The Message Complexity of Distributed Graph Optimization

Fabien Dufoulon
Lancaster University, UK

Shreyas Pai
Aalto University, Finland

Gopal Pandurangan
University of Houston, TX, USA

Sriram V. Pemmaraju
University of Iowa, IA, USA

Peter Robinson
Augusta University, GA, USA

Abstract

The *message complexity* of a distributed algorithm is the total number of messages sent by all nodes over the course of the algorithm. This paper studies the message complexity of distributed algorithms for fundamental graph optimization problems. We focus on four classical graph optimization problems: Maximum Matching (MaxM), Minimum Vertex Cover (MVC), Minimum Dominating Set (MDS), and Maximum Independent Set (MaxIS). In the sequential setting, these problems are representative of a wide spectrum of hardness of approximation. While there has been some progress in understanding the round complexity of distributed algorithms (for both exact and approximate versions) for these problems, much less is known about their message complexity and its relation with the quality of approximation. We almost fully quantify the message complexity of distributed graph optimization by showing the following results:

1. *Cubic regime:* Our first main contribution is showing essentially *cubic*, i.e., $\tilde{\Omega}(n^3)$ lower bounds[1] (where n is the number of nodes in the graph) on the message complexity of distributed *exact* computation of Minimum Vertex Cover (MVC), Minimum Dominating Set (MDS), and Maximum Independent Set (MaxIS). Our lower bounds apply to any distributed algorithm that runs in polynomial number of rounds (a mild and necessary restriction). Our result is significant since, to the best of our knowledge, this are the first $\omega(m)$ (where m is the number of edges in the graph) message lower bound known for distributed computation of such classical graph optimization problems. Our bounds are essentially tight, as all these problems can be solved trivially using $O(n^3)$ messages in polynomial rounds. All these bounds hold in the standard CONGEST model of distributed computation in which messages are of $O(\log n)$ size.
2. *Quadratic regime:* In contrast, we show that if we allow *approximate* computation then $\tilde{\Theta}(n^2)$ messages are both necessary and sufficient. Specifically, we show that $\tilde{\Omega}(n^2)$ messages are required for *constant-factor* approximation algorithms for all four problems. For MaxM and MVC, these bounds hold for any constant-factor approximation, whereas for MDS and MaxIS they hold for any approximation factor better than some specific constants. These lower bounds hold even in the LOCAL model (in which messages can be arbitrarily large) and they even apply to algorithms that take arbitrarily many rounds. We show that our lower bounds are essentially tight, by showing that if we allow approximation to within an arbitrarily small constant factor, then all these problems can be solved using $\tilde{O}(n^2)$ messages even in the CONGEST model.
3. *Linear regime:* We complement the above lower bounds by showing distributed algorithms with $\tilde{O}(n)$ message complexity that run in polylogarithmic rounds and give constant-factor approximations for all four problems on *random* graphs. These results imply that almost linear (in n) message complexity is achievable on almost all (connected) graphs of every edge density.

[1] $\tilde{\Omega}$ and $\tilde{O}$ hide a $1/\operatorname{polylog} n$ and $\operatorname{polylog} n$ factor respectively.

15th Innovations in Theoretical Computer Science Conference (ITCS 2024).
Editor: Venkatesan Guruswami; Article No. 41; pp. 41:1–41:26
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed graph algorithm, message complexity, distributed approximation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2024.41

Related Version *Full Version*: https://arxiv.org/abs/2311.14811

Funding *Fabien Dufoulon*: Supported, while a postdoctoral fellow at the University of Houston (in Houston, USA), in part by National Science Foundation (NSF) grants CCF-1540512, IIS-1633720, and CCF-1717075 and U.S.-Israel Binational Science Foundation (BSF) grant 2016419.
Shreyas Pai: Supported in part by Research Council of Finland Grant 334238 and Helsinki Institute for Information Technology HIIT.
Gopal Pandurangan: Supported in part by National Science Foundation (NSF) grants CCF-1540512, IIS-1633720, and CCF-1717075 and U.S.-Israel Binational Science Foundation (BSF) grant 2016419.
Sriram V. Pemmaraju: Partially supported by NSF grant IIS-1955939.

1 Introduction

The focus of this paper is understanding the communication cost of distributively solving graph optimization problems. The communication cost of distributed computation has been studied extensively in theoretical computer science since the seminal work of Yao [81]. This line of work studies the communication cost of computing boolean functions of the form $f(x, y)$, where the input $x \in \{0,1\}^n$ is given to Alice and the input $y \in \{0,1\}^n$ is given to Bob, and these two players (nodes) jointly compute $f(x, y)$ by communicating across a communication link (edge). The *communication complexity* of f is measured by the minimum number of bits exchanged by Alice and Bob to compute f. Boolean functions that have been studied extensively include *equality*, *set disjointness* etc.; we refer to [60, 74] for a comprehensive treatment. We note that the communication complexity of functions on graphs, e.g., connectivity, bipartiteness, maximum matching, etc., has also been studied; see [46, 50, 51, 29, 42] for some examples. In the context of a graph problem, each player gets a *portion of an input graph G*. The graph may be edge-partitioned into two parts or may be partitioned in some other arbitrary way.

Over the years many extensions and variants of this basic "two-party" communication complexity model have been studied. One important early variant is by Tiwari [77] who studied the same problem, but instead of the two players communicating via an edge, the two players communicate via an arbitrary network. Another important variant is *multi-party* communication complexity, introduced in the work of Chandra, Furst and Lipton [19], where k players are provided inputs $x_1, \ldots, x_k \in \{0,1\}^n$ and these k players want to compute some joint boolean function $f : (\{0,1\}^n)^k \to \{0,1\}$ with the goal of minimizing the total communication between the k players. In the multi-party model, the k players are connected by a communication network, which is typically a clique, but arbitrary topologies have also been considered (see for e.g., [80, 22] and the references therein).

A related, yet different line of work has been the study of the *message complexity* in distributed computing (see for e.g., [61, 67, 4, 2, 70, 31]). In this line of work, we are given an input graph G, which *also* serves as the communication network. The nodes of this network (which can be viewed as players) communicate along the edges of G to solve problems *defined on G*. The message complexity is simply the total number of messages sent by all nodes over the course of the algorithm. Usually, only messages of small size (say, $O(\log n)$ bits) are allowed, hence in such cases, message complexity is essentially the total number of bits communicated (up to a small factor). A wide variety of graph problems have been studied in this setting, including classical problems such as breadth-first search, minimum spanning tree, minimum cut, maximal independent set, $(\Delta+1)$-coloring, shortest paths, etc., and NP-complete problems such as minimum vertex cover, minimum dominating set, etc. A key difference between this line of work and the previously described research on two-party and multi-party communication complexity models is that here the input graph G is *also* the communication network and so the structure of G simultaneously determines both the difficulty of the problem and the difficulty of communication needed to solve the problem. For example, it may be that a particular problem is easier to solve on a sparse input graph G, but the sparsity of G also limits the volume of information that can be exchanged along the edges of G. A second difference is that most prior work on two-party and multi-party communication complexity by default assumes an *asynchronous* communication model. Whereas in distributed computing, a *synchronous* model of computation, i.e., a model with a global clock, is extensively used. While there has been a significant progress in our understanding of communication complexity in the 2-party and multi-party settings, our understanding of the message complexity of distributed computation of graph problems is significantly limited. We refer to Section 1.3 for more details comparing and contrasting the above lines of research.

The focus of this paper is gaining a deeper understanding of the *message complexity* of distributed algorithms for fundamental graph optimization problems. Besides message complexity, *round complexity* is also a key measure of the performance of distributed algorithms. While a rich body of literature exists on the round complexity of distributed exact and approximation algorithms for graph optimization problems [9, 59, 58, 17, 52, 24, 39, 68, 5, 18], much less is known about the message complexity of these problems and the possibility of tradeoffs between the message complexity and the quality of approximation that can be achieved for these problems.

We focus on four classical graph optimization problems: Maximum Matching (MaxM), Minimum Vertex Cover (MVC), Minimum Dominating Set (MDS), and Maximum Independent Set (MaxIS). In the sequential setting, these problems are representative of a wide spectrum of hardness of approximability. MaxM can be solved exactly in polynomial time. MVC has a simple 2-approximation algorithm, but it does not have a better than 1.3606-approximation [25]. A simple greedy algorithm provides a $O(\log\Delta)$-approximation to MDS [78], though it is known that MDS does not have a $(1-\epsilon)\cdot\ln\Delta$-approximation (where Δ is the maximum degree of the graph) for any $0<\epsilon<1$ [26]. Finally, MaxIS is known to be even harder; it does not even have an $O(n^{1-\epsilon})$-approximation for any $0<\epsilon<1$ [78]. All of these hardness of approximation results are conditional on P $\neq$ NP.

In the standard models of distributed computing such as LOCAL and CONGEST, it is assumed that processors have infinite computational power. This means that hardness of approximation results in the sequential setting do not directly translate to the distributed setting. Hardness of approximation in the distributed setting is, roughly speaking, due to the distance information has to travel or the volume of information that has to travel for nodes to produce a solution that is close enough to optimal. Researchers are starting to better understand the distributed hardness of approximation from a *round complexity* point of view, but intriguing gaps remain. For example, there is a $(2+\epsilon)$-approximation algorithm for MVC (even vertex-weighted MVC) running in $O(\log\Delta/(\epsilon\log\log\Delta))$ rounds in the CONGEST model [9]. The approximation factor was reduced to exactly 2 in [14], but at the cost of polylogarithmic factor extra rounds. These upper bounds are complemented by an $\tilde{\Omega}(n^2)$ *round* lower bound for solving MVC *exactly* in the CONGEST model [18]. Currently, the round complexity of obtaining an α-approximation for MVC, for $1<\alpha<2$, is unknown.

Even this level of understanding is lacking about the *message complexity* of distributed graph optimization. The key question addressed in this paper is this: *how does the message complexity of fundamental distributed graph optimization problems change as we move from exact algorithms to approximation algorithms?* We almost fully answer this question and the key takeaway from our results is that there is a sharp separation between the message complexity of exact and approximate solutions. Specifically, we show that $\tilde{\Theta}(n^3)$ messages are necessary and sufficient for the distributed *exact* computation of MVC, MDS, and MaxIS for algorithms that runs in a polynomial number of rounds. In contrast, we show that if we allow *approximate* computation, for a constant-approximation factor, then $\tilde{\Theta}(n^2)$ messages are both necessary and sufficient for algorithms that run in polynomial rounds for all four problems, MaxM, MVC, MDS, and MaxIS. We note that focusing on algorithms that run in polynomial rounds is hardly restrictive because any problem can be solved in polynomial rounds in standard distributed computing models (e.g., CONGEST and LOCAL) by gathering the entire input at a single node.

1.1 Distributed Computing Models

We primarily work in the *synchronous* version of a standard message-passing model of distributed computing known as CONGEST [73]. In this model, the input is a graph $G = (V, E)$, $n = |V|$, $m = |E|$, 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 the *synchronous* version of this model, it is assumed that all nodes have access to have a common global clock, and both computation and communication proceed in lockstep, i.e., in discrete time steps called *rounds*.[2] In each round, each node (i) receives messages (if any) sent to it in the previous round, (ii) performs local computation based on information it has, and (iii) sends a message (possibly different) to each of its neighbors in the graph. Processors are assumed to be arbitrarily powerful and can perform arbitrary (e.g., exponential-time) local computations in a round. We allow only small, i.e., $O(\log n)$-sized messages, to be sent per edge per round. Since each ID can be represented with $O(\log n)$ bits, each message in the CONGEST model is large enough to contain $O(1)$ IDs. We note that some of our message lower bounds also hold in the less restrictive LOCAL model, where messages sent per edge per round can be of arbitrary size.

We primarily work in the standard KT_0 (***K**nowledge **T**ill radius 0*) model, also called the *clean network model* [73], in which nodes have initial local knowledge of only themselves and do not know anything else about the network; specifically, nodes know nothing about their neighbors (e.g., IDs of neighbors). As we explain in the full paper [28], some of our lower bounds even extend to the KT_1 model, in which each node has initial knowledge of itself and the IDs of its neighbors. The point is significant because knowledge of neighbors' IDs can be used in surprising ways to reduce the message complexity of algorithms (see for e.g., the Minimum Spanning Tree (MST) algorithm of King, Kutten, and Thorup [53]). Unless explicitly specified otherwise, all the results we present are in the KT_0 CONGEST model.

1.2 Our Contributions

Our main results, which are summarized in Table 1, can be organized into 3 categories. Column 2 shows essentially tight almost *cubic* (i.e., $\tilde{\Omega}(n^3)$) lower bounds in the CONGEST model on the message complexity of computing *exact* solutions for MVC, MDS, and MaxIS

[2] We note that all our lower bounds also hold in the more general *asynchronous* model where there is no such assumption of a common clock. See Section 2.

in polynomial number of rounds.[3] This is significant since, to the best of our knowledge, these are the *first* $\omega(m)$ *message lower bounds* (where m is the number of edges in the graph) known for distributed computation of graph problems in the CONGEST model. Tight message lower bounds are known for a wide variety of problems in the KT_0 CONGEST model including important global problems such as broadcast, leader election (LE), and minimum spanning tree (MST) [61] as well as for local symmetry breaking problems such as maximal independent set (MIS), ruling sets, and $(\Delta + 1)$-coloring [66, 67]. But all of these lower bounds are either of the form $\Omega(m)$ or $\Omega(n^2)$.

Column 3 shows *quadratic* lower bounds on the message complexity of *constant-factor approximations* for MaxM, MVC, MDS, and MaxIS. These bounds hold not just for polynomial-round algorithms, but even for algorithms that use arbitrarily many rounds. Furthermore, they hold not just in the CONGEST model, but also in the LOCAL model, in which message sizes can be arbitrarily large. These quadratic lower bounds are tight because we are also able to show $\tilde{O}(n^2)$ message upper bounds for constant-approximation algorithms for all four problems, for arbitrarily small constant. To the best of our knowledge, of the four problems we consider, only MDS has been previously studied from a message complexity perspective. In [43, 45], the authors show an (expected) $O(\log \Delta)$-approximation algorithm to MDS in the KT_0 CONGEST model that uses $O(n^{1.5})$ messages, running in polylogarithmic rounds. This upper bound result shows that non-trivial approximation for MDS can be achieved in the KT_0 CONGEST model *without* communicating over most edges. The authors also show a $\tilde{\Omega}(n^{1.5})$ message lower bound for algorithms that yield an $O(1)$-approximation for MDS. Our work significantly improves on this lower bound result by showing that $\Omega(n^2)$ is a lower bound for $5/4 - \epsilon$ approximation of MDS. This indicates that message complexity can be quite sensitive to the *quality of approximation* for some problems and hence the $\Omega(n^2)$ message lower bounds shown in this paper for approximation algorithms cannot be taken for granted.

Finally, we note that we are able to extend these lower bounds (both cubic and quadratic), which are in the KT_0 model, to the KT_1 model also. We present these results in the full paper [28].

We complement our lower bounds by presenting almost-quadratic, i.e., $\tilde{O}(n^2)$, message upper bounds for computing $(1 \pm \epsilon)$-approximations to all four problems (Column 4) and essentially linear, i.e., $\tilde{O}(n)$, message complexity algorithms (Column 5) on $G(n, p)$ (Erdös-Rényi) random graphs[16] for all four problems that give constant-factor approximations with high probability. We now describe the techniques used to obtain our results in more detail.

A. Tight Cubic Lower Bounds for Exact Computation

The starting point for our cubic message lower bounds is the communication-complexity-based approach in [5, 18] that is used to show $\tilde{\Omega}(n^2)$ *round* lower bounds for exact MVC and MDS. In [18], the authors present a reduction from the 2-party communication complexity problem SetDisjointness to MVC. For any positive integer n that is a power of 2 and bit-vectors $x, y \in \{0, 1\}^{n^2}$, this reduction maps an instance (x, y) of SetDisjointness to a graph $G_{x,y}$ with $\Theta(n)$ vertices and $\Theta(n^2)$ edges such that SetDisjointness(x, y) = FALSE iff $G_{x,y}$ has a vertex cover of size at most $4n + 4 \log n - 4$. Furthermore, $G_{x,y}$ has the property that its vertex set can be partitioned into sets V_x and V_y where the subgraph $G_{x,y}[V_x]$ is

[3] It is open whether the message complexity of exactly computing MaxM is $\tilde{\Omega}(n^3)$. See Section 5.

Table 1 A summary of our message complexity lower and upper bound results. All of these results hold in the synchronous KT_0 CONGEST model, with all the message complexity lower bounds applying to all algorithms that run in polynomial rounds. Additionally, the quadratic lower bounds for approximation algorithms (Column 3) even apply in the KT_0 LOCAL model and also to algorithms that take arbitrarily many rounds. The cubic lower bounds for exact algorithms (Column 2) are tight because any problem can be trivially solved in the KT_0 CONGEST model in polynomial rounds using $O(n^3)$ messages by gathering the entire graph topology at a node. Moreover, this gathering algorithm implies that, in *random graphs*, all problems can be solved trivially in $\tilde{O}(n^2)$ messages with high probability since such graphs have $O(\log n)$ diameter with high probability. While our focus is not on round complexity, we note that our approximation algorithms for arbitrary graphs take polynomial number of rounds, while those for random graphs take polylogarithmic number of rounds. For the lower bound for approximate MaxM (Columnn 3), $\epsilon \in (\frac{1}{n^{1/3}}, 1)$; everywhere else the only restriction on ϵ is $0 < \epsilon < 1$. The lower bound for approximate MVC holds for any $c \geq 1$.

Problem	Lower Bounds Exact	Lower Bounds Approximate	Upper Bounds Approximate	Upper Bounds in Random Graphs
MaxM	Open	$\Omega(\epsilon^3 n^2)$ for ϵ-apx	$\tilde{O}(n^2/\epsilon)$ for $(1-\epsilon)$-apx	$\tilde{O}(n)$ for exact
MVC	$\tilde{\Omega}(n^3)$	$\Omega(n^2/c)$ for c-apx	$\tilde{O}(n^2/\epsilon)$ for $(1+\epsilon)$-apx	$\tilde{O}(n)$ for $(2-o(1))$-apx
MDS	$\tilde{\Omega}(n^3)$	$\Omega(n^2)$ for $(\frac{5}{4}-\epsilon)$-apx	$\tilde{O}(n^2/\epsilon)$ for $(1+\epsilon)$-apx	$\tilde{O}(n)$ for $(1+o(1))$-apx
MaxIS	$\tilde{\Omega}(n^3)$	$\Omega(n^2)$ for $\left(\frac{1}{2}+\epsilon\right)$-apx	$\tilde{O}(n^2/\epsilon)$ for $(1-\epsilon)$-apx	$\tilde{O}(n)$ for $(\frac{1}{2}-o(1))$-apx

determined completely by x (and independently of y), the subgraph $G_{x,y}[V_y]$ is completely determined by y (and independently of x), and the cut (V_x, V_y) is small, i.e., has $O(\log n)$ edges. It is then shown that if there is an algorithm $\mathcal{A}$ for solving MVC (exactly) in the KT_0 CONGEST model, then Alice and Bob can solve SetDisjointness on x, y by simulating $\mathcal{A}$. Specifically, Alice and Bob start by respectively constructing $G_{x,y}[V_x]$ and $G_{x,y}[V_y]$ using their private inputs. They then simulate $\mathcal{A}$ round-by-round, communicating with each other only when algorithm $\mathcal{A}$ sends a message from a node in V_x to V_y (or vice versa). Since the (V_x, V_y) cut has size $O(\log n)$, this means that if $\mathcal{A}$ runs in T rounds, Alice and Bob can solve MVC on $G_{x,y}$ by communicating $O(T \cdot \log^2 n)$ bits. Finally, the linear (in length of x and y) lower bound on the communication complexity of SetDisjointness [75] (even for randomized, Monte Carlo algorithms) implies an $\tilde{\Omega}(n^2)$ round lower bound on T. The approach for showing an $\tilde{\Omega}(n^2)$ round lower bound for MDS [5] is quite similar, the only difference being the construction of the lower bound graph $G_{x,y}$.

We extend the above approach to obtain an $\tilde{\Omega}(n^3)$ message lower bound using a key new idea. Our idea is to "stretch" the (V_x, V_y) cut by adding vertex subsets $V_2, V_3, \ldots, V_{\ell-1}$ to the graph $G_{x,y}$. Renaming V_x as V_1 and V_y as V_ℓ, we then replace the edges in the original cut (V_x, V_y) by edges between (V_i, V_{i+1}) for $1 \leq i \leq \ell - 1$. The size of each cut (V_i, V_{i+1}) is still small, i.e., $O(\text{polylog}(n))$ edges. The first challenge we overcome is showing that the correctness of the mapping from SetDisjointness instances (x, y) to MVC instances $G_{x,y}$ is preserved. We now explain the motivation for "stretching" the cut. Suppose there is an algorithm $\mathcal{A}$ that solves MVC on $G_{x,y}$, while sending only $o(n^2/\text{polylog}(n))$ messages across a constant-fraction of the $\ell - 1$ cuts (V_i, V_{i+1}). Then Alice and Bob can simulate $\mathcal{A}$ with low communication complexity. Specifically, Alice and Bob can coordinate to (randomly) pick one of the low-message cuts (V_i, V_{i+1}). Alice starts by constructing the subgraph of $G_{x,y}$ induced by $V_1 \cup V_2 \cup \ldots \cup V_i$ and similarly Bob constructs the subgraph of $G_{x,y}$ induced by $V_{i+1} \cup V_{i+2} \cup \ldots \cup V_\ell$. Alice and Bob can then simulate $\mathcal{A}$ round-by-round, communicating with each other only when $\mathcal{A}$ needs to send a message from V_i to V_{i+1} (or vice versa). Since

$o(n^2/\operatorname{polylog}(n))$ messages are sent across the (V_i, V_{i+1}) cut, this means that Alice and Bob only communicate $o(n^2)$ bits. Because of the mapping from SetDisjointness instances to MVC instances, Alice and Bob can use the solution to MVC obtained by simulating $\mathcal{A}$, to solve SetDisjointness in $o(n^2)$ bits, something that is not possible. This implies that algorithm $\mathcal{A}$ necessarily sends $\tilde{\Omega}(n^2)$ messages across a constant-fraction of the $\ell - 1$ cuts (V_i, V_{i+1}). By setting $\ell = \tilde{\Theta}(n)$, we obtain an $\tilde{\Omega}(n^3)$ message lower bound for $\mathcal{A}$.

The above high-level description glosses over several technical challenges. One of these is the fact that the 2-party communication complexity is asynchronous, i.e., Alice and Bob have no common notion of time, whereas we are interested in proving lower bounds for the *synchronous* KT_0 $\mathsf{CONGEST}$ model. However, in such a synchronous model of distributed computing, the *time-encoding* trick can be used to reduce messages. For example, a node can stay silent for many clock ticks and then send a single bit at clock tick t to a neighbor, thereby using just 1 bit of actual information to implicitly convey $\log t$ bits of information. To overcome this challenge, we first use the above argument in a synchronous version of the 2-party communication model, showing that Alice and Bob can simulate algorithm $\mathcal{A}$ using a small number of bits in the synchronous 2-party communication model. We then appeal to a result from [69] which shows that in 2-party communication models, synchrony can be used to compress messages, but only by a $\log(r)$-factor for r-round algorithms. Applying this result allows us to translate the communication complexity in the synchronous 2-party communication model to communication complexity in the standard 2-party communication model with a logarithmic-factor loss, if we restrict ourselves to algorithms that run in polynomial rounds.

We end this subsection by summarizing the scope of these cubic lower bounds. First, they only hold in the $\mathsf{CONGEST}$ model, and not in the LOCAL model, because the lower bound technique described above relies on edges having low bandwidth. In fact, it is easy to see that any problem can be solved using $O(n^2)$ messages in the LOCAL model because a single node can gather the entire graph topology, and broadcast it to all nodes, using $O(n^2)$ messages. Second, our use of communication complexity techniques to obtain lower bounds in the synchronous setting implies that our cubic lower bounds only hold for algorithms that run in polynomial rounds. Third, while the above argument has been sketched in the KT_0 $\mathsf{CONGEST}$ model, it can be generalized to work in the KT_1 $\mathsf{CONGEST}$ model as well. We present this generalized argument in the full paper [28].

B. Tight Quadratic Lower Bounds for Approximate Computation

Recall that we show quadratic message lower bounds for approximation algorithms not just in the KT_0 $\mathsf{CONGEST}$ model, but even in the KT_0 LOCAL model, and our bounds hold not just for polynomial-round algorithms, but *unconditionally*, i.e., even for algorithms that use arbitrarily many rounds. Unfortunately, communication-complexity-based approaches cannot be used for these types of powerful lower bounds. Communication complexity reductions typically show a lower bound of, say $\Omega(b)$ bits, on the volume of information that travels across a cut in the graph in any algorithm for the problem. However, in the KT_0 LOCAL model, this does not translate to a message complexity lower bound because there is no upper bound on the bandwidth of an edge and in fact b or more bits can travel across an edge in a *single* message in the KT_0 LOCAL model. Furthermore, as mentioned previously, since information can be encoded in clock ticks, communication-complexity-based lower bounds degrade with the number of rounds. So any message complexity lower bound obtained in the *synchronous* KT_0 $\mathsf{CONGEST}$ or LOCAL model necessarily only applies to algorithms that are round-restricted. For these reasons we use approaches different from communication-complexity-based techniques.

Our first technique, which we apply to the MaxM problem, involves showing that finding a large "planted matching" in the network is impossible without $\Omega(n)$ of the nodes identifying incident "planted matching" edges. Further, we show using a symmetry argument that identifying a specific edge incident on a node requires messages to pass over many incident edges. In general, we show in Theorem 8 that there is an inherent dependency between the number of discovered "planted matching" edges and the message complexity of any algorithm for approximating a maximum matching. We believe that this technique could be of independent interest because it can be used to show the difficulty of identifying other "planted subgraphs" in the KT_0 model using few messages. This in turn can lead to message complexity lower bounds in the KT_0 LOCAL model for other problems.

For the other problems, namely MDS, MVC, and MaxIS, we use the so-called *edge-crossing* technique, that has been used to prove a variety of distributed computing lower bounds (see [56, 4, 61, 1, 72] for some examples). For MVC and MaxIS, our constructions build upon the lower bound graphs used in [67] for proving message complexity lower bounds for MIS and $(\Delta + 1)$-coloring. Our use of this technique for MDS, which heavily borrows from communication-complexity-based lower bound constructions, seems novel. Below we sketch the 2-step approach we use to obtain the $\Omega(n^2)$ message lower bound for a $(\frac{5}{4} - \epsilon)$-approximation for MDS in the KT_0 LOCAL model.

(i) In [5] the authors present a reduction from the 2-party communication complexity problem SetDisjointness to MDS and use this to show an $\tilde{\Omega}(n^2)$ round lower bound on computing an exact MDS in the $\mathsf{CONGEST}$ model. For their round lower bound argument, they construct a family of lower bound graphs that have a small cut – of size $O(\text{polylog}(n))$ – across which $\Omega(n^2)$ bits have to flow. This small cut is needed to translate the lower bound on the number of bits to a round complexity lower bound. But, to show message complexity lower bounds, we do not need a small cut and this provides much greater flexibility in the construction of the lower bound graph family. For positive integer n, $x, y \in \{0, 1\}^{n^2}$, the construction in [5] maps the instance (x, y) of SetDisjointness to a graph $G_{x,y}$ with n vertices and $\Theta(n^2)$ edges. We take advantage of the flexibility mentioned above and extend the lower bound construction in [5] to create a relatively large gap in the size of the MDS in graphs $G_{x,y}$ for which SetDisjointness(x, y) = FALSE versus graphs $G_{x,y}$ for which SetDisjointness(x, y) = TRUE[4]. However, at this stage this is still a communication-complexity-based reduction, and as observed earlier we cannot obtain unconditional KT_0 message complexity lower bounds via this construction.

(ii) Our goal now is to circumvent the need for a communication-complexity-based reduction, while still using this lower bound graph construction. To achieve this goal, we pick a graph $G = G_{x,y}$, $x, y \in \{0, 1\}^{n^2}$ for which SetDisjointness(x, y) = TRUE. We then show that for many pairs of edges (e, e') in G, the graph $G(e, e')$ obtained by "crossing" the edges e and e' satisfies the property that $G(e, e') = G_{x',y'}$ for $x', y' \in \{0, 1\}^{n^2}$ where SetDisjointness(x', y') = FALSE. The gap in the MDS sizes mentioned earlier implies that the MDS sizes in G and $G(e, e')$ are relatively different. Finally, we rely on the well-known feature of "edge-crossing" arguments, which is that an algorithm that does not send messages on e and e' cannot distinguish between G and $G(e, e')$. This leads to the result (see Theorems 13 and 14) that $\tilde{\Omega}(n^2)$ messages are needed to obtain an $(5/4 - \epsilon)$-approximation for MDS, for any $\epsilon > 0$, in KT_0 LOCAL model.

[4] This relatively large gap is created by forcing small minimum dominating sets in both cases; 4 when SetDisjointness(x, y) = FALSE and 5 when SetDisjointness(x, y) = FALSE.

We end this subsection by briefly mentioning our technique for obtaining $\tilde{O}(n^2)$ message upper bounds for $(1 \pm \epsilon)$-approximations for all 4 problems in the KT_0 $\mathsf{CONGEST}$ model. A "ball growing" approach has been widely used in distributed computing for problems such as network decomposition in [62, 3, 65]. Combining this approach with local exponential-time computations, Ghaffari, Kuhn, and Maus [40] devised $(1 \pm \epsilon)$-approximations algorithms for covering and packing integer linear programs in the KT_0 LOCAL model, running in polylogarithmic rounds. Since all 4 problems we consider are instances of covering and packing integer linear programs, the results in [40] apply to these problems. Our contribution is to show that this KT_0 LOCAL algorithm can also be implemented in the KT_0 $\mathsf{CONGEST}$ model (i.e., using small messages) using only $\tilde{O}(n^2)$ messages, while running in polynomial time.

C. Tight Linear Bounds for Random Graphs

The $\Omega(n^2)$ message lower bounds that we show hold on some specifically constructed graph families. We complement our lower bounds by presenting essentially linear, i.e., $\tilde{O}(n)$, message complexity algorithms on $G(n, p)$ (Erdös-Rényi) random graphs[16] for all four problems that work (even) in the KT_0 $\mathsf{CONGEST}$ model and give constant-factor approximations with high probability. Our message bounds are essentially tight, since it is easy to see that $\Omega(n)$ is a message lower bound for all these problems. Furthermore, all our algorithms are fast, the algorithms for MVC, MDS, and MaxIS run in $O(\log^2 n)$ rounds, whereas the MaxM algorithm runs in $O(1)$ rounds. These results apply for all $G(n, p)$ random graphs above the connectivity threshold, i.e., $p = \Theta(\log n/n)$ (see Section 4). These results imply that *almost all* graphs[5] of every edge density (above $\Theta(\log n)$) admit very message-efficient (essentially linear) algorithms for exact (for MaxM) or constant-factor approximation (for MaxIS, MDS, and MVC). In other words, this means for the vast majority of graphs one needs to use a small fraction of the edges to solve these problems. In the full version [28], we also show that in general graphs, MaxM can be solved in $O(n)$ messages and $O(1)$ rounds in KT_0 $\mathsf{CONGEST}$ giving an expected $O((\Delta/\delta)^2)$-factor approximation, where Δ and δ are respectively the maximum and minimum degrees of the graph.

Our main technical contribution is to show that the randomized greedy MIS algorithm can be implemented in random graphs using $\tilde{O}(n)$ messages and in $O(\log^2 n)$ rounds (where the first bound holds with high probability). This implies constant-factor distributed approximation for MaxIS, MVC, and MDS within the same bounds. We note that while random graphs (above the connectivity threshold) have low diameter (i.e., $O(\log n)$), our $\tilde{O}(n)$ upper bounds are not exclusively due to this property. To compare, we point out that all of the lower bound graphs we construct for quadratic lower bounds for approximation algorithms have constant diameter.

For lack of space, many details and full proofs are deferred to the full paper [28].

[5] Since we show high probability bounds on $G(n, p)$ for every p above the connectivity threshold, one can interpret bounds on random graphs in a deterministic manner as applying to all graphs (of every edge density), except for a vanishingly small fraction. For example, $G(n, 1/2)$ is a uniform distribution on all graphs of size n and our bounds show that almost all graphs admit $\tilde{O}(n)$ message algorithms for the four problems.

1.3 Related Work

Significant progress has been made in understanding and improving the *round* complexity of fundamental "local" problems such as MIS, maximal matching, $(\Delta + 1)$-coloring, and ruling sets (see e.g., [10, 11, 13, 12, 20, 40, 37, 76, 38, 47, 15, 57, 66, 67]) in both the LOCAL and CONGEST models. This research on "local" problems has nice connections to distributed approximation for graph optimization problems and more recently this has become a highly active area of research. This line of research includes round complexity upper bounds for constant-factor and $(1 - \epsilon)$-factor approximation for MaxM [13, 8, 33, 34, 59, 64], constant-factor approximations for MVC [9, 59] and logarithmic-approximations for MDS [17, 52, 24, 39, 59, 68]. It also includes round complexity lower bounds for solving MVC, MDS, and MaxIS, approximately [59, 58, 30] as well as exactly [5, 18].

We compare and contrast only those results in communication complexity that are relevant to our work. The classical 2-party communication complexity where two parties communicate via an (asynchronous) link has been studied extensively for computing various boolean functions, including equality, set disjointness etc. See the books of [74, 60] for a detailed treatment. The work of Tiwari [77] studies the 2-party communication complexity where the two players are connected by an arbitrary network. The network is assumed to be asynchronous. Tiwari shows that the lower bound on the communication complexity of *deterministic* protocols in a n-node network can be $\tilde{\Omega}(n)$ times the standard 2-party communication complexity (where the two players are connected by a direct link). The high-level idea of Tiwari's lower bound is relating the communication complexity in a network to that of a single-link setting by arguing that the two players have to communicate across several vertex-disjoint cuts. The "stretching technique" we use to obtain cubic lower bounds uses similar ideas, but we make this work even for *randomized algorithms*, and furthermore, we also circumvent the time-encoding trick mentioned earlier, so that our bounds also hold for synchronous algorithms.

Another work that is relevant to ours is that of Chattopadhyay, Radhakrishnan, and Rudra [22] who study multi-party communication complexity where the k players are connected by an arbitrary network (see also related follow-up works [21, 23]). This work builds on the earlier work of Woodruff and Zhang [80] who study the same problems, but under the assumption that the network is a clique. In these works, in the context of graph problems, the input graph – which is *different* from the communication network (which also has k nodes) – is *edge-partitioned* among the k players and the goal is to compute some property of the input graph, e.g., whether it is connected or bipartite etc. Chattopadhyay et al. show that the lower bounds on the multi-party communication complexity of such problems on an arbitrary network connecting the k players is at least $\Omega(k)$ times the same complexity when the players are connected by a clique. They also exploit communication over several disjoint cuts to show their stronger lower bounds. While their results hold also for randomized protocols (unlike the results of Tiwari [77]), their results do not apply to the synchronous setting. It is easy to show that in this setting, one can solve all their problems in $O(m)$ messages, where m is the number of edges of the communication network. It is important to stress that the above results hold only when the input graph is *edge partitioned*. Providing the input graph using a vertex partition is closer to the distributed computing setting where one can associate vertices of the input graph (and their incident edges) to players. Indeed this is the assumption used in distributed computing models such as the congested clique [48, 49, 63, 27] and k-machine models [6, 55, 71, 7]. Generally, lower bounds in the vertex partition setting are harder to show and in fact, Drucker, Kuhn, and Oshman [27] prove that showing non-trivial lower bounds in the congested clique model will imply breakthrough circuit complexity lower bounds.

2 Tight Cubic Bounds for Exact Computations

In this section we present near cubic, i.e. $\tilde{\Omega}(n^3)$ lower bounds on the message complexity of KT_0 CONGEST algorithms that compute exact solutions to MVC and MaxIS in Section 2.1. Due to space restrictions we defer the cubic lower bound for computing an exact solution to MDS to the full version [28].

We first present a generic framework for proving message complexity lower bounds in KT_0 CONGEST model by reduction from 2-party communication complexity lower bounds. We begin by defining a lower bound graph family which we call an *ℓ-separated family of lower bound graphs*. Definition 1 is a generalization of the lower bound graph family defined in [18] which is used in to obtain round complexity lower bounds in the CONGEST model. In particular, the family defined in [18] is a 2-separated family of lower bound graphs (i.e. they only consider $\ell = 2$).

▶ **Definition 1** (ℓ-Separated Family of Lower Bound Graphs). *Let $f : X \times Y \to \{\text{TRUE}, \text{FALSE}\}$ be a function and P be a graph predicate. For an integer $\ell > 1$, a family of graphs $\{G_{x,y} = (V, E_{x,y}) \mid x \in X, y \in Y\}$ is said to be an* ℓ-separated family of lower bound graphs w.r.t. f and P *if V can be partitioned into ℓ disjoint and non-empty subsets $V_1, V_2, \ldots, V_{\ell-1}, V_\ell$ such that the following properties hold:*

1. *Only the existence or the weight of edges in $V_1 \times V_1$ depend on x;*
2. *Only the existence or the weight of edges in $V_\ell \times V_\ell$ depend on y;*
3. *For all $1 \leq i \leq \ell$, the vertices in V_i are only connected to vertices in $V_{i-1} \cup V_i \cup V_{i+1}$ (where $V_0 = V_{\ell+1} = \emptyset$).*
4. *$G_{x,y}$ satisfies the predicate P iff $f(x, y) = \text{TRUE}$.*

We will now show a theorem (see Theorem 3) which says that the existence of an n-vertex ℓ-separated family of lower bound graphs w.r.t. f and P implies a lower bound of roughly $\ell \cdot CC(f)$ on the message complexity of a KT_0 CONGEST algorithm for deciding P, where $CC(f)$ is the 2-party communication complexity of the function f. We are ignoring many technical details in the previous statement for the sake of intuition, and we will spend the rest of the section adding these details. Note that $CC(f)$ is trivially bounded by $O(n^2)$ since $f(x, y)$ can be decided by evaluating P on the n-vertex graph $G_{x,y}$, which can be represented using n^2 bits. Therefore, the extra ℓ factor crucially allows us to prove $\omega(n^2)$ lower bounds on message complexity.

We will prove Theorem 3 by efficiently simulating a CONGEST algorithm in the 2-party communication complexity model. In the standard 2-party model, there are two entities, usually called Alice and Bob. Alice has an input $x \in X$, unknown to Bob, and Bob has an input $y \in Y$, unknown to Alice. They wish to collaboratively compute a function $f(x, y)$ by following an agreed-upon protocol Π, which can be possibly randomized with error probability ε. The *communication complexity* of this protocol $CC(\Pi)$ is the number of bits communicated by the two parties for the worst-case choice of inputs $x \in X$ and $y \in Y$. The *deterministic communication complexity* of the function f, denoted as $CC^{\text{det}}(f)$ is the minimum communication complexity of the deterministic protocol Π that correctly computes f. And the *randomized ε-error communication complexity* of the function f, denoted as $CC^{\text{rand}}_{\varepsilon}(f)$ is the minimum communication complexity of the randomized protocol Π that correctly computes f with error probability at most ε. It is important to notice that this simple model is inherently asynchronous, since it does not provide the two parties with a common clock.

Since the CONGEST model is synchronous, it is helpful to first simulate the CONGEST algorithm in the *synchronous 2-party model*, where the two parties also have a common clock. The time interval between two consecutive clock ticks is called a round. The computation proceeds in rounds: at the beginning of each synchronous round, (1) both parties send (possibly different) messages to each other, (2) both parties then receive the messages sent to it in the same round, and (3) both parties perform local computation, which will determine the messages it will send in the next round. The synchronous communication complexity $SCC(\Pi)$ of an r-round protocol Π is the total number of bits sent by the two parties to compute $f(x,y)$ for the worst-case choice of inputs $x \in X$ and $y \in Y$. Note that if Alice (or Bob) decides to not send a message in a particular round, it does not contribute to the communication complexity, but Bob (or Alice) still receives some information, i.e., the fact that Alice (or Bob) chose to remain silent in this round. The *deterministic r-round synchronous communication complexity* of the function f, denoted as $SCC_r^{\mathrm{det}}(f)$ is the minimum communication complexity of an r-round deterministic protocol Π that correctly computes f. And the *randomized ε-error r-round synchronous communication complexity* of the function f, denoted as $SCC_{r,\varepsilon}^{\mathrm{rand}}(f)$ is the minimum communication complexity of the r-round randomized protocol Π that correctly computes f with error probability at most ε.

We use a known relation between $CC^{\mathrm{det}}(f)$ and $SCC_r^{\mathrm{det}}(f)$, and between $CC_\varepsilon^{\mathrm{rand}}(f)$ and $SCC_{r,\varepsilon}^{\mathrm{rand}}(f)$. To do so, we use the Synchronous Simulation Theorem (SST) from [69] to convert the synchronous 2-party protocol into an asynchronous 2-party protocol. Note that although [69] considers more than two parties, it also applies to the 2-party communication complexity setting. The below Lemma 2 is a simplified restatement of SST (Theorem 2 in [69]), obtained by setting the number of parties to 2 in both the synchronous and asynchronous models.

▶ **Lemma 2** (Theorem 2 in [69]). *Let $f : X \times Y \to \{\mathrm{TRUE}, \mathrm{FALSE}\}$ be a function that requires $CC_\varepsilon^{\mathrm{rand}}(f)$ bits for ε-error randomized protocols in the asynchronous 2-party communication complexity model, and $SCC_{\varepsilon,r}^{\mathrm{rand}}(f)$ bits for ε-error randomized, r-round protocols in the synchronous 2-party communication complexity model. The following relation holds (also for the deterministic setting),*

$$SCC_{\varepsilon,r}^{\mathrm{rand}}(f) = \Omega\left(\frac{CC_\varepsilon^{\mathrm{rand}}(f)}{1+\log r}\right).$$

We are now ready to state and prove Theorem 3. We use the existence of an ℓ-separated family of lower bound graphs w.r.t. f and P to show that a KT_0 $\mathsf{CONGEST}$ algorithm that uses r rounds and M messages implies a synchronous randomized 2-party protocol that uses $O(r)$ rounds and roughly $O(M/\ell)$ messages. In other words, the existence of the family implies that $SCC_{\varepsilon,r}^{\mathrm{rand}}(f) = O(M/\ell)$. Therefore any lower bound on the synchronous communication complexity gives a lower bound on M that is a factor ℓ larger. The synchronous 2-party protocol is pretty straightforward: Alice and Bob agree on a cut (V_A, V_B) where $V_A = V_1, \ldots, V_i$ and $V_B = V_{i+1}, \ldots, V_\ell$, where i is chosen uniformly at random in $\{1, \ldots, \ell-1\}$, and they simulate the $\mathsf{CONGEST}$ algorithm, one round at a time, by exchanging the messages sent across this cut. Then we use Lemma 2 to turn this into an asynchronous randomized 2-party protocol.

▶ **Theorem 3.** *Fix a function $f : X \times Y \to \{\mathrm{TRUE}, \mathrm{FALSE}\}$, a predicate P, a constant $0 < \delta < 1$, and a positive integer $\ell > 1$. Suppose there exists an ℓ-separated family of lower bound graphs $\{G_{x,y} = (V, E_{x,y}) \mid x \in X, y \in Y\}$ w.r.t. f and P. Then any r-round deterministic (or randomized with error probability at most constant $0 < \varepsilon < 1$) algorithm for deciding P in the KT_0 $\mathsf{CONGEST}$ model has message complexity*

$$M = \Omega\left(\frac{(\ell-1)}{\log|V|} \cdot \frac{CC_{\delta+\varepsilon}^{\mathrm{rand}}(f)}{(1+\log r)} - \frac{\ell \log \ell}{\log |V|}\right).$$

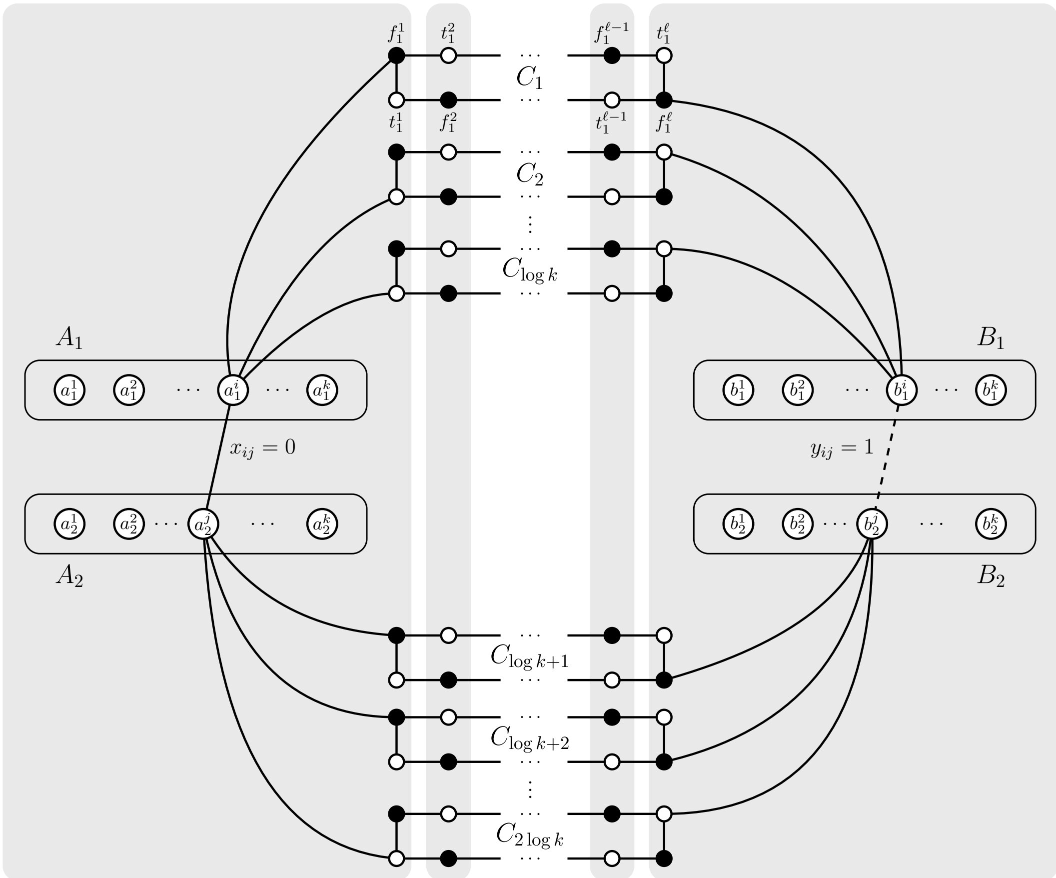

Figure 1 Illustration of one graph $G_{x,y}$ from the ℓ-separated lower bound graph family we defined to show a cubic message lower bound for exact MVC. Many edges are omitted for the sake of clarity. The gray boxes from left to right denote the sets $V_1, V_2, \dots, V_\ell$ defined in proof of Theorem 5.

2.1 Cubic Lower Bounds for Exact MVC and MaxIS

We define an ℓ-separated lower bound graph family $\{G_{x,y} \mid x \in \{0,1\}^{k^2}, y \in \{0,1\}^{k^2}\}$ w.r.t. f = SetDisjointness and predicate P which can be decided by computing the MVC (we will describe P more formally later). The SetDisjointness function is defined as: SetDisjointness(x, y) = FALSE iff there exists an index i such that $x_i = y_i = 1$. We will assume k is a power of 2 so that $\log k$ is an integer. Note that this construction is a generalization of the MVC construction in [18]. More precisely, their construction can be directly obtained from ours by setting $\ell = 2$.

For positive integer k, fix $x, y \in \{0,1\}^{k^2}$. We define the graph $G_{x,y}$ as follows. The vertex set of $G_{x,y}$ is

$$A_1 \cup A_2 \cup B_1 \cup B_2 \cup C_1 \cup C_2 \cup \cdots \cup C_{2\log k}$$

where $A_1 = \{a_1^i \mid 1 \le i \le k\}$, $A_2 = \{a_2^i \mid 1 \le i \le k\}$, $B_1 = \{b_1^i \mid 1 \le i \le k\}$ and $B_2 = \{b_2^i \mid 1 \le i \le k\}$ are called the "row vertices". And $C_i = \{t_i^j, f_i^j \mid 1 \le j \le \ell\}$ for all $1 \le i \le 2\log k$ are called the "bit gadget vertices". Therefore, $G_{x,y}$ has $4k + 4\ell \log k$ vertices.

We now describe the edges of $G_{x,y}$. The vertices in the sets A_1 form a clique, and so do the vertices in the sets A_2, B_1, B_2. Assuming ℓ is even, the vertices in the set C_i form a cycle with the following order:

$$f_i^1, t_i^2, f_i^3, t_i^4, f_i^5, \dots, t_i^{\ell-2}, f_i^{\ell-1}, t_i^\ell, f_i^\ell, t_i^{\ell-1}, f_i^{\ell-2}, \dots, t_i^5, f_i^4, t_i^3, f_i^2, t_i^1, f_i^1$$

This cycle is also illustrated for the set C_1 in Figure 1. Each vertex $a_1^i \in A_1$ is connected to bit gadgets $C_1, \ldots, C_{\log k}$ according to the binary representation of the index i. In particular, a_1^i is connected to t_h^1 if bit position h of index i is 1 and to f_h^1 if bit position h of index i is 0. Similarly, each vertex $b_1^i \in B_1$ is connected to t_h^ℓ if the bit position h of index i is 1 and to f_h^ℓ if the bit position h of index i is 0. The vertices in A_2 and B_2 are connected to $C_{\log k+1}, \ldots, C_{2\log k}$ in a symmetric manner. To be explicit, each vertex $a_2^j \in A_2$ is connected to $t_{h+\log k}^1$ if bit position h of index j is 1 and to $f_{h+\log k}^1$ if bit position h of index j is 0. Similarly, each vertex $b_2^j \in B_2$ is connected to $t_{h+\log k}^\ell$ if bit position h of index j is 1 and to $f_{h+\log k}^\ell$ if bit position h of index j is 0.

This completes the "fixed" edges in the graph, i.e., the edges that do not depend on bit vectors x and y. The edges between A_1 and A_2 depend on x as follows: $\{a_1^i, a_2^j\}$ is an edge iff $x_{ij} = 0$. The edges between B_1 and B_2 depend on y as follows: $\{b_1^i, b_2^j\}$ is an edge iff $y_{ij} = 0$. The construction is illustrated in Figure 1.

▶ **Lemma 4.** *For $x, y \in \{0,1\}^{k^2}$, if* SetDisjointness$(x, y) =$ TRUE *then the MVC of $G_{x,y}$ has size at least $4k + 2\ell \log k - 3$, and if* SetDisjointness$(x, y) =$ FALSE *then the MVC of $G_{x,y}$ has size exactly $4k + 2\ell \log k - 4$.*

▶ **Theorem 5.** *For any $0 < \varepsilon < 1/6$, any ε-error randomized Monte-Carlo r-round* KT_0 $\mathsf{CONGEST}$ *algorithm that computes an MVC or MaxIS on an n-vertex communication graph has message complexity $\tilde{\Omega}(n^3/(1 + \log r))$.*

3 Tight Quadratic Bounds for Approximate Computations

We start this section by proving $\tilde{O}(n^2)$ message complexity upper bounds for $(1 \pm \epsilon)$-approximations for all four problems, MaxM, MVC, MDS, and MaxIS. These results serve as a contrast to the cubic lower bounds for exact computation, shown in the previous section. We then show that these upper bounds are tight, by showing that $\tilde{\Omega}(n^2)$ messages are required for *constant-factor* approximation algorithms for all four problems. For MaxM and MVC, these bounds hold for any constant-factor approximation, whereas for MDS and MaxIS they hold for any approximation factor better than 5/4 and 1/2 respectively. These lower bounds hold even in the LOCAL model (in which messages can be arbitrarily large) and they apply not just to polynomial-round algorithms, but to algorithms that take arbitrarily many rounds.

3.1 Quadratic Upper Bounds for Approximate Computations

Notation. For any graph G, let $\alpha(G)$ denote the size of the largest independent set in G. For any node v in G and integer $r \geq 0$, let $B_r(v)$ denote the set of all nodes in G at distance at most r from v.

Consider the following sequential algorithm, called the "ball growing" algorithm in [40] that gives a $1/(1+\epsilon)$-approximate solution for MaxIS, for any constant $\epsilon > 0$. Let I denote the solution constructed by the algorithm; initialize I to $\emptyset$. Pick an arbitrary vertex v_1 and find a smallest radius r_1 such that $\alpha(G[B_{r_1+1}(v_1)]) \leq (1+\epsilon) \cdot \alpha(G[B_{r_1}(v_1)])$. Add a maximum-sized independent set of $G[B_{r_1}(v_1)]$ to I and delete $G[B_{r_1+1}(v_1)]$ from the graph. This completes the first iteration of the algorithm. For the second iteration, find an arbitrary vertex v_2 in the graph that remains and repeat an iteration of "ball growing" until a radius r_2 is found. Continue these iterations until the graph becomes empty. Note that each radius $r_i = O(\log n/\log(1+\epsilon))$ and the constructed solution I is an independent set of G. Furthermore, a simple argument (see [40]) shows that $|I| \geq 1/(1+\epsilon) \cdot \alpha(G)$. Note that the local computations in this algorithm do not run in polynomial-time algorithm because each v_i needs to compute the exact maximum-sized independent set in $B_r(v_i)$ for different values of r.

▶ **Lemma 6.** *There is a deterministic $1/(1+\epsilon)$-approximation algorithm for MaxIS in the KT_0 $\mathsf{CONGEST}$ model that uses $\tilde{O}(n^2/\epsilon)$ messages and runs in $O(\text{poly}(m+n))$ rounds.*

While this result is described for MaxIS, the authors of [40] also show that this "ball growing" algorithm is able to produce a $(1+\epsilon)$-approximation for MDS. Furthermore, [40] claims that this approach produces a $(1+\epsilon)$ or a $1/(1+\epsilon)$-approximation for any problem that can be expressed as certain type of packing or covering integer linear program. In addition to MaxIS and MDS, this framework also includes MaxM and MVC. So we get the following theorem.[6]

▶ **Theorem 7.** *There is a deterministic $(1-\epsilon)$-approximation algorithm for MaxIS and MaxM in the KT_0 $\mathsf{CONGEST}$ model that uses $\tilde{O}(n^2/\epsilon)$ messages and runs in $O(\text{poly}(m+n))$ rounds. Similarly, there is a a deterministic $(1+\epsilon)$-approximation algorithm for MDS and MVC in the KT_0 $\mathsf{CONGEST}$ model that uses $\tilde{O}(n^2/\epsilon)$ messages and runs in $O(\text{poly}(m+n))$ rounds.*

3.2 Unconditional Quadratic Lower Bound for MaxM Approximation

Next, we show that $\tilde{\Omega}(n^2)$ messages are required for computing a constant-factor approximation for MaxM.

▶ **Theorem 8.** *Consider any randomized algorithm for the MaxM problem in the KT_0 LOCAL model. Assume that every matched edge is output by at least one of its endpoints; either by outputting a port number or the ID of the corresponding neighbor. If, for some $\epsilon \in \left(\frac{1}{n^{1/3}}, 1\right)$, the algorithm sends at most $\frac{\epsilon^3 n^2}{7^3 \cdot 8} = O\left(\epsilon^3 n^2\right)$ messages with probability at least $1 - \frac{\epsilon}{7}$, then there exists a graph on $2n$ nodes such that the approximation ratio is at most ϵ in expectation.*

In the remainder of this section, we give a class of graphs on which finding a large matching is hard, and then we state the details of the assumed port numbering model and discuss the output specification of a given matching algorithm in this setting. We make use of these definitions when proving Theorem 8.

The Lower Bound Graph. Let $\gamma = \frac{\epsilon}{7}$. We consider the following $2n$-node graph G consisting of vertex sets A and B, where $A = \{u_1, \ldots, u_n\}$ and $B = \{v_1, \ldots, v_n\}$. We further partition A into C_A and N_A such that $N_A = \{u_1, \ldots, u_{n-\lfloor \gamma n \rfloor}\}$ and $C_A = A \setminus N_A$. Each node in N_A is connected to all nodes in C_A, whereas the nodes in C_A form a clique. Analogously, we define C_B and N_B, and the edges between them. In addition, we add the set of *valuable edges* $\{\{u_1, v_1\}, \ldots, \{u_n, v_n\}\}$. Figure 2 depicts an example of this construction. Notice that the set of valuable edges corresponds to a perfect matching of size n and hence forms an optimum solution for the maximum matching problem.

[6] In this theorem statement we use the more convenient $(1-\epsilon)$ rather than $1/(1+\epsilon)$. This is justified by the fact that $1/(1+\epsilon)$ can be written as $1-\epsilon'$, where $\epsilon/2 \leq \epsilon' \leq \epsilon$.

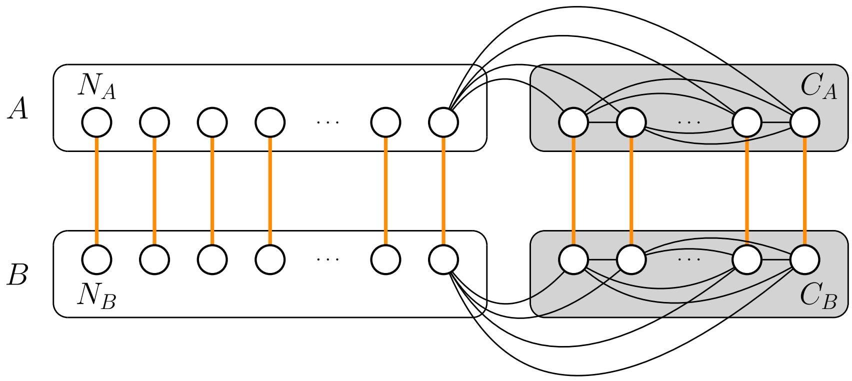

Figure 2 The lower bound construction for proving Theorem 8. There are $2n$ nodes in total equally partitioned into sets A and B. Each one of the grey-shaded areas contains $\lfloor \gamma n \rfloor$ nodes that form the cliques C_A and C_B, respectively. Every node in $N_A = A \setminus C_A$ has an edge to all nodes in C_A, and the nodes in N_B and C_B are connected similarly. The thick orange edges are the valuable edges that form a perfect matching.

The Port Numbering Model. We consider the standard port numbering model, where the incident edges of a node u are numbered $1, \ldots, \deg(u)$. This means that, in order to send a message across the edge $\{u, v\}$, node u would need to send over some port $p_{(u,v)}$, whereas node v would need to use port $p_{(v,u)}$, for some (possibly distinct) integers $p_{(u,v)} \in [\deg(u)]$[7] and $p_{(v,u)} \in [\deg(v)]$.

We say that the port $p_{(u,v)}$ is *used* if u sends a message over p or receives a message that was sent on $p_{(v,u)}$; otherwise we say that it is *unused*. A crucial property of the KT_0 assumption is that, initially, a node u does not know that it is connected to v via $p_{(u,v)}$. However, we assume that u learns that $p_{(u,v)}$ connects to v upon receiving a message directly from v. To obtain a concrete lower bound graph, we fix the node IDs to correspond to a uniformly random permutation of $[2n]$, and, for each node u, we choose an assignment of its incident ports to the corresponding endpoints by independently and uniformly selecting a random permutation of the set $[\deg(u)]$.

There are two standard ways how an algorithm may output a matching in KT_0. The first possibility is that at least one of the endpoints of each matched edge outputs the corresponding port number. The second one is that a node u outputs the ID of some neighbor v to indicate that $\{u, v\}$ is in the matching. We point out that our lower bound result holds under either output assumption.

Proof of Theorem 8. Consider any algorithm that satisfies the premise of the theorem. Suppose that it sends at most $\frac{\gamma^3 n^2}{8}$ messages with probability at least $1 - \gamma$, and let `Sparse` denote the event that this happens.

▶ **Lemma 9.** *Let $J \subseteq [n - \lfloor \gamma n \rfloor]$ be the set of indices such that, for all $i \in J$, the edge $\{u_i, v_i\}$ is not part of the computed matching. Then* $\mathbf{E}\left[|J| \mid \mathtt{Sparse}\right] \geq (1 - 5\gamma)\, n$.

[7] We use the standard notation $[m] := \{1, \ldots, m\}$.

We are now ready to complete the proof of Theorem 8. Let M be the matching computed by the algorithm and recall that the perfect matching has size n. We have

$$\begin{aligned}
\mathbf{E}[|M|] &\leq \mathbf{E}[|M| \mid \texttt{Sparse}] + n \cdot \Pr[\neg\texttt{Sparse}] \\
&\leq |C_A| + |N_A| - |J| + \gamma n \\
&\leq 2\gamma n + (n - \gamma n + 1) - (1 - 5\gamma)\, n \qquad \text{(by Lemma 9)} \\
&\leq 7\gamma n,
\end{aligned}$$

which implies the claimed upper bound on the approximation ratio since $\gamma = \frac{\epsilon}{7}$ and the maximum matching has size n.

3.3 Unconditional Quadratic Lower Bound for MDS Approximation

A well known and powerful tool for proving lower bounds in KT_0 is the notion of a port preserving crossing which is used to prove message complexity lower bounds in [4]. We use this tool to prove lower bounds for MDS approximation in this section, and also for MVC and MaxIS approximation in Sections 3.4 and 3.5.

▶ **Definition 10** (Port-Preserving Crossing). *Let H be an arbitrary graph with two edges $e = \{u, v\}$ and $e' = \{u', v'\}$ for distinct nodes u, u', v, v'. Let e be connected to u at port p and to v at port q, and similarly let e' be connected to u' at port p' and to v' at port q'. The port preserving crossing of e and e' is the graph $H_{e,e'}$ which is obtained by removing the edges e, e' from H and adding the edge $\{u, u'\}$ connected to u at port p and to u' at port p' and the edge $\{v, v'\}$ connected to v at port q and to v' at port q'.*

▶ **Theorem 11.** *Let $\mathcal{A}$ be a deterministic KT_0 LOCAL algorithm. Let H and $H_{e,e'}$ be the two graphs described in Definition 10 with the same ID assignment. If no messages pass over e and e', $\mathcal{A}$ behaves identically on the graphs H and $H_{e,e'}$.*

For the approximate MDS lower bound, we define a family $\{G_{x,y} \mid x \in \{0,1\}^{n^2}, y \in \{0,1\}^{n^2}\}$ of lower bound graphs. This construction is inspired by the construction in [5], but with a critical difference, that we highlight below. For positive integer n, let $x, y \in \{0,1\}^{n^2}$. We will now define a graph $G_{x,y}$ as follows. The vertex set of $G_{x,y}$ is

$$A_1 \cup A_2 \cup B_1 \cup B_2 \cup C_1 \cup C_2 \cup \{a^*, b^*\}$$

where $A_1 = \{a_1^i \mid 1 \leq i \leq n\}$, $A_2 = \{a_2^i \mid 1 \leq i \leq n\}$, $B_1 = \{b_1^i \mid 1 \leq i \leq n\}$, $B_2 = \{b_2^i \mid 1 \leq i \leq n\}$, $C_1 = \{c_1^i \mid 1 \leq i \leq n\}$, and $C_2 = \{c_2^i \mid 1 \leq i \leq n\}$. Therefore, $G_{x,y}$ has $6n + 2$ vertices.

We now describe the edges of $G_{x,y}$. The vertices in C_1 (and C_2) form an n-vertex clique. Each vertex $c_1^i \in C_1$ is connected to all vertices a_1^j, $j \neq i$ and to all vertices b_1^j, $j \neq i$. Similarly, each vertex $c_2^i \in C_2$ is connected to all vertices a_2^j, $j \neq i$ and to all vertices b_2^j, $j \neq i$. Vertex a^* is connected to all vertices in A_1 and vertex b^* is connected to all vertices in B_1. This completes the "fixed" edges in the graph, i.e., the edges that do not depend on bit vectors x and y. The edges between A_1 and A_2 depend on x as follows: $\{a_1^i, a_2^j\}$ is an edge iff $x_{ij} = 1$. The edges between B_1 and B_2 depend on y as follows: $\{b_1^i, b_2^j\}$ is an edge iff $y_{ij} = 1$. The construction is illustrated in Figure 3.

In [5], the authors use a similar construction to obtain an $\Omega(n^2)$ *round* lower bound for *exact* MDS. Their construction critically depends on the existence of a small cut (with $O(\text{poly}(\log n))$ edges) across the partition (V_A, V_B) of the vertex set, where $V_A \supseteq A_1 \cup A_2$ and $V_B \supseteq B_1 \cup B_2$. For a message complexity lower bound, we don't need a small cut. In

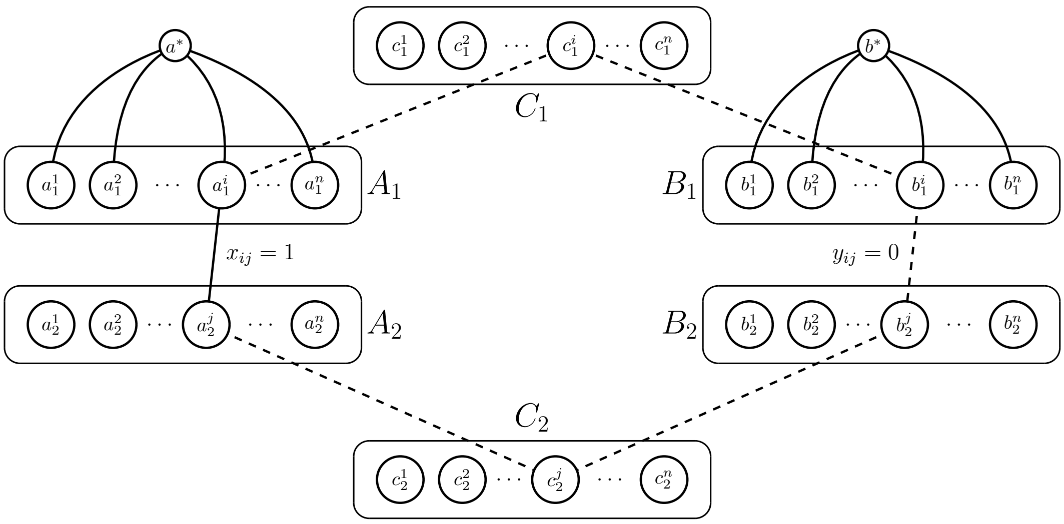

Figure 3 Illustration of the graph $G_{x,y}$ used in the MDS lower bound proof. For clarity, many edges are not shown. Dashed line segments represent the absence of edges.

fact, it can be verified that $\Omega(n^2)$ edges connect any $V_A \supseteq A_1 \cup A_2$ and $V_B \supseteq B_1 \cup B_2$ in our lower bound graph. This flexibility plays a key role in our ability to force a constant-sized dominating set in the lower bound graph and obtain a relatively large gap in the MDS sizes between different types of instances, as shown in the following lemma.

▶ **Lemma 12.** *If $\exists(i,j)$ such that $x_{ij} = y_{ij} = 1$ then $G_{x,y}$ has a dominating set of size at most 4. If $\nexists(i,j)$ such that $x_{ij} = y_{ij} = 1$ then $G_{x,y}$ has a dominating set of size at least 5.*

We will pick a member G of this family such that the subgraph $G[A_1 \cup A_2]$ is a fixed $n/2$-regular bipartite graph where for all $1 \le i \le n$, a_1^i is connected to a_2^j for all $j = i, i+1, \ldots, (i + n/2 - 1)$ (if j becomes larger than n, we wrap around back to 1). And the subgraph $G[B_1 \cup B_2]$ is the complement of $G[A_1 \cup A_2]$. We claim that that this graph has a dominating set of size at least 5. This is because G can also be viewed as $G_{x,y}$ for some string $x \in \{0,1\}^{n^2}$ and $y = \overline{x}$. Since there is no index (i,j) such that $x_{ij} = y_{ij} = 1$, by Lemma 12, G has a dominating set of size at least 5.

Let $\mathcal{A}$ be a deterministic dominating set algorithm in the KT_0 LOCAL model that uses $o(n^2)$ messages. Note that $\mathcal{A}$ outputs a dominating set of size at least 5 on G. Then, there exists an edge $e = \{a_1^i, a_2^j\}$ in G, $a_1^i \in A_1$, $a_2^j \in A_2$ such that no message passes through this edge during the execution of algorithm $\mathcal{A}$. Let $\overline{N}(a_1^i) \subseteq A_2$ be the subset of A_2 containing vertices that are not neighbors of a_1^i. Similarly, let $\overline{N}(a_2^j) \subseteq A_1$ be the subset of A_1 containing vertices that are not neighbors of a_2^j. Note that $|\overline{N}(a_1^i)| = |\overline{N}(a_2^j)| = \frac{n}{2}$, and there are $\Theta(n^2)$ edges in G between $\overline{N}(a_1^i)$ and $\overline{N}(a_2^j)$. Since $\mathcal{A}$ uses $o(n^2)$ messages, there is an edge $e' = \{a_1^p, a_2^q\}$, $a_1^p \in \overline{N}(a_2^j)$, $a_2^q \in \overline{N}(a_1^i)$, such that no message passes over edge e' during the execution of $\mathcal{A}$.

Let $G_{e,e'}$ be the graph obtained from G by crossing the edges $e = \{a_1^i, a_2^j\}$ and $e' = \{a_2^q, a_1^p\}$ according to Definition 10. In particular, e and e' are replaced by edges $\{a_1^i, a_2^q\}$ and $\{a_1^p, a_2^j\}$, with the port-numbering preserved. Note that by Theorem 11, algorithm $\mathcal{A}$ behaves identically in G and $G_{e,e'}$ and so $\mathcal{A}$ outputs a dominating set of size at least 5 for $G_{e,e'}$ also. However, $G_{e,e'}$ has a dominating set of size 4. This follows from Lemma 12, and the fact that $\{b_1^i, b_2^q\}$ (and $\{b_1^p, b_2^j\}$) is an edge in $G_{e,e'}$.

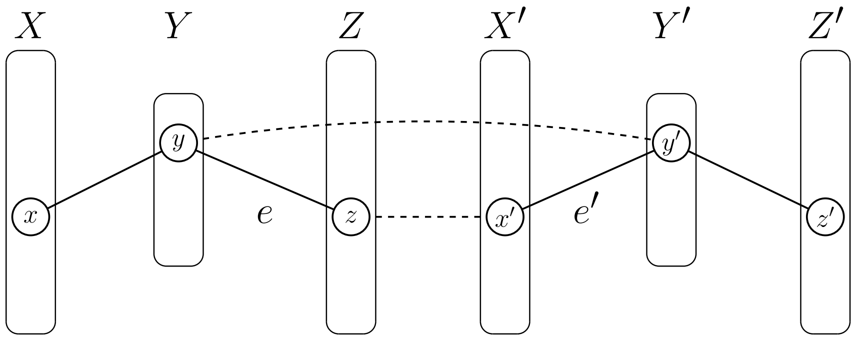

Figure 4 Illustration of the graphs $G \cup G'$, and $G_{e,e'}$ used in the MVC lower bound proof.

This means that $\mathcal{A}$ outputs a dominating set of size at least 5 for a graph with a dominating set of size at most 4. Thus, a $(5/4 - \epsilon)$-approximation, $\epsilon > 0$, for MDS requires $\Omega(n^2)$ messages.

▶ **Theorem 13.** *For any constant $\epsilon > 0$, any deterministic* KT_0 LOCAL *algorithm $\mathcal{A}$ that computes a $(5/4-\epsilon)$-approximation of MDS on n-vertex graphs has $\Omega(n^2)$ message complexity.*

▶ **Theorem 14.** *For any constant $\epsilon > 0$, any randomized Monte-Carlo* KT_0 LOCAL *algorithm $\mathcal{A}$ that computes a $(5/4 - \epsilon)$-approximation of MDS on n-vertex graphs with constant error probability $\delta < 1/2$ has $\Omega(n^2)$ message complexity.*

3.4 Unconditional Quadratic Lower Bound for MVC Approximation

Here we show that for any parameter $c \geq 1$, any randomized KT_0 LOCAL c-approximation algorithm for the minimum vertex cover (MVC) problem uses $\Omega(n^2/c)$ messages for some n-vertex graph.

Define a graph $G = (V, E)$ where V is divided into three parts X, Y, Z such that $|X| = |Z| = t$ and $|Y| = t/(4c)$. We add all possible edges between X and Y and all possible edges between Y and Z. We then add a copy $G' = (V', E')$ of G (where the three parts of V' are X', Y' and Z'). Note that G and G' have exactly $t^2/(2c)$ edges each. We will call $G \cup G'$ the *base graph*, using which we create the lower bound graphs. Let $n = |V \cup V'| = 4t + t/(2c)$, thus $t = 2cn/(8c + 1)$.

Each node in the base graph is assigned a unique ID in the range $[1, n]$, and the ports of each node are also assigned in some arbitrary way.

We create a *crossed graph* $G_{e,e'}$ by starting with the base graph $G \cup G'$ and then replacing edges $e = \{y, z\}$ and $e' = \{x', y'\}$ with edges $\{y, y'\}$ and $\{z, x'\}$ in a port preserving manner according to Definition 10. The base graph and crossed graph are illustrated in Figure 4. Note that the ID and port assignments to the nodes remain unchanged in the crossed graph.

The base and crossed graphs are similar to the lower bound graphs used in [67] to prove message complexity lower bounds for MIS and $(\Delta + 1)$-coloring (here $|Y| = |Y'| = t$). The motivation for shrinking the size of Y (and Y') is to ensure that in any approximate vertex cover of $G \cup G'$, there are lots of vertices that are not in the cover. This is made precise in the following claim.

▷ Claim 15. Any c-approximate vertex cover in $G \cup G'$ has size at most $t/2$.

Now suppose (to obtain a contradiction) that there is a deterministic KT_0 LOCAL algorithm $\mathcal{A}$ that computes a c-approximate vertex cover C in $G \cup G'$ using $o(t^2/c)$ messages.

▷ Claim 16. There exist $y \in Y$, $z \in Z \setminus C$, $x' \in X' \setminus C$ and $y' \in Y'$ such that no message passes over edges $\{y, z\}$ and $\{x', y'\}$ in algorithm $\mathcal{A}$.

▶ **Lemma 17.** *Let $y \in Y$, $z \in Z \setminus C$, $x' \in X' \setminus C$ and $y' \in Y'$ be four nodes such that no message passes over the edges $e = \{y, z\}$ and $e' = \{x', y'\}$ in algorithm $\mathcal{A}$. Then $\mathcal{A}$ cannot compute a correct vertex cover on $G_{e,e'}$.*

Claim 16 implies the existence of e and e' assumed in Lemma 17. Thus $\mathcal{A}$ must use $\Omega(t^2/c)$ messages when run on $G \cup G'$. The following theorem formally states this lower bound.

▶ **Theorem 18.** *For any constant $c \geq 1$, any deterministic KT_0 LOCAL algorithm $\mathcal{A}$ that computes a c-approximation of MVC on n-vertex graphs has $\Omega(n^2/c)$ message complexity.*

We extend this deterministic lower bound to randomized Monte-Carlo algorithms.

▶ **Theorem 19.** *Any randomized Monte-Carlo KT_0 LOCAL algorithm $\mathcal{A}$ computing a maximal matching with constant error probability $0 \leq \delta < 1/8 - o(1)$ where each matched edge is output by at least one of its end points has $\Omega(n^2)$ message complexity.*

3.5 Unconditional Quadratic Lower Bound for MaxIS Approximation

For the MaxIS approximation lower bound we use the same base graph $G \cup G'$ as the MVC lower bound but we set $|Y| = |Y'| = \epsilon t$. The crossed graph $G_{e,e'}$ obtained by crossing the edges e and e' in a port preserving manner according to Definition 10.

▶ **Theorem 20.** *For any constant $\epsilon > 0$, any deterministic KT_0 LOCAL algorithm $\mathcal{A}$ that computes a $(1/2 + \epsilon)$-approximation of MaxIS on n-vertex graphs has $\Omega(n^2)$ message complexity.*

▶ **Theorem 21.** *For any constant $\epsilon > 0$, any randomized Monte-Carlo KT_0 LOCAL algorithm $\mathcal{A}$ that computes a $(1/2 + \epsilon)$-approximation of MaxIS on n-vertex graphs with constant error probability $\delta < \epsilon^2/8 - o(1)$ has $\Omega(n^2)$ message complexity.*

4 Message-Efficient Distributed Approximation Algorithms in Random Graphs

We first provide a message- and round-efficient randomized greedy MIS in $G(n, p)$ random graphs.[8] With it, we give distributed algorithms using only $\tilde{O}(n)$ messages to compute with high probability (w.h.p.) constant-factor approximations for MaxIS, MDS, MVC and MaxM in $G(n, p)$ random graphs.

Randomized Greedy MIS. The algorithm works in $O(\log n)$ phases (where n is known to all nodes initially). Initially, all nodes start undecided. Each phase reduces the number of *undecided nodes* (i.e., not in the MIS nor neighbors of MIS nodes) by half. Phases are split into 15 iterations of $O(\log n)$ rounds, each iteration reducing the number of undecided nodes by a constant factor. In each iteration, we sample $O(\log(n)/p)$ nodes (or take all nodes if there remain less undecided nodes), referred to as *active nodes*. In two rounds, active

[8] These are graphs for which each (possible) edge $e = \{x, y\} \in V^2$ occurs independently with probability $0 < p < 1$ (where p may be a function of n).

nodes know which incident edges are part of the subgraph induced by active nodes. (As the communication graph has maximum degree $O(np)$ w.h.p, this only takes $\tilde{O}(n)$ messages.) After which, active nodes use $O(\log n)$ rounds to run distributed randomized greedy MIS [35] on this induced subgraph. Finally, nodes in the computed MIS use the last round to inform all of their neighbors.

▶ **Theorem 22.** *In $G(n, p)$ random graphs, there exists a distributed implementation (in* KT_0 $\mathsf{CONGEST}$*) of the randomized greedy MIS sequential algorithm that is correct with high probability, takes $O(\log^2 n)$ rounds and uses $\tilde{O}(n)$ messages with high probability.*

Distributed Approximation Problems. Now, we consider random graphs in the connectivity regime.[9] On such graphs (of size n), our distributed randomized greedy MIS algorithm provides an MIS of size $\sigma_n \sim \log_{1/(1-p)} np$ (see [44, 36]). From well-known results regarding random graphs [44, 36, 41, 79, 32], we obtain the approximation results summarized in the lemma below.

▶ **Lemma 23.** *In $G(n, p)$ random graphs (with $p \geq 40(\log n)/n$), $(1/2 - o(1))$-approximate MaxIS, $(1 + o(1))$-approximate MDS and $(2 - o(1))$-approximate MVC can be solved (w.h.p.) with $\tilde{O}(n)$ messages and $O(\log^2 n)$ rounds in* KT_0 $\mathsf{CONGEST}$*. Additionally, for these same graphs, a perfect (or near-perfect) matching can be computed (w.h.p.) with $\tilde{O}(n)$ messages and $\tilde{O}(n)$ rounds in* KT_0 $\mathsf{CONGEST}$*.*

5 Conclusion and Open Problems

In this work, we almost fully quantify the message complexity of four fundamental graph optimization problems – MaxM, MVC, MDS, and MaxIS. These problems represent a spectrum of hardness of approximation in the sequential setting, ranging from MaxM that is exactly solvable in polynomial-time to MaxIS that is hard to approximate even to a $O(n^{1-\epsilon})$-factor for any $\epsilon > 0$. We have shown that $\tilde{\Omega}(n^3)$ messages are needed to solve MVC, MDS, and MaxIS exactly in the KT_0 $\mathsf{CONGEST}$ model. The message complexity of exact MaxM is an intriguing open question and the lower bound technique we use to obtain the cubic bounds for MVC, MDS, and MaxIS, cannot be used for MaxM. Furthermore, there has been recent progress on improving the round complexity of exact MaxM [54], though it is not clear if techniques from this line of work can be used to obtain $o(n^3)$ message algorithms for exact MaxM. Another set of open questions relate to our quadratic lower bounds. For MDS and MaxIS, our lower bounds are for constant-factor approximations, for specific constants. Can these lower bounds be extended to any α-approximation algorithm? Such lower bounds would be a function of the graph size as well as α (similar to our lower bounds for MaxM and MVC). For MDS, such a general lower bound would have to account for the $O(n^{1.5})$ message upper bound for $O(\log \Delta)$-approximation for MDS [43, 45].

References

1 Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Christoph Lenzen. Fooling views: A new lower bound technique for distributed computations under congestion. *Distrib. Comput.*, 33(6):545–559, December 2020. doi:10.1007/s00446-020-00373-4.

2 Yehuda Afek and Eli Gafni. Time and message bounds for election in synchronous and asynchronous complete networks. *SIAM Journal on Computing*, 20(2):376–394, 1991.

[9] For sparse random graphs, $\tilde{O}(m) = \tilde{O}(n)$ messages suffice to solve MaxM, MVC, MDS, and MaxIS (with time encoding).

3 B. Awerbuch and D. Peleg. Sparse partitions. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 503–513 vol.2, 1990. doi:10.1109/FSCS.1990.89571.

4 Baruch Awerbuch, Oded Goldreich, David Peleg, and Ronen Vainish. A trade-off between information and communication in broadcast protocols. *J. ACM*, 37(2):238–256, 1990. doi:10.1145/77600.77618.

5 Nir Bachrach, Keren Censor-Hillel, Michal Dory, Yuval Efron, Dean Leitersdorf, and Ami Paz. Hardness of distributed optimization. In *PODC*, pages 238–247, 2019. doi:10.1145/3293611.3331597.

6 Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Sriram V. Pemmaraju. Near-Optimal Clustering in the k-machine model. In *ICDCN*, pages 15:1–15:10, 2018. doi:10.1145/3154273.3154317.

7 Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Sriram V. Pemmaraju. Near-optimal clustering in the k-machine model. *Theor. Comput. Sci.*, 899:80–97, 2022. doi:10.1016/J.TCS.2021.11.026.

8 Reuven Bar-Yehuda, Keren Censor-Hillel, Mohsen Ghaffari, and Gregory Schwartzman. Distributed approximation of maximum independent set and maximum matching. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, pages 165–174, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3087801.3087806.

9 Reuven Bar-Yehuda, Keren Censor-Hillel, and Gregory Schwartzman. A distributed $(2+\epsilon)$-approximation for vertex cover in $O(\log\Delta/\epsilon\log\log\Delta)$ rounds. *J. ACM*, 64(3):23:1–23:11, 2017. doi:10.1145/3060294.

10 Leonid Barenboim and Michael Elkin. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2013.

11 Leonid Barenboim, Michael Elkin, and Fabian Kuhn. Distributed $(\Delta+1)$-Coloring in Linear (in Δ) Time. *SIAM J. Comput.*, 43(1):72–95, 2014. doi:10.1137/12088848X.

12 Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The Locality of Distributed Symmetry Breaking. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 321–330, 2012. doi:10.1109/FOCS.2012.60.

13 Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The Locality of Distributed Symmetry Breaking. *J. ACM*, 63(3):20:1–20:45, 2016. doi:10.1145/2903137.

14 Ran Ben-Basat, Guy Even, Ken-ichi Kawarabayashi, and Gregory Schwartzman. A Deterministic Distributed 2-Approximation for Weighted Vertex Cover in $O(\log n\log\Delta/\log^2\log\Delta)$ Rounds. In Zvi Lotker and Boaz Patt-Shamir, editors, *Structural Information and Communication Complexity – 25th International Colloquium, SIROCCO 2018*, pages 226–236, 2018. doi:10.1007/978-3-030-01325-7_21.

15 Tushar Bisht, Kishore Kothapalli, and Sriram V. Pemmaraju. Brief announcement: Super-fast t-ruling sets. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 379–381. ACM, 2014. doi:10.1145/2611462.2611512.

16 Béla Bollobás. *Random Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2001. doi:10.1017/CBO9780511814068.

17 Keren Censor-Hillel and Michal Dory. Distributed spanner approximation. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 139–148, 2018. doi:10.1145/3212734.3212758.

18 Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and near-quadratic lower bounds for the CONGEST model. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, pages 10:1–10:16, 2017. doi:10.4230/LIPIcs.DISC.2017.10.

19 Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 94–99, New York, NY, USA, 1983. Association for Computing Machinery. doi:10.1145/800061.808737.

20 Yi-Jun Chang, Wenzheng Li, and Seth Pettie. Distributed $(\Delta+1)$-Coloring via ultrafast graph shattering. *SIAM Journal on Computing*, 49(3):497–539, 2020. doi:10.1137/19M1249527.

21 Arkadev Chattopadhyay, Michael Langberg, Shi Li, and Atri Rudra. Tight network topology dependent bounds on rounds of communication. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2524–2539. SIAM, 2017. doi:10.1137/1.9781611974782.167.

22 Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 631–640, 2014. doi:10.1109/FOCS.2014.73.

23 Arkadev Chattopadhyay and Atri Rudra. The range of topological effects on communication. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming – 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 540–551. Springer, 2015. doi:10.1007/978-3-662-47666-6_43.

24 Janosch Deurer, Fabian Kuhn, and Yannic Maus. Deterministic distributed dominating set approximation in the CONGEST model. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC)*, 2019.

25 Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

26 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 624–633, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591884.

27 Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the Power of the Congested Clique Model. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC '14, pages 367–376, New York, NY, USA, 2014. ACM. doi:10.1145/2611462.2611493.

28 Fabien Dufoulon, Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, and Peter Robinson. The Message Complexity of Distributed Graph Optimization, 2023. doi:10.48550/arXiv.2311.14811.

29 P Ďuriš and Pavel Pudlák. On the communication complexity of planarity. In *Fundamentals of Computation Theory: International Conference FCT'89 Szeged, Hungary, August 21–25, 1989 Proceedings 7*, pages 145–147. Springer, 1989.

30 Yuval Efron, Ofer Grossman, and Seri Khoury. Beyond alice and bob: Improved inapproximability for maximum independent set in congest. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC '20, pages 511–520, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3382734.3405702.

31 M. Elkin. A simple deterministic distributed mst algorithm with near-optimal time and message complexities. *J. ACM*, 67(2), April 2020. doi:10.1145/3380546.

32 P. Erdős and A Rényi. On the existence of a factor of degree one of a connected random graph. *Acta Mathematica Academiae Scientiarum Hungaricae*, 17:359–368, 1966. doi:10.1007/BF01894879.

33 Manuela Fischer. Improved deterministic distributed matching via rounding. *Distributed Computing*, 33, 2020. doi:10.1007/s00446-018-0344-4.

34 Manuela Fischer, Slobodan Mitrović, and Jara Uitto. Deterministic $(1+\epsilon)$-approximate maximum matching with poly$(1/\epsilon)$ passes in the semi-streaming model and beyond. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 248–260, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3520039.

35 Manuela Fischer and Andreas Noever. Tight analysis of parallel randomized greedy MIS. In *SODA*, pages 2152–2160, 2018.

36 Alan Frieze and Colin McDiarmid. Algorithmic theory of random graphs. *Random Structures & Algorithms*, 10(1-2):5–42, 1997. doi:10.1002/(SICI)1098-2418(199701/03)10:1/2<5::AID-RSA2>3.0.CO;2-Z.

37 Mohsen Ghaffari. An improved distributed algorithm for maximal independent set. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 270–277, USA, 2016. Society for Industrial and Applied Mathematics.

38 Mohsen Ghaffari. Distributed maximal independent set using small messages. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 805–820. SIAM, 2019. doi:10.1137/1.9781611975482.50.

39 Mohsen Ghaffari and Fabian Kuhn. Derandomizing distributed algorithms with small messages: Spanners and dominating set. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, pages 29:1–29:17, 2018. doi:10.4230/LIPIcs.DISC.2018.29.

40 Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. On the complexity of local distributed graph problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 784–797, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3055399.3055471.

41 Roman Glebov, Anita Liebenau, and Tibor Szabó. On the concentration of the domination number of the random graph. *SIAM Journal on Discrete Mathematics*, 29(3):1186–1206, 2015. doi:10.1137/12090054X.

42 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 468–485. SIAM, 2012.

43 Thorsten Götte, Christina Kolb, Christian Scheideler, and Julian Werthmann. Beep-and-sleep: Message and energy efficient set cover. In *Algorithms for Sensor Systems: 17th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2021, Lisbon, Portugal, September 9–10, 2021, Proceedings*, pages 94–110, Berlin, Heidelberg, 2021. Springer-Verlag. doi:10.1007/978-3-030-89240-1_7.

44 G. R. Grimmett and C. J. H. McDiarmid. On colouring random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 77(2):313–324, 1975. doi:10.1017/S0305004100051124.

45 Thorsten Götte, Christina Kolb, Christian Scheideler, and Julian Werthmann. Beep-and-sleep: Message and energy efficient set cover. *Theoretical Computer Science*, 950:113756, 2023. doi:10.1016/j.tcs.2023.113756.

46 András Hajnal, Wolfgang Maass, and György Turán. On the communication complexity of graph properties. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 186–191, New York, NY, USA, 1988. Association for Computing Machinery. doi:10.1145/62212.62228.

47 Magnús M. Halldórsson, Fabian Kuhn, Yannic Maus, and Tigran Tonoyan. Efficient randomized distributed coloring in congest. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1180–1193, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451089.

48 James W. Hegeman, Gopal Pandurangan, Sriram V. Pemmaraju, Vivek B. Sardeshmukh, and Michele Scquizzato. Toward Optimal Bounds in the Congested Clique: Graph Connectivity and MST. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, PODC '15, pages 91–100, New York, NY, USA, 2015. ACM. doi:10.1145/2767386.2767434.

49 James W. Hegeman and Sriram V. Pemmaraju. Lessons from the congested clique applied to mapreduce. *Theor. Comput. Sci.*, 608:268–281, 2015. doi:10.1016/j.tcs.2015.09.029.

50 Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. Communication Complexity of Approximate Matching in Distributed Graphs. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 460–473, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2015.460.

51 Gábor Ivanyos, Hartmut Klauck, Troy Lee, Miklos Santha, and Ronald de Wolf. New bounds on the classical and quantum communication complexity of some graph properties. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 148–159. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.148.

52 Lujun Jia, Rajmohan Rajaraman, and Torsten Suel. An efficient distributed algorithm for constructing small dominating sets. *Distributed Computing*, 15(4):193–205, 2002. doi:10.1007/s00446-002-0078-0.

53 Valerie King, Shay Kutten, and Mikkel Thorup. 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 '15, pages 71–80, New York, NY, USA, 2015. ACM. doi:10.1145/2767386.2767405.

54 Naoki Kitamura and Taisuke Izumi. A subquadratic-time distributed algorithm for exact maximum matching. *IEICE Transactions on Information and Systems*, E105.D(3):634–645, 2022. doi:10.1587/transinf.2021EDP7083.

55 Hartmut Klauck, Danupon Nanongkai, Gopal Pandurangan, and Peter Robinson. Distributed computation of large-scale graph problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 391–410. SIAM, 2015. doi:10.1137/1.9781611973730.28.

56 E. Korach, S. Moran, and S. Zaks. The optimality of distributive constructions of minimum weight and degree restricted spanning trees in a complete network of processors. *SIAM J. Comput.*, 16(2):231–236, 1987. doi:10.1137/0216019.

57 Kishore Kothapalli and Sriram V. Pemmaraju. Super-fast 3-ruling sets. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 136–147, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.136.

58 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. What cannot be computed locally! In *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, PODC '04, pages 300–309, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1011767.1011811.

59 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *J. ACM*, 63(2), 2016. doi:10.1145/2742012.

60 Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, New York, NY, USA, 1997.

61 Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. On the complexity of universal leader election. *J. ACM*, 62(1):7:1–7:27, 2015.

62 Nathan Linial and Michael Saks. Low diameter graph decompositions. *Combinatorica*, 13, 1993. doi:10.1007/BF01303516.

63 Zvi Lotker, Boaz Patt-Shamir, Elan Pavlov, and David Peleg. Minimum-Weight Spanning Tree Construction in $O(\log \log n)$ Communication Rounds. *SIAM Journal on Computing*, 35(1):120–131, 2005.

64 Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. Improved distributed approximate matching. *J. ACM*, 62(5), November 2015. doi:10.1145/2786753.

65 Gary L. Miller, Richard Peng, and Shen Chen Xu. Parallel graph decompositions using random shifts. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '13, pages 196–203, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2486159.2486180.

66 Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, Talal Riaz, and Peter Robinson. Symmetry Breaking in the Congest Model: Time- and Message-Efficient Algorithms for Ruling Sets. In *DISC*, pages 38:1–38:16, 2017. doi:10.4230/LIPIcs.DISC.2017.38.

67 Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, and Peter Robinson. Can we break symmetry with o(m) communication? In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*, pages 247–257. ACM, 2021. doi:10.1145/3465084.3467909.

68 Saurav Pandit and Sriram V. Pemmaraju. Rapid randomized pruning for fast greedy distributed algorithms. In Andréa W. Richa and Rachid Guerraoui, editors, *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010*, pages 325–334. ACM, 2010. doi:10.1145/1835698.1835777.

69 Gopal Pandurangan, David Peleg, and Michele Scquizzato. Message lower bounds via efficient network synchronization. *Theoretical Computer Science*, 810:82–95, 2020. Special issue on Structural Information and Communication Complexity. doi:10.1016/j.tcs.2018.11.017.

70 Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. A time- and message-optimal distributed algorithm for minimum spanning trees. *ACM Trans. Algorithms*, 16(1):13:1–13:27, 2020. doi:10.1145/3365005.

71 Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. On the distributed complexity of large-scale graph computations. *ACM Trans. Parallel Comput.*, 8(2):7:1–7:28, 2021. doi:10.1145/3460900.

72 Boaz Patt-Shamir and Mor Perry. 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*, pages 1–17, 2017. doi:10.1007/978-3-319-69084-1_1.

73 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

74 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. doi:10.1017/9781108671644.

75 Alexander A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

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

77 Prasoon Tiwari. Lower bounds on communication complexity in distributed computer networks. *J. ACM*, 34(4):921–938, October 1987. doi:10.1145/31846.32978.

78 Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

79 Ben Wieland and Anant P. Godbole. On the domination number of a random graph. *The Electronic Journal of Combinatorics [electronic only]*, 8, 2001. URL: http://eudml.org/doc/121890.

80 David P. Woodruff and Qin Zhang. When distributed computation is communication expensive. *Distributed Comput.*, 30(5):309–323, 2017. doi:10.1007/s00446-014-0218-3.

81 Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. ACM. doi:10.1145/800135.804414.