# The Distributed Complexity of Locally Checkable Labeling Problems Beyond Paths and Trees

**Yi-Jun Chang**
National University of Singapore, Singapore

## Abstract

We consider *locally checkable labeling* (LCL) problems in the LOCAL model of distributed computing. Since 2016, there has been a substantial body of work examining the possible complexities of LCL problems. For example, it has been established that there are no LCL problems exhibiting deterministic complexities falling between $\omega(\log^* n)$ and $o(\log n)$. This line of inquiry has yielded a wealth of algorithmic techniques and insights that are useful for algorithm designers.

While the complexity landscape of LCL problems on general graphs, trees, and paths is now well understood, graph classes beyond these three cases remain largely unexplored. Indeed, recent research trends have shifted towards a fine-grained study of special instances within the domains of paths and trees.

In this paper, we generalize the line of research on characterizing the complexity landscape of LCL problems to a much broader range of graph classes. We propose a conjecture that characterizes the complexity landscape of LCL problems for an *arbitrary* class of graphs that is closed under minors, and we prove a part of the conjecture.

Some highlights of our findings are as follows.

- We establish a simple characterization of the minor-closed graph classes sharing the same deterministic complexity landscape as paths, where $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$ are the only possible complexity classes.
- It is natural to conjecture that any minor-closed graph class shares the same complexity landscape as trees if and only if the graph class has bounded treewidth and unbounded pathwidth. We prove the "only if" part of the conjecture.
- For the class of graphs with pathwidth at most $k$, we show the existence of LCL problems with randomized and deterministic complexities $\Theta(n), \Theta(n^{1/2}), \Theta(n^{1/3}), \ldots, \Theta(n^{1/k})$ and the non-existence of LCL problems whose deterministic complexity is between $\omega(\log^* n)$ and $o(n^{1/k})$. Consequently, in addition to the well-known complexity landscapes for paths, trees, and general graphs, there are infinitely many different complexity landscapes among minor-closed graph classes.

**2012 ACM Subject Classification** Theory of computation → Distributed algorithms

**Keywords and phrases** Distributed graph algorithms, locality

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2024.26

**Related Version** *Full Version*: https://arxiv.org/abs/2311.06726 [20]

## 1 Introduction

In the LOCAL model of distributed computing, introduced by Linial [64], a communication network is modeled as an $n$-node graph $G = (V, E)$. In this representation, each node $v \in V$ corresponds to a computer, and each edge $e \in E$ corresponds to a communication link. In each communication round, each node sends a message to each of its neighbors, receives a message from each of its neighbors, and then performs some local computation.

The *complexity* of a distributed problem in the LOCAL model is defined as the smallest number of communication rounds needed to solve the problem, with unlimited local computation power and message sizes. Intuitively, the complexity of a distributed problem is the greatest distance that information needs to traverse within a network to attain a solution, capturing the fundamental concept of *locality* in the field of distributed computing.


15th Innovations in Theoretical Computer Science Conference (ITCS 2024).
Editor: Venkatesan Guruswami; Article No. 26; pp. 26:1–26:25
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are two variants of the LOCAL model. In the *deterministic* variant, each node has a unique identifier of $O(\log n)$ bits. In the *randomized* variant, there are no identifiers, and each node has the ability to generate local random bits. Throughout the paper, unless otherwise specified, we assume that the result under consideration applies to both the deterministic and randomized settings.

## 1.1 Locally Checkable Labeling

A distributed problem on bounded-degree graphs is a *locally checkable labeling* (LCL) if there is some constant $r$ such that the correctness of a solution can be checked locally in $r$ rounds of communication in the LOCAL model. The class of LCL problems encompasses many well-studied problems in distributed computing, including maximal independent set, maximal matching, $(\Delta + 1)$ vertex coloring, sinkless orientation, and many variants of these problems.

Formally, an LCL problem $\mathcal{P}$ is defined by the following parameters.

- An upper bound of the maximum degree $\Delta = O(1)$.
- A locality radius $r = O(1)$.
- A finite set of input labels $\Sigma_{\text{in}}$.
- A finite set of output labels $\Sigma_{\text{out}}$.
- A set of allowed configurations $\mathcal{C}$.

Each member of $\mathcal{C}$ is a graph $H = (V', E')$ whose maximum degree is at most $\Delta$ with a distinguished *center* $v \in V'$ such that each node in $H$ is within distance $r$ to $v$ and is assigned an input label from $\Sigma_{\text{in}}$ and an output label from $\Sigma_{\text{out}}$. The special case of $|\Sigma_{\text{in}}| = 1$ corresponds to the case where there is no input label. Since all of $r$, $\Delta$, $\Sigma_{\text{in}}$, and $\Sigma_{\text{out}}$ are finite, $\mathcal{C}$ is also finite.

An *instance* of an LCL problem $\mathcal{P}$ is a graph $G = (V, E)$ whose maximum degree is at most $\Delta$ where each node is assigned an input label from $\Sigma_{\text{in}}$. A *solution* for $\mathcal{P}$ on $G$ is a labeling $\phi_{\text{out}}$ that assigns to each node in $G$ an output label from $\Sigma_{\text{out}}$. The solution $\phi_{\text{out}}$ is *correct* if the $r$-radius neighborhood of each node $v \in V$ is isomorphic to an allowed configuration in $\mathcal{C}$ centered at $v$. It is straightforward to generalize the above definition to allow edge orientations and edge labels.

## 1.2 The Complexity Landscape of LCL Problems

The first systematic study of LCL problems in the LOCAL model was done by Naor and Stockmeyer [66]. They showed that randomness does not help for LCL problems whose complexity is $O(1)$, and they also showed that it is *undecidable* to determine whether an LCL problem can be solved in $O(1)$ rounds.

Since 2016, there has been a substantial body of work examining the possible complexities of LCL problems [17, 11, 3, 8, 9, 23, 24, 19, 25, 43, 53]. For example, it has been established that there are no LCL problems exhibiting deterministic complexities falling between $\omega(\log^* n)$ and $o(\log n)$ [17, 23, 67]. The complexity landscape of LCL problems on general graphs, trees, and paths is now well understood. For trees and paths, complete classifications were known: The complexity of *any* LCL problem on trees or paths must belong to one of the following complexity classes.

**Trees:** $O(1)$, $\Theta(\log^* n)$, $\Theta(\log \log n)$, $\Theta(\log n)$, and $\Theta(n^{1/k})$ for each positive integer $k$.

**Paths:** $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$.

All these complexity classes apply to both randomized and deterministic settings, except that the complexity class $\Theta(\log\log n)$ only appears in the randomized setting. Moreover, if an LCL problem has randomized complexity $\Theta(\log\log n)$ on trees, then its deterministic complexity must be $\Theta(\log n)$ on trees.

**Implications.** This line of research is not only interesting from a complexity-theoretic standpoint but has also yielded insights of relevance to algorithm designers. The derandomization theorem proved in [23] illustrates that the *graph shattering* technique [12] employed in many randomized distributed algorithms gives optimal algorithms to the LOCAL model. The *distributed constructive Lovász local lemma* problem [29] was shown [24] to be *complete* for sublogarithmic randomized complexity in a sense similar to the theory of NP-completeness, motivating a series of subsequent research on this problem [22, 37, 43, 48, 73].

The proof of some of the complexity gaps is *constructive*. For example, the proof that there is no LCL problem on trees whose complexity is $\omega(\log n)$ and $n^{o(1)}$ given in [24] demonstrates an algorithm such that for any given LCL problem $\mathcal{P}$ on trees, the algorithm either outputs a description of an $O(\log n)$-round algorithm solving $\mathcal{P}$ or decides that the complexity of $\mathcal{P}$ is $n^{\Omega(1)}$. Such a result suggests that the design of distributed algorithms could be *automated* in certain settings. Indeed, several recent research endeavors in this field have focused on attaining simple characterizations of various complexity classes of LCL problems that yield efficient algorithms for the automated design of distributed algorithms [3, 6, 5, 4, 18, 19, 25]. In particular, for LCL problems on paths without input labels, the task of designing an asymptotically optimal distributed algorithm can be done in polynomial time [18, 25].

Some of the algorithms for the automated design of distributed algorithms are practical and have been implemented. These algorithms can be used to efficiently discover non-trivial results such as an $O(1)$-round algorithm for maximal independent set on bounded-degree rooted trees [5].

**Extensions.** The study of the complexity landscape of LCL problems has been extended to other variants of the LOCAL model: online and dynamic settings [1], message size limitation [10], volume complexity [72], and node-averaged complexity [7]. Following the seminal work of Bernshteyn [13], many connections between the complexity classes of LCL problems in the LOCAL model and the complexity classes arising from *descriptive combinatorics* have been established [16, 52, 51].

## 1.3 Our Focus: Minor-Closed Graph Classes

While the complexity landscape of LCL problems on general graphs, trees, and paths is now well understood, graph classes beyond these three cases remain largely unexplored. Indeed, recent research trends in this field have shifted towards a fine-grained study of special instances within the domains of paths and trees: regular trees [5, 4, 16], rooted trees [5, 4], trees with binary input labels [6], and paths without input labels [25].

In contrast, a substantial body of work already exists concerning the design and analysis of distributed graph algorithms for various classes of networks beyond paths and trees in the LOCAL model [2, 14, 27, 32, 30, 31, 33, 35, 34, 36, 62, 75], so there currently exists a considerable gap between the complexity-theoretic and algorithmic understanding of locality in distributed computing.

To address this issue, let us consider the following generic question: Can we characterize the set of possible complexity classes of LCL problems for any given graph class $\mathcal{G}$? As any set of graphs is a graph class, it is possible to construct artificial graph classes to realize various strange complexity landscapes. To obtain meaningful interesting results, we must restrict our attention to some *natural* graph classes.

**Minor-closed graph classes.** In this work, we focus on characterizing the possible complexity classes of LCL problems on any given *minor-closed* graph class. The minor-closed graph classes are among the most prominent types of sparse graphs, covering many natural sparse graph classes, such as forests, cacti, planar graphs, bounded-genus graphs, and bounded-treewidth graphs.

A graph $H$ is a *minor* of $G$ if $H$ can be obtained from $G$ by removing nodes, removing edges, and contracting edges. Alternatively, $H$ is a minor of $G$ if there exist a partition of $V(G)$ into $k = |V(H)|$ disjoint connected clusters $\mathcal{C} = \{V_1, V_2, \ldots, V_k\}$ and a bijection between $\mathcal{C} = \{V_1, V_2, \ldots, V_k\}$ and $V(H)$ such that for each edge $e$ in $H$, the two clusters in $\mathcal{C}$ corresponding to the two endpoints of $e$ are adjacent in $G$.

Any set $\mathcal{G}$ of graphs is called a *graph class*. A graph class $\mathcal{G}$ is said to be *minor-closed* if $G \in \mathcal{G}$ implies that all minors $H$ of $G$ also belong to $\mathcal{G}$. Alternatively, a graph class $\mathcal{G}$ is minor-closed if it is closed under removing nodes, removing edges, and contracting edges.

A cornerstone result in structural graph theory is the *graph minor theorem* of Robertson and Seymour [71], which implies that for *any* minor-closed graph class $\mathcal{G}$, there is a *finite* list of graphs $H_1, H_2, \ldots, H_k$ such that $G \in \mathcal{G}$ if and only if $G$ does not contain any of $H_1, H_2, \ldots, H_k$ as a minor. Thus, any minor-closed graph class has a finite description in terms of a list of forbidden minors. The ideas developed in the proof of the graph minor theorem hold significance not only for mathematicians but also prove highly valuable in algorithm design and analysis. This has given rise to a thriving research field known as *algorithmic graph minor theory* [39, 38].

## 1.4 Our Contribution

The main contribution of this work is the formulation of a conjecture that characterizes the complexity landscape of LCL problems for an *arbitrary* class of graphs that is closed under minors. We present the conjecture in Section 2. To substantiate the conjecture, we provide a collection of results in Section 3, which collectively serve as a partial validation of the proposed conjecture. For the sake of presentation, detailed technical proofs for the assertions made in Sections 2 and 3 are left to. We conclude the paper with a discussion of potential future directions in Section 8.

## 1.5 Additional Related Work

The CONGEST model of distributed computing is a variant of the LOCAL model with an $O(\log n)$-bit message size constraint. There has been substantial research dedicated to utilizing the structural graph properties of minor-closed graph classes in the design of efficient algorithms in the CONGEST model: tree decomposition and its applications [60], low-congestion shortcut and its applications [47, 56, 57, 59, 58, 55], planarity testing and its applications [46, 50, 63], expander decomposition and its applications [21, 26].

In *local certification*, labels are assigned to nodes in a network to certify some property of the network. The certification is local in that the checking process is done by an $O(1)$-round LOCAL algorithm. Researchers have developed local certification algorithms tailored to various minor-closed graph classes [15, 41, 40, 45, 42, 65]. Akin to the study of LCL problems in the LOCAL model, some *algorithmic meta-theorems* that apply to a wide range of graph properties have been established for local certification [40, 45].

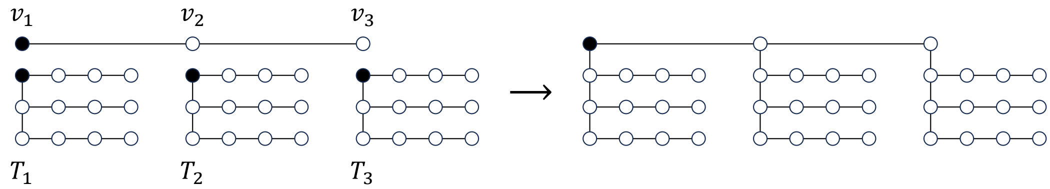


**Figure 1** The construction of $T_{3,3}$ in Definition 2.

## 2 Our Conjecture

We first present some terminology that is needed to state our conjecture.

▶ **Definition 1.** *For any two positive integers $k$ and $s$, the rooted tree $T_{k,s}$ is defined as follows.*

- *For $k = 1$, $T_{1,s}$ is an $s$-node path $(v_1, v_2, \ldots, v_s)$, where $v_1$ is designated as the root.*
- *For each $k > 1$, $T_{k,s}$ is constructed as follows. Start from an $s$-node path $(v_1, v_2, \ldots, v_s)$ and $s$ copies $T_1, T_2, \ldots, T_s$ of $T_{k-1,s}$. For each $j \in [s]$, add an edge connecting $v_j$ and the root of $T_j$. Designate $v_1$ as the root.*

Intuitively, the rooted tree $T_{k,s}$ in Definition 1 can be seen as a $k$-level hierarchical combination of $s$-node paths. The significance of $T_{k,s}$ to the complexity landscape of LCL problems lies in its role as a hard instance. Specifically, both $T_{k,s}$ and its variants have been identified as hard instances for LCL problems in the complexity class $\Theta(n^{1/k})$. These trees have been employed as lower-bound graphs in *all* existing $\Omega(n^{1/k})$ lower-bound proofs for such LCL problems [4, 5, 24, 19]. See Figure 1 for an illustration of the construction of $T_{3,3}$ from three copies $T_1$, $T_2$, and $T_3$ of $T_{2,3}$ and a 3-node path $(v_1, v_2, v_3)$, where the roots are drawn in black.

▶ **Definition 2.** *For each non-negative integer $k$, define $\mathcal{A}_k$ as the set of all minor-closed graph classes $\mathcal{G}$ meeting the following two conditions.*

**(C1)** *If $k \geq 1$, then $T_{k,s} \in \mathcal{G}$ for all positive integers $s$.*
**(C2)** *There exists a positive integer $s$ such that $T_{k+1,s} \notin \mathcal{G}$.*

For each positive integer $k$, $\mathcal{A}_k$ is the set of all minor-closed graph classes $\mathcal{G}$ that includes all of $T_{k,s}$ and excludes some of $T_{k+1,s}$. For the special case of $k = 0$, (C1) is vacuously true, so $\mathcal{A}_0$ is the set of all minor-closed graph classes $\mathcal{G}$ that excludes some of $T_{1,s}$.

**Pathwidth.** A *path decomposition* of a graph $G = (V, E)$ is a sequence $(X_1, X_2, \ldots, X_k)$ of subsets of $V$ meeting the following conditions.

**(P1)** $X_1 \cup X_2 \cup \cdots \cup X_k = V$.
**(P2)** For each node $v \in V$, there are two indices $i$ and $j$ such that $v \in X_l$ if and only if $i \leq l \leq j$.
**(P3)** For each edge $e = \{u, v\} \in E$, there is an index $i$ such that $\{u, v\} \subseteq X_i$.

The *width* of a path decomposition $(X_1, X_2, \ldots, X_k)$ is $\max_{1 \leq i \leq k} |X_i| - 1$. The *pathwidth* of a graph is the minimum width over all path decompositions of the graph. Intuitively, the pathwidth of a graph measures how similar it is to a path. A graph class $\mathcal{G}$ is said to have *bounded pathwidth* if there is a finite number $C$ such that each $G \in \mathcal{G}$ has pathwidth at most $C$.

It is well-known that for any integer $k$, the set of all graphs with pathwidth at most $k$ is a minor-closed graph class. We emphasize that it is, however, possible that a bounded-pathwidth graph class is not minor-closed. One such example is the set of all graphs with pathwidth at most $k$ and containing at least two nodes.

Throughout the paper, we write $\mathcal{A}$ to denote the set of all minor-closed graph classes whose pathwidth is bounded. Formally, $\mathcal{G} \in \mathcal{A}$ if $\mathcal{G}$ is minor-closed and there is a finite number $C$ such that the pathwidth of each $G \in \mathcal{G}$ is at most $C$.

For the sake of presentation, the proof of the following two statements is deferred to Section 4.

▶ **Proposition 3.** *$\mathcal{A} = \bigcup_{0 \le i < \infty} \mathcal{A}_i$ is a partition of $\mathcal{A}$ into disjoint sets.*

Proposition 3 shows that $\bigcup_{0 \le i < \infty} \mathcal{A}_i$ is a classification of all bounded-pathwidth minor-closed graph classes.

▶ **Proposition 4.** *For every integer $k \ge 1$, the class of all graphs with pathwidth at most $k$ is in $\mathcal{A}_k$.*

We emphasize that the converse of Proposition 4 does not hold in the following sense: $\mathcal{G} \in \mathcal{A}_k$ does not imply that each $G \in \mathcal{G}$ has pathwidth at most $k$. For example, if we let $\mathcal{G}$ be the set of all graphs with at most $k+1$ nodes, then $\mathcal{G} \in \mathcal{A}_0$ because it does not contain the $(k+2)$-node path $T_{1,k+2}$, but $\mathcal{G}$ contains the $(k+1)$-node complete graph, whose pathwidth is $k$.

**The conjecture.** For any graph class $\mathcal{G}$ and a positive integer $\Delta$, we write $\mathcal{G}^{\Delta}$ to denote the set of graphs in $\mathcal{G}$ whose maximum degree is at most $\Delta$. In the subsequent discussion, we informally say that a range of complexities is *dense* to indicate that many different complexity functions in this range can be realized by LCL problems in a way reminiscent of the famous *time hierarchy theorem* for Turing machines. We are now ready to state our conjecture that characterizes the complexity landscape of LCL problems for an *arbitrary* minor-closed graph class.

▶ **Conjecture 5.** *Let $\mathcal{G}$ be a minor-closed graph class that is not the class of all graphs, and let $\Delta \ge 3$ be an integer. The complexity landscape of* LCL *problems on $\mathcal{G}^{\Delta}$ is characterized as follows.*

- *If $\mathcal{G} \in \mathcal{A}_0$, then $O(1)$ is the only possible complexity class.*
- *If $\mathcal{G} \in \mathcal{A}_k$ for some $k > 0$, then the possible complexity classes are exactly*

$$O(1), \Theta(\log^* n), \Theta\left(n^{1/k}\right), \Theta\left(n^{1/(k-1)}\right), \ldots, \Theta(n).$$

- *If $\mathcal{G}$ has unbounded pathwidth and bounded treewidth, then $\mathcal{G}$ shares the same complexity landscape as trees. In other words, the possible complexity classes are exactly*

$$O(1), \Theta(\log^* n), \Theta(\log\log n), \Theta(\log n), \text{ and } \Theta\left(n^{1/k}\right) \text{ for all positive integers } k.$$

- *If $\mathcal{G}$ has unbounded treewidth, then the possible complexity classes are exactly*

$$O(1), \Theta(\log^* n), \Theta(\log\log n), \text{ and a dense region } [\Theta(\log n), \Theta(n)].$$

Similar to the case of trees discussed earlier, all complexity classes in Conjecture 5 apply to both randomized and deterministic settings, except that the complexity $\Theta(\log\log n)$ only exists in the randomized setting.

## 3 Supporting Evidence

In this section, we show a collection of results that prove a part of Conjecture 5. We say that an LCL problem $\mathcal{P}$ is *solvable* in a graph class $\mathcal{G}$ if all graphs in $\mathcal{G}$ admit a correct solution for $\mathcal{P}$. Let us first consider $\mathcal{A}_0$. Intuitively, $\mathcal{G} \in \mathcal{A}_0$ means that there is some length bound $\ell$ such that all graphs in $\mathcal{G}$ do not contain a minor isomorphic to the $\ell$-length path. Therefore, all members in $\mathcal{G}$ have bounded diameter, so as long as the considered LCL problem $\mathcal{P}$ is solvable, $\mathcal{P}$ can be solved in $O(1)$ rounds in the LOCAL model by a brute-force information gathering.

▶ **Theorem 6.** *Let $\Delta \geq 3$ be an integer, and let $\mathcal{G} \in \mathcal{A}_0$. All* LCL *problems that are solvable in $\mathcal{G}^\Delta$ can be solved in $O(1)$ rounds in $\mathcal{G}^\Delta$.*

Our classification $\mathcal{A} = \bigcup_{0 \leq i < \infty} \mathcal{A}_i$ makes sense from two different points of view. The first viewpoint is to consider the special role of the trees $T_{k,s}$ as hard instances in the study of the complexity of LCL problems.

Recall from an earlier discussion that *all* existing $\Omega(n^{1/k})$ lower-bound proofs for LCL in the complexity class $\Theta(n^{1/k})$ in trees are based on the tree $T_{k,s}$ or its variants. In view of the definition of $\mathcal{A}_k$, for any $\mathcal{G} \in \mathcal{A}_k$. we expect that the complexity class $\Theta(n^{1/s})$ exists for all $s \in \{1, 2, \ldots, k\}$.

▶ **Theorem 7.** *Let $k \geq 1$, $\Delta \geq 3$, and $s \in \{1, 2, \ldots, k\}$ be integers, and let $\mathcal{G} \in \mathcal{A}_k$. There is an* LCL *problem whose complexity in $\mathcal{G}^\Delta$ is $\Theta(n^{1/s})$.*

Similarly, for any $\mathcal{G} \in \mathcal{A}_k$. we expect that the complexity class $\Theta(n^{1/s})$ does not exist for all $s \in \{k+1, k+2, \ldots\}$. To establish this claim, we consider a different perspective.

In Section 5, we prove an alternative characterization of the set of graph classes $\mathcal{A}_k$ in terms of the growth rate of the size of the $d$-radius neighborhood. In particular, we show that $\mathcal{A}_k$ is precisely the set of all minor-closed graph classes $\mathcal{G}$ such that $k$ is the smallest number such that the size of the $d$-radius neighborhood of any node in any bounded-degree graph in $\mathcal{G}$ is $O(d^k)$.

The growth rate of the size of the $d$-radius neighborhood is relevant to the complexity landscape of LCL problems in that the growth rate affects the complexity gaps resulting from existing approaches. In particular, the alternative characterization of $\mathcal{A}_k$, combined with the existing proofs [23, 24, 66] to establish complexity gaps for LCL problems in general graphs, yields the following two results.

▶ **Theorem 8.** *Let $k \geq 1$ and $\Delta \geq 3$ be integers, and let $\mathcal{G} \in \mathcal{A}_k$. There is no* LCL *problem whose deterministic complexity in $\mathcal{G}^\Delta$ is between $\omega(\log^* n)$ and $o(n^{1/k})$.*

▶ **Theorem 9.** *Let $k \geq 1$ and $\Delta \geq 3$ be integers, and let $\mathcal{G} \in \mathcal{A}_k$. There is no* LCL *problem whose complexity in $\mathcal{G}^\Delta$ is between $\omega(1)$ and $o\left((\log^* n)^{1/k}\right)$.*

The proof of Theorems 6–9 are deferred to Section 6.

**An infinitude of complexity landscapes.** Theorems 7 and 8 and Proposition 4 imply that, in addition to the well-known complexity landscapes for paths, trees, and general graphs, there are infinitely many different complexity landscapes among minor-closed graph classes.

For every $\Delta \geq 3$ and $k \geq 1$, let us consider the class of graphs with maximum degree $\Delta$ and pathwidth at most $k$. For this class of graphs, there exist LCL problems with randomized and deterministic complexities $\Theta(n), \Theta(n^{1/2}), \Theta(n^{1/3}), \ldots, \Theta(n^{1/k})$, and there does not exist an LCL problem whose deterministic complexity is between $\omega(\log^* n)$ and $o(n^{1/k})$. These results already guarantee that the complexity landscapes necessarily vary for different values of $k$.

**Algorithmic implications.** Proposition 4 and Theorem 8 imply that any $n^{o(1)}$-round deterministic distributed algorithm $\mathcal{A}$ for any LCL problem on bounded-pathwidth graphs can be automatically turned into an $O(\log^* n)$-round deterministic algorithm $\mathcal{A}'$ solving the same problem on bounded-pathwidth graphs. This allows us to *automatically speed up* existing algorithms significantly on bounded-pathwidth graphs.

▶ **Corollary 10.** *The following problems can be solved in $O(\log^* n)$ rounds deterministically on graphs of bounded pathwidth and bounded degree.*

- *Constructive Lovász local lemma with the condition $epd \le 1 - \delta$, for any constant $\delta > 0$.*
- *$\Delta$ vertex coloring.*
- *$(\Delta + 1)$ edge coloring.*

**Proof.** Given the discussion above, we just need to check that these LCL problems can be solved in $n^{o(1)}$ rounds deterministically on graphs with maximum degree $\Delta = O(1)$. Indeed, constructive Lovász local lemma can be solved in polylogarithmic rounds on general graphs [73], $\Delta$ vertex coloring can be solved in $O(\log^2 n)$ rounds on bounded-degree graphs [49], and $(\Delta + 1)$ edge coloring can be solved in $\text{poly}(\Delta, \log n)$ rounds on general graphs [28]. ◀

It seems to be a highly nontrivial task to explicitly construct the $O(\log^* n)$-round algorithms whose *existence* is guaranteed in Corollary 10, as the proof of Theorem 8 is non-constructive in the sense that it does not offer an algorithm that decides between the two cases $O(\log^* n)$ and $\Omega(n^{1/k})$.

## 3.1 Path-Like Graph Classes

Theorems 7–9 allow us to completely characterize the minor-closed graph classes whose deterministic complexity landscape is identical to that of paths: $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$.

▶ **Corollary 11.** *For any minor-closed graph class $\mathcal{G}$, the possible deterministic complexity classes for* LCL *problems on bounded-degree graphs in $\mathcal{G}$ are exactly $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$ if and only if $\mathcal{G} \in \mathcal{A}_1$.*

**Proof.** Suppose $\mathcal{G} \in \mathcal{A}_1$. Then Theorems 8 and 9 show that $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$ are the only possible deterministic complexity classes for LCL problems in bounded-degree graphs in $\mathcal{G}$. Indeed, it is well-known [25] that there exist LCL problems with deterministic complexities $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$ even if we restrict ourselves to path graphs. Since $\mathcal{G} \in \mathcal{A}_1$, $\mathcal{G}$ contains all path graphs, so $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$ are exactly the possible deterministic complexity classes for LCL problems on bounded-degree graphs in $\mathcal{G}$.

Suppose $\mathcal{G} \notin \mathcal{A}_1$. If $\mathcal{G} \in \mathcal{A}_k$ for some $k \neq 1$, then Theorems 6 and 7 show that the set of possible deterministic complexity classes for LCL problems on bounded-degree graphs in $\mathcal{G}$ cannot be $O(1)$, $\Theta(\log^* n)$, and $\Theta(n)$. If $\mathcal{G} \notin \mathcal{A}_k$ for all $k$, then $\mathcal{G}$ has unbounded pathwidth. The well-known excluding forest theorem [68] implies that any minor-closed graph class $\mathcal{G}$ with unbounded pathwidth must contain all trees. Hence the complexity landscape for $\mathcal{G}^\Delta$ includes all the complexity classes for trees, such as $\Theta(\log n)$, see [24]. ◀

**The complexity of the characterization.** The graph minor theorem [71] of Robertson and Seymour implies that if $\mathcal{G}$ is a minor-closed graph class, then there is a finite list of graphs $H_1, H_2, \ldots, H_k$ such that $G \in \mathcal{G}$ if and only if $G$ does not contain any of $H_1, H_2, \ldots, H_k$ as a minor. Therefore, any minor-closed graph class $\mathcal{G}$ admits a *finite* representation by listing its finite list of forbidden minors $\{H_1, H_2, \ldots, H_k\}$, so it makes sense to consider computational problems where the input is an arbitrary minor-closed graph class.

A common method to demonstrate the simplicity of a characterization is by establishing its polynomial-time computability. Given any minor-closed graph class $\mathcal{G}$, represented by a list of forbidden minors $H_1, H_2, \ldots, H_k$, is there an efficient algorithm deciding whether $\mathcal{G}$ is path-like in the sense of Corollary 11? We show an affirmative answer to this question. In fact, we prove a more general result which shows that for any fixed index $i$, whether $\mathcal{G} \in \mathcal{A}_i$ is decidable in time polynomial in the size of the representation of $\mathcal{G}$, which is a finite list of forbidden minors $\{H_1, H_2, \ldots, H_k\}$.

▶ **Proposition 12.** *For any fixed index $i$, there is a polynomial-time algorithm that, given a list of graphs $H_1, H_2, \ldots, H_k$, decides whether the class of $\{H_1, H_2, \ldots, H_k\}$-minor-free graphs $\mathcal{G}$ is in $\mathcal{A}_i$.*

The proof of Proposition 12 is in Section 7. We remark that although the algorithm of Proposition 12 finishes in polynomial time, the algorithm is unlikely to be practical in that the list of forbidden minors for the considered minor-closed graph class $\mathcal{G}$ is often not known.

## 3.2 Tree-Like Graph Classes

It is natural to conjecture that any minor-closed graph class shares the same complexity landscape as trees if and only if the graph class has bounded treewidth and unbounded pathwidth. We prove the "only if" part of the conjecture.

▶ **Corollary 13.** *A minor-closed graph class $\mathcal{G}$ shares the same complexity landscape as trees only if $\mathcal{G}$ has bounded treewidth and unbounded pathwidth.*

**Proof.** Theorem 8 implies that if a minor-closed graph class $\mathcal{G}$ has bounded pathwidth, then the complexity class $\Theta(\log n)$ disappears. Therefore, for $\mathcal{G}$ to have the same complexity landscape as that of trees, $\mathcal{G}$ must have unbounded pathwidth.

Now suppose $\mathcal{G}$ is a minor-closed graph class with unbounded treewidth. Then the well-known excluding grid theorem [69] implies that $\mathcal{G}$ contains all planar graphs.

It was shown in [8] that for any rational number $\frac{1}{2} \le c < 1$, there exists an LCL problem $\mathcal{P}$ that is solvable in $O(n^c)$ rounds for all graphs and requires $\Omega(n^c)$ rounds to solve for planar graphs. As $\mathcal{G}$ contains all planar graphs, this implies that the complexity of $\mathcal{P}$ is $\Theta(n^c)$ in $\mathcal{G}$, This shows that $[\Theta(\sqrt{n}), \Theta(n)]$ is a *dense region* in the complexity landscape for $\mathcal{G}$. Hence the complexity landscape for $\mathcal{G}$ is different from that of trees. ◀

**The range of the dense region.** As mentioned in the above proof, we see that the dense region $[\Theta(\sqrt{n}), \Theta(n)]$ exists in the complexity landscape for any minor-closed graph class $\mathcal{G}$ that has unbounded treewidth. In the full version of the paper [20], we generalize this result to extend the range of the dense region to cover the entire interval $[\Theta(\log n), \Theta(n)]$, which is the *widest possible* due to the $\omega(\log^* n) - o(\log n)$ gap shown in [23]. Specifically, we show that there are LCL problems with the following complexities.

▶ **Theorem 14.** *For any minor-closed graph class $\mathcal{G}$ that has unbounded treewidth, there are* LCL *problems on bounded-degree graphs in $\mathcal{G}$ with the following complexities.*

- $\Theta(n^c)$*, for each rational number $c$ such that $0 \le c \le 1$.*
- $\Theta(\log^c n)$*, for each rational number $c$ such that $c \ge 1$.*

**Existing approaches.** We briefly explain the construction of the LCL problem $\mathcal{P}$ in [8] which is used in the above proof of Corollary 13. The LCL problem uses locally checkable constraints to force the underlying network to encode an execution of a linear-space-bounded Turing machine $\mathcal{M}$ in a two-dimensional grid. Suppose the time complexity of $\mathcal{M}$ on the input string $0^s$ is $T(s)$. Running a simulation of $\mathcal{M}$ on $0^s$ requires $n = s \cdot T(s)$ nodes and $t = T(s)$ rounds. For any $\frac{1}{2} \le c < 1$, the round complexity function $t = \Theta(n^c)$ can be realized with $T(s) = s^{c/(1-c)}$. Since $T(s) = \Omega(s)$, $[\Theta(\sqrt{n}), \Theta(n)]$ is the largest possible dense region resulting from this approach.

It was shown in [8] that LCL problems with complexity $\Theta(n^c)$ exist by extending this construction to higher dimensional grids. This extension is not applicable for proving Theorem 14, due to the following reason. For any graph $H$ there exists a number $d$ such that $H$ is a minor of a sufficiently large $d$-dimensional grid. If a minor-closed graph class $\mathcal{G}$ contains arbitrarily large $d$-dimensional grids for all $d$, then $\mathcal{G}$ must be the set of all graphs.

With a completely different approach, in another previous work [11], the two dense regions $[\Theta(\log\log^* n), \Theta(\log^* n)]$ and $[\Theta(\log n), \Theta(n)]$ were shown for LCL problems on general graphs. Due to a similar reason, their construction of LCL problems is also not applicable for proving Theorem 14.

The proof in [11] relies on the following graph structure. Start with an $a \times b$ grid graph whose dimensions $a$ and $b$ can be arbitrarily large. Let $u_{i,j}$ denote the node on the $i$th row and the $j$th column. For each row $i$, add an edge between $u_{i,j_1}$ and $u_{i,j_2}$ if $j_2 - j_1 = i$. This graph contains a $k$-clique as a minor given that $k \le \min\{a, b\}$. To see this, contract the $j$th column into a node $v_j$ for each $1 \le j \le k$. Then $\{v_1, v_2, \ldots, v_k\}$ forms a clique. Thus, for the results in [11] to apply to a minor-closed graph class $\mathcal{G}$, $\mathcal{G}$ must contain the $k$-clique for all $k$. Since any graph is a minor of a sufficiently large clique, it follows that $\mathcal{G}$ is necessarily the set of all graphs.

**New ideas.** To establish the dense region $[\Theta(\log n), \Theta(n)]$, in the full version of the paper [20], we modify the construction of [8] by attaching a root-to-leaf path of a complete tree to one side of the grid used in the Turing machine simulation. This enables us to increase the number of nodes to be *exponential* in the round complexity of Turing machine simulation, allowing us to realize any reasonable LCL complexity in the region $[\Theta(\log n), \Theta(n)]$ and to prove Theorem 14.

## 3.3 Summary

For convenience, in the subsequent discussion, let $\mathcal{B}$ denote the set of all minor-closed graph classes $\mathcal{G}$ that has unbounded pathwidth and bounded treewidth, and let $\mathcal{C}$ denote the set of all minor-closed graph classes $\mathcal{G}$ that has unbounded treewidth and is not the set of all graphs.

**The state of the art.** We summarize the old and new results on the complexity of LCL problems as follows. For simplicity, here we only consider the deterministic LOCAL model.

$\mathcal{A}_0$ $\quad O(1)$

$\mathcal{A}_k$ $\quad O(1) \overset{\times}{—} \Theta\left((\overset{?}{\log^*} n)^{\frac{1}{k}}\right) \overset{?}{—} \Theta(\log^* n) \overset{\times}{—} \Theta\left(n^{\frac{1}{k}}\right) \overset{?}{—} \Theta\left(n^{\frac{1}{k-1}}\right) \overset{?}{—} \cdots \overset{?}{—} \Theta\left(n^{\frac{1}{2}}\right) \overset{?}{—} \Theta(n)$

$\mathcal{B}$ $\quad O(1) \overset{\times}{—} \Theta(\log \overset{?}{\log^*} n) \overset{?}{—} \Theta(\log^* n) \overset{\times}{—} \Theta(\log n) \overset{?}{—} \cdots \overset{?}{—} \Theta\left(n^{\frac{1}{3}}\right) \overset{?}{—} \Theta\left(n^{\frac{1}{2}}\right) \overset{?}{—} \Theta(n)$

$\mathcal{C}$ $\quad O(1) \overset{\times}{—} \Theta(\log \overset{?}{\log^*} n) \overset{?}{—} \Theta(\log^* n) \overset{\times}{—} \Theta(\log n) \overset{\text{dense}}{———} \Theta(n)$

The $\omega(\log^* n) - o(\log n)$ and $\omega(1) - o(\log\log^* n)$ gaps for $\mathcal{B}$ and $\mathcal{C}$ are due to [23, 24]. The existence of the complexity class $\Theta(n^{1/k})$ for all positive integers $k$ for $\mathcal{B}$ is due to [24]. The existence of the complexity class $\Theta(\log^* n)$ for $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{B}$, and $\mathcal{C}$ is due to the well-known fact that the complexity of 3-coloring paths is $\Theta(\log^* n)$. The existence of the complexity class $\Theta(\log n)$ for $\mathcal{B}$ and $\mathcal{C}$ is due to the well-known that the complexity of $\Delta$-coloring trees is $\Theta(\log n)$. All the remaining results are due to Theorems 6–9 and 14.

**The conjecture.** For the sake of comparison, here we also illustrate the complexity landscapes described in Conjecture 5.

| | |
|---|---|
| $\mathcal{A}_0$ | $O(1)$ |
| $\mathcal{A}_k$ | $O(1) \overset{\times}{—} \Theta(\log^* n) \overset{\times}{—} \Theta\left(n^{\frac{1}{k}}\right) \overset{\times}{—} \Theta\left(n^{\frac{1}{k-1}}\right) \overset{\times}{—} \cdots \overset{\times}{—} \Theta\left(n^{\frac{1}{2}}\right) \overset{\times}{—} \Theta(n)$ |
| $\mathcal{B}$ | $O(1) \overset{\times}{—} \Theta(\log^* n) \overset{\times}{—} \Theta(\log n) \overset{\times}{—} \cdots \overset{\times}{—} \Theta\left(n^{\frac{1}{3}}\right) \overset{\times}{—} \Theta\left(n^{\frac{1}{2}}\right) \overset{\times}{—} \Theta(n)$ |
| $\mathcal{C}$ | $O(1) \overset{\times}{—} \Theta(\log^* n) \overset{\times}{—} \Theta(\log n) \overset{\text{dense}}{——} \Theta(n)$ |

## 3.4 Roadmap

In Section 4, we prove that $\bigcup_{0\le i<\infty} \mathcal{A}_i$ is a classification of all bounded-pathwidth minor-closed graph classes. In Section 5, we show a combinatorial characterization of the set of graph classes $\mathcal{A}_i$ based on the growth rate of the size of the $d$-radius neighborhood. In Section 6, we use the combinatorial characterization to prove Theorems 6–9. In Section 7, we present a polynomial-time algorithm that decides whether $\mathcal{G} \in \mathcal{A}_i$ for any given minor-closed graph class $\mathcal{G}$. Due to the page limit, the proof of the existence of the dense region $[\Theta(\log n), \Theta(n)]$ for any minor-closed graph class that has unbounded treewidth is left to the full version of the paper [20].

# 4 A Classification of Bounded-Pathwidth Networks

In this section, we prove Propositions 3 and 4. That is, $\bigcup_{0\le i<\infty} \mathcal{A}_i$ is a classification of all bounded-pathwidth minor-closed graph classes, and $\mathcal{A}_k$ contains the class of graphs of pathwidth at most $k$. We need the following well-known characterization of bounded-pathwidth minor-closed graph classes by Robertson and Seymour [68].

▶ **Theorem 15** (Excluding forest theorem [68]). *A minor-closed graph class $\mathcal{G}$ has bounded pathwidth if and only if $\mathcal{G}$ does not contain all forests.*

Propositions 3 and 4 are proved using the observation that the pathwidth of $T_{i,s}$ equals $i$ whenever $s \ge 3$. The calculation of the pathwidth of $T_{i,s}$ can be done in a way similar to the folklore calculation to show that the pathwidth of a complete ternary tree of height $d$ is precisely $d$. We still present a proof of the observation here for the sake of completeness.

▶ **Observation 16.** *For any two integers $i$ and $s$ such that $i \ge 1$ and $s \ge 3$, the pathwidth of $T_{i,s}$ is $i$.*

We need an auxiliary lemma.

▶ **Lemma 17** ([74]). *Consider a rooted tree $T$ whose root $r$ has three children $u_1$, $u_2$, and $u_3$. If the pathwidth of the subtree rooted at $u_i$ is at least $k$ for each $i \in \{1, 2, 3\}$, then the pathwidth of $T$ is at least $k + 1$.*

We are now ready to prove Observation 16.

**Proof of Observation 16.** We prove the upper bound. For the base case, $(X_1, X_2, \ldots, X_{s-1})$ with $X_i = \{v_i, v_{i+1}\}$ is a path decomposition of the $s$-node path $(v_1, v_2, \ldots, v_s)$, showing that the pathwidth of $T_{i,s}$ is at most 1.

Given that the pathwidth of $T_{i-1,s}$ is at most $i-1$, we show that the pathwidth of $T_{i,s}$ is at most $i$. Let the $s$-node path $(v_1, v_2, \ldots, v_s)$ and $s$ instances $T_1, T_2, \ldots, T_s$ of $T_{i-1,s}$ be the ones in the definition of $T_{i,s}$. For each $1 \le j \le s$, let $(X_{1,j}, X_{2,j}, \ldots, X_{k,j})$ be a path decomposition of $T_j$ of width at most $i-1$. Define $X'_{l,j} = X_{l,j} \cup \{v_j\}$. Then

$$\left(X'_{1,1}, \ldots, X'_{k,1}, \{v_1, v_2\}, X'_{1,2}, \ldots, X'_{k,2}, \{v_2, v_3\}, \ldots, \{v_{s-1}, v_s\}, X'_{1,s}, \ldots, X'_{k,s}\right)$$

is a path decomposition of $T_{i,s}$ of width at most $i$.

For the rest of the proof, we consider the lower bound. It suffices to consider the case of $s = 3$, as the pathwidth of $T_{i,s}$ is at least the pathwidth of $T_{i,3}$ for each $s \ge 3$. For the base case, it is trivial that the pathwidth of $T_{1,3}$ is at least 1.

Given that the pathwidth of $T_{i-1,3}$ is at least $i-1$, we show that the pathwidth of $T_{i,3}$ is at least $i$. Consider the path $(v_1, v_2, v_3)$ in the definition of $T_{i,3}$. We re-root the tree by setting $v_2$ as the root. Now $v_2$ has three children. Let $T$ be any subtree of $T_{i,3}$ rooted at one of the children of $v_2$. Observe that $T$ contains $T_{i-1,3}$ as a subgraph, so the pathwidth of $T$ is at least $i-1$ by induction hypothesis. Applying Lemma 17 to $T_{i,3}$ rooted at $v_2$, we obtain that the pathwidth of $T_{i,3}$ is at least $i$. ◀

Using Observation 16, we now prove Propositions 3 and 4.

▶ **Proposition 3.** *$\mathcal{A} = \bigcup_{0 \le i < \infty} \mathcal{A}_i$ is a partition of $\mathcal{A}$ into disjoint sets.*

**Proof.** First of all, Theorem 15 implies that any minor-closed graph class $\mathcal{G} \notin \mathcal{A}$ cannot belong to $\mathcal{A}_i$ for any $i$, as $\mathcal{G}$ contains all forests, so $\bigcup_{0 \le i < \infty} \mathcal{A}_i \subseteq \mathcal{A}$.

We claim that $\mathcal{A}_1, \mathcal{A}_2, \ldots$ are disjoint sets. Suppose there are two indices $i$ and $j$ such that $\mathcal{G} \in \mathcal{A}_i$ and $\mathcal{G} \in \mathcal{A}_j$ and $i < j$. Then we have $T_{j',s} \in \mathcal{G}$ for all $j'$ and $s$ such that $1 \le j' \le j$ and $s \ge 1$ by (C1) in Definition 2, as $T_{j',s}$ is a minor of $T_{j,s}$. However, $\mathcal{G} \in \mathcal{A}_i$ implies that $T_{i+1,s} \notin \mathcal{G}$ for some $s$ by (C2) in Definition 2. This is a contradiction because $1 \le i+1 \le j$.

It remains to show that each graph class $\mathcal{G} \in \mathcal{A}$ belongs to $\mathcal{A}_i$ for some index $i$. As $\mathcal{G}$ has bounded pathwidth, let $k < \infty$ be the maximum pathwidth of graphs in $\mathcal{G}$. Then $T_{k+1,s} \notin \mathcal{G}$ for all $s \ge 3$ by Observation 16. We pick $i$ to be the smallest index such that $T_{i+1,s} \notin \mathcal{G}$ for some $s$. Such an index $i$ exists, and we must have $0 \le i \le k$. As a result, $\mathcal{G} \in \mathcal{A}_i$, as both (C1) and (C2) in Definition 2 are satisfied due to our choice of $i$. ◀

▶ **Proposition 4.** *For every integer $k \ge 1$, the class of all graphs with pathwidth at most $k$ is in $\mathcal{A}_k$.*

**Proof.** Let $\mathcal{G}$ be the graph class that contains all graphs with pathwidth at most $k$. We have $T_{k,s} \in \mathcal{G}$ for all positive integers $s$ by Observation 16, so (C1) in Definition 2 is satisfied. Here we use the fact that $T_{k,s'}$ is a minor of $T_{k,s}$ whenever $s' \le s$. Since $\mathcal{G}$ does not contain any graph with pathwidth at least $k+1$, by Observation 16, $T_{k+1,s} \notin \mathcal{G}$ for all $s \ge 3$, so (C2) in Definition 2 is satisfied. ◀

## 5 The Bounded Growth Property

In this section, we prove an alternative characterization of the set of graph classes $\mathcal{A}_i$ that will be crucial in the complexity-theoretic study in Section 6.

Let $g \geq 0$. We say that a graph class $\mathcal{G}$ is *g-growth-bounded* if, for every integer $\Delta \geq 3$, there exists a constant $C_\Delta > 0$ only depending on $\Delta$ and $\mathcal{G}$ such that, for every $d \geq 1$, every $G = (V, E) \in \mathcal{G}^\Delta$, and every $r \in V$,

$$|\{v \in V \mid \mathsf{dist}(v, r) \leq d\}| \leq C_\Delta \cdot d^g.$$

In other words, $\mathcal{G}$ is $g$-growth-bounded if the $d$-radius neighborhood of any node in any bounded-degree graph in $\mathcal{G}$ has size $O(d^g)$. Recall that $\mathcal{G}^\Delta$ is the set of graphs $G \in \mathcal{G}$ with maximum degree at most $\Delta$.

The goal of this section is to prove the following result, which gives an alternative characterization of the set $\mathcal{A}_k$: It is the set of all minor-closed graph classes $\mathcal{G}$ such that $k$ is the smallest number such that $\mathcal{G}$ is $k$-growth-bounded.

▶ **Proposition 18.** *Let $\mathcal{G}$ be any minor-closed graph class. If $\mathcal{G} \notin \bigcup_{0 \leq i < \infty} \mathcal{A}_i$, then $\mathcal{G}$ is not $k$-growth-bounded for any $k < \infty$. If $\mathcal{G} \in \mathcal{A}_k$, then $k$ is the smallest number such that $\mathcal{G}$ is $k$-growth-bounded.*

We first prove an easy part of Proposition 18.

▶ **Lemma 19.** *Let $\mathcal{G}$ be any minor-closed graph class such that $\mathcal{G} \notin \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$. For any $s < k+1$, $\mathcal{G}$ is not $s$-growth-bounded.*

**Proof.** Consider the set of graphs $\mathcal{S} = \{T_{k+1,x} \mid x \geq 3\}$. By Definition 2, as $\mathcal{G} \notin \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$, $\mathcal{S}$ is a subset of $\mathcal{G}^\Delta$ for each $\Delta \geq 3$. From the graph structure of $T_{k+1,x}$, the size of the $d$-radius neighborhood of a node in a graph in $\mathcal{S}$ cannot be bounded by any function $O(d^s)$ with $s < k+1$. ◀

**Layers of nodes.** Consider the following terminology. Given a positive integer $C$ and a rooted tree $T = (V, E)$, we define a sequence $L_0^*, L_1, L_1^*, L_2, L_2^*, \ldots$ of node subsets of $T$ as follows.

- $L_0^* = V$ is the set of all nodes in $T$.
- For each $i \geq 1$, $L_i$ is the set of nodes $v$ such that the subtree rooted at $v$ contains at least $C$ nodes in $L_{i-1}^*$.
- For each $i \geq 1$, $L_i^*$ is the set of nodes $v$ having at least two children in $L_i$.

For each non-negative integer $k$, we say that a rooted tree $T$ is *$(k, C)$-limited* if $L_{k+1} = \emptyset$.

We write $N^d(r)$ to denote the set of nodes $v \in V$ with $\mathsf{dist}(v, r) \leq d$. We have the following auxiliary lemma.

▶ **Lemma 20.** *If $T$ is a $(k, C)$-limited rooted tree whose maximum degree is at most $\Delta$, then $|L_k \cap N^d(r)| \leq \Delta C d$, where $r$ is the root.*

**Proof.** Since $T$ is $(k, C)$-limited, we have $L_{k+1} = \emptyset$, so $|L_k^*| < C$. If we remove the edges between each $v \in L_k^*$ and its children in the rooted tree $T$, then the set of nodes in $L_k$ are partitioned into paths $P_1, P_2, \ldots, P_x$, as $L_k^*$ is the set of nodes in $L_k$ with at least two children in $L_k$. Here $x$ is the number of paths in the partition. Orienting each edge toward the parent, we write each path $P_i$ as $P_i = u_{i,1} \leftarrow u_{i,2} \leftarrow \cdots$.

Observe that $u_{i,1}$ is either the root $r$ of $T$ or a child of a node in $L_k$. Since the paths $P_1, P_2, \ldots, P_x$ are disjoint, at most one path $P_i$ has $u_{i,1} = r$. The number of nodes whose parent is in $L_k^*$ is at most $|L_k^*|(\Delta - 1) + 1$. Hence the total number $x$ of paths is at most $x = |L_k^*|(\Delta - 1) + 2 \leq (C-1)(\Delta - 1) + 2 \leq C\Delta$.

A necessary condition for a node $v \in L_k$ to be in $N^d(r)$ is that it is within the first $d$ nodes of some path $P_i$, we have $|L_k \cap N^d(r)| \leq xd \leq \Delta C d$. ◀

Using Lemma 20, we show that the size of the $d$-radius neighborhood of the root $r$ of any $(k, C)$-limited rooted tree $T$ is small.

▶ **Lemma 21.** *For any three integers $C \geq 1$, $k \geq 0$, and $\Delta \geq 1$, there exists a constant $K_{C,k,\Delta}$ only depending on $C$, $k$, and $\Delta$ such that the number of nodes within distance $d$ to the root in any $(k, C)$-limited rooted tree $T$ with degree at most $\Delta$ is upper bounded by $K_{C,k,\Delta} \cdot d^k$.*

**Proof.** We set $K_{C,k,\Delta}$ as follows.

$$K_{C,k,\Delta} = \begin{cases} C - 1, & \text{if } k = 0, \\ \Delta C \cdot (\Delta K_{C,k-1,\Delta} + 1) & \text{if } k \geq 1. \end{cases}$$

Let $T = (V, E)$ be any $(k, C)$-limited rooted tree, so $L_{k+1} = \emptyset$. Let $r$ be the root of $T$. We verify that the number of nodes within distance $d$ to the root in $T$ is at most $|N^d(r)| \leq K_{C,k,\Delta} \cdot d^k$.

**Base case.** For the case of $k = 0$, if $L_1 = \emptyset$, then $T$ contains at most $C - 1$ nodes, since otherwise the root of $T$ belongs to $L_1$. Therefore, $|N^d(r)| \leq |V| \leq C - 1 = K_{C,0,\Delta}$.

**Inductive step.** Now, suppose $k \geq 1$. If $L_k = \emptyset$, then $T$ is $(C, k-1)$-limited, so the number of nodes within distance $d$ to the root is at most $K_{C,k-1,\Delta} d^{k-1} \leq K_{C,k,\Delta} d^k$ by inductive hypothesis.

Suppose $L_k \neq \emptyset$, then the root $r$ belongs to $L_k$, and the number of nodes in $L_k$ within distance $d$ to the root is at most $\Delta C d$ by Lemma 20. Let $S = N^d(r) \cap L_k$ denote these nodes.

Each node in $N^d(r) \setminus S$ belongs to the tree rooted at a child $u$ of some $v \in S$. Let $S'$ be the set of nodes in $V \setminus L_k$ that are children of nodes in $S$. Then $|S'| \leq \Delta |S|$.

Since $u \notin L_k$ for each $u \in S'$, the subtree of $T$ rooted at $u$ is $(C, k-1)$-limited, so it contains at most $K_{C,k-1,\Delta} d^{k-1}$ nodes that are within distance $d$ to $u$ by inductive hypothesis. Therefore,

$$\begin{aligned} |N^d(r)| &= |N^d(r) \cap L_k| + |N^d(r) \setminus L_k| \\ &\leq |S| + |S'| \cdot K_{C,k-1,\Delta} d^{k-1} \\ &\leq \Delta C d + \Delta^2 C d \cdot K_{C,k-1,\Delta} d^{k-1} \\ &\leq \Delta C \cdot (1 + \Delta K_{C,k-1,\Delta}) d^k \\ &= K_{C,k,\Delta} d^k, \end{aligned}$$

as desired. ◀

**Rooted minors.** A *rooted* graph is a graph $G$ with a distinguished root node $r \in V$. A rooted graph can also be treated as a graph without a root. We say that $H$ is a *rooted minor* of $G$ if there exist a partition of $V(G)$ into $k = |V(H)|$ disjoint connected clusters $\mathcal{C} = \{V_1, V_2, \ldots, V_k\}$ and a bijection between $\mathcal{C} = \{V_1, V_2, \ldots, V_k\}$ and $V(H)$ satisfying the following two conditions:

- For each edge $e$ in $H$, the two clusters in $\mathcal{C}$ corresponding to the two endpoints of $e$ are adjacent in $G$.
- The cluster in $\mathcal{C}$ corresponding to the root of $H$ contains the root of $G$.

Observe that in Definition 1, $T_{i,s}$ is a rooted minor of $T_{i',s'}$ whenever $i \leq i'$ and $s \leq s'$.

We show that a rooted minor isomorphic to $T_{i+1,s}$ exists in any rooted tree $T$ that is not $(i, C)$-limited, for some sufficiently large $C$.

▶ **Lemma 22.** *For any positive integers $x$ and $\Delta$, there is a number $C_{x,\Delta} > 0$ only depending on $x$ and $\Delta$ such that for each positive integer $i$, any rooted tree $T$ with degree at most $\Delta$ that is not $(i, C_{x,\Delta})$-limited contains $T_{i+1,x}$ as a rooted minor.*

**Proof.** We define $B_1 = 1$ and $B_j = 1 + \Delta \cdot B_{j-1}$ for $j > 1$. Remember that if $T$ is not $(i, C)$-limited, then $L_{i+1} \neq \emptyset$, so $|L_i^*| \geq C$. We claim that for $C = B_x$, in any rooted tree $T$ with $|L_i^*| \geq C$, there is a path $P = (u_1, u_2, \ldots, u_k)$ meeting the following conditions.

- The first node $v_1$ of the path is the root $r$ of $T$.
- The path contains $x$ nodes in $L_i^*$. Denote these nodes as $v_1, v_2, \ldots, v_x$.
- All nodes in the path belong to $L_{i-1}$.

Such a path $P$ can be constructed as follows. We start with $u_1 = r$. Once $(u_1, u_2, \ldots, u_j)$ has been constructed, if the current path still does not contain $x$ nodes in $L_i^*$, then we extend the path by picking $u_{j+1}$ as a child of $u_j$ that maximizes the number of nodes in $L_i^*$ in the subtree rooted at $u_{j+1}$.

To analyze this construction, let $\Phi_j > 0$ be the number of nodes in $L_i^*$ in the subtree rooted at $u_j$. There are two cases.

- If $u_j \in L_i \setminus L_i^*$, then $u_j$ has exactly one child $u_{j+1}$ in $L_i$ with $\Phi_{j+1} = \Phi_j$.
- Otherwise, $u_j \in L_i^*$. Then $u_j$ has at most $\Delta$ children in $L_i$, so $\Phi_{j+1} \geq (\Phi_j - 1)/\Delta$.

Our choice of $C = B_x$ ensures that this process leads to a path containing $x$ nodes in $L_i^*$.

Now, consider such a path $P$. If $i = 0$, then $P$ already contains $T_{1,x}$ as a rooted minor, as $P$ contains at least $x$ nodes and $T_{1,x}$ is an $x$-node path.

From now on, we assume $i > 1$. Since $v_j \in L_i^*$, it has a child $w_j \in L_i$ outside of $P$. By inductive hypothesis, the subtree $T_j$ rooted at $w_j$ has a rooted minor isomorphic to $T_{i,C}$. For each $1 \leq j \leq x$, we consider the partition of the nodes in $T_j$ witnessing the fact that it has $T_{i,C}$ as a rooted minor. The rest of the nodes in $T$ can be partitioned into $x$ connected clusters $S_1, S_2, \ldots, S_x$ so that $v_j \in S_j$. We must have $r \in S_1$, and the overall partition of the nodes of $T$ shows that $T_{i+1,C}$ is a rooted minor of $T$, so we may set $C_{x,\Delta} = C = B_x$. ◀

For a graph class $\mathcal{G}$, we say that $\mathcal{G}$ is *$(k, C)$-limited* if all trees $T = (V, E)$ in $\mathcal{G}$ are $(k, C)$-limited for all choices of the root $r \in V$. It is still allowed that $\mathcal{G}$ contains graphs that are not trees.

▶ **Lemma 23.** *Let $\mathcal{G}$ be any minor-closed graph class such that for each positive integer $\Delta$ there is a constant $C > 0$ such that $\mathcal{G}^\Delta$ is $(k, C)$-limited. Then $\mathcal{G}$ is $k$-growth-bounded.*

**Proof.** Suppose $\mathcal{G}^\Delta$ is $(k, C)$-limited. For each $G = (V, E) \in \mathcal{G}$ with degree at most $\Delta$, we pick any node $r \in V$, and let $T$ be the tree corresponding to any BFS starting from $r$. Because $\mathcal{G}$ is minor-closed, we have $T \in \mathcal{G}$, so $T$ rooted at $r$ is $(k, C)$-limited. Applying Lemma 21 to the rooted tree $T$ with the parameters $C$,$k$, and $\Delta$, we conclude that the size of the $d$-radius neighborhood of $r$ in $G$ is at most $K_{C,k,\Delta} \cdot d^k$, so $\mathcal{G}$ is $k$-growth-bounded with $C_\Delta = K_{C,k,\Delta}$. ◀

▶ **Lemma 24.** *If $\mathcal{G} \in \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$, then for each integer $\Delta \geq 3$, there exists a constant $C > 0$ such that $\mathcal{G}^\Delta$ is $(k, C)$-limited.*

**Proof.** Suppose there exists $\Delta \geq 3$ such that $\mathcal{G}^\Delta$ is not $(k, C)$-limited for all $C < \infty$. We apply Lemma 22 to any tree $T \in \mathcal{G}^\Delta$ that is not $(k, C)$-limited for $C = C_{x,\Delta}$ for some choice of the root $r$. Then we obtain that $T$ contains $T_{k+1,x}$ as a minor, so $T_{k+1,x} \in \mathcal{G}$. Since this holds for all $x$, $\mathcal{G}$ contains $T_{k+1,x}$ for all $x$, implying that $\mathcal{G} \notin \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$, which is a contradiction to the assumption that $\mathcal{G} \in \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$. Hence for each integer $\Delta \geq 3$, there exists a constant $C > 0$ such that $\mathcal{G}^\Delta$ is $(k, C)$-limited. ◀

Now we are ready to prove Proposition 18.

**Proof of Proposition 18.** If $\mathcal{G} \notin \bigcup_{0 \le i < \infty} \mathcal{A}_i$, then Proposition 3 and Theorem 15 imply that $\mathcal{G}$ contains all trees. By considering complete trees, we infer that $\mathcal{G}$ is not $k$-growth-bounded for any $k < \infty$.

Now suppose $\mathcal{G} \in \mathcal{A}_k$. Since $\mathcal{G} \in \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$, Lemmas 23 and 24 implies that $\mathcal{G}$ is $k$-growth-bounded. Since $\mathcal{G} \notin \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{k-1}$, Lemma 19 implies that $\mathcal{G}$ is not $s$-growth-bounded for any $s < k$. Hence $k$ is the smallest number such that $\mathcal{G}$ is $k$-growth-bounded. ◀

## 6 The Complexity Landscape

In this section, we consider the complexity of LCL problems for the graph class $\mathcal{G}^\Delta$ with $\mathcal{G} \in \mathcal{A}_k$, for some $k \ge 0$. We first consider the case of $\mathcal{G} \in \mathcal{A}_0$.

▶ **Theorem 6.** *Let $\Delta \ge 3$ be an integer, and let $\mathcal{G} \in \mathcal{A}_0$. All LCL problems that are solvable in $\mathcal{G}^\Delta$ can be solved in $O(1)$ rounds in $\mathcal{G}^\Delta$.*

**Proof.** By Proposition 18, there is a number $C_\Delta$ such that the size of $d$-radius neighborhood of any $r \in V$ in any $G = (V, E) \in \mathcal{G}^\Delta$ is at most $C_\Delta$. As the statement holds for all $d$, we infer that each graph in $\mathcal{G}^\Delta$ has at most $C_\Delta$ nodes. Therefore, if an LCL problem is solvable in $G \in \mathcal{G}^\Delta$, then it can be solved in $C_\Delta = O(1)$ rounds. ◀

For the rest of the section, we consider $k > 0$.

▶ **Theorem 7.** *Let $k \ge 1$, $\Delta \ge 3$, and $s \in \{1, 2, \ldots, k\}$ be integers, and let $\mathcal{G} \in \mathcal{A}_k$. There is an LCL problem whose complexity in $\mathcal{G}^\Delta$ is $\Theta(n^{1/s})$.*

**Proof.** The LCL problem $\mathcal{P}_s$ considered by Chang and Pettie [24] can be solved in $O(n^{1/s})$ rounds on *all graphs*, including the ones in $\mathcal{G}$. It was shown in [24] that $\mathcal{P}_s$ requires $\Omega(n^{1/s})$ rounds to solve in the graphs $T_{s,x}$, for all $x$. Since any $\mathcal{G} \in \mathcal{A}_k$ contains $T_{s,x}$ for all $x$ by the definition of $\mathcal{A}_k$, the complexity of $\mathcal{P}_s$ in any $\mathcal{G} \in \mathcal{A}_k$ is $\Omega(n^{1/s})$. Combining the upper and lower bounds, we conclude that the complexity of $\mathcal{P}_s$ is $\Theta(n^{1/s})$ in $\mathcal{G}$. ◀

▶ **Theorem 8.** *Let $k \ge 1$ and $\Delta \ge 3$ be integers, and let $\mathcal{G} \in \mathcal{A}_k$. There is no LCL problem whose deterministic complexity in $\mathcal{G}^\Delta$ is between $\omega(\log^* n)$ and $o(n^{1/k})$.*

**Proof.** This lemma follows from a modification of the proof of the existence of the deterministic $\omega(\log^* n) - o(\log n)$ complexity gap on general graphs by Chang, Kopelowitz, and Pettie [23].

We briefly review their proof and describe the needed modification. Fix any $\mathcal{G} \in \mathcal{A}_k$. Consider any deterministic algorithm $\mathcal{A}$ solving an LCL problem $\mathcal{P}$ in $T(n) = o(n^{1/k})$ rounds for all graphs in $\mathcal{G}^\Delta$. The goal is to design an algorithm $\mathcal{A}'$ that also solves $\mathcal{P}$ for all graphs in $\mathcal{G}^\Delta$ and costs only $O(\log^* n)$ rounds.

Suppose the locality radius of $\mathcal{P}$ is $r$. If we can assign distinct $O(\log \tilde{n})$-bit identifiers to each node such that the $(T(\tilde{n}) + r)$-radius neighborhood of each node has size at most $\tilde{n}$ and does not contain repeated identifiers, then $\mathcal{P}$ can be solved in $T(\tilde{n})$ rounds by running $\mathcal{A}$ with these $O(\log \tilde{n})$-bit identifiers and pretending that the underlying network has $\tilde{n}$ nodes. To see that this strategy works, we use the property that $\mathcal{G}$ is minor-closed. We show that the output label of each node $v$ resulting from the above approach is correct. Given any node $v$ in $G$, consider a subgraph $\tilde{G}$ of $G$ that contains exactly $\tilde{n}$ nodes and contains the entire $(T(\tilde{n}) + r)$-radius neighborhood of $v$. We assign distinct $O(\log \tilde{n})$-bit identifiers to the nodes in $\tilde{G}$ in such a way that the $O(\log \tilde{n})$-bit identifiers in the $(T(\tilde{n})+r)$-radius neighborhood of $v$ are chosen to be the same in $G$ and $\tilde{G}$. Because $\mathcal{G}$ is minor-closed, we have $\tilde{G} \in \mathcal{G}$. The output labels resulting from running $\mathcal{A}$ in $G$ and $\tilde{G}$ must be the same in the $r$-radius neighborhood of $v$, so the correctness of $\mathcal{A}$ in $\tilde{G}$ implies that the output label of $v$ must be locally correct in $G$.

We show that we can choose $\tilde{n} = O(1)$ to satisfy the property that the $(T(\tilde{n})+r)$-radius neighborhood of each node has size at most $\tilde{n}$. Let $d = T(\tilde{n}) + r$. Then $d = o(\tilde{n}^{1/k})$ because $r = O(1)$ and $T(\tilde{n}) = o(\tilde{n}^{1/k})$. By Proposition 18, the size of the $d$-radius neighborhood of any node in any graph $G \in \mathcal{G}^{\Delta}$ is $O(d^k)$. Thus, the size of the $(T(\tilde{n})+r)$-radius neighborhood of each node is upper bounded by $o(\tilde{n})$, meaning that by choosing $\tilde{n}$ to be a sufficiently large constant, the size of the $(T(\tilde{n})+r)$-radius neighborhood of each node is at most $\tilde{n}$.

As $\tilde{n} = O(1)$ is a constant, the needed $O(\log \tilde{n})$-bit identifiers can be computed in $O(\log^* n)$ rounds deterministically using the coloring algorithm of [44]. Since $T(\tilde{n}) = O(1)$ is also a constant, this approach yields a new algorithm $\mathcal{A}'$ that solves $\mathcal{P}$ in just $O(\log^* n)$ rounds. ◀

▶ **Theorem 9.** *Let $k \geq 1$ and $\Delta \geq 3$ be integers, and let $\mathcal{G} \in \mathcal{A}_k$. There is no* LCL *problem whose complexity in $\mathcal{G}^{\Delta}$ is between $\omega(1)$ and $o\left((\log^* n)^{1/k}\right)$.*

**Proof.** This lemma is proved by the Ramsey-theoretic technique of Naor and Stockmeyer [66], as discussed in the paper of Chang and Pettie [24]. Although this proof considers only deterministic algorithms, the result applies to randomized algorithms as well, because it is known that randomness does not help for algorithms with round complexity $2^{O(\log^* n)}$, as shown by Chang, Kopelowitz, and Pettie [23].

Fix a graph class $\mathcal{G}^{\Delta}$ with $\mathcal{G} \in \mathcal{A}_k$. Consider a $T(n)$-round deterministic algorithm $\mathcal{A}$ solving an LCL problem $\mathcal{P}$ for a graph class $\mathcal{G}^{\Delta}$. Fix a network size $n$. Consider the following parameters.

- $\tau = T(n)$ is the round complexity of $\mathcal{A}$ on $n$-node graphs in $\mathcal{G}^{\Delta}$.
- $p$ is the maximum number of nodes in a $\tau$-radius neighborhood of a node in a graph in $\mathcal{G}^{\Delta}$. According to Proposition 18, $p = O(\tau^k)$.
- $m$ is the maximum number of nodes in a $(\tau + r)$-radius neighborhood of a node in a graph in $\mathcal{G}^{\Delta}$.
- $c$ is the number of distinct functions mapping each possible $\tau$-radius neighborhood of a graph in $\mathcal{G}^{\Delta}$, whose nodes are equipped with distinct labels drawn from some set $S$ with size $p$, to an output label of $\mathcal{P}$.

Let $R(p, m, c)$ be the minimum number of nodes guaranteeing that any edge coloring of a complete $p$-uniform hypergraph with $c$ colors contains a monochromatic clique of size $m$. It is known that $\log^* R(p, m, c) \leq p + \log^* m + \log^* c + O(1)$. Suppose the space of unique identifiers has size $n^C$. As long as $n^C \geq R(p, m, c)$, it is possible to transform $\mathcal{A}$ into an $O(1)$-round deterministic algorithm solving $\mathcal{A}$. See [24, 66] for details.

Since the part $\log^* m + \log^* c + O(1)$ is negligible comparing with $p$, if $p \ll \log^* n$, then the above transformation is possible. Since $p = O(\tau^k)$, the inequality $p \ll \log^* n$ holds whenever the round complexity of $\mathcal{A}$ is $o\left((\log^* n)^{1/k}\right)$. Hence the complexity of $\mathcal{P}$ is not within $\omega(1)$ and $o\left((\log^* n)^{1/k}\right)$ on $\mathcal{G}^{\Delta}$. ◀

## 7 The Computational Complexity of the Classification

In this section, we prove Proposition 12 by showing a polynomial-time algorithm that decides whether $\mathcal{G} \in \mathcal{A}_i$ for any given minor-closed graph class $\mathcal{G}$, where $\mathcal{G}$ is represented by a finite list of excluded minors $H_1, H_2, \ldots, H_k$.

▶ **Lemma 25.** *Let $\mathcal{G}$ be the class of $\{H_1, H_2, \dots, H_k\}$-minor-free graphs. Then $\mathcal{G} \in \mathcal{A}_0 \cup \mathcal{A}_1 \cup \dots \cup \mathcal{A}_i$ if and only if there exist an integer $s \geq 3$ and an index $j \in [k]$ such that $H_j$ is a minor of $T_{i+1,s}$.*

**Proof.** Suppose $H_j$ is a minor of $T_{i+1,s}$ for some $s \geq 3$. Then $T_{i+1,s} \notin \mathcal{G}$. For each $x \geq i+1$, $T_{i+1,s}$ is a minor of $T_{x,s}$, so $T_{x,s} \notin \mathcal{G}$, as $\mathcal{G}$ is closed under minors. Since a necessary condition for $\mathcal{G} \in \mathcal{A}_x$ is $T_{x,s} \in \mathcal{G}$, we must have $\mathcal{G} \notin \mathcal{A}_x$. Since $H_j$ is a tree, $\mathcal{G}$ has bounded pathwidth by Theorem 15. Therefore, $\mathcal{G} \in \mathcal{A} \setminus (\mathcal{A}_{i+1} \cup \mathcal{A}_{i+2} \cup \cdots) = \mathcal{A}_0 \cup \mathcal{A}_1 \cup \dots \cup \mathcal{A}_i$.

For the other direction, suppose $\mathcal{G} \in \mathcal{A}_0 \cup \mathcal{A}_1 \cup \dots \cup \mathcal{A}_i$. Then $\mathcal{G} \in \mathcal{A}_x$ for some $x \in [0, i]$. By the definition of $\mathcal{A}_x$, there is an index $s \geq 3$ such that $\mathcal{G}$ does not contain $T_{x+1,s}$. Since $T_{x+1,s}$ is a minor of $T_{i+1,s}$, $\mathcal{G}$ also does not contain $T_{i+1,s}$, so there is some graph $H_j$ in the list of excluded minors $\{H_1, H_2, \dots, H_k\}$ such that $H_j$ is a minor of $T_{i+1,s}$. ◀

▶ **Lemma 26.** *For any rooted tree $H = (V, E)$, the following two statements are equivalent.*

- *There exists a positive integer $s$ such that $H$ is a rooted minor of $T_{j,s}$.*
- *There is a choice of a node $u$ in $H$ satisfying the following requirement. Let $P$ be the unique path connecting $u$ and the root $r$ of $H$. Let $S$ be the set of nodes in $V \setminus P$ whose parent is in $P$. For each $v \in S$, there exists a positive integer $s'$ such that the subtree rooted at $v$ is a rooted minor of $T_{j-1,s'}$.*

**Proof.** Suppose $H = (V, E)$ is a rooted minor of $T_{j,s}$. Consider any clustering $\mathcal{C}$ of the node set of $T_{j,s}$ witnessing the fact that $T_{j,s}$ contains $H$ as a rooted minor. Consider the path $(v_1, v_2, \dots, v_s)$ and the $s$ rooted trees $T_1, T_2, \dots, T_s$ in the definition of $T_{j,s}$. Let $\mathcal{C}^*$ be the set of clusters in $\mathcal{C}$ containing a node in $\{v_1, v_2, \dots, v_s\}$. Then the set of nodes in $H$ corresponding to the clusters in $\mathcal{C}^*$ form a path $P$ connecting the root $r$ of $H$ to some other node $u$.

Let $S$ be the set of nodes in $V \setminus P$ whose parent is in $P$. We show that the subtree rooted at each $v \in S$ is a rooted minor of $T_{j-1,s}$. For each node $v \in S$, let $\mathcal{C}_v$ be the set of clusters in $\mathcal{C} \setminus \mathcal{C}^*$ corresponding to nodes in the subtree of $H$ rooted at $v$. The union of the clusters in $\mathcal{C}_v$ must be a connected set $U$ of nodes that are confined to one $T_i$ of the rooted trees $T_1, T_2, \dots, T_s$ in the definition of $T_{j,s}$. Let $T'$ be the rooted subtree of $T_i = T_{j-1,s}$ induced by $U$. Observe that the cluster in $\mathcal{C}_v$ corresponding to $v$ contains the root of $T'$. We extend the clustering $\mathcal{C}_v$ to cover all nodes in $T_i = T_{j-1,s}$ as follows. For each node $w$ in $T_i$ that is not covered by the clustering $\mathcal{C}_v$, let $w$ join the cluster in $\mathcal{C}_v$ that contains the unique node $w'$ in $T'$ minimizing $\mathsf{dist}(w, w')$ in $T_i$. The resulting clustering shows that the subtree rooted at $v$ is a rooted minor of $T_i = T_{j-1,s}$.

For the other direction, suppose there is a choice of a node $u$ in $H$ satisfying the following condition. Let $P = (r = u_1, u_2, \dots, u_x = u)$ be the unique path connecting the root $r$ and $u$ in $H$, where $x$ is the number of nodes in $P$. Let $S$ be the set of nodes in $V \setminus P$ whose parent is in $P$. For each node $v \in S$, there exists a positive integer $s' = s_v$ such that the subtree rooted at $v$ is a rooted minor of $T_{j-1,s'}$.

We choose $s$ to be the maximum of $|S|$ and $s_v$ among all $v \in S$. We show that $H$ is a rooted minor of $T_{j,s}$ by demonstrating a clustering of the nodes of $T_{j,s}$ meeting all the needed requirements in the definition of rooted minor. Consider the path $(v_1, v_2, \dots, v_s)$ and the $s$ rooted trees $T_1, T_2, \dots, T_s$ in the definition of $T_{j,s}$. We partition the path $(v_1, v_2, \dots, v_s)$ into $x$ subpaths $(v_1, v_2, \dots, v_s) = P_1 \circ P_2 \circ \dots \circ P_x$ such that the number of nodes in $P_i$ is at least the number of children of $u_i$ in $S$. This partition exists due to our choice of $s$.

A clustering of the nodes of $T_{j,s}$ is constructed as follows. For each child $w$ of $u_i$ such that $w \in S$, we associate $w$ with a distinct index $l_w$ such that $v_{l_w} \in P_i$. Recall that the subtree rooted at $w \in S$ is a rooted minor of $T_{j-1,s_w}$. Since $s \geq s_w$, $T_{j-1,s_w}$ is a rooted minor of

$T_{j-1,s}$, so the subtree rooted at $w \in S$ is also a rooted minor of $T_{l_w} = T_{j-1,s}$. We find a clustering of the nodes in $T_{l_w}$ witnessing that the subtree rooted at $w$ is a rooted minor of $T_{l_w}$. For each $u_i \in P$, its corresponding cluster $C$ in $T_{j,s}$ is chosen as follows. Start with the subpath $C = P_i$. For each node $v_l \in P_i$ that is not associated with any node in $S$, we add all nodes in the subtree $T_l$ to $C$.

The clustering covers all nodes in $T_{j,s}$, the root $v_1$ of $T_{j,s}$ belongs to the cluster corresponding to the root $u_1 = r$ of $H$, and each edge in $H$ corresponds to a pair of adjacent clusters, so $H$ is a rooted minor of $T_{j,s}$. ◀

Now we are ready to prove Proposition 12.

▶ **Proposition 12.** *For any fixed index $i$, there is a polynomial-time algorithm that, given a list of graphs $H_1, H_2, \ldots, H_k$, decides whether the class of $\{H_1, H_2, \ldots, H_k\}$-minor-free graphs $\mathcal{G}$ is in $\mathcal{A}_i$.*

**Proof.** In view of Lemma 25, $\mathcal{G} \in \mathcal{A}_i$ if and only if the following two conditions are met.

- If $i > 0$, then for all integers $s \geq 1$, each graph $H_j \in \{H_1, H_2, \ldots, H_k\}$ is not a minor of $T_{i,s}$.
- There exist an integer $s \geq 1$ and a graph $H_j \in \{H_1, H_2, \ldots, H_k\}$ that is a minor of $T_{i+1,s}$.

Therefore, to decide whether $\mathcal{G} \in \mathcal{A}_i$ in polynomial time, it suffices to design a polynomial-time algorithm that accomplishes the following task for a fixed index $j$.

- Given a graph $H$, decide if there exists an integer $s \geq 1$ such that $H$ is a minor of $T_{j,s}$.

We may assume that $H$ is a tree, since otherwise we can immediately decide that $H$ is not a minor of $T_{j,s}$. Moreover, the above task can be reduced to following rooted version by trying each choice of the root node in $H$.

- Given a rooted tree $H$, decide if there exists an integer $s \geq 1$ such that $H$ is a rooted minor of $T_{j,s}$.

For the rest of the proof, we design a polynomial-time algorithm for this task by induction on $j$. For the base case of $j = 1$, as $T_{1,s}$ is simply an $s$-node path, the task of checking whether $H$ is a rooted minor of $T_{1,s}$ for some $s$ can be restated as follows. Given a rooted tree $H$, check whether it is a path where its root is an endpoint of the path, that is, $H$ is isomorphic to $T_{1,s}$ for some $s$. This task is doable in polynomial time. Assuming that we have a polynomial-time algorithm for $j - 1$, the characterization of Lemma 26 yields a polynomial-time algorithm for $j$ by checking all choices of the node $u$ in the characterization of Lemma 26. ◀

**Remark.** In the *minor containment* problem, we are given two graphs $H$ and $G$, and the goal is to decide whether $H$ is a minor of $G$. The problem is well-known to be polynomial-time solvable when $H$ has constant size [61, 70]. In the minor containment instances in the above proof, $H$ is one of the forbidden minors $H_1, H_2, \ldots, H_k$. As $\{H_1, H_2, \ldots, H_k\}$ is seen as the input to the problem considered in Proposition 12, these graphs cannot be treated as constant-size graphs, so the polynomial-time algorithms in [61, 70] cannot be adapted here.

If $H$ does not have constant size, then the problem is NP-complete, even when $G$ is a tree [54]. A polynomial-time algorithm for minor containment is known for the case where $G$ is a tree and $H$ has bounded degree [54]. This polynomial-time algorithm does not apply to our setting in the above proof, as the maximum degree of the graphs $H_1, H_2, \ldots, H_k$ can be arbitrarily large.

# 8 Conclusions

In this work, we proposed a conjecture that characterizes the complexity landscape of LCL problems for any minor-closed graph class, and we proved a part of the conjecture. We believe that our classification $\mathcal{A} = \bigcup_{0 \leq i < \infty} \mathcal{A}_i$ of bounded-pathwidth minor-closed graph classes offers the right measure of similarity between a minor-closed graph class and paths from the perspective of locality of distributed computing.

A significant effort is required to completely settle the conjecture. To extend the existing techniques to establish complexity gaps for LCL problems from paths or trees to bounded-pathwidth graphs or bounded-treewidth graphs, new local algorithms to decompose these graphs are needed. One major challenge here is that both path decomposition and tree decomposition are *global* in that computing them requires at least diameter rounds, so they are not directly applicable to the LOCAL model. New results along this direction not only have complexity-theoretic implications but it is likely going to yield new algorithmic tools in the LOCAL model.

On a broader note, we hope our work will inspire others to explore structural graph theory techniques in the study of the algorithms and complexities of local distributed graph problems and build bridges between distributed computing and other areas within algorithms and combinatorics.

## References


1 Amirreza Akbari, Navid Eslami, Henrik Lievonen, Darya Melnyk, Joona Särkijärvi, and Jukka Suomela. Locality in online, dynamic, sequential, and distributed graph algorithms. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 261 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.10.

2 Saeed Akhoondian Amiri, Stefan Schmid, and Sebastian Siebertz. Distributed dominating set approximations beyond planar graphs. *ACM Transactions on Algorithms (TALG)*, 15(3):1–18, 2019.

3 Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. The distributed complexity of locally checkable problems on paths is decidable. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 262–271, 2019.

4 Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Jan Studenỳ, and Jukka Suomela. Efficient classification of locally checkable problems in regular trees. In *Proceedings of the 36th International Symposium on Distributed Computing (DISC)*, pages 8:1–8:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

5 Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Jan Studenỳ, Jukka Suomela, and Aleksandr Tereshchenko. Locally checkable problems in rooted trees. *Distributed Computing*, 36:277–311, 2023.

6 Alkida Balliu, Sebastian Brandt, Yuval Efron, Juho Hirvonen, Yannic Maus, Dennis Olivetti, and Jukka Suomela. Classification of Distributed Binary Labeling Problems. In Hagit Attiya, editor, *Proceedings of the 34th International Symposium on Distributed Computing (DISC)*, volume 179 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DISC.2020.17.

7 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, Dennis Olivetti, and Gustav Schmid. On the Node-Averaged Complexity of Locally Checkable Problems on Trees. In Rotem Oshman, editor, *Proceedings of the 37th International Symposium on Distributed Computing (DISC)*, volume 281 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DISC.2023.7.

8 Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. Almost Global Problems in the LOCAL Model. In Ulrich Schmid and Josef Widder, editors, *Proceedings of the 32nd International Symposium on Distributed Computing (DISC)*, volume 121 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.DISC.2018.9`.

9 Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. How much does randomness help with locally checkable problems? In *Proceedings of the 39th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 299–308. ACM Press, 2020. `doi:10.1145/3382734.3405715`.

10 Alkida Balliu, Keren Censor-Hillel, Yannic Maus, Dennis Olivetti, and Jukka Suomela. Locally Checkable Labelings with Small Messages. In Seth Gilbert, editor, *Proceedings of the 35th International Symposium on Distributed Computing (DISC)*, volume 209 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.DISC.2021.8`.

11 Alkida Balliu, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Dennis Olivetti, and Jukka Suomela. New classes of distributed time complexity. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 1307–1318, New York, NY, USA, 2018. ACM. `doi:10.1145/3188745.3188860`.

12 Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. *J. ACM*, 63(3):20:1–20:45, 2016.

13 Anton Bernshteyn. Distributed algorithms, the Lovász Local Lemma, and descriptive combinatorics. *Inventiones mathematicae*, pages 1–48, 2023.

14 Marthe Bonamy, Linda Cook, Carla Groenland, and Alexandra Wesolek. A Tight Local Algorithm for the Minimum Dominating Set Problem in Outerplanar Graphs. In Seth Gilbert, editor, *Proceedings of the 35th International Symposium on Distributed Computing (DISC)*, volume 209 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.DISC.2021.13`.

15 Nicolas Bousquet, Laurent Feuilloley, and Théo Pierron. Local Certification of Graph Decompositions and Applications to Minor-Free Classes. In Quentin Bramas, Vincent Gramoli, and Alessia Milani, editors, *Proceedings of the 25th International Conference on Principles of Distributed Systems (OPODIS 2021)*, volume 217 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:17, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.OPODIS.2021.22`.

16 Sebastian Brandt, Yi-Jun Chang, Jan Grebík, Christoph Grunau, Václav Rozhoň, and Zoltán Vidnyánszky. Local Problems on Trees from the Perspectives of Distributed Algorithms, Finitary Factors, and Descriptive Combinatorics. In Mark Braverman, editor, *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:26, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2022.29`.

17 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 *Proceedings 48th ACM Symposium on the Theory of Computing (STOC)*, pages 479–488, 2016.

18 Sebastian Brandt, Juho Hirvonen, Janne H Korhonen, Tuomo Lempiäinen, Patric RJ Östergård, Christopher Purcell, Joel Rybicki, Jukka Suomela, and Przemysław Uznański. LCL problems on grids. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 101–110, 2017.

19 Yi-Jun Chang. The complexity landscape of distributed locally checkable problems on trees. In *Proceedings of the 34th International Symposium on Distributed Computing (DISC)*, pages 18:1–18:17, 2020.

20 Yi-Jun Chang. The distributed complexity of locally checkable labeling problems beyond paths and trees. *arXiv preprint*, 2023. `arXiv:2311.06726`.

21 Yi-Jun Chang. Efficient distributed decomposition and routing algorithms in minor-free networks and their applications. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 55–66, 2023.

22 Yi-Jun Chang, Manuela Fischer, Mohsen Ghaffari, Jara Uitto, and Yufan Zheng. The complexity of $(\Delta+1)$ coloring in congested clique, massively parallel computation, and centralized local computation. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 471–480, 2019.

23 Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the local model. *SIAM Journal on Computing*, 48(1):122–143, 2019.

24 Yi-Jun Chang and Seth Pettie. A time hierarchy theorem for the local model. *SIAM Journal on Computing*, 48(1):33–69, 2019.

25 Yi-Jun Chang, Jan Studenỳ, and Jukka Suomela. Distributed graph problems through an automata-theoretic lens. *Theoretical Computer Science*, 951:113710, 2023.

26 Yi-Jun Chang and Hsin-Hao Su. Narrowing the local–congest gaps in sparse networks via expander decompositions. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 301–312, 2022.

27 Shiri Chechik and Doron Mukhtar. Optimal distributed coloring algorithms for planar graphs in the local model. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 787–804. SIAM, 2019.

28 Aleksander Bjørn Grodt Christiansen. The power of multi-step Vizing chains. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1013–1026, 2023.

29 K.-M. Chung, S. Pettie, and H.-H. Su. Distributed algorithms for the Lovász local lemma and graph coloring. *Distributed Computing*, 30:261–280, 2017.

30 Andrzej Czygrinow and Michał Hańćkowiak. Distributed algorithms for weighted problems in sparse graphs. *Journal of Discrete Algorithms*, 4(4):588–607, 2006. `doi:10.1016/j.jda.2005.07.006`.

31 Andrzej Czygrinow and Michał Hańćkowiak. Distributed almost exact approximations for minor-closed families. In Yossi Azar and Thomas Erlebach, editors, *Algorithms – ESA 2006*, pages 244–255, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

32 Andrzej Czygrinow and Michał Hańćkowiak. Distributed approximation algorithms for weighted problems in minor-closed families. In *Proceedings of the 13th Annual International Conference on Computing and Combinatorics (COCOON)*, pages 515–525, Berlin, Heidelberg, 2007. Springer-Verlag.

33 Andrzej Czygrinow, Michał Hańćkowiak, and Edyta Szymańska. Distributed approximation algorithms for planar graphs. In *Italian Conference on Algorithms and Complexity (CIAC)*, 2006.

34 Andrzej Czygrinow, Michal Hanćkowiak, Edyta Szymańska, Wojciech Wawrzyniak, and Marcin Witkowski. Distributed local approximation of the minimum $k$-tuple dominating set in planar graphs. In *International Conference on Principles of Distributed Systems (OPODIS)*, pages 49–59. Springer, 2014.

35 Andrzej Czygrinow, Michal Hańćkowiak, and Wojciech Wawrzyniak. Fast distributed approximations in planar graphs. In *International Symposium on Distributed Computing (DISC)*, pages 78–92. Springer, 2008.

36 Andrzej Czygrinow, Michał Hanćkowiak, Wojciech Wawrzyniak, and Marcin Witkowski. Distributed approximation algorithms for $k$-dominating set in graphs of bounded genus and linklessly embeddable graphs. *Theoretical Computer Science*, 809:327–338, 2020.

37 Peter Davies. Improved distributed algorithms for the lovász local lemma and edge coloring. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4273–4295. SIAM, 2023.

38 Erik D Demaine, Fedor V Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and $H$-minor-free graphs. *Journal of the ACM (JACM)*, 52(6):866–893, 2005.

39 Erik D Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 637–646. IEEE, 2005.

40 Laurent Feuilloley, Nicolas Bousquet, and Théo Pierron. What can be certified compactly? compact local certification of mso properties in tree-like graphs. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 131–140, 2022.

41 Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Éric Rémila, and Ioan Todinca. Compact distributed certification of planar graphs. *Algorithmica*, pages 1–30, 2021.

42 Laurent Feuilloley, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, Eric Rémila, and Ioan Todinca. Local certification of graphs with bounded genus. *Discrete Applied Mathematics*, 325:9–36, 2023.

43 M. Fischer and M. Ghaffari. Sublogarithmic distributed algorithms for Lovász local lemma with implications on complexity hierarchies. In *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, pages 18:1–18:16, 2017.

44 P. Fraigniaud, M. Heinrich, and A. Kosowski. Local conflict coloring. In *Proceedings 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 625–634, 2016.

45 Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. A meta-theorem for distributed certification. In *Proceedings of the International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 116–134. Springer, 2022.

46 Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks II: Low-congestion shortcuts, MST, and min-cut. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 202–219. SIAM, 2016.

47 Mohsen Ghaffari and Bernhard Haeupler. Low-congestion shortcuts for graphs excluding dense minors. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 213–221, 2021.

48 Mohsen Ghaffari, David G Harris, and Fabian Kuhn. On derandomizing local distributed algorithms. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 662–673. IEEE, 2018.

49 Mohsen Ghaffari, Juho Hirvonen, Fabian Kuhn, and Yannic Maus. Improved distributed delta-coloring. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 427–436, 2018.

50 Mohsen Ghaffari and Merav Parter. Near-optimal distributed DFS in planar graphs. In Andréa Richa, editor, *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, volume 91 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.DISC.2017.21`.

51 Jan Grebík and Václav Rozhoň. Local problems on grids from the perspective of distributed algorithms, finitary factors, and descriptive combinatorics. *Advances in Mathematics*, 431:109241, 2023.

52 Jan Grebík and Václav Rozhoň. Classification of local problems on paths from the perspective of descriptive combinatorics, 2021. `arXiv:2103.14112`.

53 Christoph Grunau, Václav Rozhoň, and Sebastian Brandt. The landscape of distributed complexities on trees and beyond. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 37–47, 2022.

54 Arvind Gupta and Naomi Nishimura. The parallel complexity of tree embedding problems. *Journal of Algorithms*, 18(1):176–200, 1995.

55 Bernhard Haeupler, D Ellis Hershkowitz, and David Wajc. Round-and message-optimal distributed graph algorithms. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 119–128, 2018.

56 Bernhard Haeupler, Taisuke Izumi, and Goran Zuzic. Low-congestion shortcuts without embedding. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 451–460, 2016.

57 Bernhard Haeupler, Taisuke Izumi, and Goran Zuzic. Near-optimal low-congestion shortcuts on bounded parameter graphs. In *Proceedings of the International Symposium on Distributed Computing (DISC)*, pages 158–172. Springer, 2016.

58 Bernhard Haeupler and Jason Li. Faster Distributed Shortest Path Approximations via Shortcuts. In Ulrich Schmid and Josef Widder, editors, *Proceedings of the 32nd International Symposium on Distributed Computing (DISC)*, volume 121 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.DISC.2018.33.

59 Bernhard Haeupler, Jason Li, and Goran Zuzic. Minor excluded network families admit fast distributed algorithms. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 465–474, 2018.

60 Taisuke Izumi, Naoki Kitamura, Takamasa Naruse, and Gregory Schwartzman. Fully polynomial-time distributed computation in low-treewidth graphs. In *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 11–22, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3490148.3538590.

61 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012.

62 Christoph Lenzen, Yvonne-Anne Pignolet, and Roger Wattenhofer. Distributed minimum dominating set approximations in restricted families of graphs. *Distributed computing*, 26(2): 119–137, 2013.

63 Reut Levi, Moti Medina, and Dana Ron. Property testing of planarity in the congest model. *Distributed Computing*, 34(1):15–32, 2021.

64 Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

65 Moni Naor, Merav Parter, and Eylon Yogev. The power of distributed verifiers in interactive proofs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1096–1115, 2020. doi:10.1137/1.9781611975994.67.

66 Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM J. Comput.*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

67 Seth Pettie and Hsin-Hao Su. Distributed algorithms for coloring triangle-free graphs. *Information and Computation*, 243:263–280, 2015.

68 Neil Robertson and Paul D Seymour. Graph minors. I. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

69 Neil Robertson and Paul D Seymour. Graph minors. V. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986. doi:10.1016/0095-8956(86)90030-4.

70 Neil Robertson and Paul D Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995.

71 Neil Robertson and Paul D Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. Special Issue Dedicated to Professor W.T. Tutte. doi:10.1016/j.jctb.2004.08.001.

72 Will Rosenbaum and Jukka Suomela. Seeing far vs. seeing wide: Volume complexity of local graph problems. In *Proceedings of the 39th Symposium on Principles of Distributed Computing (PODC)*, pages 89–98, 2020.

**73** Václav Rozhoň and Mohsen Ghaffari. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 350–363, 2020.

**74** Petra Scheffler. A linear algorithm for the pathwidth of trees. In *Topics in Combinatorics and Graph Theory: Essays in Honour of Gerhard Ringel*, pages 613–620, Heidelberg, 1990. Physica-Verlag HD. doi:10.1007/978-3-642-46908-4_70.

**75** Wojciech Wawrzyniak. A strengthened analysis of a local algorithm for the minimum dominating set problem in planar graphs. *Information Processing Letters*, 114(3):94–98, 2014.