avoiding all the bad events under the LLL criterion $pf(d) \leq 1$ with high probability. Then the running time of A is $\Omega(\log \log n)$ rounds.*

Since any proper d-colouring of the nodes is also a sinkless colouring, we get the following lower bound as a by-product. Our lower bound can be contrasted with Linial's classical result [28]: Linial shows that any algorithm colouring a d-regular tree of radius r in $2r/3$ rounds needs $\Omega(\sqrt{d})$ colours, and this result can be strengthened to $\Omega(d/\log d)$ colours using the graph constructions of Alon [2]. However, Linial's technique does not seem to imply any nontrivial lower bounds for the case of d colours.

COROLLARY 3. *There does not exist any Monte Carlo distributed algorithm that finds a d-colouring in d-regular, bipartite, $\Omega(\log n)$-girth graphs with high probability in $o(\log \log n)$ rounds.*

6. CONCLUSIONS

In this paper, we have shown that any Monte Carlo distributed algorithm for the sinkless orientation problem requires $\Omega(\log \log n)$ rounds even in the case of 3-regular graphs. In particular, as the existence of such orientations can be shown using the symmetric version of the Lovász local lemma with the classical $ep(d+1) \leq 1$ criterion, it follows that any algorithm for the distributed Lovász local lemma working under this criterion also requires $\Omega(\log \log n)$ communication rounds.

An exponential gap still remains between the lower bound and the best known upper bound of $O(\log n \log d)$ rounds [11, 19] for the classical version of LLL. However, Chung et al. [11] note that it is possible to break the $O(\log n)$ barrier for a weaker version of LLL. Thus, a natural avenue of further investigation would be to gain better understanding on the true complexity of the distributed Lovász local lemma and on what are the trade-offs between various criteria.

Subsequently to our work, Chang et al. [10] gave a separation result regarding the power of randomisation in the LOCAL model: d-colouring trees with maximum degree d has randomised round complexity of $\Theta(\log_d \log n)$ for any $d \geq 55$, but the deterministic round complexity is $\Theta(\log_d n)$ for any $d \geq 3$. Moreover, they showed that in bounded-degree graphs, a lower bound of $\omega(\log^* n)$ for randomised algorithms implies a bound of $\Omega(\log n)$ for deterministic algorithms. It follows that the *deterministic* complexity of finding sinkless orientations is also $\Omega(\log n)$.

7. ACKNOWLEDGEMENTS

The problem of sinkless orientations in the context of distributed computing was originally introduced by Mika Göös. We have discussed this problem and its variants with numerous people—including, at least, Laurent Feuilloley, Pierre Fraigniaud, Teemu Hankala, Joel Kaasinen, Petteri Kaski, Janne H. Korhonen, Juhana Laurinharju, Christoph Lenzen, Joseph S. B. Mitchell, Pekka Orponen, Thomas Sauerwald, Stefan Schmid, Przemysław Uznański, and Uri Zwick—many thanks to all of you! We also wish to thank the anonymous reviewers for their helpful comments and suggestions regarding the manuscript.

8. REFERENCES

[1] N. Alon. A parallel algorithmic version of the local lemma. *Random Structures & Algorithms*, 2(4):367–378, 1991.

[2] N. Alon. On constant time approximation of parameters of bounded degree graphs. In O. Goldreich, editor, *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*, pages 234–239. Springer, 2010. doi:10.1007/978-3-642-16367-8_14.

[3] N. Alon and J. H. Spencer. *The Probabilistic Method*. John Wiley & Sons, Hoboken, NJ, USA, third edition, 2008.

[4] L. Barenboim and M. Elkin. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Morgan & Claypool, 2013. doi:10.2200/S00520ED1V01Y201307DCT011.

[5] L. Barenboim, M. Elkin, and F. Kuhn. Distributed $(\Delta + 1)$-coloring in linear (in Δ) time. *SIAM Journal on Computing*, 43(1):72–95, 2014. doi:10.1137/12088848X.

[6] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. The locality of distributed symmetry breaking. In *Proc. 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, pages 321–330. IEEE Computer Society Press, 2012. doi:10.1109/FOCS.2012.60.

[7] J. Beck. An algorithmic approach to the Lovász local lemma. *Random Structures and Algorithms*, 2(4):343–365, 1991.

[8] B. Bollobás. *Extremal Graph Theory*. Academic Press, London, 1978.

[9] K. Chandrasekaran, N. Goyal, and B. Haeupler. Deterministic algorithms for the Lovász local lemma. *SIAM Journal on Computing*, 42(6):2132–2155, 2013. doi:10.1137/100799642.

[10] Y.-J. Chang, T. Kopelowitz, and S. Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model, February 2016. arXiv:1602.08166.

[11] K.-M. Chung, S. Pettie, and H.-H. Su. Distributed algorithms for the Lovász local lemma and graph coloring. In *Proc. 33rd ACM SIGACT-SIGOPS*

Symposium on Principles of Distributed Computing (PODC 2014), pages 134–143. ACM Press, 2014. doi:10.1145/2611462.2611465.

[12] A. Czumaj and C. Scheideler. Coloring nonuniform hypergraphs: A new algorithmic approach to the general Lovász local lemma. *Random Structures & Algorithms*, 17(3–4):213–237, 2000. doi:10.1002/1098-2418(200010/12)17:3/4<213::AID-RSA3>3.0.CO;2-Y.

[13] A. Czumaj and C. Scheideler. A new algorithmic approach to the general Lovász local lemma with applications to scheduling and satisfiability problems. In *Proc. 32nd Annual ACM Symposium on Theory of Computing (STOC 2000)*, pages 38–47. ACM, 2000. doi:10.1145/335305.335310.

[14] A. Czygrinow, M. Hańćkowiak, and W. Wawrzyniak. Fast distributed approximations in planar graphs. In *Proc. 22nd International Symposium on Distributed Computing (DISC 2008)*, volume 5218 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2008. doi:10.1007/978-3-540-87779-0_6.

[15] R. Diestel. *Graph Theory*. Springer, Berlin, 4th edition, 2010. http://diestel-graph-theory.com/.

[16] M. Elkin, S. Pettie, and H.-H. Su. $(2\Delta - 1)$-edge-coloring is much easier than maximal matching in the distributed setting. In *Proc. 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*, pages 355–370. SIAM, 2015. doi:10.1137/1.9781611973730.26.

[17] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 2(2):609–627, 1975.

[18] D. Gamarnik and M. Sudan. Limits of local algorithms over sparse random graphs. In *Proc. 5th Conference on Innovations in Theoretical Computer Science (ITCS 2014)*, pages 369–376. ACM Press, 2014. doi:10.1145/2554797.2554831. arXiv:arXiv:1304.1831.

[19] M. Ghaffari. An improved distributed algorithm for maximal independent set. In *Proc. 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*. SIAM, 2016. To appear. arXiv:1506.05093.

[20] M. Göös, J. Hirvonen, and J. Suomela. Lower bounds for local approximation. *Journal of the ACM*, 60(5):39:1–23, 2013. doi:10.1145/2528405. arXiv:1201.6675.

[21] M. Göös, J. Hirvonen, and J. Suomela. Linear-in-Δ lower bounds in the LOCAL model. *Distributed Computing*, 2015. doi:10.1007/s00446-015-0245-8. arXiv:1304.1007.

[22] B. Haeupler and D. G. Harris. Improved bounds and parallel algorithms for the Lovász local lemma, September 2015. arXiv:1509.06430.

[23] B. Haeupler, B. Saha, and A. Srinivasan. New constructive aspects of the Lovász local lemma. *Journal of the ACM*, 58(6):28:1–28:28, 2011. doi:10.1145/2049697.2049702. arXiv:arXiv:1001.1231v5.

[24] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally! In *Proc. 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC 2004)*, pages 300–309. ACM Press, 2004. doi:10.1145/1011767.1011811.

[25] F. Kuhn, T. Moscibroda, and R. Wattenhofer. The price of being near-sighted. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, pages 980–989. ACM Press, 2006. doi:10.1145/1109557.1109666.

[26] T. Leighton, B. Maggs, and A. W. Richa. Fast algorithms for finding O(congestion + dilation) packet routing schedules. *Combinatorica*, 19(3):375–401, 1999. doi:10.1007/s004930050061.

[27] C. Lenzen and R. Wattenhofer. Leveraging Linial's locality limit. In *Proc. 22nd International Symposium on Distributed Computing (DISC 2008)*, volume 5218 of *Lecture Notes in Computer Science*, pages 394–407. Springer, 2008. doi:10.1007/978-3-540-87779-0_27.

[28] N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992. doi:10.1137/0221015.

[29] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, 1986. doi:10.1137/0215074.

[30] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, Cambridge, UK, 2005.

[31] M. Molloy and B. Reed. Further algorithmic aspects of the local lemma. In *Proc. 30th Annual ACM Symposium on Theory of Computing (STOC 1998)*, pages 524–529. ACM, 1998. doi:10.1145/276698.276866.

[32] M. Molloy and B. Reed. *Graph Colouring and the Probabilistic Method*. Springer-Verlag, Berlin, 2002.

[33] R. Moser. A constructive proof of the Lovász local lemma. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 343–350. ACM, 2009. doi:10.1145/1536414.1536462.

[34] R. A. Moser and G. Tardos. A constructive proof of the general Lovász local lemma. *Journal of the ACM*, 57(2):11:1–11:15, 2010. doi:10.1145/1667053.1667060. arXiv:0903.0544.

[35] M. Naor. A lower bound on probabilistic algorithms for distributive ring coloring. *SIAM Journal on Discrete Mathematics*, 4(3):409–412, 1991. doi:10.1137/0404036.

[36] M. Naor and L. Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

[37] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, 2000.

[38] J. B. Shearer. On a problem of Spencer. *Combinatorica*, 5(3):241–245, 1985.

[39] A. Srinivasan. An extension of the Lovász local lemma, and its applications to integer programming. *SIAM Journal on Computing*, 36(3):609–634, 2006.

[40] A. Srinivasan. Improved algorithmic versions of the Lovász local lemma. In *Proc. 19h Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008)*, pages 611–620. SIAM, 2008.

[41] J. Suomela. Survey of local algorithms. *ACM Computing Surveys*, 45(2):24:1–40, 2013. doi:10.1145/2431211.2431223. http://www.cs.helsinki.fi/local-survey/.