Distributed Lower Bounds for Ruling Sets

Alkida Balliu
University of Freiburg
Freiburg, Germany
alkida.balliu@cs.uni-freiburg.de

Sebastian Brandt
ETH Zurich
Zurich, Switzerland
brandts@ethz.ch

Dennis Olivetti
University of Freiburg
Freiburg, Germany
dennis.olivetti@cs.uni-freiburg.de

***Abstract*—Given a graph $G = (V, E)$, an (α, β)-ruling set is a subset $S \subseteq V$ such that the distance between any two vertices in S is at least α, and the distance between any vertex in V and the closest vertex in S is at most β. We present lower bounds for distributedly computing ruling sets.**

More precisely, for the problem of computing a $(2, \beta)$-ruling set (and hence also any (α, β)-ruling set with $\alpha > 2$) in the LOCAL model of distributed computing, we show the following, where n denotes the number of vertices, Δ the maximum degree, and c is some universal constant independent of n and Δ.

- **Any deterministic algorithm requires $\Omega\left(\min\left\{\frac{\log\Delta}{\beta\log\log\Delta}, \log_\Delta n\right\}\right)$ rounds, for all $\beta \leq c \cdot \min\left\{\sqrt{\frac{\log\Delta}{\log\log\Delta}}, \log_\Delta n\right\}$. By optimizing Δ, this implies a deterministic lower bound of $\Omega\left(\sqrt{\frac{\log n}{\beta\log\log n}}\right)$ for all $\beta \leq c\sqrt[3]{\frac{\log n}{\log\log n}}$.**
- **Any randomized algorithm requires $\Omega\left(\min\left\{\frac{\log\Delta}{\beta\log\log\Delta}, \log_\Delta\log n\right\}\right)$ rounds, for all $\beta \leq c \cdot \min\left\{\sqrt{\frac{\log\Delta}{\log\log\Delta}}, \log_\Delta\log n\right\}$. By optimizing Δ, this implies a randomized lower bound of $\Omega\left(\sqrt{\frac{\log\log n}{\beta\log\log\log n}}\right)$ for all $\beta \leq c\sqrt[3]{\frac{\log\log n}{\log\log\log n}}$.**

For $\beta > 1$, this improves on the previously best lower bound of $\Omega(\log^* n)$ rounds that follows from the 30-year-old bounds of Linial [FOCS'87] and Naor [J.Disc.Math.'91] (resp. $\Omega(1)$ rounds if $\beta \in \omega(\log^* n)$). For $\beta = 1$, i.e., for the problem of computing a maximal independent set (which is nothing else than a $(2, 1)$-ruling set), our results improve on the previously best lower bound of $\Omega(\log^* n)$ *on trees*, as our bounds already hold on trees.

***Keywords*-Ruling set; maximal independent set; distributed graph algorithms; lower bounds.**

Full Version

The full version of this paper can be found at https://arxiv.org/abs/2004.08282.

I. Introduction

In this work, we study the problems of finding maximal independent sets (MIS) and ruling sets in the LOCAL model of distributed computing. In the LOCAL model, each node of the input graph is considered as a computational device and each edge as a communication link. Computation proceeds in synchronous rounds, where in each round each node can send a message of arbitrary size to each neighbor and then, after the messages arrive, perform some local computation. Each node has to terminate at some point and then output its local part of the global solution, i.e., whether it is in the MIS (resp. ruling set) or not. For a more detailed introduction to the LOCAL model, we refer the reader to Section II-A.

MIS: The problem of finding an MIS in a given graph is one of the most central and well-studied problems in the LOCAL model. Already in the '80s, the very first papers of the area [1–5] gave first upper and lower bounds for the complexity of computing an MIS, and since then there has been an abundance of papers (e.g., [6–16]) studying the problem and variants thereof. A major open question was whether an MIS can be computed deterministically in a polylogarithmic number of rounds (see, e.g., [4], or Open Problem 11.2 in the book by Barenboim and Elkin [12])—this question was finally answered in the affirmative in a very recent breakthrough by Rozhoň and Ghaffari [15] on network decompositions. In contrast, if randomization is allowed, already more than 30 years ago, Luby [2] and Alon, Babai, and Itai [3] presented $O(\log n)$-round algorithms for solving MIS, where n denotes the number of nodes of the input graph. This is still the best randomized upper bound known if the complexity is expressed solely as a function of n.

On the lower bound side, the $\Omega(\log^* n)$-round bound from the '80s and early '90s by Linial [4] and Naor [5] was the state of the art, until Kuhn, Moscibroda, and Wattenhofer (KMW) [17] proved in 2004 that there is no algorithm computing an MIS in $t = f(\Delta) + g(n)$ rounds (even allowing randomization) if $f(\Delta) \in o(\log\Delta/\log\log\Delta)$ and $g(n) \in o(\sqrt{\log n/\log\log n})$. Here, $\log^*()$ denotes the iterated logarithm and Δ the maximum node degree. Finally, last year, the KMW bounds were improved and complemented by Balliu et al. [18] who showed that $f(\Delta) + g(n)$ rounds are not sufficient for deterministic algorithms if $f(\Delta) \in o(\Delta)$ and $g(n) \in o(\log n/\log\log n)$, and not sufficient for randomized algorithms if $f(\Delta) \in o(\Delta)$ and $g(n) \in o(\log\log n/\log\log\log n)$. Due to an $O(\Delta + \log^* n)$-

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00042

round upper bound by Panconesi and Rizzi [19], the linear dependency on Δ is tight.

While the above bounds imply that the complexity of MIS on general graphs must lie in the polylogarithmic (in n) regime, the situation on trees is far less clear. Both the KMW lower bounds and the lower bounds by Balliu et al. are achieved by first proving the same bounds for the problem of finding a maximal matching and then obtaining the MIS bounds as an immediate corollary due to the fact that maximal matching on general graphs is essentially the same problem as MIS on line graphs. As the line graph of any graph with $\Delta \geq 3$ contains a cycle (of length 3), both lower bounds are not applicable on trees; in fact, as there seems to be no way around line graphs in order to transform the maximal matching bounds to MIS, there is little hope that the proofs can be adapted to work on trees. Hence, on trees, the state of the art is given by the $\Omega(\log^* n)$-round lower bounds by Linial and Naor, exhibiting a large gap to the best known deterministic upper bound of $O(\log n / \log\log n)$ rounds on trees by Barenboim and Elkin [8]. This suggests the following question.

Question 1
Is polylogarithmic time needed for deterministically computing an MIS on trees or is there a (much) faster algorithm?

Ruling sets: Ruling sets are a generalization of maximal independent sets. Let $\alpha \geq 2$, $\beta \geq 1$ be integers. An (α, β)-ruling set S is a subset of the nodes of the input graph such that the distance between any two nodes from S is at least α and any node not contained in S has a distance of at most β to the closest node in S. An MIS is a $(2, 1)$-ruling set. We observe that an (α, β)-ruling set is also an (α', β')-ruling set for any $\alpha' \leq \alpha$ and $\beta' \geq \beta$, hence finding the latter is at least as easy as finding the former. In particular, the problem of finding a $(2, \beta)$-ruling set for some $\beta > 1$ is at least as easy as the problem of finding an MIS. Moreover, as our goal is to prove lower bounds, we can safely restrict attention to $\alpha = 2$ without affecting the generality of our results.

Due to their relation to MIS (but also as interesting combinatorial objects of their own), ruling sets have been a natural object of interest in the LOCAL model and are well-studied (see, e.g., [6, 11, 14, 20–22]). In particular, the computation of ruling sets often constitutes a useful subroutine in the computation of other objects, such as maximal matching [11], maximal independent set [14], or distributed coloring [23, 24]. This is not a surprise: also the computation of an MIS is an important step in many algorithms, and it is quite natural to replace this step by the computation of a $(2, \beta)$-ruling set for some $\beta > 1$, if the latter suffices and can be computed faster. Hence, from the perspective of applications, a lower bound for MIS that also applies to such ruling sets can be considered as substantially more robust than a lower bound that cannot be extended to ruling sets.

Unfortunately, there is a simple argument why the existing lower bounds for MIS by KMW and Balliu et al. cannot be extended to $(2, \beta)$-ruling sets: as mentioned before, those lower bounds are achieved on line graphs; however, on line graphs already a $(2, 2)$-ruling set can be found in $O(\log^* n)$ rounds as shown by Kuhn, Maus, and Weidner [25]. The best lower bound for $(2, \beta)$-ruling sets follows again from the lower bounds by Linial and Naor for MIS, and stands at $\Omega(\log^* n)$, both on trees and general graphs, up to some $\beta \in \Theta(\log^* n)$. For $\beta \in \omega(\log^* n)$, no non-constant lower bound is known. In contrast, for up to polylogarithmic β, the best upper bound (expressed solely as a function of n) for computing a $(2, \beta)$-ruling set is polylogarithmic in n [6, 16, 20].

Question 2
Is polylogarithmic time needed for deterministically computing a $(2, \beta)$-ruling set (for up to polylogarithmic β) or is there a (much) faster algorithm?

Round elimination: Traditionally, proving lower bounds in the LOCAL model has been a challenging task. Until 2015, to the best of our knowledge, only about a handful of (non-trivial, non-global) lower bounds were known [4, 5, 17, 26–28], with the only lower bound (as a function of n) beyond $\Omega(\log^* n)$ being the KMW lower bound. A major obstacle seemed to be the lack of techniques that could be used to obtain (improved) lower bounds.

In 2016, things changed when it was discovered that a technique used in the proof for Linial's $\Omega(\log^* n)$-round lower bound is more widely applicable: Brandt et al. [29] used the technique, now known under the name *round elimination*, to prove lower bounds for the Lovász Local Lemma (LLL), sinkless orientation (as a special case of the LLL) and Δ-coloring. Since then, round elimination has been used to prove lower bounds for a variety of problems [18, 30–34].

In 2019, Brandt [32] showed that round elimination can, in principle, be applied to (almost) any problem that is locally checkable[1], by providing a so-called *automatic* version of round elimination, which, roughly speaking, is a blueprint for obtaining a lower bound via round elimination in which the problem of interest can be inserted. Unfortunately, for most problems, a crucial step in the general blueprint is (perhaps far) beyond the reach

[1]For a definition, see Section II-B.

of current techniques, which is the reason why we have not seen a flurry of new lower bounds in the past year. By using additional techniques inside this framework, a number of new lower bounds have been achieved [18, 33, 34], but the framework itself is still far from being well-understood. As such, we believe that obtaining a better understanding of (automatic) round elimination is one of the most promising research directions in the LOCAL model currently available and crucial for the design of new lower bounds.

Informally, the general idea of round elimination is as follows. In order to prove a lower bound for some problem Π_0 of interest, we want to find a sequence of problems

$$\Pi_0 \to \Pi_1 \to \Pi_2 \to \dots$$

such that for any two consecutive problems Π_i, Π_{i+1}, we have $T_{i+1} \leq T_i - 1$ whenever $T_i > 0$, where T_j denotes the complexity of problem Π_j for any j. In other words, Π_{i+1} is at least one round faster solvable than Π_i as long as Π_i is not 0-round solvable, which we will call the *round elimination property*. Now all that is necessary for proving a lower bound of T for problem Π_0 is to show that problem Π_{T-1} is not 0-round solvable, or equivalently, that the first 0-round solvable problem in the sequence has index at least T.

Automatic round elimination explicitly generates such a sequence of problems for any locally checkable problem Π_0, by repeatedly applying a fixed process that takes some locally checkable problem Π_i as input and returns Π_{i+1}. The main issue with the obtained sequence is that the descriptions of the problems in the sequence usually become very complicated already for small indices; without applying any additional techniques, already the size of the problem description grows roughly doubly exponential *for each subsequent problem*. Hence, it is not surprising that the crucial step of determining the first 0-round solvable problem Π_j in the sequence cannot be performed (in general) with the currently available techniques. Moreover, even if one could keep the problem description sizes reasonably small, no general method how to find the desired problem Π_j is known.[2]

Nevertheless, when studying a specific problem Π_0, it seems reasonable to try to make the problems in the sequence easier to understand. All currently known lower bound proofs via automatic round elimination follow the idea of modifying the problems in the sequence in a way that preserves the round elimination property while simplifying the problem descriptions, as suggested in [32]. The proofs can be grouped into two categories, depending on the chosen modification.

1) There exists a constant c such that each problem in the sequence can be described[3] by using at most c output labels. Examples are [18, 33, 34].
2) The size of the problem description grows doubly exponentially when going from Π_i to Π_{i+1}, for all i. Examples are [31, 32].

The idea of the second approach is to simplify the *structure*[4] of the descriptions of the problems in the sequence, but roughly preserve the *size* of the descriptions. The lower bound is achieved by showing that as long as the description size of a problem in the sequence is in $o(n)$ (or $(o(\Delta))$), the problem is not 0-round solvable. Hence, this approach only yields lower bounds of $\Omega(\log^* n)$ (resp. $\Omega(\log^* \Delta)$).

In contrast, the first approach can yield higher lower bounds, but requires finding a sequence of problems that can be described with a constant number of labels. Considering that to obtain a *good* lower bound we also must make sure that we do not reach a 0-round solvable problem too fast, for many problems such a sequence might simply not exist. In fact, characterizing the set of problems (or at least interesting subsets thereof) that admit such a sequence is an interesting open problem mentioned in [33]. For instance, while we do not have a proof, we do not believe that for MIS such a sequence yielding a polylogarithmic lower bound exists. This discussion raises the following question.

Question 3
How can we design a problem sequence satisfying the round elimination property that yields a better lower bound than $\Omega(\log^* n)$ without restricting the problem descriptions to a constant number of labels?

A. Our results

We prove the following result for deterministic algorithms.

Theorem 1. *In the LOCAL model, any deterministic algorithm that solves the $(2, \beta)$-ruling set problem requires $\Omega\left(\min\left\{\frac{\log \Delta}{\beta \log\log \Delta}, \log_\Delta n\right\}\right)$ rounds, for all $\beta \leq c \cdot \min\left\{\sqrt{\frac{\log \Delta}{\log\log \Delta}}, \log_\Delta n\right\}$, for some constant c independent of n and Δ.*

[2]Note that it is usually easy to check for a given problem whether it can be solved in 0 rounds; the difficulty lies in first obtaining a concise (parameterized) description of the problems in the sequence.

[3]The description is required to be in a certain standardized form. For details, we refer to Section II-B.

[4]For instance, the simplification could consist in transforming a problem with complicated constraints using a large number of output labels into a (much easier to understand) coloring problem with a large number of colors.

By setting $\Delta \coloneqq 2^{\sqrt{\beta \log n \log\log n}}$, we maximize our lower bound as a function of n, thereby obtaining the following corollary.

Corollary 2. *In the LOCAL model, any deterministic algorithm that solves the $(2,\beta)$-ruling set problem requires $\Omega\left(\sqrt{\frac{\log n}{\beta \log\log n}}\right)$ rounds, for all $\beta \leq c\sqrt[3]{\frac{\log n}{\log\log n}}$, for some constant c independent of n and Δ.*

This settles Question 2 for all $\beta \leq c\sqrt[3]{\frac{\log n}{\log\log n}}$. As any (α,β)-ruling set is also a $(2,\beta)$-ruling set for all $\alpha > 2$, Theorem 1 also holds for (α,β)-ruling sets. Moreover, since the given lower bounds already hold on trees, we obtain the following corollary, by setting $\beta = 1$.

Corollary 3. *In the LOCAL model, any deterministic algorithm that solves MIS on trees requires $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ rounds.*

This settles Question 1. Corollaries 2 and 3 provide the first polylogarithmic lower bounds for ruling sets, and for MIS on trees. Due to an $O(\log n/\log\log n)$-round deterministic upper bound for MIS on trees by Barenboim and Elkin [8], and a polylogarithmic deterministic upper bound for $(2,\beta)$-ruling sets on general graphs following from the work by Ghaffari et al. [16], the only remaining question for the given range of β is the exponent in the polylog.

For randomized algorithms, we prove the following.

Theorem 4. *In the LOCAL model, any randomized algorithm that solves the $(2,\beta)$-ruling set problem w.h.p.*[5] *requires $\Omega\left(\min\left\{\frac{\log\Delta}{\beta\log\log\Delta}, \log_\Delta\log n\right\}\right)$ rounds, for all $\beta \leq c\cdot\min\left\{\sqrt{\frac{\log\Delta}{\log\log\Delta}}, \log_\Delta\log n\right\}$, for some constant c independent of n and Δ.*

By setting $\Delta \coloneqq 2^{\sqrt{\beta \log\log n \log\log\log n}}$, we maximize our lower bound as a function of n, thereby obtaining the following corollary.

Corollary 5. *In the LOCAL model, any randomized algorithm that solves the $(2,\beta)$-ruling set problem w.h.p. requires $\Omega\left(\sqrt{\frac{\log\log n}{\beta\log\log\log n}}\right)$ rounds, for all $\beta \leq c\sqrt[3]{\frac{\log\log n}{\log\log\log n}}$, for some constant c independent of n and Δ.*

Again, this bound already holds on trees and we obtain the following corollary for MIS.

Corollary 6. *In the LOCAL model, any randomized algorithm that solves MIS on trees w.h.p. requires $\Omega\left(\sqrt{\frac{\log\log n}{\log\log\log n}}\right)$ rounds.*

[5]As usual, we say that an algorithm solves a problem with high probability if the global success probability is at least $1 - 1/n$.

Note that Theorem 4 implies that there is no randomized algorithm that solves the $(2,\beta)$-ruling set problem w.h.p. in $t = f(\Delta) + g(n)$ rounds if $f(\Delta) \in o\left(\frac{\log\Delta}{\beta\log\log\Delta}\right)$ and $g(n) \in o\left(\sqrt{\frac{\log\log n}{\beta\log\log\log n}}\right)$. Hence, we obtain that the $O(\log\Delta + \log\log n/\log\log\log n)$-round randomized upper bound for MIS (and hence also $(2,\beta)$-ruling set) on trees by Ghaffari [14] cannot be improved substantially in both Δ and n simultaneously, for any indicated β. Furthermore, Corollary 6 provides the first progress on Open Problem 10.15 from the book by Barenboim and Elkin [12] (on the lower bound side), asking for the randomized complexity of MIS on trees.

Our results are achieved by designing a sequence of problems with the round elimination property for $(2,\beta)$-ruling sets, where the number of used labels is non-constant. More precisely, our problem sequence will satisfy that the number of labels used in the description of problem Π_i is in $\Theta(i^\beta/(\beta!))$. In particular, for the special case of MIS, the number of used labels grows linearly. Hence, our construction of the problem sequence provides an answer to Question 3.

B. Our techniques

In order to successfully apply the round elimination technique, two main ingredients are required. The first is *finding* a good problem family: we need to define some family $\{\Pi_{i\geq 0}\}$ such that the sequence $\Pi_0 \to \Pi_1 \to \dots$ satisfies the round elimination property and Π_0 is the problem for which we want to prove a lower bound. The second ingredient is *proving* that the defined sequence indeed satisfies the desired property.

While the second ingredient is technically involved, the conceptually crucial part is the first one, designing a good sequence of problems. Usually, when applying the round elimination technique, finding the right problem family involves some guessing.[6] For instance, in [18] the problem family was found by trying to make each subsequent problem in the sequence *look very similar* to the previous one while using the same output labels in the description (see [18, Section 3.7]). In the case where each problem in the family can be described using a constant number of labels, there is even very recent software available, written by Olivetti [35], that automatically searches the space of potential problems for small Δ. Unfortunately, for the MIS problem (and for ruling sets) this approach fails, suggesting that a constant number of labels is not sufficient. Instead, we propose a more explicit and perhaps surprising approach to find the desired problem family, by first proving an *upper bound* for the problem of interest such that the proof can be "represented" via a similar sequence of problems.

[6]In rare cases the sequence suggests itself, e.g., for sinkless orientation [29] the sequence is obtained by setting $\Pi_0 = \Pi_1 = \dots$

As explained in [32, 36], the round elimination technique can also be used to find upper bounds: Instead of finding a problem sequence with the round elimination property, i.e., with the property that $T_{i+1} \leq T_i - 1$, the idea is to find a problem sequence with the property that $T_{i+1} \geq T_i - 1$. This ensures that the index j of the first 0-round solvable problem Π_j in the sequence (if such a problem exists) is an upper bound for the complexity of Π_0. Accordingly, we will call a sequence satisfying $T_{i+1} \leq T_i - 1$ a *lower bound sequence* and a sequence satisfying $T_{i+1} \geq T_i - 1$ an *upper bound sequence*. We note that the *automatic sequence* provided by *automatic* round elimination is both a lower and an upper bound sequence since there we have $T_{i+1} = T_i - 1$; in fact, it can be seen as the *tightest* sequence with the property $T_{i+1} \geq T_i - 1$. In the following, we will use this automatic sequence to informally describe the intuition behind our approach.

Intuition behind our approach: In the round elimination framework, each problem is described via a list of "allowed" configurations that specify which local output label configurations around a node or on an edge are considered correct. As mentioned before, in the automatic sequence these descriptions grow very fast. On the other hand, due to the nature of 0-round algorithms, it seems to be the case that in the first (or more generally, any) 0-round solvable problem Π_j only very few of those allowed configurations are actually required for the correctness of a given 0-round algorithm. In other words, Π_j would still remain 0-round solvable if we removed a large number of the allowed configurations; moreover, the remaining part of the problem usually has an intuitive interpretation. Assuming that the previous problems in the automatic sequence behave similarly, we obtain the following intuition for each problem Π_i with $i \leq j$:

(1) There is some small part of the problem description that has some intuitive meaning and is relevant for solving the problem in $j - i$ rounds, and
(2) there are additional allowed configurations that seem to be an artifact of the automatic process that generates the sequence.

Intuitively, Part (1) can be thought of as the *essence* of the problem, and we argue that the information encoded therein should suffice to prove lower bounds. Hence, we would like to restrict attention to Part (1).

If we had a concise description of problem Π_j and the complete automatic sequence leading to Π_j, it would be straightforward to extract Part (1) of each problem and thereby obtain a comparably simple sequence $\Pi_0^* \to \Pi_1^* \to \dots$ of problems. However, there are two issues: first, we do not have feasible access to Π_j and the preceding sequence (otherwise we would be done), and second, for technical reasons, the obtained sequence is an upper bound sequence, but not a lower bound sequence (in general), i.e., even if we had such access, the fact that $T_{i+1} \leq T_i - 1$ is not satisfied prevents us from using the sequence in a *lower bound* proof.

To solve the second issue, we make use of so-called *wildcards*, a notion introduced in [18]. We show for the case of MIS and ruling sets that, perhaps surprisingly, adding a sufficient number of wildcards to the allowed configurations in the problems from $\Pi_0^* \to \Pi_1^* \to \dots$ turns the upper bound sequence into a lower bound sequence that is "tight enough" to yield a polylogarithmic lower bound.

Our solution to the first issue is to try to design an upper bound sequence $\Pi'_0 \to \Pi'_1 \to \dots$ that is as close to the desired sequence $\Pi_0^* \to \Pi_1^* \to \dots$ as possible, and then work with problem family $\{\Pi'_{i\geq 0}\}$ instead of $\{\Pi^*_{i\geq 0}\}$. As the latter sequence is unknown, our guideline for designing $\{\Pi'_{i\geq 0}\}$ will be *simplicity*, following the above intuition that Part (1) of each Π_i (i.e., Π_i^*) is small and intuitive. A key idea in the design will be to introduce a *coloring component* into the MIS and ruling set problems. Roughly speaking, the purpose of this coloring component is that, with enough care, we can make sure that only the coloring part of the problem description grows when we go from Π'_i to Π'_{i+1}, while the MIS (resp. ruling set) part remains unchanged. This allows us to keep the structure of the problems in the sequence comparably simple, which in turn allows us to determine at which point in the sequence the problems become 0-round solvable.

Essentially, our approach reduces the task of proving lower bounds to proving upper bounds, which usually is considered to be an easier task.[7] However, the designed algorithm should also have a "simple representation" as an upper bound sequence, and this does not seem to be the case for existing ruling set algorithms. Hence, we will design a new, genuinely different ruling set algorithm that gives state-of-the-art upper bounds in terms of Δ (which is the relevant dependency for the round elimination technique, from a technical perspective) and yields a simple upper bound sequence.

Approach: To summarize, our approach works as follows. First, we prove an upper bound for finding a $(2, \beta)$-ruling set (of which MIS is a special case) that can be represented by a comparably simple upper bound sequence. To this end, we consider the initial problem Π_0 of the sequence as "$(2, \beta)$-ruling set with some coloring

[7] While the current literature uses round elimination primarily to prove lower bounds, this statement arguably also holds for lower/upper bounds *via round elimination*. One main reason is that to make a problem given in the form specified by round elimination *harder* (a technique instrumental for the design of upper bound sequences), we can simply discard allowed configurations, while to make a problem *easier* (instrumental for lower bound sequences), more complicated operations have to be used.

component" and then introduce more and more colors into the problem over the course of the sequence, in a certain hierarchical manner. Second, we insert (an increasing number of) wildcards into the problems in our sequence, and prove that this turns the upper bound sequence into a lower bound sequence that yields a polylogarithmic lower bound.

While the individual parts of our approach are technically challenging, the approach itself is surprisingly simple. Hence, we believe that this general approach does not only work for MIS and ruling sets but should also be applicable to other problems; however, as it involves, e.g., finding an upper bound proof that can be described well via a sequence of problems, obtaining new bounds using this approach is not automatic. Moreover, we think that the idea of introducing a coloring component into problems that do not seem to have any particular relation to coloring should be more widely applicable; one intuitive reason is that, similar to wildcards, it gives a relatively simple way to represent *progress* towards 0-round solvability in the sequence, which seems like a necessary ingredient for designing a lower or upper bound sequence (which we can feasibly infer bounds from).

C. Further discussion of related work

MIS: The maximal independent set problem has been widely studied in the LOCAL model. Barenboim et al. showed that, if we also consider the dependency in Δ, MIS can be solved in $O(\log^2 \Delta) + 2^{O(\sqrt{\log \log n})}$ rounds [11]. Ghaffari improved this running time to $O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ [14]. The MIS problem has been studied also in specific classes of graphs [8, 9, 11, 37]. For example, for computing MIS on trees with randomized algorithms, Lenzen and Wattenhofer showed an $O(\sqrt{\log n} \log \log n)$-round algorithm [10]. This was later improved by Barenboim et al. to $O(\sqrt{\log n \log \log n})$ [11], and then further improved to $O(\sqrt{\log n})$ by Ghaffari [14]. Barenboim et al. also showed that MIS on trees can be solved in $O(\log \Delta \log \log \Delta + \log \log n / \log \log \log n)$ rounds [11]. Ghaffari later improved this bound to $O(\log \Delta + \log \log n / \log \log \log n)$ rounds [14].

Ghaffari studied MIS also in the CONGEST[8] model, giving a randomized algorithm with a running time of $\min\{O(\log \Delta \log \log n) + 2^{O(\sqrt{\log \log n \log \log \log n})}, \log \Delta \cdot 2^{O(\sqrt{\log \log n})}\}$ rounds [38]. This was later improved by Ghaffari and Portmann [39] to $O(\log \Delta \cdot \sqrt{\log \log n}) + 2^{O(\sqrt{\log \log n})}$ rounds.

[8]The CONGEST model is the same as the LOCAL model with the difference that in CONGEST the size of the messages is bounded by $O(\log n)$ bits. We refer the reader to Section II-A for more details on these models.

While all the above algorithms are randomized, Panconesi and Srinivasan provided a deterministic algorithm for solving MIS in $2^{O(\sqrt{\log n})}$ rounds [7]. Later, Barenboim, Elkin and Kuhn showed an $O(\Delta + \log^* n)$-round algorithm [13]. Very recently, Rozhoň and Ghaffari proved that MIS can be solved deterministically in $\mathrm{poly}(\log n)$ rounds [15]. Meanwhile, the exponent of the polylog has been improved by Ghaffari et al. [16].

Ruling sets: Ruling sets have been introduced by Awerbuch et al. [6], where the authors showed how to construct $(\alpha, O(\alpha \log n))$-ruling sets in $O(\alpha \log n)$ deterministic rounds in the LOCAL model. Since then, there have been several works in this direction both in the deterministic and randomized setting, and both in the LOCAL and CONGEST models of distributed computing. In fact, as far as deterministic algorithms are concerned, Schneider, Elkin, and Wattenhofer showed how to get $(2, \beta)$-ruling sets in $O(\beta \Delta^{2/\beta} + \log^* n)$ rounds in the LOCAL model [40]. Notice that, in the LOCAL model, it is possible to get an $(\alpha, (\alpha-1)\beta)$-ruling set of a graph G by just computing a $(2, \beta)$-ruling set on the power graph $G^{\alpha-1}$. This reasoning does not directly apply to the CONGEST model, where the size of the messages is bounded by $O(\log n)$ bits. However, the algorithm of Awerbuch et al. can be modified to work in the CONGEST model. In fact, Henzinger, Krinninger, and Nanongkai sketched the arguments that show how to adapt it and get a CONGEST algorithm that gives $(\alpha, O(\alpha \log n))$-ruling sets in $O(\alpha \log n)$ rounds [41]. Later on, Kuhn, Maus, and Weidner gave a formal proof of these arguments [25]. Also, the same authors showed how to obtain $(\alpha, (\alpha-1)\lceil \log_B n \rceil)$-ruling sets ($B \geq 2$) in $O(\alpha B \log_B n)$ rounds. As a corollary, they get the same trade offs as in [40] and obtain a $(2, \beta)$-ruling set (for $\beta > 2$) in $O(\beta \Delta^{2/\beta} + \log^* n)$ rounds for the CONGEST model.

If randomness is allowed, Gfeller and Vicari showed how to compute a relaxed version of a $(2, O(\log \log \Delta))$-ruling set, where each node in the ruling set is allowed to have at most $O(\log^5 n)$ neighbors also in the ruling set, in $O(\log \log \Delta)$ rounds [21], and by then applying the algorithm of [40] on the graph induced by selected nodes, we can obtain an algorithm for $(2, \log \log n)$-ruling sets running in $O(\log \log n)$ time. Kothapalli and Pemmaraju showed how to compute $(2, 2)$-ruling sets in $O\left(\frac{\log \Delta}{(\log n)^{\varepsilon}} + (\log n)^{1/2+\varepsilon}\right)$ rounds, for any $\varepsilon > 0$ [42]. One year later, Bisht, Kothapalli, and Pemmaraju provided a sparsifying procedure that can be used, together with some MIS algorithm, to obtain $(2, \beta)$-ruling sets (in a runtime that depends on the respective MIS algorithm) [22]. For instance, by combining this sparsifying procedure with the MIS algorithm by Barenboim et al. [11], a $(2, \beta)$-ruling set can be computed in $O(\beta \log^{1/(\beta-1/2)} \Delta) + 2^{O(\sqrt{\log \log n})}$ rounds.

By using the improved MIS algorithm by Ghaffari [14] instead, we obtain a runtime of $O(\beta \log^{1/\beta} \Delta) + 2^{O(\sqrt{\log \log n})}$ rounds, which can in turn be improved to $O(\beta \log^{1/\beta} \Delta) + \mathrm{poly}(\log \log n)$ rounds by making use of the $\mathrm{poly}(\log n)$-round network decomposition algorithm by Rozhoň and Ghaffari [15]. Lastly, Pai et al. studied randomized ruling sets in the CONGEST model. They showed how to compute $(2,3)$-ruling sets in $O(\log n / \log \log n)$ rounds, and $(2,2)$-ruling sets in $O(\log \Delta (\log n)^{1/2+\varepsilon} + \varepsilon \log n \log \log n)$ rounds [43].

II. Preliminaries

A. Model

The LOCAL model: The model of computation used in this paper is the widely studied LOCAL model of distributed computing [44]. In this model, each node of the input graph has a unique identifier from 1 to $\mathrm{poly}\, n$, and the computation proceeds in synchronous rounds. At each round, each node can send a message of arbitrary size to each neighbor, and, after receiving the messages from its neighbors, perform some local computation of arbitrary complexity. In the LOCAL model, each node knows initially its unique identifier and its degree. As commonly done in this context, we also assume that each node knows the number of nodes n in the graph (or a polynomial upper bound of it) and the maximum degree Δ. Clearly, this can make the task of proving lower bounds only harder. Each node executes the same algorithm (which is what we call a distributed algorithm), and each node has to terminate at some point and then output its local part of the global solution, e.g., in the case of MIS whether the node is in the MIS or not. The runtime of such a distributed algorithm is the number of synchronous round until the last node terminates. In the randomized version of the LOCAL model, each node additionally has access to a stream of private random bits. We will study Monte Carlo algorithms that solve the desired problem with high probability, that is, the global success probability must be at least $1 - 1/n$.

Another well-studied model in the area of distributed computing is the CONGEST model [44], which is defined as the LOCAL model with the only difference that the size of each message sent between the nodes is restricted to $O(\log n)$ bits. As the CONGEST model is strictly weaker than the LOCAL model, our lower bounds hold also in the CONGEST model.

The Port Numbering model: Our results hold in the LOCAL model of distributed computing, however, for technical reasons we pass through the Port Numbering (PN) model, in the sense that we first show how to obtain our results in the PN model, and then lift them to the LOCAL model. The PN model is a variant of the LOCAL model where nodes do not have identifiers, but each node v has an internal ordering of its incident edges given by an arbitrary assignment of (pairwise distinct) so-called *port numbers* from 1 to $\deg(v)$ to the edges. This model is also synchronous, and, as in the LOCAL model, the size of the messages and the computational power of each node is not bounded. In the randomized version of the PN model, each node has access to a stream of private random bits and we require that randomized algorithms succeed with high probability.

To be able to apply the round elimination framework, we also need that *edges* have port numbers; in other words, we assume that an orientation of the edges is given. However, this is just a technical detail that does not have any effect on our argumentation, and as such we will ignore it in the following. Note that, in the LOCAL model, such an edge orientation can be obtained from the unique identifiers in one round; therefore also the presented upper bounds do not change asymptotically if we assume that an edge orientation is given.

B. Problems

In the round elimination framework a problem is characterized by an alphabet Σ of labels, a *node constraint* $\mathcal{N}$ and an *edge constraint* $\mathcal{E}$. We will only consider problems defined on Δ-regular graphs in this formalism, since, as we will later see, this is enough for our purposes. The node constraint $\mathcal{N}$ is a collection of words of length Δ over the alphabet Σ, and the edge constraint $\mathcal{E}$ is a collection of words of length 2 over Σ. The same label can appear several times in a word and the order of the elements that compose a word does not matter, hence each word technically is a multiset. We call a word in $\mathcal{N}$ a *node configuration* and a word in $\mathcal{E}$ an *edge configuration*.

Let $G = (V, E)$ be our input graph and let $A = \{(v, e) \in V \times E \mid v \in e\}$ be the set that contains all pairs $(\text{node}, \text{incident edge})$. The output for a problem in this formalism is given by a labeling of each $(v, e) \in A$ with one element from Σ. Put otherwise, each node has to output an element of the set Σ on each incident edge. We say that such an output is *correct* if it satisfies $\mathcal{N}$ and $\mathcal{E}$, i.e., for each node $v' \in V$, the collection of Δ output labels assigned to the $(v, e) \in A$ with $v = v'$ is a node configuration listed in $\mathcal{N}$, and for each edge $e' \in E$, the two output labels assigned to the (v, e) with $e = e'$ is an edge configuration listed in $\mathcal{E}$.

We use regular expressions to represent (collections of) node and edges configurations. For example, the expression $\mathsf{P}\mathsf{O}^{\Delta-1}$ describes a node configuration that consists of exactly one label P and $\Delta - 1$ labels O. Similarly, the expression $\mathsf{M}[\mathsf{P}\mathsf{O}]$ describes a collection of edge configurations that consists of one label M and the other label can be either P or O, i.e., $\mathsf{M}[\mathsf{P}\mathsf{O}] = \{\mathsf{M}\mathsf{P}, \mathsf{M}\mathsf{O}\}$. We call a part of an expression such as

$[\mathsf{PO}]$, where we have a choice between different labels, a *disjunction*. While technically an expression containing a disjunction describes a set of configurations, we will use the term *configuration* also for such an expression, for simplicity. In order to explicitly specify that the expression contains a disjunction, we will use the term *condensed configuration*. Moreover, we will say that a configuration is *contained in* a condensed configuration if we can obtain the former from the latter by picking a choice in each disjunction.

With a few exceptions, all problems from a large class of problems of interest in the LOCAL model, so-called *locally checkable* problems, can be described in this formalism. A locally checkable problem is simply a problem for which the correctness of a solution can be verified by checking whether the $O(1)$-hop neighborhood of each node is locally correct. For technical reasons, locally checkable problems whose definitions involve small cycles (such as determining for each node whether it is contained in a triangle) cannot be described in the above formalism. Hence, for simplicity, in the remainder of the paper we will use the term "locally checkable" for (locally checkable) problems that are not of this kind.

In the following we present two examples highlighting how we arrive at the description of a problem in the new formalism. In the full version of this paper, we show more formally that the given descriptions capture the MIS and ruling set problems.

Example: Let us see, for example, how we can describe the MIS problem in this formalism. We define $\Sigma = \{\mathsf{M}, \mathsf{P}, \mathsf{O}\}$. We will use the node constraint to represent whether a node is in the independent set or not. Nodes that are in the independent set must output the label M (as in "in the MIS") on all incident edges. For nodes that are not in the independent set, we have to make sure that at least one neighbor is in the independet set. To this end, we require that nodes that are not in the independent set *point* to a neighbor that is in the independent set, thereby ensuring maximality. In other words, these nodes must output a label P (as in "pointer") on exactly one incident edge and the label O (as in "other") on all the other $\Delta-1$ incident edges. Now the edge constraint must guarantee that no two neighbors are in the MIS, hence $\mathsf{MM} \notin \mathcal{E}$, and that a pointer points to a node that is in the MIS, hence $\mathsf{PM} \in \mathcal{E}$, but $\mathsf{PP} \notin \mathcal{E}$, and $\mathsf{PO} \notin \mathcal{E}$. In order to capture the situation where a node not in the MIS has several neighbors in the MIS, we must allow $\mathsf{MO} \in \mathcal{E}$. Also, since two nodes not in the MIS may be neighbors, $\mathsf{OO} \in \mathcal{E}$. This leads to the following formal definition of the node and edge constraint.

$$\begin{array}{ll} \mathcal{N}\colon & \mathcal{E}\colon \\ \mathsf{M}^{\Delta} & \mathsf{M}[\mathsf{PO}] \\ \mathsf{PO}^{\Delta-1} & \mathsf{OO} \end{array}$$

Example: In order to encode the $(2,2)$-ruling set problem we need to use a larger set of labels compared to the one used for the MIS problem. Let $\Sigma = \{\mathsf{M}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{O}_1, \mathsf{O}_2\}$. Intuitively, similarly as before, the M label can be seen as the "I am in the ruling set" label, while the labels P_1 and P_2 are "pointer" labels that are used to point to nodes in the ruling set and to nodes that are at distance 1 from a node in the ruling set. Notice that, as a $(2,1)$-ruling set (i.e., MIS) solves the $(2,2)$-ruling set problem, the encoding of the $(2,2)$-ruling set problem will contain the node and edge configurations of the MIS problem. For instance, a node in the ruling set will output M^{Δ}. Nodes at distance 1 from a node in the ruling set may output either $\mathsf{P}_1\mathsf{O}_1^{\Delta-1}$ or $\mathsf{P}_2\mathsf{O}_2^{\Delta-1}$, but those at distance 2 must output $\mathsf{P}_2\mathsf{O}_2^{\Delta-1}$. On the edge side, we must guarantee that, for any pair of nodes in the ruling set, they do not share an edge, hence $\mathsf{MM} \notin \mathcal{E}$. Also, a pointer of type 1 must point to a node in the ruling set, while a pointer of type 2 must point to a node at distance at most 1 from a node in the ruling set, hence $\mathsf{M}[\mathsf{P}_1\mathsf{P}_2] \in \mathcal{E}$ and $\mathsf{O}_1\mathsf{P}_2 \in \mathcal{E}$. On the other hand, we want to forbid bad pointing. In fact, nodes at distance 1 from a node in the ruling set must not be able to point to a node that is not in the ruling set, hence $\mathsf{P}_1[\mathsf{O}_1\mathsf{O}_2\mathsf{P}_1\mathsf{P}_2] \notin \mathcal{E}$. Also, nodes at distance 2 from a node in the ruling set must not point to another node that is at distance 2 as well, hence $\mathsf{P}_2[\mathsf{O}_2\mathsf{P}_2] \notin \mathcal{E}$. More precisely, the $(2,2)$-ruling set problem can be encoded in the formalism as follows.

$$\begin{array}{ll} \mathcal{N}\colon & \mathcal{E}\colon \\ \mathsf{M}^{\Delta} & \mathsf{M}[\mathsf{P}_1\mathsf{O}_1\mathsf{P}_2] \\ \mathsf{P}_1\mathsf{O}_1^{\Delta-1} & \mathsf{O}_1[\mathsf{O}_1\mathsf{O}_2\mathsf{P}_2] \\ \mathsf{P}_2\mathsf{O}_2^{\Delta-1} & \mathsf{O}_2\mathsf{O}_2 \end{array} \tag{1}$$

C. Round elimination

In our proofs, we will use the result of [32, Theorem 4.3], that is at the core of the round elimination technique. On a high level, this theorem says that, on Δ-regular high-girth graphs, given a locally checkable problem Π with time complexity T, there exists a locally checkable problem Π'' with time complexity $T-1$. The procedure of showing this theorem goes through an intermediate problem, that we call Π'. Given Π, Brandt [32] shows how to construct first Π' and then Π''. In the following, we will formally define these problems. In the full version of this paper, we provide an example where we compute Π' and Π'' starting from a specific problem

Π. Let Σ_Π, $\mathcal{N}_\Pi$, and $\mathcal{E}_\Pi$ be the alphabet of labels, the node constraint, and the edge constraint for problem Π, respectively.

Problem Π': In order to define problem Π', we must define the alphabet $\Sigma_{\Pi'}$, the node constraint $\mathcal{N}_{\Pi'}$, and the edge constraint $\mathcal{E}_{\Pi'}$.

- $\Sigma_{\Pi'}$: The set of labels for Π' is the set of all non-empty subsets of Σ_Π, i.e., $\Sigma_{\Pi'} = 2^{\Sigma_\Pi} \setminus \{\{\}\}$.
- $\mathcal{E}_{\Pi'}$: We construct the edge constraint in the following way. Consider a configuration $\mathsf{A}_1\mathsf{A}_2$, where $\mathsf{A}_1, \mathsf{A}_2 \in \Sigma_{\Pi'}$, such that, for all $(\mathsf{a}_1, \mathsf{a}_2) \in \mathsf{A}_1 \times \mathsf{A}_2$, it holds that $\mathsf{a}_1\mathsf{a}_2 \in \mathcal{E}_\Pi$ (notice that, by construction of $\Sigma_{\Pi'}$, it holds that $\mathsf{a}_1, \mathsf{a}_2 \in \Sigma_\Pi$). Let $\mathcal{A}$ be the collection of all such configurations. We call a configuration $\mathsf{A}_1\mathsf{A}_2 \in \mathcal{A}$ *non-maximal* if there exists another configuration $\mathsf{A}'_1\mathsf{A}'_2 \in \mathcal{A}$ such that $\mathsf{A}_i \subseteq \mathsf{A}'_i$ for all $i \in \{1,2\}$, and $\mathsf{A}_i \subsetneq \mathsf{A}'_i$ for at least one $i \in \{1,2\}$. In other words, if we have a configuration $\mathsf{A}'_1\mathsf{A}'_2 \in \mathcal{A}$ that is obtained from $\mathsf{A}_1\mathsf{A}_2$ by adding at least one element to at least one of A_1 and A_2, then we say that $\mathsf{A}_1\mathsf{A}_2$ is non-maximal. We delete all non-maximal configurations from $\mathcal{S}$, and what remains is our set $\mathcal{E}_{\Pi'}$ of configurations.
- $\mathcal{N}_{\Pi'}$: Consider a configuration $\mathsf{B}_1\mathsf{B}_2\dots\mathsf{B}_\Delta$ where $\mathsf{B}_i \in \Sigma_{\Pi'}$ for all $i \in \{1,\dots,\Delta\}$, such that there exists a tuple $(\mathsf{b}_1,\dots,\mathsf{b}_\Delta) \in \mathsf{B}_1 \times \dots \times \mathsf{B}_\Delta$ such that $\mathsf{b}_1\mathsf{b}_2\dots\mathsf{b}_\Delta \in \mathcal{N}_\Pi$. Let $\mathcal{B}$ be the collection of all such configurations. We delete from the set $\mathcal{B}$ all configurations that contain some set B_i that does not appear in any configuration in $\mathcal{E}_{\Pi'}$. The modified set $\mathcal{B}$ is our set $\mathcal{N}_{\Pi'}$.

For simplicity, we can (and will) assume that all labels that occur neither in $\mathcal{E}_{\Pi'}$, nor in $\mathcal{N}_{\Pi'}$, are also removed from $\Sigma_{\Pi'}$.

Problem Π'': Similarly as before, we need to define the alphabet $\Sigma_{\Pi''}$, the node constraint $\mathcal{N}_{\Pi''}$, and the edge constraint $\mathcal{E}_{\Pi''}$.

- $\Sigma_{\Pi''}$: The set of labels for Π'' is the set of all non-empty subsets of $\Sigma_{\Pi'}$, i.e., $\Sigma_{\Pi''} = 2^{\Sigma_{\Pi'}} \setminus \{\{\}\}$.
- $\mathcal{N}_{\Pi''}$: The node constraint is constructed as follows. Consider a configuration $\mathsf{B}_1\mathsf{B}_2\dots\mathsf{B}_\Delta$ where $\mathsf{B}_i \in \Sigma_{\Pi''}$ for all $i \in \{1,\dots,\Delta\}$, such that for all $(\mathsf{b}_1,\dots,\mathsf{b}_\Delta) \in \mathsf{B}_1 \times \dots \times \mathsf{B}_\Delta$ it holds that $\mathsf{b}_1\mathsf{b}_2\dots\mathsf{b}_\Delta \in \mathcal{N}_{\Pi'}$. Let $\mathcal{B}$ be the collection of all such configurations. We delete from $\mathcal{B}$ all non-maximal configurations, i.e., all those configurations $\mathsf{B}_1\dots\mathsf{B}_\Delta$ such that there exists some other configuration $\mathsf{B}'_1\dots\mathsf{B}'_\Delta$ that is obtained from the former by adding at least one element to at least one of the B_i sets. After performing these deletions, we set $\mathcal{N}_{\Pi''} = \mathcal{B}$.
- $\mathcal{E}_{\Pi''}$: Consider a configuration $\mathsf{A}_1\mathsf{A}_2$, where $\mathsf{A}_1, \mathsf{A}_2 \in \Sigma_{\Pi''}$, such that there exists a pair $(\mathsf{a}_1, \mathsf{a}_2) \in \mathsf{A}_1 \times \mathsf{A}_2$ such that $\mathsf{a}_1\mathsf{a}_2 \in \mathcal{E}_{\Pi'}$. Let $\mathcal{A}$ be the collection of all such configurations. We delete from the set $\mathcal{A}$ all configurations that contain some set A_1 or A_2 that does not appear in any configuration in $\mathcal{N}_{\Pi''}$, then we set $\mathcal{E}_{\Pi''} = \mathcal{A}$.

Again, we can (and will) assume that all labels that occur neither in $\mathcal{N}_{\Pi''}$, nor in $\mathcal{E}_{\Pi''}$, are also removed from $\Sigma_{\Pi''}$.

As Π' is uniquely defined by Π, we can define a function $\mathcal{R}(\cdot)$ that takes Π as input and returns Π'. Similarly, as Π'' is uniquely defined by Π', we can define a function $\overline{\mathcal{R}}(\cdot)$ that takes Π' as input and returns Π''. With these definitions, we have $\Pi'' = \overline{\mathcal{R}}(\mathcal{R}(\Pi))$. Note that $\overline{\mathcal{R}}(\cdot)$ can take any problem as input that is of the form specified by round elimination—it is not necessary that the input problem has been obtained by applying $\mathcal{R}(\cdot)$ to some problem.

Now [32, Theorem 4.3] provides the following relation between a problem Π and $\overline{\mathcal{R}}(\mathcal{R}(\Pi))$ that provides the fundament for automatic round elimination. For technical reasons, the theorem itself only holds in the port numbering model, but we will show later how to lift the obtained bounds to the LOCAL model.

Theorem 7 ([32], rephrased)**.** *Let $T > 0$. Consider a class $\mathcal{G}$ of graphs[9] with girth at least $2T + 2$, and some locally checkable problem Π. Then, there exists an algorithm that solves problem Π on $\mathcal{G}$ in T rounds if and only if there exists an algorithm that solves problem $\overline{\mathcal{R}}(\mathcal{R}(\Pi))$ in $T - 1$ rounds.*

In more technical detail, for any pair (n, Δ), Theorem 7 holds for graph classes $\mathcal{G} = \mathcal{G}(n, \Delta)$ consisting of n-node graphs with maximum degree Δ and girth at least $T = T(n, \Delta) > 0$. However, for simplicity, we will usually omit the dependency on n and Δ. We note that Theorem 7 also holds if we add a proper input vertex coloring to the setting. Moreover, we will assume that the input graphs satisfy the given girth requirement whenever we apply Theorem 7.

An interesting fact that we have not seen mentioned in [32] (or any other work) is that the equivalence breaks only *in one direction* when we go from high-girth graphs to general graphs: it is straightforward to go through the proof of [32, Theorem 4.3] and check that even on general graphs, Π can be solved in 1 round given a solution to $\overline{\mathcal{R}}(\mathcal{R}(\Pi))$. In other words, $\overline{\mathcal{R}}(\mathcal{R}(\Pi))$ is *at most* one round faster solvable than Π. Hence, any upper bound achieved via automatic round elimination holds on general graphs, both in the port numbering model and the LOCAL model (as the latter is a stronger model).

[9]Technically, the class of graphs has to satisfy a certain property, called t-independence in [32], but since it is straightforward to check that our considered class of Δ-regular high-girth graphs satisfies this property, we omit this detail.

In particular, this is true for our upper bounds for ruling sets.

Generalizing to non-regular graphs: As mentioned before, in this paper we will restrict attention to regular graphs. Since we are proving lower bounds, this does not affect the generality of our results; however, for the upper bound we prove along the way, some additional step is required to lift the bound to general graphs. In its full generality, the round elimination framework can also be applied to non-regular graphs, and the arguments in our upper bound would essentially remain the same; however, describing the framework formally is somewhat cumbersome. Hence, we will choose a different route to show that our upper bound holds on general graphs: we will present a "human-understandable" version of the algorithm obtained by round elimination for which it will be easy to check that its correctness is not affected by having nodes of different degrees.

D. Roadmap

In the full version of the paper, we will start by defining a family of problems $\Pi_{\Delta,\beta}(v,x)$, for which we will later show how it relates to the $(2,\beta)$-ruling set problem. The parameter $v = [v_0,\dots,v_\beta]$ is a list of non-negative numbers, that can be interpreted as a number of colors. Intuitively, the problem $\Pi_{\Delta,\beta}(v,x)$ can be solved in 0 rounds if we are given some vertex coloring with $\mathrm{size}(v) \coloneqq \sum_{i=0}^{\beta} v_i$ colors. The parameter x is some relaxation parameter: we will allow nodes to violate edge constraints on at most x of their incident edges.

We will use the round elimination theorem to relate problems of this family. In a first step, we will compute the problem that we obtain by applying our operator $\mathcal{R}(\cdot)$ to $\Pi_{\Delta,\beta}(v,x)$.

In a second step, we will prove upper bounds for the $(2,\beta)$-ruling set problem. We will consider a subset of the problems of the family, that is, those where parameter x is set to be 0. We start this second step by showing that $\overline{\mathcal{R}}(\Pi'_{\Delta,\beta}(v,0))$ is at least as easy as some other problem of the family, that is $\Pi_{\Delta,\beta}(v',0)$, where v' is the inclusive prefix sum of v (i.e., $v'_i = \sum_{j\le i} v_j$). The round elimination theorem will imply that, given a solution for $\Pi_{\Delta,\beta}(v',0)$, we can obtain a solution for $\Pi_{\Delta,\beta}(v,0)$ in at most one round of communication. We will finally combine multiple steps of such reasoning to obtain upper bounds: we will show how parameter v evolves over multiple steps. Crucially, a solution for $\Pi_{\Delta,\beta}([1,0,\dots,0],0)$ will directly imply a solution for the $(2,\beta)$-ruling set problem, and by repeatedly applying the round elimination theorem we will obtain some problem $\Pi_{\Delta,\beta}(v',0)$ where $\mathrm{size}(v')$ is at least as large as the number of colors in the given vertex coloring. We will first prove an upper bound on the number of steps required to obtain such a problem, thereby giving an upper bound on the time complexity of the algorithm. Then, we will provide a human-understandable version of the round-elimination-generated algorithm, in order to argue that this algorithm does not only work on regular graphs, but on all graphs. In particular, we will show that, given a c-coloring, the minimum t such that $\binom{\beta+t}{\beta} \ge c$ is an upper bound for $(2,\beta)$-ruling sets.

In a third step, we will prove lower bounds for the $(2,\beta)$-ruling set problem. The main idea here will be to show that, by increasing parameter x, we can essentially relate the problems of the family in the same way as we do for the upper bounds. That is, we can get the same evolution of parameter v as in the upper bound, at the price of increasing parameter x. Essentially, this will allow us to use the ideas obtained from the upper bound to get a lower bound.

Finally, we will show how to lift the obtained lower bounds from the port numbering model to the LOCAL model.

III. Open problems

In this work, we proved that the deterministic complexity of computing $(2,\beta)$-ruling sets is at least $\mathrm{poly}\log n$, unless β is too large. Combined with existing $\mathrm{poly}\log n$ upper bounds, our results imply that the deterministic complexity of ruling sets lies in the $\mathrm{poly}\log n$ region. An interesting open question is how many $\log$ factors are required exactly.

Another open question concerns the techniques that we use: We first prove a lower bound for a constant-radius checkable version of the ruling set problem, and then transform this bound into a lower bound for the original problem, where we lose an additive β-factor. Hence, currently our technique is not capable to prove lower bounds for $(2,\beta)$-ruling sets where β is so large that the problem can be solved in $o(\beta)$ rounds. An open question is to get rid of this restriction. For example, can we show that finding $(2,(\log n)^{1/3})$-ruling sets requires $\Omega(\log^{1/3} n/\log^{1/2}\log n)$ rounds with deterministic algorithms?

While we now have a good picture of the dependency of the complexity of $(2,\beta)$-ruling sets on n, the dependency on Δ is far less clear. Even in the case where a $(\Delta+1)$-vertex coloring is provided, the current best upper bound is $O(\beta\Delta^{1/\beta})$ rounds. Note that, for constant values of β, we get a complexity that is polynomial in Δ, while the lower bound that we provide lies in the $\mathrm{poly}\log\Delta$ region. Hence, there is an exponential gap between the current upper and lower bounds, as a function of Δ. We know that, on general graphs, any algorithm that solves MIS in time $f(\Delta)+g(n)$, must have $f(\Delta)=\Omega(\Delta)$, or both

$g(n) = \Omega(\log\log n / \log\log\log n)$ for randomized algorithms and $g(n) = \Omega(\log n / \log\log n)$ for deterministic ones [18], and we think that a necessary step for really understanding the Δ-dependency of $(2, \beta)$-ruling sets is to first prove an $\Omega(\Delta)$ lower bound for MIS on trees.

Finally, a number of interesting open questions revolve around our new technique of proving a lower bound via an upper bound. Can we characterize the problems that allow such an approach? What properties does an algorithm have to have to be well-representable as an upper bound sequence? Can the related technique of introducing a coloring component into non-coloring problems be successfully applied to other problems? We believe that finding answers to these and related questions will constitute an important step towards a better understanding of the round elimination technique.

ACKNOWLEDGMENTS

We are very thankful to the anonymous reviewers for their fruitful comments. We would like to thank Mohsen Ghaffari, Fabian Kuhn, Yannic Maus, and Julian Portmann for helpful comments on related works.

REFERENCES

[1] R. M. Karp and A. Wigderson, "A fast parallel algorithm for the maximal independent set problem," *Journal of ACM*, vol. 32, pp. 762–773, 1985.

[2] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM Journal on Computing*, vol. 15, no. 4, pp. 1036–1053, 1986.

[3] N. Alon, L. Babai, and A. Itai, "A fast and simple randomized parallel algorithm for the maximal independent set problem," *Journal of Algorithms*, vol. 7, no. 4, pp. 567–583, 1986.

[4] N. Linial, "Locality in Distributed Graph Algorithms," *SIAM Journal on Computing*, vol. 21, no. 1, pp. 193–201, 1992.

[5] M. Naor, "A lower bound on probabilistic algorithms for distributive ring coloring," *SIAM Journal on Discrete Mathematics*, vol. 4, no. 3, pp. 409–412, 1991.

[6] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin, "Network decomposition and locality in distributed computation," in *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS 1989)*, 1989, pp. 364–369.

[7] A. Panconesi and A. Srinivasan, "On the Complexity of Distributed Network Decomposition," *Journal of Algorithms*, vol. 20, no. 2, pp. 356–374, 1996.

[8] L. Barenboim and M. Elkin, "Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition," *Distributed Comput.*, vol. 22, pp. 363–379, 2010.

[9] J. Schneider and R. Wattenhofer, "An optimal maximal independent set algorithm for bounded-independence graphs," *Distributed Computing*, vol. 22, pp. 349–361, 2010.

[10] C. Lenzen and R. Wattenhofer, "MIS on trees," in *Proceedings of the 2011 Annual ACM Symposium on Principles of Distributed (PODC)*, 2011, pp. 41–48.

[11] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider, "The Locality of Distributed Symmetry Breaking," *Journal of the ACM*, vol. 63, no. 3, pp. 1–45, 2016.

[12] L. Barenboim and M. Elkin, *Distributed Graph Coloring: Fundamentals and Recent Developments*, 2013, vol. 4, no. 1.

[13] L. Barenboim, M. Elkin, and F. Kuhn, "Distributed (Δ+1)-Coloring in Linear (in Δ) Time," *SIAM Journal on Computing*, vol. 43, no. 1, pp. 72–95, 2014.

[14] M. Ghaffari, "An Improved Distributed Algorithm for Maximal Independent Set," in *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*. Society for Industrial and Applied Mathematics, 2016, pp. 270–277.

[15] V. Rozhoň and M. Ghaffari, "Polylogarithmic-time deterministic network decomposition and distributed derandomization," in *Proceedings of 52nd Annual ACM Symposium on Theory of Computing (STOC 2020)*, 2020.

[16] M. Ghaffari, C. Grunau, and V. Rozhon, "Improved deterministic network decomposition," *CoRR*, vol. abs/2007.08253, 2020.

[17] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Local computation: Lower and upper bounds," *Journal of ACM*, vol. 63, no. 2, pp. 17:1–17:44, 2016.

[18] A. Balliu, S. Brandt, J. Hirvonen, D. Olivetti, M. Rabie, and J. Suomela, "Lower bounds for maximal matchings and maximal independent sets," in *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, 2019, pp. 481–497.

[19] A. Panconesi and R. Rizzi, "Some simple distributed algorithms for sparse networks," *Distributed Computing*, vol. 14, no. 2, pp. 97–100, 2001.

[20] J. Schneider and R. Wattenhofer, "A new technique for distributed symmetry breaking," in *Proceedings of the 2010 Annual ACM Symposium on Principles of Distributed Computing, (PODC)*, 2010, pp. 257–266.

[21] B. Gfeller and E. Vicari, "A randomized distributed algorithm for the maximal independent set problem in growth-bounded graphs," in *Proceedings of the*

2007 ACM Symposium on Principles of Distributed Computing (PODC), 2007, pp. 53–60.

[22] T. Bisht, K. Kothapalli, and S. V. Pemmaraju, "Brief announcement: Super-fast t-ruling sets," in *Proceedings of the 2014 Symposium on Principles of Distributed Computing (PODC)*, 2014, pp. 379–381.

[23] M. Ghaffari, J. Hirvonen, F. Kuhn, and Y. Maus, "Improved distributed Δ-coloring," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, (PODC 2018)*, 2018, pp. 427–436.

[24] Y. Chang, W. Li, and S. Pettie, "An optimal distributed (Δ+1)-coloring algorithm?" in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, (STOC 2018)*, 2018, pp. 445–456.

[25] F. Kuhn, Y. Maus, and S. Weidner, "Deterministic distributed ruling sets of line graphs," in *Structural Information and Communication Complexity - 25th International Colloquium, SIROCCO 2018, Ma'ale HaHamisha, Israel, June 18-21, 2018, Revised Selected Papers*, 2018, pp. 193–208.

[26] K. Chung, S. Pettie, and H. Su, "Distributed algorithms for the Lovász local lemma and graph coloring," *Distributed Computing*, vol. 30, no. 4, pp. 261–280, 2017.

[27] M. Göös, J. Hirvonen, and J. Suomela, "Linear-in-delta lower bounds in the LOCAL model," in *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing, PODC*, 2014, pp. 86–95.

[28] M. Naor and L. J. Stockmeyer, "What can be computed locally?" *SIAM Journal on Computing*, vol. 24, no. 6, pp. 1259–1277, 1995.

[29] S. Brandt, O. Fischer, J. Hirvonen, B. Keller, T. Lempiäinen, J. Rybicki, J. Suomela, and J. Uitto, "A lower bound for the distributed Lovász local lemma," in *Proceedings of the 48th ACM Symposium on Theory of Computing (STOC 2016)*. ACM Press, 2016, pp. 479–488.

[30] Y.-J. Chang, Q. He, W. Li, S. Pettie, and J. Uitto, "Distributed edge coloring and a special case of the constructive Lovász local lemma," *ACM Transactions on Algorithms*, vol. 16, no. 1, pp. 8:1–8:51, 2020.

[31] A. Balliu, J. Hirvonen, D. Olivetti, and J. Suomela, "Hardness of minimal symmetry breaking in distributed computing," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, (PODC)*, 2019, pp. 369–378.

[32] S. Brandt, "An automatic speedup theorem for distributed problems," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, 2019, pp. 379–388.

[33] S. Brandt and D. Olivetti, "Truly tight-in-Δ bounds for bipartite maximal matching and variants," pp. 69–78, 2020.

[34] A. Balliu, S. Brandt, Y. Efron, J. Hirvonen, Y. Maus, D. Olivetti, and J. Suomela, "Classification of distributed binary labeling problems," in *Proceedings of the 34th International Symposium on Distributed Computing (DISC)*, 2020.

[35] D. Olivetti, "Round Eliminator: a tool for automatic speedup simulation," 2019. [Online]. Available: https://github.com/olidennis/round-eliminator

[36] P. Fraigniaud, M. Heinrich, and A. Kosowski, "Local Conflict Coloring," in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2016, pp. 625–634.

[37] L. Barenboim and M. Elkin, "Deterministic distributed vertex coloring in polylogarithmic time," *Journal of ACM*, vol. 58, pp. 23:1–23:25, 2011.

[38] M. Ghaffari, "Distributed maximal independent set using small messages," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA 2019)*, 2019, pp. 805–820.

[39] M. Ghaffari and J. Portmann, "Improved Network Decompositions Using Small Messages with Applications on MIS, Neighborhood Covers, and Beyond," in *Proceedings of the 33rd International Symposium on Distributed Computing (DISC 2019)*, 2019, pp. 18:1–18:16.

[40] J. Schneider, M. Elkin, and R. Wattenhofer, "Symmetry breaking depending on the chromatic number or the neighborhood growth," *Theoretical Computer Science*, vol. 509, pp. 40–50, 2013.

[41] M. Henzinger, S. Krinninger, and D. Nanongkai, "A deterministic almost-tight distributed algorithm for approximating single-source shortest paths," in *Proceedings 48th Symp. on Theory of Computing (STOC 2016)*, 2016, pp. 489–498.

[42] K. Kothapalli and S. V. Pemmaraju, "Super-fast 3-ruling sets," in *Annual Conference on Foundations of Software Technology and Theoretical Computer Science (IARCS 2012)*, vol. 18, 2012, pp. 136–147.

[43] S. Pai, G. Pandurangan, S. V. Pemmaraju, T. Riaz, and P. Robinson, "Symmetry breaking in the CONGEST model: Time- and message-efficient algorithms for ruling sets," in *Proceedings of 31st International Symposium on Distributed Computing, (DISC 2017)*, vol. 91, 2017, pp. 38:1–38:16.

[44] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.