Can We Break Symmetry with $o(m)$ Communication?*

Shreyas Pai
shreyas-pai@uiowa.edu
The University of Iowa
Iowa City, IA, USA

Gopal Pandurangan†
gopalpandurangan@gmail.com
University of Houston
Houston, TX, USA

Sriram V. Pemmaraju‡
sriram-pemmaraju@uiowa.edu
The University of Iowa
Iowa City, IA, USA

Peter Robinson§
peter.robinson@cityu.edu.hk
City University of Hong Kong
Hong Kong SAR, China

ABSTRACT

We study the communication cost (or *message complexity*) of fundamental distributed symmetry breaking problems, namely, coloring and MIS. While significant progress has been made in understanding and improving the running time of such problems, much less is known about the message complexity of these problems. In fact, all known algorithms need at least $\Omega(m)$ communication for these problems, where m is the number of edges in the graph. We address the following question in this paper: *can we solve problems such as coloring and MIS using sublinear, i.e., $o(m)$ communication, and if so under what conditions?*

In a classical result, Awerbuch, Goldreich, Peleg, and Vainish [JACM 1990] showed that fundamental *global* problems such as broadcast and spanning tree construction require at least $\Omega(m)$ messages in the KT-1 CONGEST model (i.e., CONGEST model in which nodes have initial knowledge of the neighbors' ID's) when algorithms are restricted to be comparison-based (i.e., algorithms in which node ID's can only be compared). Thirty five years after this result, King, Kutten, and Thorup [PODC 2015] showed that one can solve the above problems using $\tilde{O}(n)$ messages (n is the number of nodes in the graph) in $\tilde{O}(n)$ rounds in the KT-1 CONGEST model if *non-comparison-based* algorithms are permitted. An important implication of this result is that one can use the synchronous nature of the KT-1 CONGEST model, using silence to convey information, and solve *any* graph problem using non-comparison-based algorithms with $\tilde{O}(n)$ messages, but this takes an *exponential* number of rounds. In the asynchronous model, even this is not possible.

In contrast, much less is known about the message complexity of *local* symmetry breaking problems such as coloring and MIS. Our paper fills this gap by presenting the following results.

Lower bounds: In the KT-1 CONGEST model, we show that any comparison-based algorithm, even a randomized Monte-Carlo algorithm with constant success probability, requires $\Omega(n^2)$ messages in the worst case to solve either $(\Delta+1)$-coloring or MIS, *regardless* of the number of rounds. We also show that $\Omega(n)$ is a lower bound on the number of messages for any $(\Delta+1)$-coloring or MIS algorithm, even non-comparison-based, and even with nodes having initial knowledge of up to a constant radius.

Upper bounds: In the KT-1 CONGEST model, we present the following randomized non-comparison-based algorithms for coloring that, with high probability, use $o(m)$ messages and run in polynomially many rounds.

(a) A $(\Delta+1)$-coloring algorithm that uses $\tilde{O}(n^{1.5})$ messages, while running in $\tilde{O}(D+\sqrt{n})$ rounds, where D is the graph diameter. Our result also implies an *asynchronous* algorithm for $(\Delta+1)$-coloring with the same message bound but running in $\tilde{O}(n)$ rounds.

(b) For any constant $\varepsilon > 0$, a $(1+\varepsilon)\Delta$-coloring algorithm that uses $\tilde{O}(n/\varepsilon^2)$ messages, while running in $\tilde{O}(n)$ rounds.

If we increase our input knowledge slightly to radius 2, i.e., in the KT-2 CONGEST model, we obtain:

(c) A randomized comparison-based MIS algorithm that uses $\tilde{O}(n^{1.5})$ messages. while running in $\tilde{O}(\sqrt{n})$ rounds.

While our lower bound results can be viewed as counterparts to the classical result of Awerbuch, Goldreich, Peleg, and Vainish [JACM 90], but for local problems, our algorithms are the first-known algorithms for coloring and MIS that take $o(m)$ messages and run in polynomially many rounds.

CCS CONCEPTS

• **Theory of computation** → **Models of computation**; **Distributed algorithms**; *Graph algorithms analysis*; • **Mathematics of computing** → *Discrete mathematics.*

KEYWORDS

Distributed Graph Algorithms; Congest Model; Message Complexity; Symmetry Breaking; MIS; Coloring

*A full version of this paper is available on arxiv [29]: https://arxiv.org/abs/2105.08917

†Gopal Pandurangan was supported, in part, by NSF grants IIS-1633720, CCF-1717075, CCF-1540512, and BSF grant 2016419.

‡Sriram V. Pemmaraju was supported, in part, by NSF grant IIS-1955939.

§Peter Robinson was partially supported by a grant from the Research Grants Council (HKSAR) [Project No. CityU 11213620], as well as by a grant from the City University of Hong Kong [Project No. 7200639/CS].

PODC '21, July 26–30, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8548-0/21/07.
https://doi.org/10.1145/3465084.3467909

ACM Reference Format:
Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, and Peter Robinson. 2021. Can We Break Symmetry with $o(m)$ Communication?. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC '21), July 26–30, 2021, Virtual Event, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3465084.3467909

1 INTRODUCTION

There has been significant interest over the last decade in obtaining communication-efficient algorithms for fundamental problems in distributed computing. In the CONGEST model, which is a message-passing model with small-sized messages (typically $O(\log n)$-sized, where n is the number of nodes in the network), communication cost is usually measured by the number of messages. In the so-called *clean network* model, a.k.a. the KT-0 (**K**nowledge **T**ill radius 0) model, where nodes have intial knowledge of only themselves and don't even know the ID's of neighbors, Kutten et al. [20] showed that $\Omega(m)$ (m is the number of edges in the network) is a lower bound for the message complexity for fundamental *global* problems such as leader election, broadcast, spanning tree, and mimimum spanning tree (MST) construction. This lower bound applies even for randomized Monte Carlo algorithms. For all these problems, there exist algorithms that (essentially) match this message lower bound; in fact, these also have optimal time complexity (of D, the network diameter) in the CONGEST model (see e.g., [8, 20, 31]).

The clean network model does not capture many real world networks such as the Internet and peer-to-peer networks where nodes typically have knowledge of identities (i.e., IP addresses) of other nodes. Also, there has been a lot of recent interest in "all-to-all" communication models such as the congested clique [22], Massively Parallel Computing (MPC) [14], and k-machine model [17], where each machine is assumed to have knowledge of ID's of all other machines. Motivated by these applications and models, there has been a lot of recent interest in studying message-efficient algorithms under the so-called KT-1 version of the CONGEST model, where nodes have initial knowledge of the IDs of their neighbors, but no other knowledge of their neighbors. An immediate question that arises is whether the $\Omega(m)$ message lower bound also holds in the KT-1 model; or whether sublinear, i.e., $o(m)$ message complexity is possible.

The above question was partially answered in a seminal paper by Awerbuch et al. [1] who initiated the study of trade-offs between the message complexity and *initial knowledge* of distributed algorithms that solve *global* problems, such as broadcast and spanning tree construction. For any integer $\rho > 0$, in the KT-ρ version of the CONGEST model (in short, KT-ρ CONGEST), each node v is provided initial knowledge of (i) the IDs of all nodes at distance at most ρ from v and (ii) the neighborhood of every node at distance at most $\rho - 1$ from v. The bounds in this paper [1] are for *comparison-based algorithms*, i.e., algorithms in which local computations on IDs are restricted to comparisons only. This means that operations on IDs such as those used in the Cole-Vishkin coloring algorithm [6] or applying random hash functions to IDs are disallowed. Comparison-based algorithms are quite natural and indeed, most distributed algorithms (with few notable exceptions such as Cole-Vishkin [6] and hash-functions based algorithms of King et al [16]) are comparison-based. For the KT-1 CONGEST model the authors show that $\Omega(m)$ messages are needed for any *comparison-based* algorithm (even randomized) that solves broadcast. Furthermore, in the KT-ρ CONGEST model, $\Omega\left(\min\left\{m, n^{\frac{1+\Theta(1)}{\rho}}\right\}\right)$ messages are needed for any comparison-based algorithm that solves broadcast. The paper also shows matching upper bounds for comparison-based algorithms for broadcast. These lower bounds also hold for non-comparison based algorithms, where the size of the IDs is very large and grows independently with respect to message size, time, and randomness. This paper left open the possibility of circumventing the lower bound if one uses non-comparison based algorithms on more natural ID spaces typically used in distributed algorithms (as assumed in the current paper), where IDs are drawn from a polynomial-sized ID space.

Nearly 35 years later, the above question was settled by King et al. [16] who showed that the Awerbuch et al. lower bounds "break" if the assumption that the algorithms be comparison-based is dropped and one uses ID space that is of polynomial size.[1] Specifically, it is shown in [16] that the Spanning Tree (and hence broadcast) and Minimum Spanning Tree (MST) problem can be solved using $\tilde{O}(n)$ messages in KT-1 CONGEST model.[2] In followup papers, it is shown that these problems can be solved with $o(m)$ messages, but with a higher message bound of $\tilde{O}(n^{1.5})$, even in the *asynchronous* CONGEST KT-1 model [24, 25]. Using the King et al. [16] result, it is possible to solve *any* graph problem (including symmetry breaking problems) using randomized *non-comparison* based algorithms in $\tilde{O}(n)$ messages. However, this takes an *exponential* number of rounds. This is done by building a spanning tree using the algorithm of King et al. and then using time-encoding to convey the entire topology to the root of the spanning tree. The root then locally computes the result and disseminates it to the entire network, again using time-encoding (e.g., see [33] for details). Time-encoding uses silence to convey information and takes at least exponential (in m) rounds. Note that this works only in *synchronous* setting and not in the asynchronous model. Hence, designing algorithms that use $\tilde{O}(n)$ (or even $o(m)$) messages for other graph problems, including local symmetry breaking problems, regardless of the number of rounds, in the asynchronous CONGEST KT-1 model is open. Designing algorithms that use small number of messages is also relevant from a practical point of view, especially in the context of designing *energy-efficient* algorithms for resource-constrained networks such as ad hoc wireless and sensor networks, where number of messages exchanged is correlated to the energy spent by the algorithm.

Motivated by the above results, we initiate a similar study, but for fundamental *local symmetry breaking* problems, such as $(\Delta + 1)$-coloring and Maximal Independent Set (MIS). These problems have been studied extensively for over four decades. Significant progress has been made in understanding and improving the *running time* (round complexity) of these problems (see e.g., [2–4, 9, 10, 13, 35] and the references therein); however, much less is known with respect to message complexity. For $(\Delta + 1)$-coloring and MIS, to the best of our knowledge, all known distributed algorithms use at least

[1] This can be relaxed to allow even exponential-sized ID space: by using fingerprinting technique [15, 16], with high probability, one can map n IDs in exponential ID space to distinct IDs in polynomial ID space.

[2] We use $\tilde{O}(f(n))$ as short for $O(f(n) \cdot \text{poly}\log n)$ and $\tilde{\Omega}(g(n))$ as short for $\Omega(g(n)/(\text{poly}\log n))$.

		$(\Delta+1)$-coloring		MIS	
		C	NC	C	NC
KT-1	Lower Bound	$\Omega(m)^*$	$\Omega(n)$	$\Omega(m)^*$	$\Omega(n)$
	Upper Bound	$\tilde{O}(m)^*$	$\tilde{O}(n^{1.5})$	$\tilde{O}(m)^*$	$\tilde{O}(m)$
KT-2	Lower Bound	$\Omega(n)$	$\Omega(n)$	$\Omega(n)$	$\Omega(n)$
	Upper Bound	$\tilde{O}(m)$	$\tilde{O}(n^{1.5})$	$\tilde{O}(n^{1.5})$	$\tilde{O}(n^{1.5})$

Figure 1: A summary of lower and upper bounds results in this paper. The notation "C" and "NC" stand for comparison-based and non-comparison-based respectively. The comparison based upper bounds of $\tilde{O}(m)$ are not from this paper; but are immediately implied by a number of well-known MIS and coloring algorithms (e.g., [23, 39]). The lower bounds in the KT-2 column hold also in KT-ρ for any constant $\rho \geq 1$. The cells marked with * are the ones where the lower bound and upper bound are tight, i.e., within $O(\mathbf{poly}(\log n))$ factor. Closing the gaps in the other regimes are interesting open problems.

$\Omega(m)$ messages. The overarching question we address in this paper is whether these problems can be solved using $o(m)$ messages in the Congest model and if so, under what conditions.

Our paper presents both negative and positive answers for the above question and shows results in three general directions. First, we show that even though the *round complexity* of local symmetry breaking problems is provably much smaller than the round complexity of global problems, comparison-based algorithms for local symmetry breaking problems require *as many messages* as they do for global problems in the KT-1 Congest model. Second, we show that if we drop the requirement that our algorithms be comparison-based only, then it is possible to design algorithms for local symmetry breaking problems in the KT-1 Congest model that use far fewer messages. Third, as we increase ρ, the radius of initial knowledge, to just two, i.e., in the KT-2 Congest model, it is possible to design algorithms for local symmetry breaking problems that use even fewer messages. The specific results that illustrate these three directions are presented in the next subsection.

1.1 Main Results

We present new lower and upper bounds on the message complexity for two fundamental symmetry breaking problems, namely, coloring and MIS. See Figure 1 for a summary.

Lower bounds: In the KT-1 Congest model, we show that any comparison-based algorithm, even a randomized Monte Carlo algorithm with constant success probability, requires $\Omega(n^2)$ messages in the worst case to solve either $(\Delta+1)$-coloring or MIS, regardless of the number of rounds. Our result can be considered as a counterpart to the classical result of Awerbuch et al. [1], but for local problems. We also show that in the KT-ρ Congest model, for any constant $\rho \geq 1$, $(\Delta+1)$-coloring and MIS require $\Omega(n)$ messages even for non-comparison-based and Monte Carlo randomized algorithms with constant success probability.

Upper bounds: In the KT-1 Congest model, we present the following randomized non-comparison-based algorithms for coloring that with high probability[3] (w.h.p.) use $o(m)$ messages and run in polynomially many rounds.

- **(a)** A $(\Delta+1)$-coloring algorithm that uses $\tilde{O}(n^{1.5})$ messages, while running in $\tilde{O}(D+\sqrt{n})$ rounds, where D is the graph diameter. Our result also implies an *asynchronous* algorithm for $(\Delta+1)$-coloring with the same message bound but running in $\tilde{O}(n)$ rounds.
- **(b)** A $(1+\varepsilon)\Delta$-coloring algorithm that uses $\tilde{O}(n/\varepsilon^2)$ messages, while running in $\tilde{O}(n)$ rounds.

If we increase our input knowledge slightly, i.e., we work in the KT-2 Congest model, where nodes have initial knowledge of their *two hop-neighborhood*, then we get the following additional and stronger result.

- **(c)** A *comparison-based* algorithm for MIS that uses $\tilde{O}(n^{1.5})$ messages, while running in $\tilde{O}(\sqrt{n})$ rounds.

Our algorithms for coloring and MIS are the first-known algorithms that take $o(m)$ messages and running in polynomial number of rounds.

1.2 Other Related Work

Several recent papers (see e.g., [11, 12, 24, 25] have studied message-efficient algorithms for *global* problems, namely, construction of spanning tree, minimum spanning tree, broadcasting and leader election, in the KT-1 Congest model inspired by the work of King et al. [16]. We note that all these are non-comparison-based algorithms. We use these prior algorithms for our non-comparison-based algorithms in the KT-1 and KT-2 models. In a recent paper, Robinson [33] shows non-trivial lower bounds on the message complexity of constructing graph spanners in the Congest KT-1 model.

In contrast to global problems, much less is known about obtaining sublinear, i.e., $o(m)$ algorithms for local problems, such as MIS and coloring. Pai et al. [28] showed that MIS has a fundamental lower bound of $\Omega(n^2)$ messages in the Congest KT-0 model (even for randomized algorithms). However, this result does not extend to the KT-1 model. In contrast, they also showed that the 2-ruling set problem (note that MIS is 1-ruling set) can be solved using $\tilde{O}(n)$ messages in the KT-0 model in polynomial time. To the best of our knowledge, we are not aware of other results on the message complexity (in particular, lower bounds and sublinear upper bounds) on fundamental symmetry breaking problems, vis-a-vis the initial input knowledge.

Recently, [34] initiated the study of volume complexity of distributed graph problems. The volume complexity measures the size of the network (i.e., number of nodes) that a node must learn about in order to compute its output. This measure of complexity of distributed algorithms does not seem to have any (non-trivial) connections to the message complexity (which instead quantifies the total number of used edges), and vice versa.

[3] This refers to probability at least $1-n^{-c}$ for constant $c \geq 1$.

1.3 Technical Contributions

- **Lower bounds:** To obtain our KT-1 Congest lower bounds for comparison-based algorithms for $(\Delta + 1)$-coloring and MIS, we start with the machinery introduced by Awerbuch et al. [1] for proving their KT-1 Congest lower bounds for comparison-based algorithms for broadcast. At the core of their approach is an indistinguishability argument that uses edge crossings. Edge crossings have been used numerous times to prove a variety of distributed computing lower bounds (see [19, 20, 28, 30, 32] for some examples). However, in the KT-1 Congest model, indistinguishability arguments via edge crossing are more challenging because when an edge incident on a node is crossed, the node is exposed to a new ID due to KT-1. For symmetry breaking problems, there is a further challenge due to the fact that multiple outputs are possible and the indistinguishability argument needs to work for all outputs. Finally, since we want to show our lower bounds even for Monte Carlo algorithms with constant success probability, we require our indistinguishability arguments to apply to a large fraction of edge crossings (so as to be able to apply Yao's lemma [26, 38]). The lower bound graph family and ID assignment we design, overcomes all of these challenges. We use a unified construction that works for both $(\Delta + 1)$-coloring and MIS and we expect this construction to work for other symmetry breaking problems such as maximal matching and edge coloring.
- **Upper bounds:** Our upper bounds are largely obtained by exploiting the fact that shared (or public) randomness combined with KT-1 is a powerful way of eliminating the need to communicate over a large number of edges.[4] Specifically, we start with the recent coloring algorithm of Chang et al. [5] that works efficiently in the MPC model. Roughly speaking, this algorithm starts with a probabilistic step; by randomly partitioning the nodes and the color palette. Then, after this probabilistic step, a large number of edges become inactive for the rest of algorithm. This property is crucial to ensuring that the algorithm is efficient in the MPC model. After the probabilistic step, nodes exchange their state with neighbors in so that every node can determine which of its incident edges to render inactive. This state exchange is cheap in the MPC model, but is costly with respect to messages in the Congest model. We show how to simulate this coloring algorithm in the Congest model without the costly exchange of state. Instead we use shared randomness with limited dependence combined with KT-1.

1.4 Preliminaries

1.4.1 KT-ρ Congest model. We work in the synchronous, message-passing model of distributed computing, known as the Congest model. The input is a graph $G = (V, E)$, $n = |V|$, which also serves as the communication network. Nodes in the graph are processors with unique IDs from a space whose size is polynomial in n. In each round, each node can send an $O(\log n)$-bit message to each of its neighbors. Since we are interested in message complexity, the initial knowledge of the nodes is important. For any integer $\rho > 0$, in the KT-ρ Congest model each node v is provided initial knowledge of (i) the IDs of all nodes at distance at most ρ from v and (ii) the neighborhood of every vertex at distance at most $\rho - 1$ from v.

1.4.2 Comparison-based Algorithms. Often, the outcome of a distributed algorithm does not depend on specific values of node IDs, but may depend on the relative ordering of IDs. For example, node IDs of endpoints may be used to break ties between edges of the same weight vying to join a minimum spanning tree. In this case, only the ordering of the IDs matters, not their specific values. Since this type of behavior is characteristic of many distributed algorithms, Awerbuch et al. [1] formally define these as *comparison-based* algorithms. In comparison-based algorithms, the algorithm executed by each node contains two types of variables: ID-type variables and *ordinary* variables. In the KT-ρ Congest model, the ID-type variables at a node v will store the IDs of all nodes within ρ hops of v. Nodes can send ID-type variables in messages, but since messages in the Congest model are restricted to be $O(\log n)$ bits long, each message can contain only a constant number of ID-type variables. The local computations at any node may involve operations of the following two forms only:

(1) Comparing two ID-type variables I_i, I_j and storing the result of the comparison in an ordinary variable.
(2) Performing an arbitrary computation on ordinary variables and storing the result in another ordinary variable.

Note that if randomization is allowed, then nodes can choose to ignore the node IDs and choose a new set of ($O(\log n)$-length) IDs and do arbitrary computations with them. These are still comparison-based algorithms.[5]

1.4.3 Efficient Broadcasting in the KT-1 Congest model. As explained earlier, shared randomness along with initial knowledge, plays a key role in making our algorithms message-efficient. We use a graph structure called a *danner* introduced by Gmyr and Pandurangan [12] to share random bits among the nodes in the graph in a message-efficient fashion. Their specific result is stated in the following theorem.

Theorem 1.1 (Gmyr and Pandurangan [12]). *Given an n-vertex, m-edge, diameter D, graph $G = (V, E)$ and a parameter $\delta \in [0, 1]$, there is a randomized algorithm in the KT-1 Congest model, that constructs a spanning subgraph (i.e., a danner) H of G such that H has $\tilde{O}(\min\{m, n^{1+\delta}\})$ edges and diameter $\tilde{O}(D + n^{1-\delta})$ with high probability. This construction uses $\tilde{O}(\min\{m, n^{1+\delta}\})$ messages and runs in $\tilde{O}(n^{1-\delta})$ rounds with high probability.*

We need the following corollary of this theorem.

Corollary 1.2. *Given an n-vertex, m-edge, diameter D graph $G = (V, E)$ and a parameter $\delta \in [0, 1]$, there exists a randomized algorithm to solve the leader election and broadcast problems in the synchronous KT-1 Congest model using $\tilde{O}(\min\{m, n^{1+\delta}\})$ messages and in $\tilde{O}(D + n^{1-\delta})$ rounds with high probability.*

[4] Note that we do not a priori assume shared randomness, but only private randomness (as is usual), but use the danner structure (Section 1.4.3) to share privately generated random bits throughput the graph.

[5] However, note that such randomly chosen node IDs are unknown to neighbors and if the algorithm uses only those IDs then this becomes effectively the KT0 model where bounds are already known [1, 28].

We use this corollary to share $O(\text{poly}\log n)$ random bits in a message-efficient manner by first electing a leader and then having the leader locally generate the random bits and broadcasting them. The message and time complexities for this operation are given by the above corollary. We note that the above danner bounds hold in KT-1 Congest model, which is *synchronous*.

1.4.4 Tail inequalities and hash functions with limited independence. To obtain message-efficient algorithms in the KT-1 model, we make use of hash functions with limited independence. These hash functions use c-wise independence and hence we use the following tail inequalities and properties of such hash functions.
The following tail inequalities are from [36].

Lemma 1.3. *Let $c \geq 4$ be an even integer. Suppose $Z_1, Z_2, \ldots, Z_t$ are c-wise independent random variables taking values in $[0,1]$. Let $Z = \sum_{i=1}^{t} Z_i$ and $\mu = \mathbb{E}[Z]$, and let $\lambda > 0$. Then,*

$$Pr[|Z - \mu| \geq \lambda] \leq 2\left(\frac{ct}{\lambda^2}\right)^{c/2}.$$

Lemma 1.4. *Suppose that X is the summation of n, c-wise independent 0-1 random variables, each with mean p. Let μ satisfy $\mu \geq \mathbb{E}[X] = np$. Then,*

$$Pr[X \geq (1+\delta)\mu] \leq exp(-\min\{c, \delta^2\mu\}).$$

The following is Definition 7 in [7].

Definition 1.5. For $N, L, c \in \mathbb{N}$, such that $c \leq N$, a family of functions $\mathcal{H} = \{h : [N] \to [L]\}$ is c-wise independent if for all distinct $x_1, x_2, \ldots, x_c \in [N]$, the random variables $h(x_1), h(x_2), \ldots, h(x_c)$ are independent and uniformly distributed in $[L]$ when h is chosen uniformly at random from H.

The following lemma appears as Corollary 3.34 in [37].

Lemma 1.6. *For every a, b, c, there is a family of c-wise independent hash functions $\mathcal{H} = \{h : \{0,1\}^a \to \{0,1\}^b\}$ such that choosing a random function from $\mathcal{H}$ takes $c \cdot \max\{a, b\}$ random bits, and evaluating a function from $\mathcal{H}$ takes poly(a, b, c) computation.*

2 MESSAGE COMPLEXITY LOWER BOUNDS

2.1 Technical Preliminaries

We now state key definitions and notation from Awerbuch et al. [1] which we will use in our proofs of the $\Omega(m)$ message lower bounds for $(\Delta+1)$-coloring and MIS, for comparison-based algorithms, in the KT-1 Congest model.

Definition 2.1 (Executions). We denote the execution of a Congest algorithm (or protocol) $\mathcal{A}$ on a graph $G(V, E)$ with an ID-assignment ϕ by $EX(\mathcal{A}, G, \phi)$. This execution contains (i) the messages sent and received by the nodes in each round and (ii) a snapshot of the local state of each node in each round. We denote the state of a node v in the beginning of round i of the execution $EX(\mathcal{A}, G, \phi)$ by $L_i(EX, v)$.

The *decoded representation* of an execution is obtained by replacing each occurrence of an ID value $\phi(v)$ by v in the execution. This decoded representation allows us to define a similarity of executions. We denote the decoded representations of all messages sent during round i of an execution $EX(\mathcal{A}, G, \phi)$ as $h_i(EX(\mathcal{A}, G, \phi))$.

Definition 2.2 (Similar executions). Two executions of a Congest algorithm $\mathcal{A}$ on graphs $G(V, E)$ and $G'(V, E')$ with ID-assignments ϕ and ϕ' are *similar* if they have the same decoded representation. Likewise, we say that two messages are *similar* if their decoded representations are the same.

A crucial element of our lower bound proof consists of taking two graphs $G(V, E)$ and $G'(V', E')$, where G' is obtained from G by "crossing" a pair of edges in G, and showing that the executions of any comparison-based algorithm, on G and G' are similar. Showing similarity of executions requires that the "crossing" of edges remains, in a certain sense, hidden from the algorithm. Below, we define what it means for an algorithm to *utilize* an edge. Later on we will be able to show that if the edges being "crossed" are not utilized by the algorithm, then the edge "crossing" is hidden from the algorithm. One way an algorithm utilizes an edge is by sending a message across it. But, this notion of utilization does not suffice in the KT-1 model. We need the stronger notion, defined below.

Definition 2.3 (Utilized Edge). An edge $e = \{u, v\}$ is utilized if any one of the following happens during the course of the algorithm: (i) a message is sent along e, (ii) the node u sends or receives `ID` $\phi(v)$, or (iii) the node v sends or receives `ID` $\phi(u)$.

By definition, the number of utilized edges is an upper bound on the number of edges along which a message sent. Using a charging argument, Awerbuch et al. [1] show that the number of utilized edges is also upper bounded by a constant times the number of edges along which a message sent. We restate their claim here.

Lemma 2.4 (Lemma 3.4 of [1]). *Let m_u denote the number of utilized edges in an execution $EX(\mathcal{A}, G, \phi)$. Then the message complexity of the execution is $\Omega(m_u)$.*

2.2 Lower Bound Graph Construction and ID Assignments

We now describe the construction of lower bound graphs that we use for our $\Omega(n^2)$ message complexity lower bounds. The same construction works for both the $(\Delta+1)$-coloring and MIS lower bounds. Recall that these bounds are for comparison-based algorithms in the KT-1 Congest model.

We start with a graph $G(X, Y, Z, E)$ such that $|X| = |Y| = |Z| = t$ and the subgraphs of G induced by $X \cup Y$ and $Y \cup Z$ are both isomorphic to the complete bipartite graph $K_{t,t}$. Thus, $|E| = 2t^2$. We then add a copy $G'(X', Y', Z', E')$ of G and consider the graph $G \cup G'$. We call this the *base graph*. Let $V = X \cup Y \cup Z$ and $V' = X' \cup Y' \cup Z'$. For each $v \in V$, the corresponding copy in V' is named v'. Let $n = |V \cup V'|$. Thus $t = n/6$. From the base graph $G \cup G'$, we obtain a *crossed graph* as follows. For a vertex $y \in Y$, *cross* an edge $e = \{y, z\}$ in G, where $z \in Z$ with the edge $e' = \{x', y'\}$ in G' where $x' \in X'$ to obtain the graph $G_{e,e'}$. When we cross the edge $e = \{y, z\} \in E$ with $e' = \{x', y'\} \in E'$, the resulting *crossed graph* $G_{e,e'}$ has vertex set $V \cup V'$ and edge set $(E \cup E' \setminus \{e, e'\}) \cup \{\{y, y'\}, \{x', z\}\}$. The base graph $G \cup G'$ and the crossed graph $G_{e,e'}$ for edges $e \in E, e' \in E'$ are illustrated in Figure 2.

We now define appropriate ID-assignments for the base graph and the crossed graph. Let S be an arbitrary totally ordered set such that $|S| = 40t$, and let $\overline{S}$ be the sorted list of elements in S in ascending order. We will assign distinct elements in S as `ID`'s to

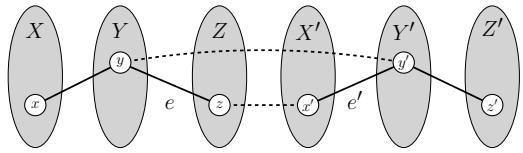

Figure 2: This figure shows the base graph $G \cup G'$ and the crossed graph $G_{e,e'}$, described in Section 2.2

the base graph and the crossed graph. We use a short-hand and say that the ID of a vertex v is $i \in [0, 40t)$, when we mean that the ID of v is $\overline{S}[i]$. Note that since $\overline{S}$ is sorted in ascending order, the relative ordering of the indices is the same as that of the corresponding ID's in $\overline{S}$.

Let $\phi : V \to [0, 40t)$ be an ID assignment such that $\phi(v)$ is even for all $v \in V$ and additionally $\phi(v) \in [0, 2t)$ if $v \in X$, $\phi(v) \in [10t, 12t)$ if $v \in Y$, and $\phi(v) \in [20t, 22t)$ if $v \in Z$. For a vertex $y \in Y$ and pair of incident edges $e = \{y, z\}$ and $e' = \{x', y'\}$, we define a "shifted" ID assignment $\phi'_{e,e'}$ for the vertex set V' of G'. We motivate this "shifted" assignment and define it precisely further below. But for now, assuming $\phi'_{e,e'}$ is defined, we define the ID assignment $\psi_{e,e'} : V \cup V' \to [0, 40t)$ as just the union of ϕ and $\phi'_{e,e'}$, i.e., $\psi_{e,e'}(v) = \phi(v)$ for all $v \in V$ and $\psi_{e,e'}(v') = \phi'_{e,e'}(v')$ for all $v' \in V'$. Our first goal in this subsection is to show that these two executions

$$EX = EX(\mathcal{A}, G \cup G', \psi_{e,e'}); \qquad EX_{e,e'} = EX(\mathcal{A}, G_{e,e'}, \psi_{e,e'})$$

on the base graph $G \cup G'$ and the crossed graph $G_{e,e'}$ are similar for any comparison-based algorithm $\mathcal{A}$.

For the executions EX and $EX_{e,e'}$ to be similar, it must be the case that the crossing of edges e and e' is hidden from algorithm $\mathcal{A}$. To achieve this, the ID assignment $\phi'_{e,e'}$ of V' must be carefully chosen. For example, vertex z has neighbor y in $G \cup G'$, but has neighbor x' in $G_{e,e'}$ (see Figure 2). In the KT-1 model, z's initial local knowledge consists of vertex y in $G \cup G'$ and vertex x' in $G_{e,e'}$. Therefore, for $\mathcal{A}$ to not distinguish between these two situations, it must be the case that the ID of x' is "adjacent" to the ID of y. To achieve this, without disrupting other constraints on the relative order of ID's, we start by assigning vertices in X' the ID's of their corresponding vertices in X and then "shift" these by $(\phi(y) - \phi(x)) + 1$. As a result, vertex x' ends up with ID $\phi(y) + 1$. A similar "shift" is performed to obtain the ID's of vertex set Y', though this time the "shift" is by the amount $(\phi(z) - \phi(y)) + 1$ because we want vertex y' to be "adjacent" to vertex z. The "shift" for vertex set Z' just needs to be so that the ID assignment is disjoint, We now define the ID assignment $\phi'_{e,e'} : V' \to [0, 40t)$ as

$$\phi'_{e,e'}(v') = \begin{cases} \phi(v) + (\phi(y) - \phi(x)) + 1, & \text{if } v' \in X' \\ \phi(v) + (\phi(z) - \phi(y)) + 1, & \text{if } v' \in Y' \\ \phi(v) + 10t + 1, & \text{if } v' \in Z' \end{cases} \tag{1}$$

Note that the IDs of all vertices in each of the parts, X', Y', and Z', are "shifted" by the same amount, though IDs in different parts may be "shifted" by different amounts.

The following observations about $\phi'_{e,e'}$ are easy to verify.

(i) The ranges of ϕ and $\phi'_{e,e'}$ are disjoint.

(ii) Moreover, $\phi'_{e,e'}(v) \in [8t + 1, 14t + 1]$ if $v \in X'$, $\phi'_{e,e'}(v) \in [18t + 1, 24t + 1]$ if $v \in Y'$, and $\phi'_{e,e'}(v) \in [30t + 1, 32t + 1]$ if $v \in Z'$.

(iii) For any $u, v \in V$, $u \neq v$, $\phi(u) < \phi(v)$ iff $\phi'_{e,e'}(u') < \phi'_{e,e'}(v')$.

Item (iii) is simply saying that the ID ordering on V' induced by $\phi'_{e,e'}$ is the same as the ID ordering induced by ϕ with respect to the corresponding vertices in V. This follows from the fact that the ID's of vertices in X' are obtained by shifting the ID's of vertices in X by the same amount, thus preserving the relative ordering of ID's in X and X'. Similarly, for vertex sets Y' and Z'. Furthermore, even though the ID's of different sets, X', Y', and Z' are obtained by "shifting" by different amounts, the "shifting" also ensures that ID's in X' remain less than ID's in Y', which in turn remain less than ID's in Z'.

To prove that EX and $EX_{e,e'}$ are similar, we need two intermediate ID assignments for the set $V \cup V'$. Recall that edge $e = \{y, z\}$ and edge $e' = \{x', y'\}$.

(i) Define $\psi_{e,e',x}$ to be the ID assignment $\psi_{e,e'}$ except for interchanging the values of x' and y (i.e. $\psi_{e,e',x}(y) = \phi'_{e,e'}(x')$ and $\psi_{e,e',x}(x') = \phi(y)$).

(ii) Define $\psi_{e,e',z}$ analogously as $\psi_{e,e'}$ except for interchanging the values of y' and z (i.e. $\psi_{e,e',z}(z) = \phi'_{e,e'}(y')$ and $\psi_{e,e',z}(y') = \phi(z)$).

Using these ID assignments, we define two intermediate executions on the base graph $G \cup G'$.

$$EX_{e,e',x} = EX(\mathcal{A}, G \cup G', \psi_{e,e',x});$$
$$EX_{e,e',z} = EX(\mathcal{A}, G \cup G', \psi_{e,e',z})$$

The following lemma, which shows that the executions EX, $EX_{e,e',x}$, and $EX_{e,e',y}$ are similar, critically uses the fact that the ID assignment $\psi_{e,e'}$ shifts the ID's of vertices in $X' \cup Y' \cup Z'$ so that the ID of x' becomes "adjacent" to the ID of y and the ID of y' becomes "adjacent" to the ID of z.

Lemma 2.5. *For any $x \in X$, $y \in Y$, $z \in Z$ and edges $e = \{y, z\}$ and $e' = \{x', y'\}$, the executions EX, $EX_{e,e',x}$ and $EX_{e,e',z}$ are similar.*

Proof. All three executions have the same input graph $G \cup G'$. The execution pair EX and $EX_{e,e',x}$ have the same ID assignment except for the vertices x' and y, which have their ID's swapped. Note that by definition of $\psi'_{e,e'}$ and $\phi'_{e,e'}$ in (1), we have

$$\psi_{e,e'}(x') = \phi'_{e,e'}(x') = \phi(x) + (\phi(y) - \phi(x)) + 1 = \phi(y) + 1.$$

Furthermore, $\psi_{e,e'}(y) = \phi(y)$. Therefore, when we swap the ID's of x' and y in $\psi_{e,e'}$ to obtain $\psi_{e,e',x}$, there is no change in the relative ordering of ID's and therefore the executions EX and $EX_{e,e',x}$ are similar.

A similar argument holds for the execution pair EX and $EX_{e,e',z}$. By the definition of $\psi'_{e,e'}$ and $\phi'_{e,e'}$ in (1), we have

$$\psi_{e,e'}(y') = \phi'_{e,e'}(y') = \phi(y) + (\phi(z) - \phi(y)) + 1 = \phi(z) + 1$$

and $\psi_{e,e'}(z) = \phi(z)$. Thus the relative ordering of ID's in $\psi_{e,e'}$ and $\psi_{e,e',z}$ is the same and therefore the executions EX and $EX_{e,e',x}$ are similar.

The lemma follows because similarity of executions is transitive. □

We can derive the final tool we need by directly appealing to a lemma in Awerbuch et al. [1]. Informally, the lemma shows that if edges $e = \{y, z\}$ and $\{x', y'\}$ are not utilized in the execution EX of algorithm $\mathcal{A}$, then executions EX and $EX_{e,e'}$ are similar. The main obstacle is that the initial knowledge vertices x', y, y', z is different in EX and $EX_{e,e'}$ so a direct inductive proof like in Lemma 2.5 does not work. But we can use the intermediate executions of Lemma 2.5 to show the similarity for these four vertices. For all other vertices, we can do a direct inductive argument.

Lemma 2.6 (Restatement of Lemma 3.8 of [1]). *Let $x \in X, y \in Y$, and $z \in Z$ be arbitrary vertices and let $e = \{y, z\}$ and $e' = \{x', y'\}$. Suppose that during the first r rounds of the execution EX both e and e' are not utilized. Then the following hold for every round $1 \leq i \leq r$ of the executions EX, $EX_{e,e',x}$, $EX_{e,e',z}$ and $EX_{e,e'}$:*

1. *The states of the nodes in the beginning of the round, i.e. $L_i(\cdot, \cdot)$ satisfy:*
   (a) *For every processor $w \in V \setminus \{y, z, y', x'\}$, $L_i(EX_{e,e'}, w) = L_i(EX, w)$.*
   (b) *For $u \in \{x', z\}$, $L_i(EX_{e,e'}, u) = L_i(EX_{e,e',x}, u)$.*
   (c) *For $v \in \{y, y'\}$, $L_i(EX_{e,e'}, v) = L_i(EX_{e,e',z}, v)$.*
2. *The messages sent during the round are similar, i.e., $h_i(EX) = h_i(EX_{e,e',x}) = h_i(EX_{e,e',z}) = h_i(EX_{e,e'})$.*
3. *In $EX_{e,e'}$, no messages are sent during the round over the edges $\{x', z\}$ and $\{y, y'\}$.*

Corollary 2.7. *Suppose that during the execution EX neither of the edges $e = \{y, z\}$ and $e' = \{x', y'\}$ are utilized, for some vertices $x \in X$, $y \in Y$, and $z \in Z$. Then the executions EX and $EX_{e,e'}$ are similar and furthermore in $EX_{e,e'}$, no messages are sent through the edges $\{y, y'\}$ and $\{x', z\}$.*

In the next subsections, we will show that this similarity leads to a contradiction with respect to correctness for problems such as $(\Delta + 1)$-coloring and MIS. This in turn will imply a constraint on the behavior of algorithm $\mathcal{A}$: for every pair of edges $e = \{y, z\}$ and $e' = \{x', y'\}$, at least one of the edges is utilized by $\mathcal{A}$. This in turn will lead to the message complexity lower bound we desire.

2.3 $\Omega(m)$ message lower bound for $(\Delta + 1)$-Coloring in KT-1 Congest

Now that we have shown that EX and $EX_{e,e'}$ are similar if e and e' are not utilized by algorithm $\mathcal{A}$, we will show that for some problems this leads to a contradiction. The intuition for this is simple. Let ϕ and ϕ' be ID assignments for V and V' respectively, that consistently order the vertices, i.e., $\phi(u) < \phi(v)$ iff $\phi'(u') < \phi'(v')$ for all $u, v \in V$. Since G and G' are isomorphic, it is easy to show that $EX_G = EX(\mathcal{A}, G, \phi)$ and $EX_{G'} = EX(\mathcal{A}, G', \phi')$ are similar. This is shown below in Lemma 2.8 below. Now consider the base graph $G \cup G'$ and the ID assignment $\psi_{e,e'}$ of $V \cup V'$. Lemma 2.8 implies that corresponding vertices v and v' have the same local states after execution $EX = EX(\mathcal{A}, G \cup G', \psi_{e,e'})$ completes. Since EX and $EX_{e,e'} = EX(\mathcal{A}, G_{e,e'}, \psi_{e,e'})$ are similar, this also implies that vertices v and v' have the same local states after execution $EX_{e,e'}$. But, in the crossed graph $G_{e,e'}$, y and y' are neighbors. For problems in which neighboring vertices ought not to have the same local state (e.g., neighboring vertices cannot have the same color in a solution to the vertex coloring problem), this is a contradiction.

Lemma 2.8. *Consider an arbitrary vertex $y \in Y$ and an arbitrary pair of edges $e = \{y, z\}$, $z \in Z$ and $e' = \{x', y'\}$, $x' \in X'$. For any comparison-based algorithm $\mathcal{A}$ in the KT-1 Congest model, the executions $EX_G = EX(\mathcal{A}, G, \phi)$ and $EX_{G'} = EX(\mathcal{A}, G', \phi'_{e,e'})$ are similar.*

Proof. Since the input graphs G and G' are copies of each other, the only thing that is different between the two executions is the ID assignments. However, Property (iii) of the ID assignment $\phi'_{e,e'}$ above implies that every ID comparison by $\mathcal{A}$ on G yields the same result as the corresponding ID comparison on G'. Therefore, by an inductive argument it can be shown that at the beginning of each round, the state of each vertex v in G is the same as the state of the corresponding vertex v' in G' and the messages received by these vertices are also be the same. This gives us that the executions EX_G and $EX_{G'}$ are similar. □

Lemma 2.9. *Let $x \in X, y \in Y$, and $z \in Z$ be three vertices such that the edges $e = \{y, z)\}$ and $e' = \{x', y'\}$ are not utilized in the execution EX. Then, algorithm $\mathcal{A}$ computes an incorrect $(\Delta + 1)$-coloring for the crossed graph $G_{e,e'}$.*

Proof. In the execution EX, since the input graph has two disconnected components G and G', Lemma 2.8 gives us that the color of a vertex v in G is the same as the color of the corresponding vertex v' in G'. Since the edges $e = \{y, z\}$ and $e' = \{x', y'\}$ are not utilized in $G \cup G'$, applying Corollary 2.7, $\mathcal{A}$ will compute the same coloring in the graph $G_{e,e'}$ as it will in $G \cup G'$. This implies a monochromatic edge $\{y, y'\}$ in $G_{e,e'}$ which contradicts the correctness of the algorithm. □

Theorem 2.10 (Deterministic Lower Bound). *Let $\mathcal{A}$ be a deterministic comparison-based algorithm that computes a $(\Delta+1)$-coloring. Then the message complexity of $\mathcal{A}$ is $\Omega(n^2)$. This holds even if the vertices know the size of the network.*

Proof. Suppose that $\mathcal{A}$ is a deterministic comparison-based algorithm that computes a $(\Delta + 1)$-coloring and has message complexity $o(n^2)$. Then by Lemma 2.4, the number of edges utilized by $\mathcal{A}$ is $o(n^2)$. This implies that there exists a $y \in Y$ and edges $e = \{y, z\}$ and $e' = \{x', y'\}$ such that e and e' are not utilized when $\mathcal{A}$ executes on $G \cup G'$. By Lemma 2.9 this implies that $\mathcal{A}$ computes an incorrect $(\Delta + 1)$-coloring for $G_{e,e'}$. □

We now extend this lower bound to Monte Carlo randomized algorithms, even with constant error probability. To do this we strengthen Lemma 2.9 so that it applies not just to a single crossed graph, but to the entire family of crossed graphs. Let $\mathcal{F}$ denote the family of all crossed graphs, i.e., $\mathcal{F} = \{G_{e,e'} \mid e = \{y, z\}, e' = \{x', y'\}, x, y, z, \in V\}$. Note that $|\mathcal{F}| = t^3$ because there are t choices for y and for each choice of y, there are t^2 choices for e and e'.

Lemma 2.11. *Let $\mathcal{A}$ be a deterministic comparison-based KT-1 Congest algorithm that computes a $(\Delta + 1)$-coloring correctly on at least a constant δ fraction of graphs in the family $\mathcal{F}$. Then the message complexity of $\mathcal{A}$ is $\Omega(\delta n^2)$. This holds even if the vertices know the size of the network.*

Proof. Assume for the sake of contradiction that the message complexity of $\mathcal{A}$ is $o(\delta n^2)$. By Lemma 2.4, we have that $\mathcal{A}$ utilizes

$o(\delta n^2)$ edges in any graph that it runs on. Specifically consider the execution EX of algorithm $\mathcal{A}$ on input graph $G \cup G'$ and ID assignment $\psi_{e,e'}$ where e, e' denote a graph $G_{e,e'}$ in the family $\mathcal{F}$.

Since $\mathcal{A}$ utilizes $o(\delta n^2)$ edges, there can only be $o(n) = o(t)$ vertices in Y such that more than $cn/6 = ct$ incident edges are utilized, for some constant c to be determined later. Recall that $t = n/6$. The rest of the $t - o(t)$ vertices in Y have less than ct incident edges that are utilized. By Lemma 2.8 the same statement holds for the corresponding vertices in Y' because in EX, the two graphs G and G' that form the input graph are disconnected, which implies the executions of $\mathcal{A}$ on G and G' are similar.

So for each such vertex $y \in Y$, there are at most $(c^2/4)t^2$ edge pairs of the form $e = \{y, z\}$, $e' = \{x', y'\}$ such that e, e' are utilized. Therefore, by Lemma 2.9, the algorithm computes an incorrect $(\Delta+1)$-coloring on at least $(1-o(1))(1-(c^2/4)) = 1-(c^2/4)-o(1)$-fraction of the graphs in $\mathcal{F}$ (since for each $y \in Y$ there are exactly t^2 graphs in $\mathcal{F}$). Setting $c = \sqrt{2\delta}$, the algorithm computes an incorrect $(\Delta+1)$-coloring on at least $1 - \delta/2 - o(1)$-fraction of the graphs in $\mathcal{F}$. This is a contradiction if $1 - \delta < 1 - \delta/2 - o(1)$ or $\delta > o(1)$. Since δ is a constant, we get a contradiction. □

A simple application of Yao's lemma [26, 38] with the uniform distribution on all the graphs in the family $\mathcal{F}$ gives the following theorem.

Theorem 2.12 (Randomized Lower Bound). *Let $\mathcal{A}$ be a randomized Monte-Carlo comparison based KT-1 Congest algorithm that computes a $(\Delta+1)$-coloring with probability of error less than a constant $\epsilon \in [0, 1)$. Then the worst case message complexity of $\mathcal{A}$ is $\Omega((1-\epsilon)n^2)$. This holds even if the vertices know the size of the network.*

2.4 $\Omega(m)$ message lower bound for MIS in KT-1 Congest

In this section, we show analogous theorems for MIS. The proofs are omitted either due to space constraints (see [29]) or because they are similar to the proofs in the previous section.

Lemma 2.13. *Let $x \in X$, $y \in Y$, and $z \in Z$ be three vertices such that the edges $e = \{y, z\}$ and $e' = \{x', y'\}$ are not utilized in the execution EX. Then, algorithm $\mathcal{A}$ computes an incorrect MIS on $G_{e,e'}$.*

Theorem 2.14 (Deterministic Lower Bound). *Let $\mathcal{A}$ be a deterministic comparison-based KT-1 Congest algorithm that solves the MIS problem. Then the message complexity of $\mathcal{A}$ is $\Omega(n^2)$. This holds even if the vertices know the size of the network.*

Lemma 2.15. *Let $\mathcal{A}$ be a deterministic comparison-based KT-1 Congest algorithm that computes an MIS correctly on at least a constant δ fraction of graphs in the family $\mathcal{F}$. Then the message complexity of $\mathcal{A}$ is $\Omega(\delta n^2)$. This holds even if the vertices know the size of the network.*

Theorem 2.16 (Randomized Lower Bound). *Let $\mathcal{A}$ be a randomized Monte-Carlo comparison-based KT-1 Congest algorithm that computes an MIS with probability of error less than a constant $\epsilon \in [0, 1)$. Then the worst case message complexity of $\mathcal{A}$ is $\Omega((1-\epsilon)n^2)$. This holds even if the vertices know the size of the network.*

2.5 $\Omega(n)$ message lower bound in KT-ρ Congest

The $\Omega(m)$ lower bounds we have proved apply to comparison-based algorithms in the KT-1 Congest model. We now prove a weaker $\Omega(n)$ message complexity bound for $(\Delta+1)$-coloring and MIS, but these apply more generally, to all algorithms (even non-comparison-based) and to the KT-ρ Congest model, for any constant ρ.

Theorem 2.17. *Any randomized Monte Carlo algorithm that computes an MIS or a $(\Delta+1)$-vertex coloring with probability at least $\frac{5}{8}$, requires $\Omega(n)$ messages in expectation in the KT-ρ Congest model, for any constant ρ.*

Proof. Similarly to [21, 27], we assume without loss of generality that algorithms follow the general framework that all nodes perform their coin flips initially and only exchange their current local state (including coin flips) without performing any other local computation until the very last round.

For the given constant ρ, define the constant k to be the smallest integer such that

$$\log^*(k) \geq 2(\rho+3).$$

Consider an n-node graph G consisting of the disjoint union of n/k cycles each of k nodes.[6] For each cycle C_i, we fix a set of IDs R_i from some integer range of size k such that all ID ranges assigned to the cycles are pairwise disjoint. We will equip the nodes of each cycle C_i with k unique IDs given by some permutation of R_i, as described below.

Suppose towards a contradiction that there exists an algorithm B_ρ that computes a 3-coloring on G while sending $o(n)$ messages in expectation. We provide additional power to the algorithm by revealing, to each node u, the coin flips of the nodes in its ρ-neighborhood. Since there are $n/k = \Omega(n)$ cycles but the expected message complexity of B_ρ is $o(n)$, it holds that, with probability at least $\frac{3}{4}$, there exists a cycle C_j such that the nodes in C_j do not send any messages at all when executing B_ρ; call this event Mute. Note that the probability bound on Mute holds for any possible asssignment of IDs to the nodes.

We now condition on Mute occurring. Consider any node $u \in C_j$ and observe that the output of u is a function of its initial knowledge, i.e., its random coin flips and the local state of its ρ-neighborhood. We point out that, even though that u also has knowledge of n, it is easy to see that this does not have any impact on the output of the algorithm. It follows that the execution of algorithm B_ρ at u can be simulated by a *canonical ρ-round algorithm* B_0 under the $KT-0$ assumption that exchanges messages for the first ρ rounds without performing any other computation and then outputs the color at u obtained by locally computing the state transitions of the nodes in u's ρ-hop neighborhood according to B_ρ. Clearly, the output at u follows the exact same probability distribution when executing B_ρ under the KT-ρ assumption as it does when executing algorithm B_0 under the KT-0 assumption.

A straightforward consequence of [27] is that, for each cycle C_i, and any canonical ρ-round algorithm, there exists some permutation of the set R_i, denoted I_i, such that the algorithm fails to yield a valid coloring with some probability greater than $\frac{1}{2}$ (assuming

[6] For simplicity, we assume that n/k is an integer.

KT-0) when the nodes in C_i are assigned the IDs in I_i, where this probability is taken over the coin flips of the nodes in C_i. Let I be the resulting ID assignment if we assign the IDs of the nodes in each cycle C_i according to I_i.

Now consider the execution of B_ρ on graph G with ID assignment I. Since event Mute occurs with probability at least $\frac{3}{4}$, the above implies that algorithm B_ρ fails to yield a valid coloring on at least one of the cycles with probability $> \frac{3}{8}$, yielding a contradiction. □

3 UPPER BOUNDS IN KT-1 CONGEST

3.1 $(\Delta+1)$-Coloring using $\tilde{O}(n^{1.5})$ Messages

In this section we present a $(\Delta+1)$-list-coloring algorithm in the KT-1 Congest model that uses $\tilde{O}(n^{1.5})$ messages. This algorithm is obtained by utilizing – with some modifications – the simple graph partitioning technique introduced recently by Chang et al. [5]. This technique is central to the fast $(\Delta+1)$-coloring algorithms that Chang et al. [5] obtain in different models of computation, namely Congested Clique, MPC, and Centralized Local Computation.

The Chang et al. [5] graph partitioning scheme is as follows. Let $\Psi(v)$ denote the palette of vertex $v \in V$ and let $k = \sqrt{\Delta}$.

- **Vertex set partition:** We partition V into $B_1, \ldots, B_k, L$ as follows. Include each $v \in V$ in the set L with probability $q = \Theta\left(\frac{\sqrt{\log n}}{\Delta^{1/4}}\right)$. Then each remaining vertex joins one of $B_1, \ldots, B_k$ uniformly at random.
- **Palette partition:** Let $C = \bigcup_{v \in V} \Psi(v)$ be the set of all colors. We partition C into k sets $C_1, \ldots, C_k$ where each color $c \in C$ joins one of the k sets uniformly at random.

Chang et al. [5] then show that whp, the output of the partitioning scheme satisfies the following 4 properties, assuming that $\Delta = \omega(\log^2 n)$. These properties allow us to color each set B_i using palette C_i, in parallel, and then recursively color the set L until it becomes small enough to color trivially.

(i) Size of Each Part: $|E(G[B_i])| = O(|V|)$, for each $i \in [k]$. Also, $|L| = O(q|V|) = O\left(\frac{\sqrt{\log n}}{\Delta^{1/4}}\right) \cdot |V|$.

(ii) Available Colors in B_i: For each $i \in \{1, \ldots, k\}$ and $v \in B_i$, let the number of available colors in v in the subgraph B_i be $g_i(v) := |\Psi(v) \cap C_i|$. Then $g_i(v) \geq \Delta_i + 1$, where $\Delta_i := \max_{v \in B_i} \deg_{B_i}(v)$.

(iii) Available Colors in L: For each $v \in L$, define $g_L(v) := |\Psi(v)| - (\deg_G(v) - \deg_L(v))$. It is required that $g_L(v) \geq \max\{\deg_L(v), \Delta_L - \Delta_L^{3/4}\} + 1$ for each $v \in L$, where $\Delta_L := \max_{v \in L} \deg_L(v)$. Note that $g_L(v)$ represents a lower bound on the number of available colors in v's palette *after* all of $B_1, \ldots, B_k$ have been colored.

(iv) Remaining Degrees: The maximum degrees of B_i and L are $\deg_{B_i}(v) \leq \Delta_i = O(\sqrt{\Delta})$ and $\deg_L(v) \leq \Delta_L = O(q\Delta) = O\left(\frac{\sqrt{\log n}}{\Delta^{1/4}}\right) \cdot \Delta$. For each vertex, we have that $\deg_{B_i}(v) \leq \max\{O(\log n), O(1/\sqrt{\Delta}) \cdot \deg(v)\}$ and also have $\deg_L(v) \leq \max\{O(\log n), O(q) \cdot \deg(v)\}$.

We now present our algorithm, which takes as input an n-vertex graph G with maximum degree Δ and diameter D. The algorithm runs in the KT-1 Congest model and produces a $(\Delta+1)$-list-coloring of G using $\tilde{O}(n^{1.5})$ messages and running in $\tilde{O}(D + \sqrt{n})$ rounds.

Algorithm 1: KT-1 $(\Delta+1)$-Coloring Algorithm:

1 For $\delta = 1/2$, build a danner H, elect a leader ℓ, and have the leader broadcast a string R of $O(\log^2 n)$ random bits.
2 Nodes use the $O(\log^2 n)$ bits of R to sample three $O(\log n)$-wise independent hash functions: (a) h_L, to decide whether to join L, (b) h, to decide which set B_i to join, and (c) h_c, to determine which color goes into which part C_i.
3 Nodes execute a randomized algorithm for list coloring by Johansson [39] in each B_i in parallel.
4 Using the danner H, we can check whether the induced graph $G[L]$ has $\tilde{O}(n)$ edges.
5 If it does, we execute the list coloring algorithm by Johansson [39] on $G[L]$.
6 If not, we recursively run this algorithm on $G[L]$ with the same parameter n.

The "full independence" version of the following lemma is proved in [5]. We provide a brief sketch of the changes required in this proof to make a version with limited independence go through.

Lemma 3.1. *Properties (i)-(iv) mentioned above hold w.h.p., even when the partitioning of vertices and colors is done using $O(\log n)$-wise independence, as described in Line 2 of Algorithm 1.*

Proof. Chang et al. [5] show that this lemma holds when the vertex partitioning is done using full independence, while the color partitioning is done using $O(\log n)$-wise independence. A closer look at their proof reveals that all four properties are shown using Chernoff bounds, and these bounds can be safely replaced by limited dependence Chernoff bounds described in Lemma 1.4. Therefore the four properties hold whp even when the partitioning of both vertices and colors is done using $O(\log n)$-wise independence. □

The following lemma is proved in [5] and given that Properties (i)-(iv) hold in the limited independence setting we use, it goes through without any changes.

Lemma 3.2. *The algorithm makes $O(1)$ recursive calls w.h.p.*

Theorem 3.3. *Given as input an n-vertex graph G with maximum degree Δ and diameter D, Algorithm 1 runs in the KT-1 Congest model and produces a $(\Delta+1)$-list-coloring of G using $\tilde{O}(n^{1.5})$ messages and running in $\tilde{O}(D + \sqrt{n})$ rounds.*

Due to space constraints, the proof of the above theorem appears in the full version [29].

3.1.1 Asynchronous KT-1 Congest algorithm. The $(\Delta+1)$-coloring in the Congest KT-1 mode described above (Algorithm 1) has a natural counterpart in the asynchronous version of the Congest KT-1 model. The details appear in the full version [29] leading to the following theorem.

Theorem 3.4. *Given as input an n-vertex graph G with maximum degree Δ, there is an algorithm that runs in the asynchronous KT-1 Congest model and produces a $(\Delta+1)$-list-coloring of G using $\tilde{O}(n^{1.5})$ messages and running in $\tilde{O}(n)$ rounds.*

3.2 $(1+\epsilon)\Delta$-Coloring using $\tilde{O}(n)$ Messages

In this section, we show that for any $\epsilon > 0$, there is an algorithm that can compute a $(1+\epsilon)\Delta$-coloring in the KT-1 Congest model in $\tilde{O}(n)$ rounds, using $\tilde{O}(n/\epsilon^2)$ messages. Due to space constraints, the proofs in this section appear in the full version [29].

At the beginning of the algorithm, for a large enough constant C, one node generates $(C/\epsilon)\cdot \log^3 n$ random bits and shares it with all other nodes using a danner [12], using $\tilde{O}(n/\epsilon)$ messages and $\tilde{O}(n)$ rounds in the KT-1 Congest model (cf. Corollary 1.2). In the following algorithm, each node v that has not already permanently colored itself, will use random bit string s_i in Phase i to first select a random hash function h_i from a family of $\Theta(\log n)$-wise independent hash functions $\mathcal{H} = \{h : [\text{poly}(n)] \to [(1+\epsilon)\Delta]\}$. Node v will then compute $h_i(\texttt{ID}_v)$ to pick a random color from the palette $[(1+\epsilon)\Delta]$. Note that the length of s_i is $\Theta(\log^2 n)$ and by Lemma 1.6, this number of random bits suffice to pick a $\Theta(\log n)$-wise independent hash function with domain size $\text{poly}(n)$ and range size $(1+\epsilon)\Delta$. In Corollary 3.6, it is shown that Algorithm 2 runs in $O(\log n/\epsilon)$ phases and therefore $r = \Theta(\log n/\epsilon)$ random bit strings suffice.

Algorithm 2: KT-1 $(1+\epsilon)\Delta$-Coloring (One phase):

1. Each active node (i.e., which has not been colored yet) chooses a random candidate color from $(1+\varepsilon)\Delta$ color palette.
2. It makes this color permanent if it is sure that none of its neighbors has chosen this color yet.
3. If unsuccessful in choosing a permanent color, go to step 1.

In step 2, we will show that a node has to check only a small subset of its neighbors in any phase. First, we will show that the probability of success in each phase is large.

Lemma 3.5. *In any phase, a node chooses a color that has not been chosen by any of its neighbors in this phase or in any previous phases with probability at least $\varepsilon/(1+\varepsilon) \approx \varepsilon$ (for small ε). Hence there will be no conflict with the chosen color and hence the node will successfully color itself. Thus, a node successfully colors itself in $O(\log n/\varepsilon)$ rounds whp.*

Corollary 3.6. *Whp, all nodes successfully color themselves in $O(\log n/\varepsilon)$ rounds.*

Implementing step 2 with small message complexity:

Lemma 3.7. *In each phase, each node exchanges at most $O(\log^2 n/\varepsilon)$ messages whp.*

Theorem 3.8. *There is a coloring algorithm that achieves $(1+\varepsilon)\Delta$ coloring using $O(n\log^3 n/\varepsilon^2)$ messages whp in KT1 model (with shared randomness).*

4 AN MIS ALGORITHM USING $\tilde{O}(n^{1.5})$ MESSAGES IN KT-2 CONGEST

We now give a high-level overview of Algorithm 3 that uses KT-2 knowledge to compute an MIS using only $O(n^{1.5}\log^2 n)$ messages while taking $\tilde{O}(\sqrt{n})$ rounds; the full details are explained in the full version [29]. We first sample a set S of $\Theta(\sqrt{n})$ nodes and then add these nodes to the independent set according to the randomized greedy MIS algorithm. Since S was chosen randomly, this has the same effect as performing $\Theta(\sqrt{n})$ iterations of the sequential randomized greedy algorithm, which is known to reduce the maximum degree in the remnant graph to $\tilde{O}(\sqrt{n})$ (see [18]). Then, each node $u \in S$ that entered the independent set informs its 2-hop neighbors. It is crucial that node u uses its KT-2 knowledge to convey this information, as otherwise the same 2-hop neighbor v might receive u's message from multiple 1-hop neighbors of u, which may result in $\omega(n)$ messages being sent on behalf of u. Finally, we compute an MIS on the (sparsified) remnant graph using Luby's algorithm.

Algorithm 3: KT-2 MIS Algorithm:

1. **Sample $O(\sqrt{n})$ vertices:** Add each node to a set S with probability $c/\sqrt{n}$.
2. **Run Randomized Greedy MIS:** Each node in S chooses a random rank at the start of the algorithm. In the parallel version of Greedy, a node enters the MIS as soon as it is a local maximum among undecided neighbors in S.
3. **Inform 2-hop Neighbors:** Each node $u \in S$ that enters the MIS u uses KT-2 knowledge to inform all of its 2-hop neighbors that it has joined the MIS.
4. **Pruning Inactive Edges:** Each node $v \in V$ uses its own KT-2 knowledge to either deactivate itself if a 1-hop neighbor has joined the MIS or deactivate edges incident on the 1-hop neighbors that are neighbors with a node that joins the MIS.
5. **Finishing Up:** All nodes in the remnant graph know which of their neighbors are deactivated and so we can run Luby's algorithm on the remnant graph.

Theorem 4.1. *Algorithm 3 computes a correct MIS. Moreover, it uses $O(n^{1.5}\log^2 n)$ messages and runs in $\tilde{O}(\sqrt{n})$ rounds with high probability.*

5 CONCLUSION

In this paper, we initiate the study of the message complexity of two fundamental symmetry breaking problems, MIS and $(\Delta+1)$-coloring. We show that while it is impossible to obtain $o(m)$ message complexity in the KT-1 Congest model using comparison-based algorithms, one can do so by either using non-comparison based algorithms or by slightly increasing the input knowledge, i.e., in the KT-2 Congest model.

Several key open questions arise from our work. The first is whether one can obtain an $o(m)$-message, non-comparison-based algorithm for MIS in the KT-1 Congest model, running in polynomial time. We have shown that this is possible for $(\Delta+1)$-coloring. The second is whether one can obtain (nearly optimal) $\tilde{O}(n)$-message (non-comparison-based) algorithms for MIS and $(\Delta+1)$-coloring in the KT-1 Congest model, running in polynomial time. The question is open for MIS even in the KT-2 Congest model. Another important issue is reducing the running time of our algorithms. In particular, can we make them run in polylog n rounds, for the same message bounds?

REFERENCES

[1] Baruch Awerbuch, Oded Goldreich, David Peleg, and Ronen Vainish. 1988. A Tradeoff between Information and Communication in Broadcast Protocols. 319 LNCS, 2 (1988), 369–379. https://doi.org/10.1007/BFb0040404

[2] Leonid Barenboim and Michael Elkin. 2013. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Morgan & Claypool Publishers.

[3] Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. 2012. The Locality of Distributed Symmetry Breaking. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS '12)*. IEEE Computer Society, USA, 321–330. https://doi.org/10.1109/FOCS.2012.60

[4] Yi-Jun Chang, Wenzheng Li, and Seth Pettie. 2018. An optimal distributed (Δ+1)-coloring algorithm?. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 445–456.

[5] Yi-Jun Chang, Manuela Fischer, Mohsen Ghaffari, Jara Uitto, and Yufan Zheng. 2019. The Complexity of (Δ + 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 '19)*. Association for Computing Machinery, New York, NY, USA, 471–480. https://doi.org/10.1145/3293611.3331607

[6] R Cole and U Vishkin. 1986. Deterministic Coin Tossing and Accelerating Cascades: Micro and Macro Techniques for Designing Parallel Algorithms. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (STOC '86)*. Association for Computing Machinery, New York, NY, USA, 206–219. https://doi.org/10.1145/12130.12151

[7] Artur Czumaj, Peter Davies, and Merav Parter. 2020. Simple, Deterministic, Constant-Round Coloring in the Congested Clique. *Proceedings of the 39th Symposium on Principles of Distributed Computing* (Jul 2020). https://doi.org/10.1145/3382734.3405751

[8] Michael Elkin. 2020. A Simple Deterministic Distributed MST Algorithm with Near-Optimal Time and Message Complexities. *J. ACM* 67, 2 (2020), 13:1–13:15.

[9] Mohsen Ghaffari. 2016. An Improved Distributed Algorithm for Maximal Independent Set. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. 270–277.

[10] Mohsen Ghaffari. 2019. Distributed Maximal Independent Set Using Small Messages. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '19)*. Society for Industrial and Applied Mathematics, USA, 805–820.

[11] Mohsen Ghaffari and Fabian Kuhn. 2018. Distributed MST and Broadcast with Fewer Messages, and Faster Gossiping. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018 (LIPIcs)*, Ulrich Schmid and Josef Widder (Eds.), Vol. 121. 30:1–30:12.

[12] Robert Gmyr and Gopal Pandurangan. 2018. Time-Message Trade-Offs in Distributed Algorithms. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018 (LIPIcs)*, Ulrich Schmid and Josef Widder (Eds.), Vol. 121. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 32:1–32:18. https://doi.org/10.4230/LIPIcs.DISC.2018.32

[13] Magnús M. Halldórsson, Fabian Kuhn, Yannic Maus, and Tigran Tonoyan. 2020. Efficient Randomized Distributed Coloring in CONGEST. *CoRR* abs/2012.14169 (2020). https://arxiv.org/abs/2012.14169

[14] Howard Karloff, Siddharth Suri, and Sergei Vassilvitskii. 2010. A model of computation for mapreduce. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 938–948.

[15] Richard M. Karp and Michael O. Rabin. 1987. Efficient Randomized Pattern-Matching Algorithms. *IBM J. Res. Dev.* 31, 2 (1987), 249–260.

[16] Valerie King, Shay Kutten, and Mikkel Thorup. 2015. Construction and Impromptu Repair of an MST in a Distributed Network with o(m) Communication. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, Chryssis Georgiou and Paul G. Spirakis (Eds.). ACM, 71–80. https://doi.org/10.1145/2767386.2767405

[17] Hartmut Klauck, Danupon Nanongkai, Gopal Pandurangan, and Peter Robinson. 2015. Distributed Computation of Large-Scale Graph Problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '15)*. Society for Industrial and Applied Mathematics, USA, 391–410.

[18] Christian Konrad. 2018. MIS in the Congested Clique Model in O(log log Δ) Rounds. *CoRR* abs/1802.07647 (2018). arXiv:1802.07647 http://arxiv.org/abs/1802.07647

[19] E. Korach, S. Moran, and S. Zaks. 1987. The Optimality of Distributive Constructions of Minimum Weight and Degree Restricted Spanning Trees in a Complete Network of Processors. *SIAM J. Comput.* 16, 2 (April 1987), 231–236. https://doi.org/10.1137/0216019

[20] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. 2015. On the Complexity of Universal Leader Election. *J. ACM* 62, 1 (2015), 7:1–7:27.

[21] Nathan Linial. 1992. Locality in Distributed Graph Algorithms. *SIAM J. Comput.* 21, 1 (1992), 193–201. https://doi.org/10.1137/0221015

[22] Zvi Lotker, Boaz Patt-Shamir, Elan Pavlov, and David Peleg. 2005. Minimum-Weight Spanning Tree Construction in O(Log Log n) Communication Rounds. *SIAM J. Comput.* 35, 1 (July 2005), 120–131. https://doi.org/10.1137/S0097539704441848

[23] M Luby. 1985. A Simple Parallel Algorithm for the Maximal Independent Set Problem. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC '85)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/22145.22146

[24] Ali Mashreghi and Valerie King. 2018. Broadcast and Minimum Spanning Tree with o(m) Messages in the Asynchronous CONGEST Model. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018 (LIPIcs)*, Vol. 121. 37:1–37:17.

[25] Ali Mashreghi and Valerie King. 2019. Brief Announcement: Faster Asynchronous MST and Low Diameter Tree Construction with Sublinear Communication. In *33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary (LIPIcs)*, Jukka Suomela (Ed.), Vol. 146. 49:1–49:3.

[26] Rajeev Motwani and Prabhakar Raghavan. 1995. *Randomized Algorithms*. Cambridge University Press, USA.

[27] Moni Naor. 1991. A Lower Bound on Probabilistic Algorithms for Distributive Ring Coloring. *SIAM J. Discret. Math.* 4, 3 (1991), 409–412. https://doi.org/10.1137/0404036

[28] Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, Talal Riaz, and Peter Robinson. 2017. Symmetry Breaking in the Congest Model: Time- and Message-Efficient Algorithms for Ruling Sets. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria (LIPIcs)*, Vol. 91. 38:1–38:16.

[29] Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, and Peter Robinson. 2021. Can We Break Symmetry with o(m) Communication? arXiv:cs.DC/2105.08917

[30] Shreyas Pai and Sriram V. Pemmaraju. 2020. Connectivity Lower Bounds in Broadcast Congested Clique. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020) (Leibniz International Proceedings in Informatics (LIPIcs))*, Nitin Saxena and Sunil Simon (Eds.), Vol. 182. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 32:1–32:17. https://doi.org/10.4230/LIPIcs.FSTTCS.2020.32

[31] Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. 2017. A time- and message-optimal distributed algorithm for minimum spanning trees. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. ACM, 743–756.

[32] Boaz Patt-Shamir and Mor Perry. 2017. Proof-Labeling Schemes: Broadcast, Unicast and in Between. In *Stabilization, Safety, and Security of Distributed Systems - 19th International Symposium, SSS 2017, Boston, MA, USA, November 5-8, 2017, Proceedings (Lecture Notes in Computer Science)*, Paul G. Spirakis and Philippas Tsigas (Eds.), Vol. 10616. Springer, 1–17. https://doi.org/10.1007/978-3-319-69084-1_1

[33] Peter Robinson. 2021. Being Fast Means Being Chatty: The Local Information Cost of Graph Spanners. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

[34] Will Rosenbaum and Jukka Suomela. 2020. Seeing Far vs. Seeing Wide: Volume Complexity of Local Graph Problems. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*. 89–98.

[35] Václav Rozhon and Mohsen Ghaffari. 2020. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*. 350–363.

[36] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. 1993. *Chernoff-Hoeffding Bounds for Applications with Limited Independence*. Society for Industrial and Applied Mathematics, USA, 331–340.

[37] Salil P. Vadhan. 2012. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science* 7, 1–3 (2012), 1–336. https://doi.org/10.1561/0400000010

[38] Andrew Chi-Chin Yao. 1977. Probabilistic Computations: Toward a Unified Measure of Complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (SFCS '77)*. IEEE Computer Society, USA, 222–227. https://doi.org/10.1109/SFCS.1977.24

[39] Öjvind Johansson. 1999. Simple Distributed (Δ + 1)-Coloring of Graphs. *Information Processing Letters 70* 70 (1999), 229–232.