Parameterized Distributed Algorithms

Ran Ben-Basat
Harvard University, Cambridge, MA, United States
ran@seas.harvard.edu

Ken-ichi Kawarabayashi
National Institute of Informatics, Tokyo, Japan
k_keniti@nii.ac.jp

Gregory Schwartzman
National Institute of Informatics, Tokyo, Japan
greg@nii.ac.jp

Abstract

In this work, we initiate a thorough study of graph optimization problems parameterized by the *output size* in the distributed setting. In such a problem, an algorithm decides whether a solution of size bounded by k exists and if so, it finds one. We study fundamental problems, including Minimum Vertex Cover (*MVC*), Maximum Independent Set (*MaxIS*), Maximum Matching (*MaxM*), and many others, in both the LOCAL and CONGEST distributed computation models. We present lower bounds for the round complexity of solving parameterized problems in both models, together with optimal and near-optimal upper bounds.

Our results extend beyond the scope of parameterized problems. We show that any LOCAL $(1+\epsilon)$-approximation algorithm for the above problems must take $\Omega(\epsilon^{-1})$ rounds. Joined with the $(\epsilon^{-1}\log n)^{O(1)}$ rounds algorithm of [18] and the $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ lower bound of [22], the lower bounds match the upper bound up to polynomial factors in both parameters. We also show that our parameterized approach reduces the runtime of exact and approximate CONGEST algorithms for *MVC* and *MaxM* if the optimal solution is small, without knowing its size beforehand. Finally, we propose the first $o(n^2)$ rounds CONGEST algorithms that approximate *MVC* within a factor strictly smaller than 2.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed Algorithms, Approximation Algorithms, Parameterized Algorithms

Digital Object Identifier 10.4230/LIPIcs.DISC.2019.6

Related Version `https://arxiv.org/abs/1807.04900`

Funding *Ran Ben-Basat*: Supported by the Zuckerman Institute, the Technion Hiroshi Fujiwara Cyber Security Research center, and the Israel Cyber Directorate.
Ken-ichi Kawarabayashi: Supported by JSPS Kakenhi Grant Number JP18H05291.
Gregory Schwartzman: Supported by JSPS Kakenhi Grant Number JP19K20216 and JP18H05291.

Acknowledgements The authors thank the anonymous reviewers for their helpful remarks. We also thank Keren Censor-Hillel, Guy Even, Seri Khoury, and Ariel Kulik for helpful discussion and comments.

1 Introduction

We initiate the study of distributed algorithms for graph optimization problems parameterized on output size which are fundamental both in the sequential and distributed settings. When we refer to parametrized algorithms in this paper, we consider the output size as the parameter, which is called the *standard parametrization* [27]. Broadly speaking, in these problems we aim to find some underlying graph structure (e.g., a set of nodes) that abides a set of constraints (e.g., cover all edges) while minimizing the cost.

33rd International Symposium on Distributed Computing (DISC 2019).
Editor: Jukka Suomela; Article No. 6; pp. 6:1–6:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

While parameterized algorithms have received much attention in the sequential setting, not even the most fundamental problems (e.g., Vertex Cover) have a distributed counterpart[1]. We present parameterized upper and lower bounds for fundamental problems such as Minimum Vertex Cover, Maximum Matching and many more.

Motivation. In the sequential setting, some combinatorial optimization problems are known to have polynomial time algorithms (e.g., Maximum Matching), while others are NP-complete (e.g., Minimum Vertex Cover). To deal with the hardness of finding an optimal solution to these problems, the field of parameterized complexity asks what is the best running time we may achieve with respect to some parameter of the problem, instead of the size of the input instance. This parameter, usually denoted by k, is typically taken to be the size of the solution. Thus, even a running time that is exponential in k may be acceptable for small values of k.

In the distributed setting, a network of nodes, which is represented by a communication graph $G(V, E)$, aims to solve some graph problem with respect to G. Computation proceeds in synchronous rounds, in each of which every vertex can send a message to each of its neighbors. The running time of the algorithm is measured as the number of communication rounds it takes to finish. There are two primary communication models: LOCAL, which allows sending messages of unbounded size, and CONGEST, which limits messages to $O(\log n)$ bits (where $n = |V|$).

The above definition implies that the notion of "hardness" in the distributed setting is different. Because we do not take the computation time of the nodes into account, we can solve any problem in $O(D)$ rounds of the LOCAL model (where D is the diameter of the graph), and $O(n^2)$ rounds of the CONGEST model. This is achieved by having every node in the graph learn the entire graph topology and then solve the problem. Indeed, there exist "hard" problems in the distributed setting, where the dependence on D and n is rather large.

There are several lower bounds for distributed combinatorial optimization problems for both the LOCAL and the CONGEST models. For example, [28] provides lower bounds of $\widetilde{\Omega}(\sqrt{n} + D)$ (Where $\widetilde{\Omega}$ hides polylogarithmic factors in n), for a range of problems including MST, min-cut, min s-t cut and many more. There are also $\widetilde{\Omega}(n)$ bounds in the CONGEST model for problems such as approximating the network's diameter [1] and finding weighted all-pairs shortest paths [10]. Recently, the first near-quadratic (in n) lower bounds were shown by [10] for computing exact Vertex Cover and Maximum Independent Set.

The above shows that, similar to the sequential setting, the distributed setting also has many "hard" problems that can benefit from the parameterized complexity lens. Recently, the study of parameterized algorithms for *MVC* and *MaxM* was also initiated in the streaming environment by [11, 12]. This provides further motivation for our work, showing that indeed non-standard models of computation can benefit from parameterized algorithms.

1.1 Our results

Given the above motivation we consider the following fundamental problems (See Section 2 for formal definitions): Minimum Vertex Cover (*MVC*), Maximum Independent Set (*MaxIS*), Minimum Dominating Set (*MDS*), Minimum Feedback Vertex Set (*MFVS*), Maximum Matching (*MaxM*), Minimum Edge Dominating Set (*MEDS*), Minimum Feedback Edge Set (*MFES*). We use $\mathcal{P}$ to denote this problem set.

[1] Many parameters other than the output size have been considered, such as the maximum degree, arboricity, etc.

The problems are considered in both the LOCAL and CONGEST model, where we present lower bounds for the round complexity of solving parameterized problems in both models, together with optimal and near-optimal upper bounds. We also extend existing results [10,22] to the parameterized setting. Some of these extensions are rather direct, but are presented to provide a complete picture of the parameterized distributed complexity landscape.

Our results also extend beyond the scope of parameterized problems. We show that any LOCAL $(1+\epsilon)$-approximation algorithm for the above problems must take $\Omega(\epsilon^{-1})$ rounds. Joined with the $(\epsilon^{-1}\log n)^{O(1)}$ rounds algorithm of [18] and the $\Omega\left(\sqrt{\log n/\log\log n}\right)$ lower bound of [22], the lower bounds match the upper bound up to polynomial factors in both parameters. We also show that our parameterized approach reduces the runtime of exact and approximate CONGEST algorithms for *MVC* and *MaxM* if the optimal solution is small, without knowing its size beforehand. Finally, we propose the first $o(n^2)$ rounds CONGEST algorithms that approximate *MVC* within a factor strictly smaller than 2.

We note that considering parameterized algorithms in the distributed setting presents interesting challenges and unique opportunities compared to the classical sequential environment. In essence, we consider the *communication cost* of solving a parameterized problem on a network. On one hand, we have much more resources (and unlimited computational power), but on the other we need to deal with synchronizations and bandwidth restrictions.

We consider combinatorial minimization (maximization) problems where the size of the solution is upper bounded (lower bounded) by k. A parameterized distributed algorithm is given a parameter k (which is known to all nodes), and must output a solution of size at most k (at least k for maximization problems) if such a solution exists. Otherwise, *all vertices* must output that no such solution exists. A similar definition is given for parameterized approximation problems (see Section 2 for more details). For every $P \in \mathcal{P}$ we denote by k-P its parameterized variant and by k-$\mathcal{P}$ all these problems.

Lower bounds (partially deferred to the full version [7]). We show that the problem of *MVC* can be reduced to *MFVS* and *MFES* via standard reductions which also hold in the CONGEST model. There are no known results for *MFVS* and *MFES* in the distributed setting, so the above reductions, albeit simple, immediately imply that all existing lower bounds for *MVC* also apply for *MFVS* and *MFES*. Using the fact that *MFVS* and *MFES* have a global nature, we can achieve stronger lower bounds for the problems. Specifically, we show that no reasonable approximation can be achieved for *MFVS* and *MFES* in $o(D)$ rounds in the LOCAL model. This is formalized in the following theorem.

▶ **Theorem 1.** *For any $k \in \mathbb{N}^+$, any algorithm that solves MFVS or MFES on a graph with a solution of size k to within an additive error of $O(n/D)$ must take $\Omega(D)$ rounds in the LOCAL model.*

Our main result is a novel lower bound for $(1+\epsilon)$-approximation for all problems in $\mathcal{P}$. Our lower bound states that any $(1+\epsilon)$-approximation (deterministic or randomized) algorithm in the LOCAL model for any problem in $\mathcal{P}$ requires $\Omega(\epsilon^{-1})$ rounds. Usually, lower bounds in the distributed setting are given as a function of the input (size, max degree), and not as a factor of approximation ratio. Our lower bound also applies to *Max-Cut* and *Max-DiCut*, whose parameterized variants are not considered in this paper, and thus are not in $\mathcal{P}$. We state the following theorem.

▶ **Theorem 9.** *For any $\epsilon = \Omega(1/n)$ and $\delta < 1/2 - EXP(-\Omega(n\epsilon))$, any Monte-Carlo LOCAL algorithm that computes a $(1+\epsilon)$-multiplicative approximation with probability at least $1-\delta$ for a problem $P \in \{MVC, MaxM, MaxIS, MDS, MEDS, Max\text{-}DiCut, Max\text{-}Cut, MFVS, MFVS\}$ requires $\Omega(\epsilon^{-1})$ rounds.*

In the full version [7], we show a more involved lower bound construction which extends our lower bound to *Max-k-Cut*[2] where $k = O(1)$. Our lower bound also has implications for non-parameterized algorithms.

The problem of finding a maximum matching in the distributed setting is a fundamental problem in the field that received much attention [2,3,15,25,26]. Despite the existence of a variety of approximation algorithms for the problem, no non-trivial result is known for computing an exact solution for general graphs.

Our lower bounds also have implications for computing a $(1+\epsilon)$-approximation of *MVC*, *MaxM* and *MaxIS* in the LOCAL model. Combined with the $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ lower bound of [22], we can express a lower bound to the problem as $\Omega\left(\epsilon^{-1}+\sqrt{\frac{\log n}{\log\log n}}\right) = (\epsilon^{-1}\log n)^{\Omega(1)}$. Together with the result of [18], which presents an $(\epsilon^{-1}\log n)^{O(1)}$ upper bound, this implies that the complexity of computing a $(1+\epsilon)$-approximation is given by $(\epsilon^{-1}\log n)^{\Theta(1)}$.

Finally, we show a simple and generic way of extending lower bounds to the parameterized setting. The problem with many of the existing lower bounds (e.g., [10,22]) is that the size of the solution is $\tilde{O}(n)$ (linear up to polylogarithmic factors). Thus, it might be the case that if the solution is substantially smaller than the input we might achieve a much faster running time. We show that by simply attaching a large graph to the lower bound graph we can achieve the same lower bounds as a function of k, rather than n. This allows us to restate our $(1+\epsilon)$-approximation lower bound and the bounds of [22] and [10] as a function of $k \ll n$. We show that these lower bounds hold for parameterized problems as defined in this paper.

▶ **Theorem 2.** *There exists a family of graphs $G_{k,n}(V,E)$, such that for any $\epsilon = \Omega(1/k)$ and $\delta < 1/2 - EXP(-\Omega(k\epsilon))$, any Monte-Carlo LOCAL algorithm that computes a $(1+\epsilon)$-multiplicative approximation with probability $1-\delta$ for some k-$P \in k$-$\mathcal{P}$ requires $\Omega(\epsilon^{-1})$ rounds. Here, $n = |V|$ can be arbitrarily larger than k.*

▶ **Theorem 3.** *There exists $\underline{k} \in \mathbb{N}$ such that for any $\underline{k} \le k \le 0.99n$, there exists a family of graphs $G_{k,n}(V,E)$, such that any algorithm that solves k-MVC on $G_{k,n}$ in the CONGEST model requires $\Omega(k^2/\log k \log n)$ rounds, where $n = |V|$ can be arbitrarily larger than k.*

▶ **Theorem 4.** *There exists a family of graphs $G_{k,n}(V,E)$, for sufficiently large k, such that any algorithm that computes a constant approximation for k-MVC, k-MaxM, or k-MaxIS for $G_{k,n}$ in the LOCAL model requires $\Omega\left(\sqrt{\log k/\log\log k}\right)$ rounds, where $n = |V|$ can be arbitrarily larger than k.*

Upper bounds. We first define the family of problems whose optimal solution is lower bounded by the graph diameter (*DLB*, see Section 2 for a formal definition). If the optimal solution size (*OPT*) is small, then for minimization DLB problems we can learn the entire graph in $O(OPT)$ LOCAL rounds. The problem is actually for the case when the optimal solution is large, and all vertices in the graph must output that no k-sized solution exists. Here we introduce an auxiliary result which we use as a building block for all of our algorithms.

▶ **Theorem 12.** *There exists an $O(k)$ rounds deterministic algorithm in the CONGEST model that terminates with all vertices outputting* SMALL *if the diameter is bounded by k, and* LARGE *if the diameter is larger than $2k$. If the diameter is between $k+1$ and $2k$, the vertices answer unanimously, but may return either* SMALL *or* LARGE.

[2] In *Max-k-Cut*, we wish to divide the vertices into k sets $\{A_i\}_{i=1}^{k}$ such that the number of edges whose endpoints are in different sets is maximized. Notice that this is not a parameterized problem.

Table 1 A summary of our round complexity results for *k-MVC* and *k-MaxM*. All lower bounds hold for randomized algorithms as well as deterministic.

Variant	**Upper Bound**		**Lower Bound**	
	LOCAL	CONGEST	LOCAL	CONGEST
Exact	$O(k)$ [*det.*]	$O\left(k+\frac{k^2\log k}{\log n}\right)$ [*rand.*] $O\left(k^2\right)$ [*det.*]	$\Omega(k)$	$\Omega\left(k+\frac{k^2}{\log k\log n}\right)$ **k-MVC* only
$(1+\epsilon)$-approx. $\forall\epsilon=\Omega(1/k)$			$\Omega\left(\epsilon^{-1}+\sqrt{\frac{\log k}{\log\log k}}\right)$	
$(2-\epsilon)$-approx. $\forall\epsilon\in[1/k,1]$		$O\left(k+\frac{(k\epsilon)^2\log(k\epsilon)}{\log n}\right)$ [*rand.*] $O(k+(k\epsilon)^2)$ [*det.*]		

Using the above, we can check the diameter, have all vertices reject if it is too large, and otherwise have a leader learn the entire graph in $O(k)$ rounds. As for maximization problems (such as *MaxM* and *MaxIS*) the challenge is somewhat different, as the parameter k does not bound the diameter for legal instances. We first check whether the diameter is at most $2k$ or at least $4k$. If it is bounded by $2k$, we can learn the entire graph. Otherwise, we note that any *maximal* solution has size at least k and is a legal solution to the parameterized problem. Thus, we can efficiently compute a maximal solution by having every node/edge which is a local minimum (according to id) enter the matching/independent set. We repeat this k times and finish (this also works in the CONGEST model). We formalize this in the following theorem.

▶ **Theorem 13.** *There exist $O(k)$ rounds LOCAL algorithms for k-MaxM, k-MaxIS, and any minimization problem k-P for $P \in$ DLB.*

Next, we consider the problems of *k-MVC* and *k-MaxM* as case studies for the CONGEST model. We show deterministic upper bound of $O(k^2)$ for *k-MVC* and *k-MaxM* (For *k-MVC* this is near-tight according to Theorem 3). We also note that as the complement of a *k-MVC* is a *k-MaxIS* we have a near-tight upper bound of $O(n-k)^2$ for the problem. This means that if k is large (e.g., $k = n - \log n$) the problem is easy to solve. In the CONGEST model, we first verify that the diameter is indeed small. If it is large, we proceed as we did in the LOCAL model for both problems. For *k-MVC*, we use a standard kernelization procedure to reduce the size of the graph. This is done by adding every node of degree larger than k into the cover. The remaining graph has a bounded diameter and a small number of edges; thus we use a leader node to collect the entire graph. The problem of *k-MaxM* is more challenging, as we do not use existing kernelization techniques. Instead, we introduce a new augmentation-based approach for the parameterized problem.

We then show how with the help of randomization we can achieve a running time of $O(k + k^2 \log k/\log n)$ for both problems. Note that for $k \ll n$ this can be a substantial, up to quadratic, improvement. Further, for *k-MVC* it brings our round complexity to within a $O(\log^2 k)$ factor from the lower bound.

Approximations. We consider approximation algorithms, in the CONGEST model, for parameterized *MVC* and *MaxM*. We make non-trivial use of the Fidelity Preserving Transformation framework [14] and simultaneously apply multiple reduction rules that reduce the parameter from k to $O(k\epsilon)$. Using this technique, we derive $(2-\epsilon)$-approximations that run faster than our exact algorithms for any $\epsilon = o(1)$. We summarize our other results in Table 1.

Table 2 Our CONGEST round complexity for deterministic *MVC* and *MaxM*, where ϵ is a positive constant (the actual dependency in ϵ^{-1} is logarithmic). Here, *OPT* is the size of the optimal solution and is *not known* to the algorithm.

	Exact	$(1+\epsilon)$-approx.	2-approx.	$(2+\epsilon)$-approx.
MVC	$O\left(\min\left\{OPT^2 \log OPT, n^2\right\}\right)$	$O\left(OPT^2\right)$	$O\left(\min\left\{OPT \log OPT, \frac{\log n \log \Delta}{\log^2 \log \Delta}\right\}\right)$	$O\left(\min\left\{OPT, \frac{\log \Delta}{\log \log \Delta}\right\}\right)$
MaxM			$O\left(\min\left\{OPT \log OPT, \Delta + \log^* n\right\}\right)$	$O\left(\min\left\{OPT, \Delta + \log^* n\right\}\right)$

Applications to non-parameterized algorithms (deferred to the full version [7]). We show that our algorithms can also imply faster non-parameterized algorithms if the optimal solution is small, without needing to know its size. Specifically, we combine our exact and approximation algorithms for parameterized k-*MVC* and k-*MaxM* with doubling and a partial binary search for the value of k. Additionally, our solutions can determine whether to run the existing non-parameterized algorithm or follow the parameterized approach. This results in an algorithm whose runtime is the minimum between current approaches and the number of rounds required for the binary search. Our results are presented in Table 2.

We also present *deterministic* algorithms for *MVC*, *MaxM* in the CONGEST model with an approximation ratio strictly better than 2. Namely, our algorithms terminate in $O(OPT \log OPT) = O(n \log n)$ rounds and provide an approximation ratio of $2 - 1/\sqrt{OPT}$. Here, *OPT* is the size of the optimal solution and is not known to the algorithm. This is the first non-trivial $(2-\epsilon)$-approximation for *MVC*.[3]

1.2 Related work

Distributed Matching and Covering. Both *MVC* and *MaxM* have received significant attention in the distributed setting. We survey on the results relevant to this paper. We start with existing lower bounds. In [10] a family of graphs of increasing size is presented, such that computing an *MVC* for any graph in the family requires $\Omega(n^2/\log^2 n)$ rounds in the CONGEST model. In [22] a family of graphs is introduced such that any constant approximation for *MVC* requires $\Omega\left(\min\left\{\sqrt{\log n/\log\log n}, \log\Delta/\log\log\Delta\right\}\right)$ rounds in the LOCAL model. Both bounds hold for deterministic and randomized algorithms.

For *MVC*, no non-trivial exact distributed algorithms are known. As for approximations, an optimal (for constant values of ϵ) $(2+\epsilon)$-approximate deterministic algorithm (for the weighted variant) in the CONGEST model running in $O(\epsilon^{-1}\log\Delta/\log\log\Delta)$ rounds is given in [4]. [5] then improved the dependency on ϵ to $O(\log\Delta/\log\log\Delta + \log\epsilon^{-1}\log\Delta/\log^2\log\Delta)$, which also results in a faster 2-approximation algorithm by setting $\epsilon = 1/n$. Recently, this was improved further with a $O(\log n)$ rounds deterministic 2-approximation algorithm [6]. In LOCAL, a randomized $(1+\epsilon)$-approximation in $(\epsilon^{-1}\log n)^{O(1)}$ rounds is due to [18].

For *MaxM*, there are no non-trivial known lower bounds for the exact problem. At the time this paper was first made public, the best lower bound for approximations was due to [22], and no exact non-trivial solution was known for the problem in both the LOCAL and CONGEST models. Independently and simultaneously to our results, [2] show an exact algorithm for *bipartite* graphs running in $O(n \log n)$ rounds in the CONGEST model. They also show an $\tilde{\Omega}(D + \sqrt{n})$ lower bound for computing an $O(1 + \Theta(1/\sqrt{n}))$-approximation for unweighted *fractional* matching in the CONGEST model.

[3] When this paper was first made public no better than 2 deterministic approximation for *MaxM* was known in the CONGEST model. Independently and simultaneously [2] provide a deterministic algorithm in the CONGEST model achieving a $(1.5+\epsilon)$-approximation for the problem.

As for approximations, much is known. We survey the results for the unweighted case. An optimal randomized $(1+\epsilon)$-approximation in the CONGEST model, running in $O(\log\Delta/\log\log\Delta)$ rounds for constant ϵ is given in [3]. As for deterministic algorithms, the best known results in the LOCAL model are due to [15] and [20]. In [15] a maximal matching algorithm running in $O(\log^2\Delta\log n)$ rounds and a $(2+\epsilon)$-approximate algorithm running in $O(\log^2\Delta\log 1/\epsilon+\log^* n)$ are given. In [20], a $(1+\epsilon)$-approximation randomized algorithm that runs in $\widetilde{O}(\epsilon^{-3}\log\Delta)$ rounds and a deterministic $\widetilde{O}(\epsilon^{-5}\log^3\Delta+\epsilon^{-1}\log^* n)$ round $(1+\epsilon)$-approximation algorithm are presented. Recently, [2] showed a deterministic algorithm running in $O(\epsilon^{-2}\log(\Delta W)+\epsilon^{-1}(\log^2(\Delta/\epsilon)+\log^* n))$ CONGEST rounds, achieving a $(1+\epsilon)(g/(g-1))$-approximation for weighted maximum matching, where g is the length of the shortest odd cycle in the graph.

Distributed parameterized algorithms. Parameterized distributed algorithms were previously considered for detection problems. Namely, in [21] it was shown the detecting k-paths and trees on k-nodes can be done deterministically in $O(k2^k)$ rounds of the broadcast CONGEST model. Similar results were obtained independently by [13] in the context of distributed property testing [9]. Other distributed algorithms have running times which are parameterized by topological properties of the input graph, see for example [17, 19].

2 Preliminaries

In this paper, we consider the classic and parameterized variants of several popular graph packing and covering problems, in the LOCAL and CONGEST models. A solution to these problems is either a vertex-set or an edge-set. In vertex-set solutions, we require that each vertex will know if it is in the solution or not. For edge-set problems, each vertex must know which of its edges are in the solution, and both end-points of an edge must agree. Computation takes place in synchronous rounds during which each node first receives messages from its neighbors, then perform local computation, and finally send messages to its neighbors. Each of the messages sent to a node neighbor may be different from the others, while the size of messages is unbounded in the LOCAL model or of size $O(\log n)$ in the CONGEST. In both models, the communication graph is identical to the graph on which the problem is solved. That is, two nodes may send messages to each other only if they share an edge. In this paper, we consider the following problems: Minimum Vertex Cover (*MVC*), Maximum Independent Set (*MaxIS*), Minimum Dominating Set (*MDS*), Minimum Feedback Vertex Set (*MFVS*), Maximum Matching (*MaxM*), Minimum Edge Dominating Set (*MEDS*), Minimum Feedback Edge Set (*MFES*). For completeness, we provide formal problem definitions in the full version [7]. We use $\mathcal{P}=\{MVC, MaxM, MaxIS, MDS, MEDS, MFVS, MFES\}$ to denote the above set of problems. Given a parameter k, a parameterized algorithm computes a solution of size bounded by k if such exists; otherwise, the nodes must report that no such solution exists. For a problem $P\in\mathcal{P}$, we denote by k-P (e.g., k-*MVC*) the parameterized variant of the problem. We use k-$\mathcal{P}$ to denote all these problems. We note that all nodes know k when the algorithm starts and that the result may not be the optimal solution to the problem.

▶ **Definition 5.** *An algorithm for a* minimization *(respectively,* maximization*) problem k-P must find a solution of size at most (respectively, at least) k if such exists. If no such solution exists then all nodes must report so when the algorithm terminates.*

We now generalize our definition to account for randomized and approximation algorithm.

▶ **Definition 6.** *For $\alpha \geq 1$, an α-approximation algorithm for a* minimization *(respectively,* maximization*) problem k-P must find a solution of size at most αk (respectively, at least k/α) if a solution of size k exists. Otherwise, all nodes must report that no k-sized solution exists.*

▶ **Definition 7.** *Given $\delta \geq 0$ and $\alpha \geq 1$, an α-approximation Monte Carlo algorithm for a problem k-P terminates with an α-approximate solution to k-P with probability at least $1 - \delta$.*

We now define the notion of *Diameter-Lower-Bounded* (*DLB*) problems. Intuitively, this class contains all problems whose optimal solution size is $\Omega(D)$, which allows efficient algorithms for their parameterized variants. For example, *DLB* includes *MVC*, *MaxIS*, *MaxM*, *MDS*, and *MEDS*, but not *MFVS* and *MFES*. Roughly speaking, these problems admit efficient LOCAL parameterized algorithms as the parameter limits the radius that each node needs to see for solving the problem.

▶ **Definition 8.** *An optimization problem P is in DLB if for any input graph G of diameter D, the size of an optimal solution to P is of size $\Omega(D)$.*

3 Lower Bounds

In this section, we present lower bounds for a large family of classical distributed graph problems. In appendix 3, we generalize the results to randomized algorithms, present additional lower bounds, and show that how classic bounds translate to the parameterized problems.

Here, we provide a construction that implies lower bounds for approximating all problems in $\mathcal{P}$. Our lower bounds dictate that any algorithm that computes a $(1+\epsilon)$-approximation, for $\epsilon = \Omega(1/n)$, requires at least $\Omega(\epsilon^{-1})$ rounds in the LOCAL model, even for randomized algorithms. We note that for all of these problems, no known lower bounds [22, 24] imply superlogarithmic lower bound (which we can get, e.g., for $\epsilon = n^{-2/3}$) was known for approximations. Further, for some problems, such as *MaxM*, no such lower bound is known even for exact solutions in the CONGEST model.

We then generalize our approach and show that even in the parameterized variants of the problems (where the optimal solution is bounded by k), $\Omega(\epsilon^{-1})$ rounds are needed for an arbitrarily large graphs (where $n \gg k$). Our approach is based on the observation that for any x-round algorithm, there exists an input graph of many distinct $\Theta(x)$-long paths, such that the algorithm has an additive error on each of the paths, which accumulates over all paths in the construction. Intuitively, we show that for any set of n node identifiers and any algorithm that takes $o(\epsilon^{-1})$ rounds, it is possible to assign identifiers to nodes such that the approximation ratio would be larger than $1+\epsilon$. Our goal in this section is to prove the following theorem.

▶ **Theorem 9.** *For any $\epsilon = \Omega(1/n)$ and $\delta < 1/2 - EXP(-\Omega(n\epsilon))$, any Monte-Carlo LOCAL algorithm that computes a $(1+\epsilon)$-multiplicative approximation with probability at least $1-\delta$ for a problem $P \in \{MVC, MaxM, MaxIS, MDS, MEDS, Max\text{-}DiCut, Max\text{-}Cut, MFVS, MFVS\}$ requires $\Omega(\epsilon^{-1})$ rounds.*

3.1 Basic Construction

We start with lower bounds for the non-parameterized variants of the problems, where the optimal solution may be of size $\Theta(n)$. For integer parameters $r, \ell > 10$, the graph

$$G_{r,\ell} = (\{v_0\} \cup \{v_{i,j} \mid i \in [r], j \in [\ell+1]\}, \{\{v_0, v_{i,0}\} \mid i \in [r]\} \cup \{\{v_{i,j}, v_{i,j+1}\} \mid i \in [r], j \in [\ell]\})$$

consists of r disjoint paths of length ℓ, whose initial nodes are connected to a central vertex v_0. We also consider the digraph $\vec{G}_{r,\ell}$ in which each edge is oriented away from v_0; i.e.,

$$\vec{G}_{r,\ell} = (\{v_0\} \cup \{v_{i,j} \mid i \in [r], j \in [\ell+1]\}, \{(v_0, v_{i,0}) \mid i \in [r]\} \cup \{(v_{i,j}, v_{i,j+1}) \mid i \in [r], j \in [\ell]\})$$

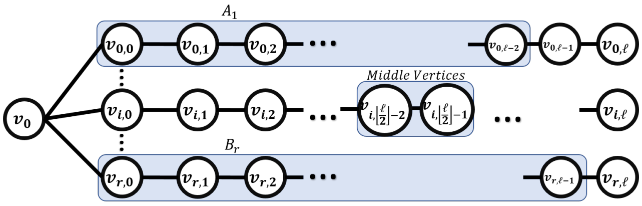

Figure 1 Our basic construction. The middle vertices' ($v_{i,\lfloor \ell/2\rfloor-2}$ and $v_{i,\lfloor \ell/2\rfloor-1}$) output remains the same if we reverse the identifiers along A_i and the algorithm takes less than $\lfloor \ell/2\rfloor - 3$ rounds. The output of $v_{i,\lfloor \ell/2\rfloor}$ also remains the same if B_i is flipped. However, since the distance of these vertices from v_0 change, the output is suboptimal for at least one of the orderings.

We present our construction in Figure 1. $G_{r,\ell}$ has $n = r(\ell+1)+1$ vertices and a diameter (as $r > 1$) of 2ℓ. Observe that the optimal solutions of *MVC*, *MaxM*, *MaxIS*, *MDS*, and *MEDS* on $G_{r,\ell}$ (for $r, \ell \geq 3$) have values of $\Theta(r\ell) = \Theta(n)$. For every path $i \in [r]$, let $A_i = \langle v_{i,0}, \ldots, v_{i,\ell-2}\rangle$ and $B_i = \langle v_{i,0}, \ldots, v_{i,\ell-1}\rangle$ denote the longest odd and even length subpaths that do not include v_0 and $v_{i,\ell}$. Given a path P of vertices with assigned identifiers, we denote by P^R a reversal in the order of identifiers. For example, if the identifiers assigned to A_0 were $\langle 0, \ldots, \ell-2\rangle$, then those of A_0^R would be $\langle \ell-2, \ell-3, \ldots, 0\rangle$. This reversal of identifiers along a path plays a crucial role in our lower bounds. Intuitively, if the number of rounds is less than $\ell/2-3$ and we reverse A_i or B_i, the output of the middle vertices *would change* to reflect the mirror image they observe. We show that this implies that on either the original identifier assignment or its reversal, the algorithm must find a sub-optimal solution to the i'th path (where the choice of whether to flip A_i or B_i depends if the output is a vertex set or edge set). In turn, this would sum up to a solution that is far from the optimum by at least an r-additive factor. As the optimal solution is of size $\Theta(r\ell)$, this implies a multiplicative error of $1+\Theta(r/(r\ell)) = 1+1/\ell$.

We show that for arbitrarily large graphs with an optimal solution of size $\Theta(n)$, any x-round algorithm must have an additive error of $\Omega(n/x)$.

▶ **Lemma 10.** *Let $x, r \in \mathbb{N}^+$ be integers larger than 10, and let $\mathcal{I}$ be a set of $n = (2x+4)r+1$ node identifiers. For any deterministic LOCAL algorithm for MVC, MaxM, MaxIS, MDS, MEDS, Max-DiCut, or Max-Cut that terminates in x rounds on $G_{r,2x+3}$, there exists an assignment of vertex identifiers for which the algorithm has an additive error of $\Omega(n/x)$.*

Proof. First, let us characterize the optimal solutions for each of the problems on $G_{r,2x+3}$ (or $\vec{G}_{r,2x+3}$ for *Max-DiCut*). For simplicity of presentation, we assume that $\mathcal{I} = [n]$ and (x mod 6) $= 0$ although the result holds for any $\mathcal{I}$ and x. We have

$$
\begin{aligned}
&OPT_{MVC} = \{v_{i,2j} \mid i \in [r], j \in [x+2]\}\,,\\
&OPT_{MM} = \{\{v_{i,2j}, v_{i,2j+1}\} \mid i \in [r], j \in [x+2]\}\,,\\
&OPT_{MaxIS} = OPT_{\textit{Max-Cut}} = OPT_{\textit{Max-DiCut}} = \{v_0\} \cup \{v_{i,2j+1} \mid i \in [r], j \in [x+2]\}\,,^{4}\\
&OPT_{MDS} = \{v_{i,3j} \mid i \in [r], j \in [2x/3+1]\}\,,\\
&OPT_{MEDS} = \{\{v_0, v_{i,0}\} \mid i \in [r]\} \cup \{\{v_{i,3j+2}, v_{i,3j+3}\} \mid i \in [r], j \in [2x/3]\}\,.^{5}
\end{aligned}
$$

[4] For *Max-Cut*, the complement solution $V \setminus \{v_0\} \cup \{v_{i,2j+1} \mid i \in [r], j \in [x+2]\}$ is also optimal, but the correctness would follow from similar arguments.

[5] Each edge of the form $\{v_0, v_{i,0}\}$ (for $i \in [r]$) may each be replaced by the $\{v_{i,0}, v_{i,1}\}$ edge. Unlike the other problems, the optimal solution here is not unique, but this does not affect the proof.

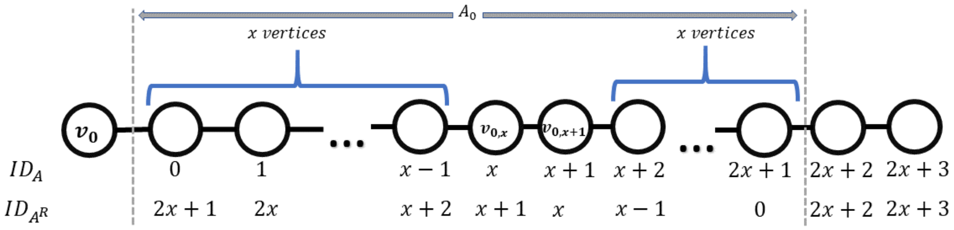
A_0
x vertices
x vertices
v_0
$v_{0,x}$
$v_{0,x+1}$
ID_A
0 1 $x-1$ x $x+1$ $x+2$ $2x+1$ $2x+2$ $2x+3$
ID_{A^R}
$2x+1$ $2x$ $x+2$ $x+1$ x $x-1$ 0 $2x+2$ $2x+3$

Figure 2 Before the reversal (in A_0), vertices $v_{0,x}$ and $v_{0,x+1}$ have identifiers x and $x+1$. When reversing A_0, the vertices switch identifiers and must switch their output. That is, $out_A(v_{0,x}) = out_{A^R}(v_{0,x+1})$ and $out_A(v_{0,x+1}) = out_{A^R}(v_{0,x})$ in any algorithm that takes fewer than x communication rounds.

For example, this means that the only optimal solution to *MVC* picks all vertices whose distance from v_0 is odd. Next, consider a path $i \in [r]$ and consider the case where every vertex $v_{i,j}$ has identifier $(2x+4)i+j$. From the point of view of the node with identifier $(2x+4)i+x$ (which is $v_{i,x}$ in this assignment), in its x-hop neighborhood it has nodes with identifiers $(2x+4)i, (2x+4)i+1, \ldots, (2x+4)i+x-1$ on one port (side) and identifiers $(2x+4)i+x+1, (2x+4)i+x+2, \ldots, (2x+4)i+2x+2$ on the other. On the other hand, if we reverse A_i (i.e., assign $v_{i,j} \in A_i$ with identifier $(2x+4)i+2x+1-j$), the view of $v_{i,x}$ remains exactly the same. That is, the node observes the exact same topology and vertex identifiers in both cases. Since the algorithm is deterministic, the output of $(2x+4)i+x$ must remain the same for both identifier assignments, even though now it is placed in $v_{i,x+1}$! Similarly, reversing A_i would mean that the node with identifier $(2x+4)i+x+1$ (which changes places from $v_{i,x+1}$ to $v_{i,x}$ after the reversal) also provides the same output in both cases. Therefore, reversing A_i would switch the outputs of $v_{i,x+1}$ and $v_{i,x}$. This implies that the output of the algorithm is suboptimal for either A_i or A_i^R for *MVC*, *MaxM*, *MaxIS*, *Max-Cut*, and *Max-DiCut*. We illustrate this reversal on path 0 in Figure 2. Repeating this argument for B_i, we get that its reversal would switch the outputs of $v_{i,x}$ and $v_{i,x+2}$, making the algorithm err in *MDS* (as $v_{i,x}$ is in the optimal cover while $v_{i,x+2}$ is not).

For MEDS, every u, v that share an edge must agree whether it is in the solution or not. In an optimal solution the edge $\{v_{i,x}, v_{i,x+1}\}$ must be in the dominating set while $\{v_{i,x+1}, v_{i,x+2}\}$ must not. However, by reversing B_i the identifiers of $v_{i,x}$ and $v_{i,x+2}$ switch, changing edge added from $\{v_{i,x}, v_{i,x+1}\}$ to $\{v_{i,x+1}, v_{i,x+2}\}$ or vice versa, implying an error for B_i or B_i^R.

As we showed that there exists an identifier assignment that "fools" the algorithm on every path $i \in [r]$, we conclude that the algorithm has an additive error of at least $r = (n-1)/(2x+4) = \Omega(n/x)$. ◀

Lower bounds for MFVS and MFES follow from the reduction to *MVC* which is presented in the full version [7]. Since the optimal solution to all problems on $G_{r,2x+4}$ is of size $\Theta(n)$, we have that the algorithms have an approximation ratio of $1+\Theta((n/x)/n) = 1+\Theta(1/x)$. Plugging $\epsilon = \Theta(1/x)$ we conclude the following.

▶ **Corollary 11.** *For any $\epsilon = \Omega(1/n)$, any deterministic LOCAL algorithm that computes a $(1+\epsilon)$-multiplicative approximation for any $P \in \mathcal{P}$ requires $\Omega(\epsilon^{-1})$ rounds.*

4 Upper Bound Warmup – Parameterized Diameter Approximation

In this section, we illustrate the concept of parameterized algorithms with the classic problem of diameter approximation. This procedure will also play an important role in all our algorithms. Computing the exact diameter of a graph in the CONGEST model is costly. Specifically, it is known that computing a $(3/2-\epsilon)$-approximation of the diameter, or even distinguishing between diameter 3 or 4, requires $\widetilde{\Omega}(n)$ CONGEST rounds[6] [1, 8, 16, 23]. Computing a 2-approximation for the diameter is straightforward in $O(D)$ rounds, by finding the depth of any shortest paths tree. However, we wish to devise algorithms whose round complexity is bounded by some function $f(k)$, even if no solution of size k exists. Therefore, we now show that it is possible to compute a 2-approximation for the *parameterized version* of the diameter computation problem.

▶ **Theorem 12.** *There exists an $O(k)$ rounds deterministic algorithm in the CONGEST model that terminates with all vertices outputting* SMALL *if the diameter is bounded by k, and* LARGE *if the diameter is larger than $2k$. If the diameter is between $k+1$ and $2k$, the vertices answer unanimously, but may return either* SMALL *or* LARGE*.*

Proof. Our algorithm starts with k rounds, such that in every round each vertex broadcasts the minimal identifier it has learned about (initially its own identifier). After this stage terminates, each vertex v has learned the minimal identifier x_v in its k-hop neighborhood.

Next follows $2k+1$ rounds such that in each round each vertex v broadcasts y_v and z_v, which are the minimal and maximal x_u identifier it has seen so far. That is, we start with $y_v = z_v = x_v$ and at each round we set $y_v \leftarrow \min\{y_v, \min\{y_u \mid u \in N(v)\}\}$ and $z_v \leftarrow \max\{z_v, \max\{z_u \mid u \in N(v)\}\}$. When this ends, each vertex returns *SMALL* if $y_v = z_v$ and *LARGE* otherwise. Clearly, the entire execution takes $O(k)$ rounds.

For correctness, observe that if the diameter is bounded by k then all x_v's are identical to the globally minimal identifier. Next, assume that the diameter is at least $2k+1$, and fix some vertex v. This means that there exist a vertex u whose distance is exactly $k+1$ with respect to x_v, and thus at most $2k+1$ from v. Since the first stage of the algorithm runs for k rounds, we have that $x_u \neq x_v$. Therefore, after $k+1$ rounds of the second stage we have that $y_{x_v} \neq z_{x_v}$, and after additional k rounds $y_v \neq z_v$ and thus v outputs *LARGE*. Finally, if the diameter is between $k+1$ and $2k$, then all vertices have the same y_v and z_v values and thus answer unanimously. ◀

5 Parameterized Problems Upper Bounds

5.1 LOCAL Algorithms

Our first result is for diameter lower bounded problem in the LOCAL model. We show that any minimization problem k-$P \in DLB$ can be solved in $O(k)$ LOCAL rounds. To that end, we first use Theorem 12 to check whether the diameter is at most ck, or at least $2ck$, where c is a constant such that a diameter of at least ck implies that no solution of size k exists. If the diameter is larger than $2ck$, the algorithm terminates and reports that no k-sized solution exists. Otherwise, we collect the entire graph at a leader vertex v which computes the optimal solution. If the solution is of size at most k, v sends it to all vertices. If no solution of size k exists, v notifies the other vertices.

[6] Where $\widetilde{\Omega}(n)$ hides factors polylogarithmic in n.

The above approach does not necessarily work for maximization problems as the existence of a k-sized solution does not imply a bounded diameter. Nevertheless, we now show that *k-MaxM* and *k-MaxIS* have $O(k)$ rounds algorithms. For this, we first check whether the diameter is at most $2k$ or at least $4k$. If the diameter is small, we can still collect the graph and solve the problem locally. Otherwise, we use the fact that *any* maximal matching or Independent Set in a graph with a diameter larger than $2k$ must be of size at least k. Since the maximal matching or independent set may be too large, we run just k iterations of extending the solution. For *k-MaxM*, at each iteration, any edge that neither of its endpoints is matched and that is a local minimum (with respect to the identifiers of its endpoints) joins the matching. We are guaranteed that the matching grows by at least a single edge at each round and thus after k iterations the algorithm terminates. Similarly, for *k-MaxIS*, at each iteration, every vertex that neither of its endpoints is in the independent set and is a local minimum enters the set. We summarize this in the following theorem.

▶ **Theorem 13.** *There exist $O(k)$ rounds LOCAL algorithms for k-MaxM, k-MaxIS, and any minimization problem k-P for $P \in$ DLB.*

5.2 CONGEST algorithms for k-MVC

Here we state our results for *k-MVC* in the CONGEST model. Our first algorithm is deterministic and aims to solve the exact variant of *k-MVC*. Intuitively, it works in two phases; first, it checks that the diameter is $O(k)$, if not the algorithm rejects. Knowing that the diameter is bounded by $O(k)$, we proceed by calculating a solution *assuming there exists a solution of size at most k*. If this assumption holds, we are guaranteed to find such a solution. We run the above for just enough rounds to guarantee that if a k-sized solution exists we will find such a solution. Finally, we check that the size of the solution returned by the algorithm is indeed bounded by k.

We first show a procedure that solves the problem *if* a solution of size k exists. If no such solution exists, this procedure may not terminate in time or compute a cover larger than k.

▶ **Lemma 14.** *There exists a deterministic algorithm that if a k-sized cover exists: (1) terminates in $O(k^2)$ CONGEST rounds and (2) finds such a cover.*

Proof. Given that there exists a k-sized cover, the diameter of the graph is bounded by $2k$. Therefore, we can compute a unique leader and a BFS tree rooted at that leader in $O(k)$ rounds. Our first observation is that every v with a degree larger than k must be in any k-sized cover. Thus, every such vertex immediately goes into the cover and gets removed together with all of its adjacent edges. If a vertex has degree 0, it terminates (without entering the cover). Denote the remaining graph by $G' = (V', E')$.

For our analysis, let us fix some vertex cover $X \subseteq V'$ of size k and denote the remaining vertices by $A = V' \setminus X$. We note that the set A is an independent set. Thus, all edges in the graph are either between vertices in X or between A and X. We note that $|X| \leq k$, and now we aim to bound the number of remaining edges. We now show that $|E'| \leq k^2$.

As all vertices with degrees greater than k have been added to the cover and removed, all remaining vertices have a degree of at most k. Because all remaining edges in the graph are of the form $(x, v) \in E', x \in X, v \in A$ or $(u, v) \in E', u, v \in A$, we may immediately bound the number of remaining edges, $|E'| \leq k^2$. We now can just learn the entire graph in time $O(|E'| + |D|) = O(|E'|) = O(k^2)$ using pipelining. The leader vertex computes the optimal cover for G' and notifies all vertices in G' whether they should join it. ◀

▶ **Theorem 15.** *There exists a deterministic algorithm for k-MVC that terminates in $O(k^2)$ rounds.*

Proof. Our algorithm first uses Theorem 12 to estimate the diameter. Specifically, we first apply it for $k' = 2k$. If the vertices report LARGE, we follow the same approach as in the algorithm in the LOCAL model and reject. Otherwise, we proceed knowing the diameter is bounded by $4k$. For the case the diameter is bounded we can compute a unique leader and a BFS tree rooted at that leader in $O(k)$ rounds.

Let $c = O(1)$ such that the algorithm provided in Lemma 14 is guaranteed to terminate in ck^2 rounds for *any* graph G with a cover of size k. We run the algorithm for ck^2 rounds; if the procedure did not terminate, all vertices report that no k-sized cover exists. Finally, we count the number of nodes in the cover using the BFS tree and verify that indeed the size of the solution in G is bounded by k. If any node in the tree sees more than k identifiers of vertices that joined the cover, it notifies all vertices that the solution is invalid and thus no k-sized solution exists. ◀

A Randomized Algorithm. While we show a deterministic LOCAL algorithm for k-*MVC* that is optimal even if randomization is allowed, we have a gap of $\Theta(\min\{k, \log k \log n\})$ in our CONGEST round complexity. We now present a randomized algorithm with a $O\left(k + \frac{k^2 \log k}{\log n}\right)$ round complexity, thereby reducing the gap to $O(\log^2 k)$. This is achieved by the observation that while node identifiers are of length $\Theta(\log n)$, we can replace each node identifier with an $O(\log k)$-bit *fingerprint.* If there exists a cover of size k, there are at most $k + k^2 \leq 2k^2$ vertices in G' (after our reduction rule), and we can use ($\mathfrak{b} = (c+4)\log k + 1$)-bit fingerprints, for some $c > 0$, and get that the probability of collision (that two vertices have the same fingerprint) is at most $\binom{2k^2}{2} 2^{-\mathfrak{b}} < k^4 2^{-\mathfrak{b}+1} = k^{-c}$. Next, we run our deterministic algorithm, where each vertex considers its fingerprint as an identifier. Observe that since $|E'| \leq k^2$ and each edge encoding now requires $O(\log k)$ bits (for $c = O(1)$), the overall amount of bits sent to the leader is $O(|E'| \log k) = O(k^2 \log k)$. Since the diameter of the graph is $O(k)$, and $O(\log n)$ bits may be transmitted on every round on each edge, we use pipelining to get the round complexity in Theorem 16. Note that we only use fingerprints for the part of the algorithm which requires time quadratic in k. That is, checking the size of the diameter and validating the size of the solution are still done using the original identifiers.

Finally, if no cover of size k exists, we may have more than $2k^2$ vertices in G' and may get fingerprint collisions. Nevertheless, when using the BFS tree to send the (fingerprinted) identifiers to the leader, we can identify that such a collision happened and the leader can notify all vertices to report that no k-sized cover exists.

▶ **Theorem 16.** *For any $\delta = k^{-O(1)}$, there exists a randomized algorithm for k-MVC that terminates in $O\left(k + \frac{k^2 \log k}{\log n}\right)$ rounds, while being correct with probability at least $1 - \delta$.*

Approximations. As we may add all nodes of degree more than k to the cover, this bounds Δ, the maximum degree in the remaining graph, by k. We can now apply the algorithm of [5] which runs in $O(\log \Delta / \log \log \Delta + \log \epsilon^{-1} \log \Delta / \log^2 \log \Delta)$ and achieves a $(2+\epsilon)$-approximation. This immediately results in a deterministic $O(\log k / \log \log k + \log \epsilon^{-1} \log k / \log^2 \log k)$-round $(2+\epsilon)$-approximation algorithm in the CONGEST model. Further, since there exists a cover of size $OPT \leq k$, setting $\epsilon = 1/(k+1)$ implies that the resulting cover is of size $\lfloor OPT(2+\epsilon) \rfloor \leq 2OPT + \lfloor k/(k+1) \rfloor = 2OPT$. Thus, we conclude that our algorithm computes a 2-approximation for the problem in $O(\log^2 k/\log^2\log k)$ rounds. Unfortunately, while this succeeds if there indeed exists a solution of size k, validating the size of the solution takes $O(k)$ rounds.

We now expand the discussion and propose an algorithm that computes a $(2-\epsilon)$-approximation. For $\epsilon = o(1)$, this gives a better round complexity than our exact algorithm, while for $\epsilon = O(1/\sqrt{k})$ it improves the approximation ratio of the above 2-approximation while still terminating in $O(k)$ rounds. In the full version [7], we use this algorithm to derive the first $(2-\epsilon)$-approximation algorithm for the (non-parametric) *MVC* problem that runs in $o(n^2)$ rounds.

▶ **Theorem 17.** $\forall\epsilon \in [1/k, 1]$*, there exists a deterministic CONGEST algorithm for k-MVC that computes a* $(2-\epsilon)$*-approximation in* $O\left(k + (k\epsilon)^2\right)$ *rounds. For any* $\delta = (k\epsilon)^{-O(1)}$*, there also exists a randomized algorithm that terminates in* $O\left(k + \frac{(k\epsilon)^2\log(k\epsilon)}{\log n}\right)$ *rounds and errs with probability* $\leq \delta$*.*

Proof. We utilize the framework of Fidelity Preserving Transformations [14]. Intuitively, if there exists a vertex cover of size k in the original graph, and we remove two vertices u, v that share an edge, then the new graph must have a cover of size at most $k-1$. This allows us to reduce the parameter at the cost of introducing error (we add both u and v to the cover, while an optimal solution may only include one of them). This process is called $(1,1)$*-reduction step* as it reduces the parameter by 1 and increases the size of the cover (compared with an optimal solution) by at most 1. Roughly speaking, we repeat the process until the parameter reduces to $k\epsilon$, at which point we run an exact algorithm on the remaining graph.

[14] proved that for any $\alpha \leq 2$, repeating a $(1,1)$-reduction step until the parameter reduces to $k(2-\alpha)$ allows computing an α-approximation by finding an exact solution to the resulting subgraph and adding all vertices that have an edge that was reduced in the process. For our purpose, we set $\alpha = 2-\epsilon$; thus, the exact algorithms only need to find a cover of size $k\epsilon$.

Our algorithm begins by checking the diameter is $O(k)$ and finding a leader vertex v. This is doable in $O(k)$ rounds as having a vertex cover of size k guarantees that the diameter is $O(k)$. We proceed with applying the $(1,1)$-reduction steps. To that end, we compute a maximal matching M and send it to v, which requires $O(k)$ rounds. If $|M| \leq k(1-\epsilon)$, v instructs all matched vertices to enter the cover, and the algorithm terminates with a solution of size at most $2k(1-\epsilon) < k(2-\epsilon)$, as needed. If $|M| > k(1-\epsilon)$, the leader selects an arbitrary submatching $M' \subset M$ of size $k(1-\epsilon)$ and the reduction rules are simultaneously applied for every $e \in M'$. The remaining graph has a cover of size at most $k\epsilon$, at which point we apply the exact algorithms. Finally, we validate the size of the solution as in the above algorithms. By Theorems 15 and 16, we establish the correctness and runtime of our algorithms. ◀

5.3 CONGEST algorithms for *k-MaxM*

In the full version [7], we present deterministic and randomized upper bounds for the exact and approximate variants of the *k-MaxM* matching problem. Our exact algorithm first checks that the diameter is $O(k)$ and then uses a leader node to iteratively compute augmenting paths which extend the matching until it becomes of size k. Our approximations are derived from the Fidelity Preserving Transformations framework [14] and leverage the fact that a maximal matching (or a matching of size k) can be found in $O(k)$ rounds. We then sends the maximal matching to a leader node which arbitrarily selects a $(k/2 - O(k\epsilon))$-sized submatching and sends it to all nodes which then find the exact maximum matching on the residual graph.

References

1 Amir Abboud, Keren Censor-Hillel, and Seri Khoury. Near-Linear Lower Bounds for Distributed Distance Computations, Even in Sparse Networks. In *DISC*, 2016.

2 Mohamad Ahmadi, Fabian Kuhn, and Rotem Oshman. Distributed Approximate Maximum Matching in the CONGEST Model. In *DISC*, 2018.

3 R. Bar-Yehuda, K. Censor-Hillel, M. Ghaffari, and G. Schwartzman. Distributed Approximation of Maximum Independent Set and Maximum Matching. In *PODC*, 2017.

4 Reuven Bar-Yehuda, Keren Censor-Hillel, and Gregory Schwartzman. A Distributed $(2+\epsilon)$-Approximation for Vertex Cover in O(log Δ / ϵ log log Δ) Rounds. *J. ACM*, 2017.

5 R. Ben-Basat, G. Even, K.-i. Kawarabayashi, and G. Schwartzman. A Deterministic Distributed 2-Approximation for Weighted Vertex Cover in $O(\log n \log \Delta / \log^2 \log \Delta)$ Rounds. In *SIROCCO*, 2018.

6 R. Ben-Basat, G. Even, K.-i. Kawarabayashi, and G. Schwartzman. Optimal Distributed Covering Algorithms. In *DISC*, 2019.

7 Ran Ben-Basat, Ken-ichi Kawarabayashi, and Gregory Schwartzman. Parameterized Distributed Algorithms. `arXiv:1807.04900`.

8 Karl Bringmann and Sebastian Krinninger. A Note on Hardness of Diameter Approximation. *Information Processing Letters*, 2018.

9 Keren Censor-Hillel, Eldar Fischer, Gregory Schwartzman, and Yadu Vasudev. Fast Distributed Algorithms for Testing Graph Properties. In *DISC*, 2016.

10 Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and Near-Quadratic Lower Bounds for the CONGEST Model. In *DISC*, 2017.

11 R. Chitnis, G. Cormode, H. Esfandiari, M. Hajiaghayi, A. McGregor, M. Monemizadeh, and S. Vorotnikova. Kernelization via Sampling with Applications to Finding Matchings and Related Problems in Dynamic Graph Streams. In *SODA*, 2016.

12 R. H. Chitnis, G. Cormode, M. T. Hajiaghayi, and M. Monemizadeh. Parameterized Streaming: Maximal Matching and Vertex Cover. In *SODA*, 2015.

13 Guy Even, Orr Fischer, Pierre Fraigniaud, Tzlil Gonen, Reut Levi, Moti Medina, Pedro Montealegre, Dennis Olivetti, Rotem Oshman, Ivan Rapaport, and Ioan Todinca. Three Notes on Distributed Property Testing. In *DISC*, 2017.

14 M. R. Fellows, A. Kulik, F. Rosamond, and H. Shachnai. Parameterized approximation via fidelity preserving transformations. *J. of Computer and System Sciences*, 2018.

15 Manuela Fischer. Improved Deterministic Distributed Matching via Rounding. In *DISC*, 2017.

16 Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks Cannot Compute Their Diameter in Sublinear Time. In *SODA*, 2012.

17 Mohsen Ghaffari and Bernhard Haeupler. Distributed Algorithms for Planar Networks II: Low-Congestion Shortcuts, MST, and Min-Cut. In *SODA*, 2016.

18 Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. On the Complexity of Local Distributed Graph Problems. In *STOC 2017*, 2017.

19 Bernhard Haeupler, Jason Li, and Goran Zuzic. Minor Excluded Network Families Admit Fast Distributed Algorithms. In *PODC*, 2018.

20 D. G. Harris. Distributed approximation algorithms for maximum matching in graphs and hypergraphs. *ArXiv e-prints*, abs/1807.07645, 2018. `arXiv:1807.07645`.

21 Janne H. Korhonen and Joel Rybicki. Deterministic Subgraph Detection in Broadcast CONGEST. In *OPODIS*, 2017.

22 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local Computation: Lower and Upper Bounds. *J. ACM*, 2016.

23 Christoph Lenzen and Boaz Patt-Shamir. Fast Routing Table Construction Using Small Messages: Extended Abstract. In *STOC*, 2013.

24 Christoph Lenzen and Roger Wattenhofer. Leveraging Linial's locality limit. In *DISC*, 2008.

25 Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. Improved Distributed Approximate Matching. *J. ACM*, 2015.

26 Zvi Lotker, Boaz Patt-Shamir, and Adi Rosén. Distributed Approximate Matching. *SIAM J. Comput.*, 2009.

27 Dániel Marx. Parameterized complexity and approximation algorithms. *The Comp. J.*, 2008.

28 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. In *STOC*, 2011.