Narrowing the LOCAL–CONGEST Gaps in Sparse Networks via Expander Decompositions

Yi-Jun Chang[*]
cyijun@nus.edu.sg
National University of Singapore
Singapore

Hsin-Hao Su[†]
hsinhao.su@bc.edu
Boston College
USA

ABSTRACT

Many combinatorial optimization problems, including maximum weighted matching and maximum independent set, can be approximated within $(1\pm\epsilon)$ factors in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds in the LOCAL model via network decompositions [Ghaffari, Kuhn, and Maus, STOC 2018]. These approaches, however, require sending messages of unlimited size, so they do not extend to the more realistic CONGEST model, which restricts the message size to be $O(\log n)$ bits. For example, despite the long line of research devoted to the distributed matching problem, it still remains a major open problem whether an $(1-\epsilon)$-approximate maximum weighted matching can be computed in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds in the CONGEST model.

In this paper, we develop a generic framework for obtaining $\mathrm{poly}(\log n, 1/\epsilon)$-round $(1\pm\epsilon)$-approximation algorithms for many combinatorial optimization problems, including maximum weighted matching, maximum independent set, and correlation clustering, in graphs excluding a fixed minor in the CONGEST model. This class of graphs covers many sparse network classes that have been studied in the literature, including planar graphs, bounded-genus graphs, and bounded-treewidth graphs.

Furthermore, we show that our framework can be applied to give an efficient distributed property testing algorithm for an arbitrary minor-closed graph property that is closed under taking disjoint union, significantly generalizing the previous distributed property testing algorithm for planarity in [Levi, Medina, and Ron, PODC 2018 & Distributed Computing 2021].

Our framework uses distributed expander decomposition algorithms [Chang and Saranurak, FOCS 2020] to decompose the graph into clusters of high conductance. We show that any graph excluding a fixed minor admits small edge separators. Using this result, we show the existence of a high-degree vertex in each cluster in an expander decomposition, which allows the entire graph topology of the cluster to be routed to a vertex. Similar to the use of network decompositions in the LOCAL model, the vertex will be able to perform any local computation on the subgraph induced by the cluster and broadcast the result over the cluster.

CCS CONCEPTS

• **Theory of computation → Distributed algorithms**.

KEYWORDS

distributed graph algorithms, expander decompositions, matching

ACM Reference Format:
Yi-Jun Chang and Hsin-Hao Su. 2022. Narrowing the LOCAL–CONGEST Gaps in Sparse Networks via Expander Decompositions. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC '22), July 25–29, 2022, Salerno, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3519270.3538423

1 INTRODUCTION

The LOCAL and CONGEST models are two prominent vertex-centric models for studying distributed graph algorithms. In these models, vertices host processors and operate in synchronized rounds. In each round, each vertex sends a message to each of its neighbors, receives messages from its neighbors, and performs local computations. The time complexity of an algorithm is defined to be the number of rounds used. The main difference between the two models is the restriction on the message size. In the LOCAL model, we allow messages of unlimited size to be sent across each link; while in the CONGEST model, an upper bound of $O(\log n)$ bits is imposed on the message size, where n is the number of nodes. Algorithms designed for the vertex-centric models can be optimized by Pregel-like systems [75] such as GraphX [56] and Gigraph [54] to process massive graph data, see [76] for a comprehensive survey. Since algorithms designed for the CONGEST model use smaller messages, it is likely they will be converted to more efficient processes than their counterparts in the LOCAL model.

Combinatorial optimization problems, such as matching and independent set, are central in the area of distributed graph algorithms. Many combinatorial optimization problems are known to be efficiently solvable in the LOCAL model. Ghaffari, Kuhn, and Maus [49] gave a general framework for approximating packing and covering integer linear programming problems within $(1\pm\epsilon)$ of the optimality in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds. The framework covers, for example, the maximum weighted matching problem and the maximum independent set problem. With the recent breakthrough of Rozhoň and Ghaffari [83] on deterministic network decompositions, their approach can even be implemented deterministically. The approach of [49], however, requires sending messages of unlimited size, so the complexities of many of these problems remain to be tackled in the CONGEST model. For example, in contrary to the LOCAL model, it is still unclear whether a $\mathrm{poly}(\log n, 1/\epsilon)$-round $(1-\epsilon)$-approximate algorithm for maximum weighted matching exists in the CONGEST model. Moreover, it is known that some

[*]Supported by NUS ODPRT Grant No. R-252-000-C04-133.
[†]Supported by NSF Grant No. CCF-2008422.

PODC '22, July 25–29, 2022, Salerno, Italy

ACM ISBN 978-1-4503-9262-4/22/07.
https://doi.org/10.1145/3519270.3538423

problems cannot be computed efficiently in the CONGEST model in general [8, 36]. For example, there is a constant $\epsilon > 0$ such that finding an $(1-\epsilon)$-approximate maximum independent set requires $\tilde{\Omega}(n^2)$ rounds.

Our Contribution. We develop a new tool set for solving combinatorial optimization problems in the CONGEST model on a wide range of sparse network classes that have been studied in the literature. Our framework applies to any graph classes that are minor closed, covering many natural graph classes such as planar graphs, bounded-genus graphs, and bounded-treewidth graphs.

Our approach is as follows. We use an expander decomposition to decompose the graph into components of high conductance. The existence of small edge separators guarantees the existence of a high-degree vertex in each component, which allows the entire graph topology of the component to be routed to a vertex. Similar to the use of the network decompositions in the LOCAL model, the vertex will then be able to solve the problem locally and broadcast the result over the component.

We show that our framework can be applied to give efficient algorithms to solve various combinatorial optimization problems, property testing problems, and graph decomposition problems in the CONGEST model, narrowing the gaps of these problems between the CONGEST model and the LOCAL model in H-minor-free networks.

Notation. Throughout this paper, $n = |V|$ denotes the number of the vertices and $\Delta = \max_{v \in V} \deg(v)$ denotes the maximum degree of the graph $G = (V, E)$ under consideration. We say that an algorithm succeeds with high probability (w.h.p.) if it succeeds with probability $1 - 1/\mathrm{poly}(n)$. We write $\tilde{O}(\cdot)$, $\tilde{\Omega}(\cdot)$, and $\tilde{\Theta}(\cdot)$ to compress a $\log^{\pm O(1)} n$ factor.

1.1 Our Results

Matching. A *matching* is a set of edges that do not share endpoints. Given a weighted graph $G = (V, E, w)$, the maximum weight matching (MWM) problem is to compute a matching M with the maximum weight, where the weight of M is defined as $\sum_{e \in M} w(e)$. Given an unweighted graph $G = (V, E)$, the maximum cardinality matching (MCM) problem is to compute a matching M such that $|M|$ is maximized. Clearly, the MCM problem is a special case of the MWM problem. For MWM, we assume that all the edge weights $w(e)$ are positive integers, and we write W to denote the maximum weight $\max_{e \in E} w(e)$.

In the CONGEST model, [10, 74] showed that a $(1-\epsilon)$ approximate MCM can be computed in rounds with exponential dependencies on $(1/\epsilon)$. Very recently, and independently from our work, [43] showed that a $(1-\epsilon)$-approximate MCM can be computed in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds. However, for the MWM problem in general graphs, currently the best approximation ratio one can get in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds is $(2/3-\epsilon)$ by the rounding approach of [2]. Using exponential in $(1/\epsilon)$ rounds, recently [41] showed that a $(1-\epsilon)$-approximate MWM can be computed in general graphs. Also in bipartite graphs, a $(1-\epsilon)$-approximate MWM is known to be obtainable in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds [42, 74].

On the other hand, in the LOCAL model, many fast $\mathrm{poly}(\log n, 1/\epsilon)$-round algorithms for computing a $(1-\epsilon)$-approximate MWM in general graphs have been developed [44, 48–50, 62, 79, 83] (see Table 1 in the full version [21] for a more detailed survey). Using our framework, we obtain the first $\mathrm{poly}(\log n, 1/\epsilon)$-round algorithms for computing $(1-\epsilon)$-approximate MWM in non-trivial graph classes outside bipartite graphs in the CONGEST model.

Theorem 1.1. *A $(1-\varepsilon)$-approximate maximum weighted matching of an H-minor-free network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability in the* CONGEST *model.*

Throughout the paper, although the hidden leading constant in the round complexity of our H-minor-free networks algorithms depend on H, we emphasize that the constants $O(1)$ in the exponents of the round complexity $\varepsilon^{-O(1)} \log^{O(1)} n$ are independent of H.

Maximum Independent Set. An independent set is a set of non-adjacent vertices. The maximum independent set (MaxIS) problem is to find an independent set whose cardinality is maximum over all possible independent sets. Note that a maximal independent set is a $(1/\Delta)$-approximation to the MaxIS problem. Therefore, in the CONGEST model, a $(1/\Delta)$-approximate solution can be computed in $\mathrm{MIS}(n, \Delta)$ time, where $\mathrm{MIS}(n, \Delta)$ is the number rounds needed to compute a maximal independent set in the CONGEST model. The weighted version of the problem was considered in [10], and they gave an algorithm that finds a $(1/\Delta)$-approximate weighted MaxIS in $O(\mathrm{MIS}(n, \Delta) \cdot \log W)$ rounds, where W is the maximum weight. Later, it was shown in [66] that a $((1-\epsilon)/\Delta)$-approximate weighted MaxIS can be computed in $\mathrm{poly}(\log \log n) \cdot O(1/\epsilon)$ rounds with high probability. Moreover, they also showed that a $((1-\epsilon)/8\alpha)$-approximate weighted MaxIS in graphs of arboricity α can be obtained in $\tilde{O}(\log n/\epsilon)$ rounds with high probability. For the unweighted version, [66] also showed that a $((1-\epsilon)/\Delta)$-approximate MaxIS can be computed in $O(1/\epsilon)$ rounds with high probability when $\Delta \le n/\log n$.

In the LOCAL model, Ghaffari, Kuhn, Maus [49] showed that an $(1-\epsilon)$-approximation to the MaxIS problem can be computed in $\mathrm{poly}(\log n, 1/\epsilon)$ rounds. No analogous $(1-\epsilon)$-approximation algorithms are known in the CONGEST model as there are lower bounds showing algorithms with constant approximation ratios require $n^{\Theta(1)}$ rounds [8, 36]. Using our framework, we show:

Theorem 1.2. *A $(1-\varepsilon)$-approximate maximum independent set of an H-minor-free network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log \log n})}$ rounds deterministically in the* CONGEST *model.*

Correlation Clustering. The correlation clustering problem introduced by Bansal, Blum, and Chawla [9] is known to have various applications in spam detection, gene clustering, chat disentanglement, and co-reference resolution [6, 13, 23, 37, 38]. In this problem, each edge is labeled with a positive label or a negative label that denotes whether the two endpoints of the edge are positively correlated or negatively correlated.

The goal is to partition the vertices V into clusters $V_1, V_2, \cdots, V_k$ such that they are as *consistent* with the labels as possible. Let E^+

denote the positively-labeled edges and E^- denote the negatively-labeled edges. There are two versions of the problem: In the agreement maximization version, the goal is to maximize $\sum_{i=1}^{k} |E^+ \cap (V_i \times V_i)| + \sum_{1 \le i < j \le k} |E^- \cap (V_i \times V_j)|$. In the disagreement minimization version, the goal is to minimize $\sum_{i=1}^{k} |E^- \cap (V_i \times V_i)| + \sum_{1 \le i < j \le k} |E^+ \cap (V_i \times V_j)|$. Note that two versions of the problem are equivalent if one is looking for the exact solution.

We focus on approximate solutions for the agreement maximization version of the problem. In the centralized setting, the problem is shown to be APX-Hard in general graphs [22, 39]. In particular, Charikar, Guruswami, and Wirth [22] showed that it is NP-hard to approximate the problem within a factor of $115/116+\epsilon$ for any $\epsilon > 0$. On the positive side, they gave a 0.7664-approximation algorithm for the problem. Later, Swamy [85] gave a 0.7666-approxmation algorithm for the problem. In the distributed setting, while there are $O(1)$-approximation parallel algorithms on complete graphs [14, 23, 80] for the disagreement minimization problem, to our knowledge, no efficient algorithms for the CONGEST model or the LOCAL model have been proposed outside of complete graphs for both versions of the problem.[1] Using our framework, we show:

Theorem 1.3. *A $(1-\varepsilon)$-approximate agreement maximization correlation clustering of an H-minor-free network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log \log n})}$ rounds deterministically in the* CONGEST *model.*

In addition to approximation algorithms for combinatorial optimization problems, we demonstrate applications of our framework to the realm of property testing and graph decompositions.

Property Testing. A graph property $\mathcal{P}$ is a set of graphs. We say that a graph G has property $\mathcal{P}$ if $G \in \mathcal{P}$. We say that an n-vertex graph $G = (V, E)$ is *ε-far* from having property $\mathcal{P}$ if removing and adding at most $\epsilon|E|$ edges cannot turn G into a graph in $\mathcal{P}$. The study of property testing in the distributed setting was initiated by Censor-Hillel, Fischer, Schwartzman, and Vasudev [17]. We say that a distributed property testing algorithm $\mathcal{A}$ for a property $\mathcal{P}$ with proximity parameter ε is correct if it satisfies the following.

- If G has property $\mathcal{P}$, then all vertices output Accept.
- If G is ε-far from having property $\mathcal{P}$, then at least one vertex outputs Reject.

Levi, Medina, and Ron [72] showed an $\varepsilon^{-O(1)} \cdot O(\log n)$-round randomized distributed algorithm for property testing of planarity in the CONGEST model with *one-sided error.* If G has property $\mathcal{P}$, then all vertices output Accept. If G is ε-far from having property $\mathcal{P}$, then at least one vertex outputs Reject with high probability. Their algorithm uses the distributed planarity testing algorithm of Ghaffari and Haeupler [45] as a subroutine.

Using our framework, we give a simple proof that distributed property testing of planarity can be solved in $\mathrm{poly}(1/\varepsilon, \log n)$ rounds in the randomized setting and in $n^{o(1)} \cdot \mathrm{poly}(1/\varepsilon)$ rounds in the deterministic setting. More generally, our algorithm can be generalized to testing an *arbitrary* minor-closed graph property that is closed under taking disjoint union.

Theorem 1.4. *Distributed property testing for any minor-closed graph property $\mathcal{P}$ that is closed under taking disjoint union can be solved in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log \log n})}$ rounds deterministically in the* CONGEST *model.*

Graph Decompositions. An (ε, D) *low-diameter decomposition* of a graph $G = (V, E)$ is a partition of the vertex set $V = V_1 \cup V_2 \cup \cdots \cup V_k$ such that the number of inter-cluster edges is at most $\varepsilon|E|$ and the diameter of the induced subgraph $G[V_i]$ is at most D for each $1 \le i \le k$.

It is well-known [1, 40, 68] that for any H-minor-free graph, a low-diameter decomposition with $D = O(\varepsilon^{-1})$ exists, where the hidden constant in $O(\cdot)$ depends only on H. It is straightforward to see that the inverse linear dependence $D = O(\varepsilon^{-1})$ on ε is the best possible by considering cycle graphs.

In the distributed setting, Czygrinow, Hańćkowiak, and Wawrzyniak [29] designed a distributed algorithm that computes a low-diameter decomposition with $D = \varepsilon^{-O(1)}$ in $\varepsilon^{-O(1)} \cdot O(\log^* n)$ rounds for planar networks. Their algorithm also applies to the edge-weighted setting where the guarantee of the algorithm is that the summation of the weights of inter-cluster edges is at most ε-fraction of the summation of the weights of all edges. Although they presented their algorithm in the LOCAL model, the algorithm also works in the CONGEST model. Levi, Medina, and Ron [72] also designed a distributed algorithm that computes a low-diameter decomposition with $D = \varepsilon^{-O(1)}$ in $\varepsilon^{-O(1)} \cdot O(\log n)$ rounds for H-minor-free networks, which is used in their distributed algorithm for property testing of planarity.

Using our framework, we improve the inverse polynomial dependence $D = \varepsilon^{-O(1)}$ on ε to the optimal $D = O(\varepsilon^{-1})$. We present a simple proof that a low-diameter decomposition with $D = O(\varepsilon^{-1})$ can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log \log n})}$ rounds deterministically in the CONGEST model.

Theorem 1.5. *Given an H-minor free network, a low-diameter decomposition with $D = O(\varepsilon^{-1})$ can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log \log n})}$ rounds deterministically in the* CONGEST *model.*

1.2 Our Framework

Our framework of algorithm design is based on the recently developed distributed constructions of expander decompositions in the CONGEST model [19, 20]. We say that a graph is an *ϕ-expander* if its conductance is at least ϕ. An (ε, ϕ) *expander decomposition* of a graph is a removal of at most ε fraction of the edges such that each remaining connected component is an ϕ-expander. Intuitively, the conductance of a graph measures how well-connected it is. In particular, any random walk converges quickly to its stationary distribution in a high-conductance graph. Expander decompositions have a wide range of applications in theoretical computer science, including linear system solvers [84], unique games [7, 81], minimum cut [67], property testing [55, 69], and dynamic algorithms [24, 78].

We say that H is a *minor* of G if H can be obtained from G by iteratively removing vertices and edges and contracting edges. We write $H \preceq G$ if H is a minor of G. We say that G is *H-minor-free* if

[1] It is, however, not hard to see that a $\mathrm{poly}(\log n, 1/\epsilon)$-round $(1-\epsilon)$-approximate algorithm for the agreement maximization problem in general graphs can be obtained via low diameter decompositions in the LOCAL model.

$H \not\leq G$. A class of graphs $\mathcal{G}$ is *minor-closed* if $G \in \mathcal{G}$ implies $H \in \mathcal{G}$ for any $H \leq G$. Many natural graph classes, such as planar graphs, bounded-genus graphs, and bounded-treewidth graphs, are minor-closed. The graph minor theorem of Robertson and Seymour [82] implies that for any minor-closed family of graphs $\mathcal{G}$, there exists a *finite* set of *forbidden minors* $\mathcal{H}$ such that $G \notin \mathcal{G}$ if and only if $H \leq G$ for some $H \in \mathcal{H}$. For example, if $\mathcal{G}$ is the set of all planar graphs, then $\mathcal{H} = \{K_5, K_{3,3}\}$. That is, G is planar if and only if G is $K_{3,3}$-minor-free and K_5-minor-free. The graph minor theorem also implies that a minor-closed family of graphs must be a subset of the family of H-minor free graphs for some fixed graph H.

In this paper, we focus on the class of H-minor-free networks for any fixed H. The idea of our framework is that we want to use expander decompositions in the CONGEST model in a way similar to the use of low-diameter decompositions [73, 83] in the LOCAL model. That is, for each low-diameter cluster V_i, we want to gather the graph topology $G[V_i]$ to a vertex $v_i^* \in V_i$ so that v_i^* can run any sequential algorithm on $G[V_i]$ locally and broadcast the result to all other vertices in V_i. This approach clearly requires sending messages of unlimited size in the general case.

An *edge separator* of a graph is a cut $\{S, V \setminus S\}$ such that $\min\{|S|, |V \setminus S|\} \geq |V|/3$. The *size* of an edge separator $\{S, V \setminus S\}$ is the number of edges crossing S and $V \setminus S$. If $G[V_i]$ is an ϕ-expander and admits a small edge separator, then there must exist a high-degree vertex $v_i^* \in V_i$, so the connectivity property of a ϕ-expander allows us to design an efficient routing algorithm to let v_i^* gather the entire graph topology of $G[V_i]$. These properties are shown in Section 2.

It is known [33, 77] that planar graphs admit an edge separator of size $O(\sqrt{\Delta|V|})$. More generally, any graph that can be embedded on a surface of genus g has an edge separator of size $O(\sqrt{g\Delta|V|})$ [86]. In this paper, we generalize these results to show that all H-minor-free graphs admit an edge separator of size $O(\sqrt{\Delta|V|})$, so the approach discussed above is applicable to all H-minor-free graphs.

Theorem 1.6. *For any H-minor-free graph $G = (V, E)$, there is a cut S such that $\min\{|S|, |V \setminus S|\} \geq n/3$ and $|\partial(S)| = O(\sqrt{\Delta n})$, where the hidden constant in $O(\cdot)$ depends only on H.*

1.3 Applying Our Framework

We show that by using our framework, many unweighted optimization problems can be approximated within $(1 \pm \epsilon)$ factors in a straightforward manner. Then, we use the MWM problem as an example to demonstrate that our framework can be applied to solve weighted problems as well.

Unweighted Problems. As a warm up, to illustrate how our framework can be used, we first describe how to use our framework to obtain simple poly$(1/\epsilon, \log n)$-round $(1 - \epsilon)$-approximate algorithms for MCM in planar graphs, as well as other unweighted problems such as the MaxIS problem and the correlation clustering problem in H-minor-free graphs.

The idea behind these algorithms is very simple. If the size of an optimal solution is linear in the number of vertices, then we can simply let each cluster of an expander decomposition to compute its local optimal solution by letting a high-degree vertex in the cluster learn the graph topology of the cluster. We just need to show that ignoring the $\epsilon|E|$ inter-cluster edges only worsens the quality solution by a factor of at most $(1-\varepsilon)$. It is conceivable that this holds for many unweighted problems. Indeed, this is true for the MaxIS problem in $O(1)$-arboricity graphs and the agreement maximization correlation clustering problem in general graphs, so our framework immediately gives efficient $(1 - \epsilon)$-approximate algorithms for these problems in H-minor-free graphs, see Sections 3.1 and 3.3 for details.

For the case of the MCM problem, the size of an optimal solution is not linear in the number of vertices in general, but it is possible to preprocess the graph so that the size of an optimal solution is linear in the number of the vertices by using the preprocessing procedure of [27] for planar graphs. By doing so, we obtain a $(1-\varepsilon)$-approximate algorithm for MCM on planar graphs. We describe such an approach in Section 3.2.

Weighted Matching. Extending the framework to weighted problems is significantly more challenging, because when applying the expander decomposition in Theorem 2.6 we do not have control over which edges we will remove. For unweighted problems, the $\epsilon|E|$ edges that we remove usually can only cause a small degrade on the optimal solution. However, in the weighted problem, the small fraction of edges could have very high weights. As a result, the optimal solution could become much worse after removing those edges.

To overcome this obstacle, instead of applying the decomposition only once in the beginning, we embed our method into Duan and Pettie's sequential scaling algorithm [34] for approximating MWM. Roughly speaking, their scaling algorithm is a primal-dual algorithm that consists of multiple iterations. It processes the subgraphs from the ones induced by higher weight edges to the ones induced by lower weight edges over the iterations. Each iteration consists of non-trivial steps such as the augmentation step as well as the blossom shrinking step that are not easily implementable in the CONGEST model, but implementable in linear time in the centralized setting. For example, the augmentation step involves finding a maximal set of augmenting paths in the working subgraph. Since the length of an augmenting path can as large as $\Theta(n)$, it would not be possible to find it in poly$(\log n, 1/\epsilon)$ rounds in the CONGEST model.

We apply our expander decomposition framework to the working subgraph before some of the non-trivial steps. Instead of physically removing the inter-component edges from the graph, we add or subtract a small weight to the edges so they are no longer the "tight" edges (i.e. the edges in the working subgraph) in the primal-dual algorithm. We show that adding or subtracting the small weights would only degrade the optimal solution slightly. Moreover, we show this allows us to process each component independently (e.g. the long augmenting paths mentioned in the previous paragraph would be broken). Each component can then route the topology to a vertex and let the vertex perform the non-trivial steps locally and broadcast the result back.

This summarizes the high-level idea. However, there are several technical challenges such as that the expander decomposition may cut through some intermediate structures (i.e. the active blossoms). In addition, similar to the aforementioned unweighted case, we also need to preprocess the working subgraph before running the expander decomposition to ensure the number of inter-component edges is small relative to the solution. The planar graph preprocessing procedure of [27] does not work for H-minor free graphs in general. We discuss how we resolve these issues in the full version [21] of the paper.

1.4 Related Work

Chang, Pettie, Saranurak, and Zhang [19] gave the first application of expander decompositions to the CONGEST model of distributed computing. They designed a distributed algorithm for constructing an expander decomposition and applied it to give a near-optimal distributed algorithm for the *triangle listing* problem, based on the following framework. First construct an (ε, ϕ) expander decomposition to partition the vertex set $V = V_1 \cup V_2 \cup \cdots \cup V_k$ into ϕ-expanders. Using existing routing algorithms [51, 52] for ϕ-expanders, existing distributed triangle listing algorithms that make use of non-local communication can be simulated in ϕ-expanders with small overhead. Based on this approach, all triangles containing at least one edge in $G[V_i]$ can be listed efficiently, for all high-conductance clusters $G[V_i]$ in parallel. Finally, the remaining ε-fraction of the inter-cluster edges are handled using recursive calls.

Subsequent to the work of [19], expander decomposition has been applied to numerous other problems in the CONGEST model via this framework of algorithm design [16, 18, 35, 63, 70]. So far, all applications of distributed expander decomposition have been confined to the distributed subgraph finding problems [15], except the work of Daga, Henzinger, Nanongkai, and Saranurak [32], where they designed a sublinear-round exact min-cut algorithm in the CONGEST model by incorporating distributed expander decomposition into the sequential min-cut algorithm of Kawarabayashi and Thorup [67].

Distributed Algorithms on Minor-closed Networks. Many real-world networks have sparse structures. Over the past few years, much of the research effort has been devoted to designing efficient distributed algorithms in LOCAL and CONGEST utilizing structural properties of sparse networks, and many natural graph classes studied in the literature, such as planar graphs, bounded-genus graphs, and bounded-treewidth graphs, are minor-closed, so they can be characterized by a finite list of excluded minors.

Distributed Approximation. There is a long line of research studying distributed approximation on graphs with an excluded minor [3, 5, 12, 25–31, 71, 88]. Czygrinow, Hańćkowiak, and Wawrzyniak [29] showed that an $(1 \pm \varepsilon)$-approximation of maximum matching, maximum independent set, and minimum dominating set of a planar graph can be constructed in $O(\log^* n)$ rounds deterministically in the LOCAL model, for any constant $\varepsilon > 0$. The algorithm for minimum dominating set was later extended to k-dominating set on bounded-genus graphs [5, 29, 30]. These algorithms are based on a generic approach [5, 25, 26, 29–31] using low-diameter decompositions. A common ingredient shared by all these algorithms is a computation of an $(1 \pm \varepsilon)$-approximate solution of each low-diameter cluster via a brute-force information gathering, requiring sending unbounded-size messages and confining all these algorithms to the LOCAL model. Our framework which is based on expander decompositions provides an opportunity to extend this line of research to the CONGEST model.

Low-congestion Shortcuts and its Applications. There is a line of work designing efficient algorithms on networks with an excluded minor via low-congestion shortcuts [46, 47, 53, 57–61]. Given a partition of the vertex set V of a graph $G = (V, E)$ into connected clusters $V = V_1 \cup V_2 \cup \cdots \cup V_k$, a low-congestion shortcut with congestion c and dilation d is a set of subgraphs $H_1, H_2, \ldots, H_k$ such that the diameter of $G[V_i] + H_i$ is at most d and each edge belongs to at most c subgraphs H_i. Here $G[V_i] + H_i$ denotes the subgraph of G induced by the union of the edges in $G[V_i]$ and H_i. For any clustering of an H-minor-free graph, there is an $\tilde{O}(D)$-round CONGEST algorithm computing a low-congestion shortcut with $c = O(D \log n)$ and $d = O(D)$, where D is the diameter of the graph [47]. As a result, many graph problems, including minimum spanning tree, minimum cut, and shortest-path approximations, can be solved in near-optimal $\tilde{O}(D)$ rounds in CONGEST on any H-minor-free graph [46, 57].

The type of problems efficiently solvable via low-congestion shortcuts is fundamentally very different from the type of problems efficiently solvable via our framework. Low-congestion shortcut is useful in designing near-optimal $\tilde{O}(D)$-round algorithms for *global* problems that already require $\Omega(D)$ rounds to solve. Our framework is useful in designing algorithms that take $\log^{O(1)} n$ or $n^{o(1)}$ rounds for *local* problems that do not have the $\Omega(D)$ lower bound.

1.5 Organization

Our framework of algorithm design based on expander decompositions is presented in Section 2. Using this framework, in Section 3, we give $\mathrm{poly}(1/\varepsilon, \log n)$-round randomized algorithms and $n^{o(1)} \cdot \mathrm{poly}(1/\varepsilon)$-round deterministic algorithms for various optimization, property testing, and graph decomposition problems on planar or H-minor-free networks, proving Theorems 1.2 to 1.5. Due to the page constraint, the proofs of Theorems 1.1 and 1.6 are left to the full version [21] of the paper.

2 GRAPH PARTITIONING

Let $G = (V, E)$ be a graph. Consider the following graph terminology regarding a subset $S \subseteq V$.

$$\begin{aligned}
\mathrm{vol}(S) &= \sum_{v \in S} \deg(v), \\
\partial(S) &= E(S, V \setminus S) \\
&= \{e = \{u, v\} \in E \mid \{u, v\} \cap S \neq \emptyset \text{ and } \{u, v\} \cap (V \setminus S) \neq \emptyset\}, \\
\Phi(S) &= \begin{cases} 0, & S = \emptyset \text{ or } S = V, \\ \frac{|\partial(S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}, & S \neq \emptyset \text{ and } S \neq V. \end{cases}
\end{aligned}$$

We call $\mathrm{vol}(S)$ the *volume* of the vertex set S. When S is interpreted as a cut $\{S, V \setminus S\}$, we call $\Phi(S)$ the *conductance* of the cut S.

Graph Conductance. The *conductance* of a graph G is defined as

$$\Phi(G) = \min_{S \subseteq V \text{ s.t. } S \neq \emptyset \text{ and } S \neq V} \Phi(S).$$

In other words, $\Phi(G)$ is the minimum value of $\Phi(S)$ over all non-trivial cuts $S \subseteq V$.

Mixing Time. A *uniform lazy random walk* starting at a vertex $v \in V$ is described by the following probability distribution, where $N(u)$ denotes the set of neighbors of u.

$$p_0^v(u) = \begin{cases} 1, & u = v, \\ 0, & u \neq v, \end{cases}$$

$$p_i^v(u) = \frac{1}{2} \cdot p_{i-1}^v(u) + \frac{1}{2\deg(u)} \cdot \sum_{w \in N(u)} p_{i-1}^v(w), \quad \text{for} \quad i \geq 1.$$

If G is connected, then the stationary distribution of a uniform lazy random walk is $\pi(u) = \deg(u)/\mathrm{vol}(V)$, regardless of the starting vertex v. The *mixing time* $\tau_{\mathrm{mix}}(G)$ of G is defined as the minimum number t such that $|p_t^v(u) - \pi(u)| \leq \pi(u)/|V|$ for all $u \in V$ and $v \in V$. The following relation [64] between the mixing time $\tau_{\mathrm{mix}}(G)$ and conductance $\Phi(G)$ is well-known:

$$\Theta\left(\frac{1}{\Phi(G)}\right) \leq \tau_{\mathrm{mix}}(G) \leq \Theta\left(\frac{\log|V|}{\Phi(G)^2}\right).$$

Expander Decompositions. We say that G is an *ϕ-expander* if $\Phi(G) \geq \phi$. An (ε, ϕ) *expander decomposition* of a graph is a removal of at most ε fraction of the edges such that each remaining connected component has conductance at least ϕ. Formally, an (ε, ϕ) expander decomposition of G is a partition $E = E_1 \cup E_2 \cup \cdots \cup E_k \cup E^{\mathrm{r}}$ of the edge set E meeting the following requirements.

- The set of inter-cluster edges E^{r} satisfies $|E^{\mathrm{r}}| \leq \varepsilon|E|$.
- We write $V_i \subseteq V$ to denote the set of vertices incident to an edge in E_i. It is required that $V = V_1 \cup V_2 \cup \cdots \cup V_k$ partitions the vertex set V and $G_i = (V_i, E_i)$ has conductance $\Phi(G_i) \geq \phi$ for each $1 \leq i \leq k$.

Existentially, it is well known that for any n-vertex graph, an (ϵ, ϕ)-expander decomposition exists for any $0 < \epsilon < 1$ and $\phi = \Omega(\epsilon/\log n)$ [55, 65, 84], and this bound is tight. After removing any constant fraction of the edges in a hypercube, some remaining component must have conductance at most $O(1/\log n)$ [4].

The following distributed algorithms for constructing expander decompositions are due to Chang and Saranurak [20].

Theorem 2.1. *For any $0 < \varepsilon < 1$, an (ε, ϕ) expander decomposition of a graph $G = (V, E)$ with $\phi = \varepsilon^{O(1)} \log^{-O(1)} n$ can be constructed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability.*

Theorem 2.2. *For any $0 < \varepsilon < 1$, an (ε, ϕ) expander decomposition of a graph $G = (V, E)$ with $\phi = \varepsilon^{O(1)} 2^{-O(\sqrt{\log n \log\log n})}$ can be constructed in $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically.*

2.1 Existence of a High-degree Vertex

Let $E = E_1 \cup E_2 \cup \cdots \cup E_k \cup E^{\mathrm{r}}$ be any (ε, ϕ) expander decomposition. Let $G_i = (V_i, E_i)$ be the subgraph of G induced by E_i. Let Δ_i be the maximum degree of the graph G_i. Recall that $V = V_1 \cup V_2 \cup \cdots \cup V_k$ partitions the vertex set V and observe that G_i is a subgraph of $G[V_i]$, the subgraph of G induced by the vertex set V_i. We will show that if G is H-minor-free, then there must exist a vertex in each $G[V_i]$ whose degree is $\Omega(\phi^2)|V_i|$ for any (ε, ϕ) expander decomposition of G.

Edge Separators. An *edge separator* of a graph is a cut $\{S, V \setminus S\}$ such that

$$\min\{|S|, |V \setminus S|\} \geq |V|/3.$$

The *size* of an edge separator is the number of cut edges $|\partial(S)|$. In the full version [21] of the paper, we show that any H-minor-free graph G admits an edge separator of size $O(\sqrt{\Delta|V|})$, where the hidden constant in $O(\cdot)$ depends only on H. The following lemma is a consequence of this result.

Lemma 2.3. *If G is H-minor-free, then $\Delta_i = \Omega(\phi^2)|V_i|$ for each cluster $G_i = (V_i, E_i)$ of any (ε, ϕ) expander decomposition of G. The hidden constant in $\Omega(\cdot)$ depends only on H.*

Proof. We focus on the H-minor-free graph $G_i = (V_i, E_i)$ in the proof. Consider any $O(\sqrt{\Delta_i|V_i|})$-size edge separator S of G_i. The fact that $\min\{|S|, |V_i \setminus S|\} \geq |V_i|/3$ implies

$$\min\{\mathrm{vol}(S), \mathrm{vol}(V_i \setminus S)\} \geq \min\{|S|, |V_i \setminus S|\} = \Omega(|V_i|),$$

and hence

$$\begin{aligned} \phi \leq \Phi(G_i) \leq \Phi(S) &= \frac{|\partial(S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}} \\ &= O\left(\frac{\sqrt{\Delta_i|V_i|}}{V_i}\right) = O\left(\sqrt{\frac{\Delta_i}{|V_i|}}\right), \end{aligned}$$

which implies $\Delta_i = \Omega(\phi^2)|V_i|$. □

2.2 Routing

Select v_i^* as any vertex $v \in V_i$ that has the maximum degree Δ_i in G_i. We show that the bound given by Lemma 2.3 implies an efficient algorithm for v_i^* to learn the entire graph topology of $G[V_i]$.

Expander Routing. Consider a routing task where each vertex v is the source and the destination of at most $L \cdot \deg(v)$ $O(\log n)$-bit messages. If G is an ϕ-expander, then such a task can be solved in $L \cdot \tau_{\mathrm{mix}} \cdot 2^{O(\sqrt{\log n})} = L \cdot \phi^{-2} \cdot 2^{O(\sqrt{\log n})}$ rounds with high probability [51, 52] or $L \cdot \phi^{-O(1)} \cdot 2^{O(\log^{2/3} n \log^{1/3} \log n)}$ rounds deterministically [20].

Edge Density of H-minor-free Graphs. The *edge density* of a graph G is $|E|/|V|$. It is well-known that any H-minor-free graph has edge density $O(1)$, where the constant $O(1)$ depends only on H. Specifically, Thomason [87] showed that any K_t-minor-free graph $G = (V, E)$ satisfies $|E| = O(t\sqrt{\log t}) \cdot |V|$. Moreover, Barenboim and Elkin [11] showed that given an upper bound d on the edge density of a graph $G = (V, E)$, its edge set E can be oriented such that the out-degree of each vertex is at most $O(d)$ in $O(\log n)$ rounds. As a result, for any H-minor-free graph G, in $O(\log n)$ rounds we can orient its edges such that each $v \in V_i$ has out-degree $O(1)$.

Information Gathering. In view of the above discussion, the task of letting v_i^* learn the entire graph topology of $G[V_i]$ can be reduced routing $O(1)$ messages of $O(\log n)$ bits from each $v \in V_i$ to v_i^*, as we can first spend $O(\log n)$ rounds to find an edge orientation of $G[V_i]$ with $O(1)$ out-degree, and then each vertex $v \in V_i$ only has to send information about its outgoing edges in $G[V_i]$ to v_i^*.

By Lemma 2.3, if G is H-minor-free, then the degree of v_i^* in $G_i = (V_i, E_i)$ is $\Omega(\phi^2)|V_i|$, so the number of $O(\log n)$-bit messages sent to v_i^* in this routing task is $O(\phi^{-2}) \cdot \deg_{G_i}(v_i^*)$. Therefore, using expander routing, this routing task can be solved in $\phi^{-2} \cdot 2^{O(\sqrt{\log n})}$ rounds with high probability [51, 52] or $\phi^{-O(1)} \cdot 2^{O(\log^{2/3} n \log^{1/3} \log n)}$ rounds deterministically [20].

We provide faster randomized and deterministic algorithms for this task in Lemmas 2.4 and 2.5. Due to Lemma 2.3, the conditions in Lemmas 2.4 and 2.5 are satisfied. In these lemmas, n denotes the number of vertices in the underlying network G, not the number of vertices in one cluster G_i. The above discussion on the edge density of H-minor-free graphs implies that for any H-minor-free graph G, the degree of v_i^* in $G_i = (V_i, E_i)$ is $\Omega(\phi^2)|V_i| = \Omega(\phi^2)|E_i|$ by Lemma 2.3.

Lemma 2.4. *Suppose* $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$. *In* $O(\phi^{-4}\log^3 n)$ *rounds, an* $O(\log n)$*-bit message from each* $v \in V_i$ *can be routed via the edges* E_i *to* v_i^* *with high probability.*

Proof. The algorithm runs a lazy random walk of length $O(\phi^{-4}\log^2 n)$ from each vertex $v \in V_i$ in parallel. We claim that each step of the lazy random walk can be simulated in $O(\log n)$ rounds with probability $1 - 1/\text{poly}(n)$, so the overall round complexity is $O(\phi^{-4}\log^3 n)$. To prove this claim, observe that for each edge $e \in E_i$ and for each j, the expected number of random walks traversing e in the jth step is $O(1)$, so a Chernoff bound implies that this number is at most $O(\log n)$ with probability $1 - 1/\text{poly}(n)$. By a union bound over all $e \in E_i$ and $1 \leq j \leq O(\phi^{-4}\log^2 n)$, the number of $O(\log n)$-bit messages sent along each edge in each step is $O(\log n)$ with probability $1 - 1/\text{poly}(n)$.

For the correctness of the algorithm, we show that with probability $1 - 1/\text{poly}(n)$ each random walk passes v_i^*, so in the end v_i^* receives all the messages. After $\tau_{\text{mix}} = O(\phi^{-2}\log n)$ lazy random walk steps, it lands at a random vertex according to the degree distribution $\pi(u) = \deg_{G_i}(u)/2|E_i|$, up to a small additive error $\pm\pi(u)/n$. In particular, it lands at v_i^* with probability $\Omega(\deg_{G_i}(v_i^*)/|E_i|) = \Omega(\phi^2)$. Thus, after $s = O(\phi^{-2}\log n)$ segments of random walks of length $\tau_{\text{mix}} = O(\phi^{-2}\log n)$, the probability that the walk never reach v_i^* is at most $(1 - \Omega(\phi^2))^s = n^{-\Omega(1)}$. □

Lemma 2.5. *Suppose* $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$. *In* $O(\phi^{-18}) \cdot 2^{O(\sqrt{\log n})}$ *rounds, an* $O(\log n)$*-bit message from each* $v \in V_i$ *can be routed via the edges* E_i *to* v_i^* *deterministically.*

Proof. Although this routing task can be solved in $\phi^{-O(1)} \cdot 2^{O(\log^{2/3} n \log^{1/3} \log n)}$ rounds using deterministic expander routing [20], we provide a faster and more direct algorithm via an almost maximal flow algorithm [20, Lemma D.10].[2]

In order to apply [20, Lemma D.10], we need to do some preprocessing to the graph G_i. Let G_i' be the result of replacing each vertex v in G_i' by a $\deg_{G_i}(v)$-vertex graph X_v with $\Theta(1)$ conductance and $\Theta(1)$ maximum degree in such a way that the $\deg_{G_i}(v)$ edges in E_i incident to v are attached to distinct $\deg_{G_i}(v)$ vertices in X_v. Observe that the new graph G_i' has maximum degree $O(1)$.

Define the *sparsity* of a cut S of a graph $G = (V, E)$ as $\Psi(S) = \frac{|\partial(S)|}{\min\{|S|,|V \setminus S|\}}$ if $S \neq \emptyset$ and $S \neq V$. Define the *sparsity* of a graph $G = (V, E)$ as the minimum sparsity over all cuts $S \subseteq V$ with $S \neq \emptyset$ and $S \neq V$. Then we must have $\Psi(G_i') = \Theta(\Phi(G_i)) = \Omega(\phi)$ [20, Lemma C.2].

Next, define G_i'' as the result of replacing each vertex $u \in X_{v_i^*}$ by an $O(\phi^{-2})$-vertex graph Y_u with $\Theta(1)$ conductance and $\Theta(1)$ maximum degree such that $T = \bigcup_{u \in X_{v_i^*}} Y_u$ constitutes more than half of the vertices in G_i''. It is clear that the new graph G_i'' has sparsity $\Omega(\phi^3)$ [20, Lemma C.1] and maximum degree $O(1)$.

Now, let S be the vertices in G_i'' that are not in T. Then the original routing problem is reduced to finding a set of paths from each $v \in S$ to an arbitrary vertex in T. Suppose that this set of paths satisfies that the maximum path length is d and each vertex belongs to at most c paths, then the routing can be done with an additional $O(cd)$ rounds. As $|S| < |T|$, [20, Lemma D.10] shows that such a set of paths with $c = O(\Delta\psi^{-1}\log^{3/2} n)$ and $d = O(\Delta^2\psi^{-2}) \cdot 2^{O(\sqrt{\log n})}$ can be found in $t = O(\Delta^6\psi^{-6}) \cdot 2^{O(\sqrt{\log n})}$ rounds. Here $\Delta = O(1)$ is the maximum degree and $\psi = \Omega(\phi^3)$ is the sparsity. Therefore, the overall round complexity of routing is $O(t + cd) = O(\phi^{-18}) \cdot 2^{O(\sqrt{\log n})}$. □

Observe that the routing algorithms of Lemmas 2.4 and 2.5 can also be used to deliver an $O(\log n)$-bit message from v_i^* to each vertex $v \in V_i$ in G_i by reversing the routing procedure.

2.3 Summary

We summarize our results as a theorem.

Theorem 2.6. *Given any parameter* $0 < \varepsilon < 1$, *there is an algorithm for finding a partition* $V = V_1 \cup V_2 \cup \cdots \cup V_k$ *of the vertex set of an* H*-minor-free graph* $G = (V, E)$ *with the following properties.*

Inter-cluster Edges: *The number of inter-cluster edges is at most* $\varepsilon \min\{|V|, |E|\}$.

Construction Time: *The round complexity for partitioning the graph is* $\varepsilon^{-O(1)} \log^{O(1)} n$ *in the randomized setting and is* $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ *in the deterministic setting.*

Routing Time: *Each cluster* V_i *has a leader* $v_i^* \in V_i$ *that knows the entire graph topology of* $G[V_i]$. *Furthermore, we can let* v_i^* *exchange a distinct* $O(\log n)$*-bit message with each vertex* $v \in V_i$ *in* $\varepsilon^{-O(1)} \log^{O(1)} n$ *rounds in the randomized setting and in* $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ *rounds in the deterministic setting.*

Proof. Since G is H-minor-free, there is a constant $t = O(1)$ depending only on H such that $|E|/|V| \leq t$ [87]. The partition $V = V_1 \cup V_2 \cup \cdots \cup V_k$ is constructed using the expander decomposition algorithms of Theorems 2.1 and 2.2 with parameter $\varepsilon' = \varepsilon/t \leq \varepsilon$, so the requirement on the construction time is met. The upper bound on the number of inter-cluster edges $\varepsilon'|E| = \varepsilon|E|/t \leq \varepsilon|V|$ follows from the definition of an (ε, ϕ) expander decomposition.

For each cluster $G_i = (V_i, E_i)$ in the expander decomposition, in $O(\phi^{-1}\log n)$ rounds the vertices in G_i can select a vertex $v_i^* \in V_i$ that has the maximum degree in G_i. The algorithm for selecting v_i^* is as follows. In the first step, each vertex v in G_i broadcasts $(\text{ID}(v), \deg_{G_i}(v))$ to its neighbors in G_i. After that, in each round each vertex v maintains a pair $(\text{ID}(u), \deg_{G_i}(u))$ that has the highest $\deg_{G_i}(u)$ over all pairs that v has received, breaking the tie by comparing $\text{ID}(u)$, and v broadcasts this pair $(\text{ID}(u), \deg_{G_i}(u))$ to all its neighbors in G_i. The graph G_i has diameter $O(\phi^{-1}\log n)$ because G_i is an ϕ-expander. Therefore, after $O(\phi^{-1}\log n)$ rounds of communication, all vertices in G_i agree with the same pair $(\text{ID}(u), \deg_{G_i}(u))$ and we may set $v_i^* = u$.

[2] See the full version arXiv:2007.14898v1 of [20].

For learning the graph topology of $G[V_i]$ and routing, we apply the routing algorithms of Lemmas 2.4 and 2.5 to $G[V_i]$, in parallel for all $1 \leq i \leq k$. In view of Theorems 2.1 and 2.2, we use $\phi = \varepsilon^{O(1)} \log^{-O(1)} n$ in the randomized setting and $\phi = \varepsilon^{O(1)} 2^{-O(\sqrt{\log n \log \log n})}$ in the deterministic setting. Due to Lemma 2.3, the conditions in Lemmas 2.4 and 2.5 are satisfied. The requirement on the routing time is met in view of the round complexities specified in Lemmas 2.4 and 2.5. □

The Behavior of a Failed Execution. We briefly discuss the behavior of the algorithm of Theorem 2.6 when it fails. For example, if G is not H-minor-free, then the algorithm of Theorem 2.6 might not work successfully. Note that the choice of the parameter t in the algorithm of Theorem 2.6 depends only on H, regardless of whether the underlying graph is H-minor-free.

Even if G is H-minor-free, the algorithm might fail with a probability of $1/\text{poly}(n)$ in the randomized setting. Understanding the behavior of a failed execution of the algorithm of Theorem 2.6 is crucial to its application in property testing, which we will discuss in Section 3.4, as there is no guarantee that the underlying network G is H-minor-free.

The algorithm of Theorem 2.6 has two parts, the clustering step and the routing step.

Clustering Step. In the clustering step we may assume that the algorithm always outputs a clustering $V = V_1 \cup V_2 \cup \cdots \cup V_k$, even in a failed execution. In particular, if a vertex v is not assigned to any cluster, then v simply assign itself to the cluster $\{v\}$.

In a successful execution, each $G[V_i]$ has diameter $O(\phi^{-1} \log n)$ because G_i is an ϕ-expander and G_i is the result of removing some edges from $G[V_i]$. We can also guarantee that each cluster has this property even in a failed execution, as follows. Choose $b = O(\phi^{-1} \log n)$ be any upper bound on the cluster diameter for a successful execution of an expander decomposition algorithm. The number b depends only on ϕ and n. in $O(\phi^{-1} \log n)$ rounds, we run the following algorithm. Using b rounds, each vertex v computes the maximum $\text{ID}(u)$ over all vertices u within distance b to v in $G[V_i]$. After that, each vertex v compares its result with its neighbors in $G[V_i]$, and then v marks itself $*$ if there is a disagreement. Finally, each vertex $v \in V_i$ checks in $2b+1$ rounds whether there is a vertex $u \in V_i$ within distance $2b+1$ to v that is marked $*$. If such a vertex u exists, then v also marks itself $*$. It is that there are two possible outcomes. Either all vertices in V_i are marked $*$, or all vertices in V_i are not marked $*$. If the diameter of $G[V_i]$ is at most b, then all vertices in V_i are not marked $*$. If the diameter of $G[V_i]$ is at least $2b+1$, then all vertices in V_i are marked $*$. Hence if a vertex v is marked $*$, it knows that the clustering step has failed, in which case we can let v reset its cluster to be $\{v\}$.

For the number of inter-cluster edges, if G is H-minor-free, then the upper bound $\varepsilon \min\{|V|, |E|\}$ is always satisfied in the deterministic setting and it is satisfied with probability $1 - 1/\text{poly}(n)$ in the randomized setting. Recall that the algorithm of Theorem 2.6 is based on an expander decomposition algorithm with parameter $\varepsilon' \leq \varepsilon$. Since the expander decomposition algorithm does not rely on the assumption that G is H-minor-free, the weaker upper bound $\varepsilon|E|$ on the number of inter-cluster edges holds regardless of whether the input graph G is H-minor-free or not.

Routing Step. We distinguish between different reasons for the routing algorithms of Lemmas 2.4 and 2.5 to fail. The first reason of failure is that the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ for Lemmas 2.4 and 2.5 is not satisfied. In view of Lemma 2.3, the only possibility that this condition is not met is when G is not H-minor-free. In view of the above discussion, each cluster $G[V_i]$ always has diameter $O(\phi^{-1} \log n)$, so whether the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ is satisfied can be checked in $O(\phi^{-1} \log n)$ rounds.

Even if the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ is met, the routing algorithms of Lemmas 2.4 and 2.5 might still fail. There are two possible reasons. One reason is that $\Phi(G_i) < \phi$ is too small due to an error in the expander decomposition algorithm in the clustering step. The other reason is because that in the randomized setting there is a small probability that the algorithm might fail. In either case, the failure occurs with probability $1/\text{poly}(n)$, regardless of whether G is H-minor-free.

In an failed execution of the routing algorithms of Lemmas 2.4 and 2.5, only a subset of all messages are delivered. To detect a failure of delivery of a message, we can simply reverse the execution of the algorithm. Once a vertex $v \in V_i$ detects that some of its messages are not successfully delivered, it broadcasts to all vertices in V_i that the routing algorithm has failed in $O(\phi^{-1} \log n)$ rounds. Hence we can assume that all vertices in a cluster V_i know whether the routing algorithm is successful.

3 APPLICATIONS

Using Theorem 2.6, we give $\text{poly}(1/\varepsilon, \log n)$-round randomized algorithms and $n^{o(1)} \cdot \text{poly}(1/\varepsilon)$-round deterministic algorithms for various optimization, property testing, and graph decomposition problems on planar or H-minor-free networks.

3.1 Maximum Independent Set

We design an efficient algorithm for computing a $(1-\varepsilon)$-approximate maximum independent set of any H-minor-free network by combining Theorem 2.6 with the approach of Czygrinow, Hańćkowiak, and Wawrzyniak [29].

Let G be an H-minor-free graph. Let $\alpha(G)$ denote the size of the maximum independent set of G. Recall that any H-minor-free graph has edge density $d = O(1)$, where the constant $O(1)$ depends only on H [87]. For any H-minor-free graph, as $|E|/|V| \leq d$, its minimum degree is at most $2d$. Hence $\alpha(G) = \Theta(n)$, as an independent set I of size at least $n/(2d+1)$ can be computed by repeatedly adding a minimum-degree vertex v to I and removing all its neighboring vertices. For example, if G is planar, then $\alpha(G) \geq n/4$ due to the four color theorem.

Run the algorithm of Theorem 2.6 on G with parameter $\varepsilon' = \varepsilon/(2d+1)$ to partition the vertices into $V = V_1 \cup V_2 \cup \ldots \cup V_k$. For each V_i, we route the entire graph topology of $G[V_i]$ into v_i^* and let v_i^* compute the maximum independent set I_i of $G[V_i]$ locally. Then, v_i^* sends a message to each vertex in V_i to inform if it is in I_i.

Let $I = I_1 \cup \ldots \cup I_k$. For each edge $e = \{u, v\}$, if both u and v are in I, we add one of u and v to Z. Since this can only happen if e is an inter-cluster edge, we have $|Z| \leq \varepsilon' \cdot n$. Let $I' = I \setminus Z$. Clearly, I' is an independent set.

Using the fact that $\alpha(G) \geq n/(2d+1)$ and $\varepsilon' = \varepsilon/(2d+1)$, we have

$$|I'| = \left(\sum_{i=1}^{k} |I_i|\right) - |Z| \geq \alpha(G) - |Z| \geq \alpha(G) - \varepsilon' \cdot n$$
$$\geq \alpha(G) - \varepsilon\alpha(G) = (1-\varepsilon)\alpha(G).$$

Hence we conclude the following theorem.

Theorem 1.2. *A $(1-\varepsilon)$-approximate maximum independent set of an H-minor-free network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically in the* CONGEST *model.*

3.2 Maximum Cardinality Matching in Planar Graphs

We show that a $(1-\varepsilon)$-approximate maximum cardinality matching of a planar network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically.

Let G be a planar graph. We begin by creating a new graph $\bar{G} = (\bar{V}, \bar{E})$ from G by removing some vertices from G such that the sizes of the maximum matching are the same in G and in $\bar{G}$. Moreover, the size of maximum matching is at least $\Omega(|\bar{V}|)$. After applying the algorithm of Theorem 2.6 on $\bar{G}$, the leader v_i^* of each cluster gathers the graph topology of the cluster and compute the maximum matching locally.

A *k-star* is a subgraph induced by the vertices $\{x, v_1, \ldots, v_k\}$ where for each $1 \leq i \leq k$, $\deg(v_i) = 1$ and there is an edge connecting x and v_i. A *k-double-star* is a subgraph induced by the vertices $\{x, y, v_1, \ldots, v_k\}$ where for each $1 \leq i \leq k$, $\deg(v_i) = 2$ and there are two edges $\{x, v_i\}$ and $\{y, v_i\}$. The graph $\bar{G}$ is created by eliminating all 2-stars and 3-double-stars from G by the following procedure.

To eliminate 2-stars, every vertex u with degree 1 sends a token (u) to its neighbor. Then every vertex who has received more than one tokens bounces all the tokens, *except one of them*, back to their originators. Vertices of degree 1 whose token was bounced back are removed from G. To eliminate 3-double stars, every vertex u with exactly two neighbors u_1, u_2 sends a token $(u, (u_1, u_2))$ to their neighbors. Every vertex then aggregate the tokens based on the second coordinate, the 2-tuple (u_1, u_2). If there are more than 2 tokens with the same second coordinate, all *but two of them* are bounced back to their originators. Vertices whose tokens were bounced back are removed from G.

Note that the eliminations of 2-stars and 3-double-stars do not change the size of the maximum matching. Moreover, we have the following property:

Lemma 3.1 ([27, Lemma 6]). *Let $G = (V, E)$ be a planar graph with $n = |V|$ and no isolated vertices. If G contains no 2-stars and 3-double-stars then the size of the maximum matching of G is $\Omega(n)$.*

By Lemma 3.1, the size of the maximum matching of $|\bar{V}|$ is at least $c \cdot |\bar{V}|$ for some constant $c > 0$. Czygrinow, Hańćkowiak, and Wawrzyniak [29] used Lemma 3.1 to design an $O(\log^* n)$-round deterministic algorithm for computing a $(1-\varepsilon)$-approximate maximum cardinality matching in the LOCAL model, for any constant $\varepsilon > 0$. We show that Theorem 2.6 allows us to obtain efficient matching algorithms in the CONGEST model as well.

Now we run the algorithm of Theorem 2.6 on $\bar{G}$ with parameter $\varepsilon' = c \cdot \varepsilon$ to partition the vertices into $\bar{V} = V_1 \cup V_2 \cup \ldots \cup V_k$. For each V_i, we route the entire graph topology of $\bar{G}[V_i]$ into v_i^* and let it compute the maximum matching M_i of $\bar{G}[V_i]$ locally. We claim that the union of the matching $M = M_1 \cup M_2 \cup \ldots \cup M_k$ is an $(1-\varepsilon)$-approximate maximum matching.

Let M^* be a maximum matching. Let $M_i^* = M^* \cap (V_i \times V_i)$ be M^* restricted to V_i. We have

$$|M| = \sum_{i=1}^{k} |M_i|$$
$$\geq \sum_{i=1}^{k} |M_i^*| = |M^*| - (\text{\# inter-cluster } M^*\text{-edges})$$
$$\geq |M^*| - \varepsilon' \cdot |\bar{V}|$$
$$= |M^*| - \varepsilon \cdot c \cdot |\bar{V}| \geq |M^*| - \varepsilon|M^*|$$
$$= (1-\varepsilon)|M^*|.$$

Hence we conclude the following theorem.

Theorem 3.2. *A $(1-\varepsilon)$-approximate maximum matching of a planar network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically in the* CONGEST *model.*

In the full version [21] of the paper, we will generalize this result to the more difficult maximum weighted matching problem for an arbitrary H-minor-free graph to prove Theorem 1.1.

3.3 Correlation Clustering

In the agreement maximization correlation clustering problem, the edge set is partitioned into $E = E^+ \cup E^-$ two parts, and the goal is to compute a clustering of the vertices $V = V_1 \cup V_2 \cup \cdots \cup V_k$ maximizing $\sum_{i=1}^{k} |E^+ \cap (V_i \times V_i)| + \sum_{1 \leq i < j \leq k} |E^+ \cap (V_i \times V_j)|$, which is the number of intra-cluster E^+-edges plus the number of inter-cluster E^--edges.

Given a partition $E = E^+ \cup E^-$ of the edges in G, let $\gamma(G)$ denote the optimal value for the agreement maximization correlation clustering problem. Note that $\gamma(G) \geq |E|/2$ if G is connected. This is because if $|E^+| \geq |E|/2$, we can put each vertex as a standalone cluster and get a score of at least $|E|/2$. Otherwise, putting every vertex in the same cluster yields a score of at least $|E|/2$.

Apply the algorithm of Theorem 2.6 on G with parameter $\epsilon' = \epsilon/2$ to partition the vertices into $V_1 \ldots V_k$. For each V_i, route the entire graph topology of $G[V_i]$ into v_i^* and let v_i^* compute an optimal correlation clustering C_i of $G[V_i]$. Let C be the union of $C_1, C_2, \ldots, C_k$. Let C^* be an optimal clustering and C_i^* be the restriction of C^* to V_i. Formally, C_i^* is constructed by adding $C \cap V_i$ to C_i^* for each cluster $C \in C^*$ such that $C \cap V_i \neq \emptyset$.

Using the fact that $\gamma(G) \geq |E|/2$ and $\epsilon' = \epsilon/2$, we have

$$\text{score}(C) \geq \sum_{i=1}^{k} \text{score}(C_i) \geq \sum_{i=1}^{k} \text{score}(C_i^*)$$
$$\geq \text{score}(C^*) - \epsilon'|E| \geq \gamma(G) - \epsilon\gamma(G) = (1-\epsilon)\gamma(G).$$

Hence we conclude the following theorem. Note that the requirement that G is H-minor-free is only used in applying Theorem 2.6.

Theorem 1.3. *A $(1 - \varepsilon)$-approximate agreement maximization correlation clustering of an H-minor-free network G can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically in the* CONGEST *model.*

3.4 Property Testing

We design an efficient algorithm for testing an arbitrary minor-closed property $\mathcal{P}$ that is closed under taking disjoint union. This covers many natural graph classes, including planar graphs, outerplanar graphs, graphs with treewidth at most w, and H-minor-free graphs for a fixed connected graph H.

We pick s to be the smallest positive integer such that $K_s \notin \mathcal{P}$, i.e. the s-vertex clique does not have property $\mathcal{P}$. If such a number s does not exist, then $\mathcal{P}$ contains the set of all cliques. Since $\mathcal{P}$ is minor-closed and any finite graph is a minor of some clique, $\mathcal{P}$ must be the trivial property that contains all graphs, in which case we have a trivial property tester that works by letting each vertex output Accept.

Algorithm. From now on we assume that s exists, and we let $H = K_s$ be the s-clique. Our property testing algorithm applies Theorem 2.6 under the assumption that the underlying graph is H-minor-free, and then each vertex v makes its decision as follows.

- Suppose that the routing algorithm of Theorem 2.6 works successfully for a cluster V_i, then v_i^* knows the graph topology of $G[V_i]$. We let v_i^* locally check whether $G[V_i]$ has property $\mathcal{P}$ and broadcast the results to all vertices in V_i. If $G[V_i]$ does not have property $\mathcal{P}$, then all vertices in V_i outputs Reject, otherwise all vertices in V_i outputs Accept.
- Suppose that the routing algorithm of Theorem 2.6 does not work successfully for a cluster V_i. If it fails because the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ is not met, then all vertices in V_i outputs Reject, otherwise all vertices in V_i outputs Accept.

Here we recall from the discussion in Section 2.3 that each cluster V_i is able to check whether the routing algorithm works successfully and whether the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ is met. Therefore, each vertex v is able to decide whether to output Accept or Reject in the above algorithm.

Analysis. Suppose G has property $\mathcal{P}$. Because $\mathcal{P}$ is minor-closed, $G[V_i]$ also has property $\mathcal{P}$. Moreover, $G \in \mathcal{P}$ implies that G is H-minor-free. Recall the discussion in Section 2.3 that the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ is not met only when G is not H-minor-free. Therefore, from the description of our algorithm, all vertices will output Accept with probability one.

Suppose G is ε-far from having property $\mathcal{P}$. There are two cases. The first case is that the algorithm of Theorem 2.6 does not fail. Recall that the algorithm of Theorem 2.6 is based on an expander decomposition algorithm with parameter $\varepsilon' \leq \varepsilon$. Therefore, as long as the execution of the expander decomposition algorithm is successful, the number of inter-cluster edges is at most $\varepsilon|E|$, regardless of whether the input graph G is H-minor-free or not. Since G is ε-far from having property $\mathcal{P}$, the graph G' resulting from removing all inter-cluster edges also does not have property $\mathcal{P}$. Since G' is the disjoint union of all clusters $G[V_i]$ and $\mathcal{P}$ is closed under taking disjoint union, there must be at least one cluster $G[V_i]$ that does not have property $\mathcal{P}$, so all vertices V_i in this cluster will output Reject, as required.

The second case is that the algorithm of Theorem 2.6 fails. If it fails because the condition $\deg_{G_i}(v_i^*) = \Omega(\phi^2)|E_i|$ is not met, then all vertices in V_i outputs Reject, as required. As discussed in Section 2.3, the probability that algorithm of Theorem 2.6 fails due to other reasons is at most $1/\text{poly}(n)$, so the probability that all vertices in the graph output Accept is at most $1/\text{poly}(n)$. In particular, a failure in the expander decomposition algorithm might cause the number of inter-cluster edges to be significantly higher than $\varepsilon|E|$, potentially causing all $G[V_i]$ to have property $\mathcal{P}$. Although such a failure might not be detected, it occurs with probability at most $1/\text{poly}(n)$.

Hence we conclude the following theorem. Note that in the randomized setting our algorithm has one-sided error in that all vertices output Accept with probability one if G has property $\mathcal{P}$.

Theorem 1.4. *Distributed property testing for any minor-closed graph property $\mathcal{P}$ that is closed under taking disjoint union can be solved in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically in the* CONGEST *model.*

Lower Bound. In the full version [21] of the paper, we give a concrete example of a minor-closed property $\mathcal{P}$ that is *not* closed under taking disjoint union and requires $\Omega(n)$ rounds to test even for constant $\varepsilon > 0$ and in the LOCAL model. This shows that the requirement in Theorem 1.4 that the graph property is closed under taking disjoint union is, in a sense, necessary.

3.5 Low-diameter Decompositions

Using Theorem 2.6, we design an efficient algorithm that finds a partition of the vertex set $V = V_1 \cup V_2 \cup \cdots \cup V_k$ such that the number of inter-cluster edges $\sum_{1 \leq i \leq k} |\partial(V_i)|/2$ is at most $\varepsilon|E|$ and the diameter of the induced subgraph $G[V_i]$ is at most $D = O(\varepsilon^{-1})$ for each $1 \leq i \leq k$.

We first run Theorem 2.6 with parameter $\tilde{\varepsilon} = \varepsilon/2$ to obtain a clustering $V = V_1 \cup V_2 \cup \cdots \cup V_k$ such that the number of inter-cluster edges is at most $\tilde{\varepsilon}|E| \leq \varepsilon|E|/2$. We then refine the cluster V_i by letting v_i^* compute a low-diameter decomposition of $G[V_i]$ with $\tilde{\varepsilon} = \varepsilon/2$ and $\tilde{D} = O(\tilde{\varepsilon}^{-1})$ using any known sequential algorithm [1, 40, 68] for this task. Hence each cluster in the final clustering has diameter $O(\varepsilon^{-1})$. This step introduces at most $\varepsilon|E|/2$ inter-cluster edges, so the total number of inter-cluster edges is at most $\varepsilon|E|/2 + \varepsilon|E|/2 = \varepsilon|E|$, as required.

Theorem 1.5. *Given an H-minor free network, a low-diameter decomposition with $D = O(\varepsilon^{-1})$ can be computed in $\varepsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\varepsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically in the* CONGEST *model.*

REFERENCES

[1] Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. 2019. Cops, Robbers, and Threatening Skeletons: Padded Decomposition for

Minor-Free Graphs. *SIAM J. Comput.* 48, 3 (2019), 1120–1145. https://doi.org/10.1137/17M1112406

[2] Mohamad Ahmadi, Fabian Kuhn, and Rotem Oshman. 2018. Distributed Approximate Maximum Matching in the CONGEST Model. In *Proc. 32nd International Symposium on Distributed Computing (DISC).* 6:1–6:17.

[3] Saeed Akhoondian Amiri, Stefan Schmid, and Sebastian Siebertz. 2016. A local constant factor MDS approximation for bounded genus graphs. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC).* 227–233.

[4] Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. 2018. Graph Clustering using Effective Resistance. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA.* 41:1–41:16.

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

[6] A. Arasu, C. Ré, and D. Suciu. 2009. Large-Scale Deduplication with Constraints Using Dedupalog. In *2009 IEEE 25th International Conference on Data Engineering.* 952–963.

[7] Sanjeev Arora, Boaz Barak, and David Steurer. 2015. Subexponential Algorithms for Unique Games and Related Problems. *J. ACM* 62, 5, Article 42 (Nov. 2015), 25 pages.

[8] Nir Bachrach, Keren Censor-Hillel, Michal Dory, Yuval Efron, Dean Leitersdorf, and Ami Paz. 2019. Hardness of Distributed Optimization. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC).* 238–247.

[9] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation Clustering. *Machine Learning* 56, 1-3 (2004), 89–113.

[10] Reuven Bar-Yehuda, Keren Censor-Hillel, Mohsen Ghaffari, and Gregory Schwartzman. 2017. Distributed Approximation of Maximum Independent Set and Maximum Matching. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC).* 165–174.

[11] Leonid Barenboim and Michael Elkin. 2010. Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition. *Distributed Computing* 22, 5-6 (2010), 363–379.

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

[13] Francesco Bonchi, David Garcia-Soriano, and Edo Liberty. 2014. Correlation Clustering: From Theory to Practice. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14).* 1972–1972.

[14] Mélanie Cambus, Davin Choo, Havu Miikonen, and Jara Uitto. 2021. Massively Parallel Correlation Clustering in Bounded Arboricity Graphs. In *35th International Symposium on Distributed Computing (DISC)*, Vol. 209. 15:1–15:18.

[15] Keren Censor-Hillel. 2021. Distributed Subgraph Finding: Progress and Challenges. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 198)*, Nikhil Bansal, Emanuela Merelli, and James Worrell (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 3:1–3:14. https://doi.org/10.4230/LIPIcs.ICALP.2021.3

[16] Keren Censor-Hillel, Yi-Jun Chang, François Le Gall, and Dean Leitersdorf. 2021. Tight distributed listing of cliques. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA).* SIAM, 2878–2891.

[17] Keren Censor-Hillel, Eldar Fischer, Gregory Schwartzman, and Yadu Vasudev. 2019. Fast distributed algorithms for testing graph properties. *Distributed Computing* 32, 1 (2019), 41–57.

[18] Keren Censor-Hillel, François Le Gall, and Dean Leitersdorf. 2020. On Distributed Listing of Cliques. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC).* 474–482.

[19] Yi-Jun Chang, Seth Pettie, Thatchaphol Saranurak, and Hengjie Zhang. 2021. Near-Optimal Distributed Triangle Enumeration via Expander Decompositions. *J. ACM* 68, 3, Article 21 (May 2021). https://doi.org/10.1145/3446330

[20] Yi-Jun Chang and Thatchaphol Saranurak. 2020. Deterministic Distributed Expander Decomposition and Routing with Applications in Distributed Derandomization. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS).*

[21] Yi-Jun Chang and Hsin-Hao Su. 2022. Narrowing the LOCAL–CONGEST Gaps in Sparse Networks via Expander Decompositions. *arXiv preprint arXiv:2205.08093* (2022).

[22] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. 2005. Clustering with qualitative information. *J. Comput. System Sci.* 71, 3 (2005), 360 – 383.

[23] Flavio Chierichetti, Nilesh Dalvi, and Ravi Kumar. 2014. Correlation Clustering in MapReduce. In *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 641–650.

[24] Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. 2020. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *Proeedings of the IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS).* IEEE, 1158–1167.

[25] Andrzej Czygrinow and Michał Hańćkowiak. 2006. Distributed Almost Exact Approximations for Minor-Closed Families. In *Algorithms – ESA 2006*, Yossi Azar and Thomas Erlebach (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 244–255.

[26] A. Czygrinow and M. Hańćkowiak. 2007. Distributed Approximation Algorithms for Weighted Problems in Minor-Closed Families. In *Proceedings of the 13th Annual International Conference on Computing and Combinatorics (COCOON).* Springer-Verlag, Berlin, Heidelberg, 515–525.

[27] Andrzej Czygrinow, Michał Hańćkowiak, and Edyta Szymańska. 2006. Distributed Approximation Algorithms for Planar Graphs. In *Italian Conference on Algorithms and Complexity (CIAC).*

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

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

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

[31] A. Czygrinow and M. Hańćkowiak. 2006. Distributed algorithms for weighted problems in sparse graphs. *Journal of Discrete Algorithms* 4, 4 (2006), 588–607. https://doi.org/10.1016/j.jda.2005.07.006

[32] Mohit Daga, Monika Henzinger, Danupon Nanongkai, and Thatchaphol Saranurak. 2019. Distributed Edge Connectivity in Sublinear Time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Phoenix, AZ, USA) *(STOC 2019).* Association for Computing Machinery, New York, NY, USA, 343–354.

[33] K. Diks, H.N. Djidjev, O. Sykora, and I. Vrto. 1993. Edge Separators of Planar and Outerplanar Graphs With Applications. *Journal of Algorithms* 14, 2 (1993), 258–279. https://doi.org/10.1006/jagm.1993.1013

[34] R. Duan and S. Pettie. 2014. Linear Time Approximation for Maximum Weight Matching. *J. ACM* 61, 1 (2014). Article 1.

[35] Talya Eden, Nimrod Fiat, Orr Fischer, Fabian Kuhn, and Rotem Oshman. 2019. Sublinear-Time Distributed Algorithms for Detecting Small Cliques and Even Cycles. In *Proceedings of the International Symposium on Distributed Computing (DISC).* 15:1–15:16.

[36] Yuval Efron, Ofer Grossman, and Seri Khoury. 2020. Beyond Alice and Bob: Improved Inapproximability for Maximum Independent Set in CONGEST. In *ACM Symposium on Principles of Distributed Computing (PODC)*, Yuval Emek and Christian Cachin (Eds.). 511–520.

[37] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 19, 1 (2007), 1–16.

[38] Micha Elsner and Warren Schudy. 2009. Bounding and Comparing Methods for Correlation Clustering Beyond ILP. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing.* Association for Computational Linguistics, Boulder, Colorado, 19–27. https://www.aclweb.org/anthology/W09-1803

[39] Dotan Emanuel, Amos Fiat, and Uri Zwick. 2003. Correlation Clustering – Minimizing Disagreements on Arbitrary Weighted Graphs. In *Proc. European Symposium on Algorithms (ESA).* 208–220.

[40] Jittat Fakcharoenphol and Kunal Talwar. 2003. An Improved Decomposition Theorem for Graphs Excluding a Fixed Minor. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 36–46.

[41] Salwa Faour, Marc Fuchs, and Fabian Kuhn. 2021. Distributed CONGEST Approximation of Weighted Vertex Covers and Matchings. *CoRR* abs/2111.10577 (2021). arXiv:2111.10577 https://arxiv.org/abs/2111.10577

[42] Manuela Fischer. 2018. Improved deterministic distributed matching via rounding. *Distributed Computing* (2018), 1–13.

[43] Manuela Fischer, Slobodan Mitrovic, and Jara Uitto. 2021. Deterministic $(1+\epsilon)$-Approximate Maximum Matching with poly$(1/\epsilon)$ Passes in the Semi-Streaming Model. *To appear in STOC 2022* abs/2106.04179 (2021).

[44] Mohsen Ghaffari, Christoph Grunau, and Václav Rozhon. 2021. Improved Deterministic Network Decomposition. In *Proc. 2021 ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 2904–2923.

[45] Mohsen Ghaffari and Bernhard Haeupler. 2016. Distributed algorithms for planar networks I: Planar embedding. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC).* 29–38.

[46] Mohsen Ghaffari and Bernhard Haeupler. 2016. 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). SIAM, 202–219.

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

[48] Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. 2018. On Derandomizing Local Distributed Algorithms. In *Proc. 59th IEEE Symposium on Foundations of Computer Science (FOCS).* 662–673.

[49] M. Ghaffari, F. Kuhn, and Y. Maus. 2017. On the complexity of local distributed graph problems. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC).* 784–797. https://doi.org/10.1145/3055399.3055471

[50] Mohsen Ghaffari, Fabian Kuhn, Yannic Maus, and Jara Uitto. 2018. Deterministic Distributed Edge-coloring with Fewer Colors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing* (Los Angeles, CA, USA) *(STOC 2018).* ACM, New York, NY, USA, 418–430. https://doi.org/10.1145/3188745.3188906

[51] Mohsen Ghaffari, Fabian Kuhn, and Hsin-Hao Su. 2017. Distributed MST and Routing in Almost Mixing Time. In *Proceedings 37th ACM Symposium on Principles of Distributed Computing (PODC).* 131–140.

[52] Mohsen Ghaffari and Jason Li. 2018. New Distributed Algorithms in Almost Mixing Time via Transformations from Parallel Algorithms. In *Proceedings 32nd International Symposium on Distributed Computing (DISC) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 121),* Ulrich Schmid and Josef Widder (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 31:1–31:16.

[53] Mohsen Ghaffari and Merav Parter. 2017. Near-optimal distributed DFS in planar graphs. In *31st International Symposium on Distributed Computing (DISC 2017).* Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[54] Apache Giraph. [n.d.]. http://giraph.apache.org.

[55] Oded Goldreich and Dana Ron. 1999. A Sublinear Bipartiteness Tester for Bounded Degree Graphs. *Combinatorica* 19, 3 (01 Mar 1999), 335–373.

[56] Joseph E. Gonzalez, Reynold S. Xin, Ankur Dave, Daniel Crankshaw, Michael J. Franklin, and Ion Stoica. 2014. GraphX: Graph Processing in a Distributed Dataflow Framework. In *Proc. 11th USENIX Conference on Operating Systems Design and Implementation (OSDI).* 599–613.

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

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

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

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

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

[62] David G. Harris. 2019. Distributed approximation algorithms for maximum matching in graphs and hypergraphs. In *Proc. 60th IEEE Symposium on Foundations of Computer Science (FOCS).* 700–724.

[63] Taisuke Izumi, François Le Gall, and Frédéric Magniez. 2020. Quantum Distributed Algorithm for Triangle Finding in the CONGEST Model. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 154),* Christophe Paul and Markus Bläser (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 23:1–23:13. https://doi.org/10.4230/LIPIcs.STACS.2020.23

[64] Mark Jerrum and Alistair Sinclair. 1989. Approximating the Permanent. *SIAM J. Comput.* 18, 6 (1989), 1149–1178.

[65] Ravi Kannan, Santosh Vempala, and Adrian Vetta. 2004. On clusterings: good, bad and spectral. *J. ACM* 51, 3 (May 2004), 497–515. https://doi.org/10.1145/990308.990313

[66] Ken-ichi Kawarabayashi, Seri Khoury, Aaron Schild, and Gregory Schwartzman. 2020. Improved Distributed Approximations for Maximum Independent Set. In *34th International Symposium on Distributed Computing (DISC),* Vol. 179. 35:1–35:16.

[67] Ken-Ichi Kawarabayashi and Mikkel Thorup. 2018. Deterministic Edge Connectivity in Near-Linear Time. *J. ACM* 66, 1, Article 4 (Dec. 2018), 4:1–4:50 pages.

[68] Philip Klein, Serge A Plotkin, and Satish Rao. 1993. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the 25th annual ACM symposium on Theory of computing (STOC).* 682–690.

[69] A. Kumar, C. Seshadhri, and A. Stolman. 2018. Finding forbidden minors in sublinear time: a $O(n^{1/2+o(1)})$-query one-sided tester for minor closed properties on bounded degree graphs. In *Proceedings 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS).* 509–520.

[70] François Le Gall and Masayuki Miyamoto. 2021. Lower Bounds for Induced Cycle Detection in Distributed Computing. In *Proceedings of the 32nd International Symposium on Algorithms and Computation (ISAAC).*

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

[72] Reut Levi, Moti Medina, and Dana Ron. 2021. Property testing of planarity in the CONGEST model. *Distributed Computing* 34, 1 (2021), 15–32.

[73] Nathan Linial and Michael Saks. 1993. Low diameter graph decompositions. *Combinatorica* 13, 4 (01 Dec 1993), 441–454.

[74] Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. 2015. Improved Distributed Approximate Matching. *J. ACM* 62, 5, Article 38 (2015), 17 pages.

[75] Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. 2010. Pregel: A System for Large-scale Graph Processing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD).* 135–146.

[76] Robert Ryan McCune, Tim Weninger, and Greg Madey. 2015. Thinking Like a Vertex: A Survey of Vertex-Centric Frameworks for Large-Scale Distributed Graph Processing. *ACM Comput. Surv.* 48, 2, Article 25 (Oct. 2015), 39 pages.

[77] G. L. Miller. 1986. Finding Small Simple Cycle Separators for 2-Connected Planar Graphs. *J. Comput. Syst. Sci.* 32, 3 (1986), 265–279.

[78] Danupon Nanongkai, Thatchaphol Saranurak, and Christian Wulff-Nilsen. 2017. Dynamic minimum spanning forest with subpolynomial worst-case update time. In *Proceedings of IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE, 950–961.

[79] Tim Nieberg. 2008. Local, Distributed Weighted Matching on General and Wireless Topologies. In *Proc. 5th Int'l Workshop on Foundations of Mobile Computing (DIALM-POMC) (DIALM-POMC '08).* 87–92.

[80] Xinghao Pan, Dimitris Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. 2015. Parallel Correlation Clustering on Big Graphs. In *Proc. 28th International Conference on Neural Information Processing Systems (NIPS)* (Montreal, Canada). 82–90.

[81] Prasad Raghavendra and David Steurer. 2010. Graph Expansion and the Unique Games Conjecture. In *Proceedings 42nd ACM Symposium on Theory of Computing (STOC).* 755–764.

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

[83] Václav Rozhoň and Mohsen Ghaffari. 2020. Polylogarithmic-Time Deterministic Network Decomposition and Distributed Derandomization. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC).* 350–363.

[84] Daniel A. Spielman and Shang-Hua Teng. 2004. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings 36th Annual ACM Symposium on Theory of Computing (STOC).* 81–90.

[85] Chaitanya Swamy. 2004. Correlation Clustering: Maximizing Agreements via Semidefinite Programming. In *Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (New Orleans, Louisiana). 526–527.

[86] Ondrej Sýkora and Imrich Vrťo. 1993. Edge separators for graphs of bounded genus with applications. *Theoretical Computer Science* 112, 2 (1993), 419–429. https://doi.org/10.1016/0304-3975(93)90031-N

[87] Andrew Thomason. 2001. The Extremal Function for Complete Minors. *Journal of Combinatorial Theory, Series B* 81, 2 (2001), 318–338. https://doi.org/10.1006/jctb.2000.2013

[88] Wojciech Wawrzyniak. 2014. A strengthened analysis of a local algorithm for the minimum dominating set problem in planar graphs. *Inform. Process. Lett.* 114, 3 (2014), 94–98.