Ruling Sets in Random Order and Adversarial Streams

Sepehr Assadi ✉
Department of Computer Science, Rutgers University, Piscataway, NJ, USA

Aditi Dudeja ✉
Department of Computer Science, Rutgers University, Piscataway, NJ, USA

Abstract

The goal of this paper is to understand the complexity of a key symmetry breaking problem, namely the (α, β)-ruling set problem in the graph streaming model. Given a graph $G = (V, E)$, an (α, β)-ruling set is a subset $I \subseteq V$ such that the distance between any two vertices in I is at least α and the distance between a vertex in V and the closest vertex in I is at most β. This is a fundamental problem in distributed computing where it finds applications as a useful subroutine for other problems such as maximal matching, distributed colouring, or shortest paths. Additionally, it is a generalization of MIS, which is a $(2, 1)$-ruling set.

Our main results are two algorithms for $(2, 2)$-ruling sets:

1. In adversarial streams, where the order in which edges arrive is arbitrary, we give an algorithm with $\tilde{O}(n^{4/3})$ space, improving upon the best known algorithm due to Konrad et al. [DISC 2019], with space $\tilde{O}(n^{3/2})$.
2. In random-order streams, where the edges arrive in a random order, we give a semi-streaming algorithm, that is an algorithm that takes $\tilde{O}(n)$ space.

Finally, we present new algorithms and lower bounds for (α, β)-ruling sets for other values of α and β. Our algorithms improve and generalize the previous work of Konrad et al. [DISC 2019] for $(2, \beta)$-ruling sets, while our lower bound establishes the impossibility of obtaining any non-trivial streaming algorithm for $(\alpha, \alpha - 1)$-ruling sets for all <u>even</u> $\alpha > 2$.

2012 ACM Subject Classification Theory of computation → Sketching and sampling; Theory of computation → Lower bounds and information complexity; Theory of computation → Random order and robust communication complexity

Keywords and phrases Symmetry breaking, Ruling sets, Lower bounds, Communication Complexity

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.6

Funding *Sepehr Assadi*: Research supported in part by a NSF CAREER Grant CCF-2047061, and a gift from Google Research.
Aditi Dudeja: This work was done while funded by NSF Award 1942010.

Acknowledgements We thank anonymous reviewers for their insightful feedback.

1 Introduction

The goal of this paper is to understand the complexity of *symmetry breaking problems*, specifically the (α, β)-ruling set problem, which is defined as follows.

▶ **Definition 1.** *Given a graph $G = (V, E)$, an (α, β)-ruling set is a subset $I \subseteq V$ such that the distance between any two vertices in I is at least α and the distance between a vertex in V and the closest vertex in I is at most β.*

Ruling sets are a generalization of *maximal independent sets*: MIS is a $(2, 1)$-ruling set. Ruling sets have been well-studied in numerous distributed setting such as the CONGEST and LOCAL models, and the k-machine model (see e.g. [7, 8, 10, 17, 18, 24, 31]). Moreover,

35th International Symposium on Distributed Computing (DISC 2021).
Editor: Seth Gilbert; Article No. 6; pp. 6:1–6:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ruling sets are used as a subroutine in algorithms solving other interesting graph problems such as distributed coloring [12], maximal matching [8], maximal independent set [18], or shortest paths [19]. In many applications where they are sufficient and easier to compute, ruling sets can replace MIS computation. (see e.g. [8, 27]).

We study this problem in the graph streaming model, which is motivated by the fact that modern data sets are too large to fit in a computer's random access memory. A streaming algorithm processes its data sequentially, in one or a small number of passes. It has a memory sublinear in the size of input. The distributed and graph streaming models have a lot in common, with techniques used in one model often finding applications in the other. For example L_0-sampling [20] and graph sketches [3] which were first used in the context of streaming algorithms, now find application in several distributed settings too: [30] use linear sketches to obtain algorithms for MST in the k-machine model, and [21] use them in the Congested Clique model to give an optimal algorithm for the same problem. There are numerous examples of tools from communication and information complexity being used to establish lower bounds for streaming as well as distributed computing problems (see, e.g. [1] and the references therein).

Many of the classical symmetry breaking problems have been studied in the semi-streaming model. For example, [6] and [13] studied the MIS problem in the semi-streaming setting and obtained an $\Omega(n^2)$ lower bound on space for one pass streaming algorithms. If we increase the number of passes, then correlation clustering of [2] enables us to get a semi-streaming algorithm in $O(\log\log n)$ passes. Similar to the case of distributed computing where the problem of computing a $(\Delta+1)$-colouring is easier than computing an MIS, [6] also gave a one-pass semi streaming algorithm for $\Delta+1$ coloring. [24] studied the problem of $(2,\beta)$-ruling sets in the semi-streaming setting, establishing that for $\beta \geq 2$, the problem of computing a $(2,\beta)$-ruling set is strictly easier than computing an MIS. We extend this line of study of symmetry breaking problems by considering random order streams, and giving improved algorithms and lower bounds for adversarial streams for various values of α and β.

Our Results. Our main result is a semi-streaming algorithm for $(2,2)$-ruling sets and thus $(2,\beta)$-ruling sets for all $\beta \geq 2$ in random-order streams. The random order model is motivated by the fact that it gives rise to a natural notion of *average-case analysis*, which explains why some streaming problems may have strong lower bounds, while being efficiently solvable in practice (see [28]). As a result, several problems such as matching (see [23, 16, 22, 15, 5, 4, 9]), connectivity [11] or properties of bounded degree graphs [29] have been studied extensively in this model. With this, we state our first result.

▶ **Result 1.** *There is an $\tilde{O}(n)$-space streaming algorithm that computes a $(2,2)$-ruling set (and therefore $(2,\beta)$-ruling set for $\beta \geq 2$) of any graph with high probability when the edges arrive in a random order. This also gives a semi-streaming algorithm for (α,β)-ruling sets for $\beta \geq \alpha \geq 2$ when the edges arrive in a random order.*

Result 1 is the first example of a semi-streaming algorithm for any $(2,\beta)$-ruling set for $\beta = O(1)$ (although we make the assumption that the edges arrive in a random order). This bound is optimal since $\Omega(n)$ space is needed to output the solution. While it is known that MIS requires $\Omega(n^2)$ space when the edges arrive in the stream in an arbitrary order, no such barriers are known for random-order stream. Thus resolving the complexity of MIS in this model remains an interesting open question.

Our second contribution is improving the $\tilde{O}(n^{1+\frac{1}{2^\beta-1}})$-space algorithm of [24] for $(2,\beta)$-ruling sets when edges arrive in the stream in an arbitrary order.

▶ **Result 2.** *There is an* $\tilde{O}(\beta \cdot n^{1+\frac{1}{2^{\beta}-1}})$ *space streaming algorithm that computes a* $(2, \beta)$*-ruling set of any graph with high probability when the edges arrive in an arbitrary order. This also gives a streaming algorithm that computes an* (α, β)*-ruling set (for* $\beta \geq \alpha \geq 2$*) with high probability using* $\tilde{O}(n^{1+\frac{1}{2^{\beta-\alpha+2}-1}})$ *space when the edges arrive in an arbitrary order*

The improvement in Result 2 is the most significant for $\alpha = \beta = 2$, where we get a bound of $\tilde{O}(n^{4/3})$, improving upon the previously known best bound of $\tilde{O}(n^{3/2})$. Our final result is a lower bound for $(\alpha, \alpha-1)$-ruling sets for all even $\alpha > 2$.

▶ **Result 3.** *Any randomized constant error one-pass streaming algorithm in the adversarial order model that computes a* $(\alpha, \alpha-1)$*-ruling set for any even* $\alpha > 2$ *requires* $\Omega(n^2/\alpha^2)$ *space.*

Landscape of (α, β)-ruling sets. The algorithm of Result 2 gives non-trivial streaming algorithms (with space complexity much smaller than input size), for all $\beta \geq \alpha \geq 2$. On the other hand, Result 3 establishes that the problem of computing (α, β)-ruling sets when $\beta = \alpha - 1$ for even $\alpha > 2$, does not admit non-trivial streaming algorithms. When $\beta < \alpha - 1$, an (α, β)-ruling sets may not even exist. Thus, our results give a relatively complete picture of which setting of parameters α and β the problem of computing (α, β)-ruling sets admits non-trivial streaming algorithms.

2 Preliminaries

From now on, we use β-ruling sets to denote $(2, \beta)$-ruling sets. For a graph G we use n to denote $|V(G)|$ and m to denote $|E(G)|$. For a subgraph K of G and a vertex $v \in K$, we define $E_K(v)$ to be the set of edges incident on v in K, $\deg_K(v)$ to be the number of neighbours of v in K and $N_K(v)$ to be the neighbours of v in K. For $u, v \in V(G)$, let $\text{dist}(u, v)$ denote the length of the shortest path from u to v.

Concentration Results. In our proofs, we will use negatively associated random variables.

▶ **Definition 2.** *We say a collection of random variables* $\{X_1, X_2, \cdots, X_k\}$ *are* negatively associated *if for any two disjoint index sets* $I, J \subseteq [n]$ *and two functions* f, g *both monotone increasing or both monotone decreasing, it holds*

$$\mathbb{E}\left[f(X_i : i \in I) \cdot g(X_j : j \in J)\right] \leq \mathbb{E}\left[f(X_i : i \in I)\right] \cdot \mathbb{E}\left[g(X_j : j \in J)\right]$$

We will apply our concentration bounds to randomly chosen subsets of vertices of a fixed size, or to segments of stream. Since the stream is random-ordered, both of these situations can be thought of as sampling without replacement. Thus, we can apply Chernoff for negatively associated random variables (see for example [32]).

▶ **Proposition 3** ([14])**.** [Chernoff Bound] *Let* $X_1, \cdots, X_k$ *be negatively associated* 0-1 *random variables. Let* $X = \sum_{i=1}^{k} X_i$*. Let* $\mu = \mathbb{E}\left[\sum_{i=1}^{k} X_i\right]$ *and let* $\mu_{\min} \leq \mu \leq \mu_{\max}$*. Then, for all* $\delta \in (0, 1)$*, we have:*

$$\Pr\left(X \geq (1+\delta)\mu_{\max}\right) \leq \exp\left(-\delta^2 \mu_{\max}/3\right) \text{ and } \Pr\left(X \leq (1-\delta)\mu_{\min}\right) \leq \exp\left(-\delta^2 \mu_{\min}/2\right).$$

Communication Complexity. In order to prove lower bounds on the space complexity, we will reduce Index_t to our problem, which we formally define.

▶ **Definition 4.** *In the two-party communication problem* Index_t*, Alice holds a* t*-bit string* $X \in \{0,1\}^t$ *and Bob holds an index* $\sigma \in [t]$*. Alice sends a single message to Bob, who upon receipt outputs* X_σ*.*

We shall use the following well-known distribution for Index_t.

Distribution 1 Distribution $\mathcal{D}_{\mathsf{Ind}}$.

1 Pick $X \in \{0,1\}^t$ uniformly at random.
2 Pick $\sigma \in [t]$ uniformly at random.

If an instance of Index_t is drawn from $\mathcal{D}_{\mathsf{Ind}}$, then we have the following bound on the message complexity of any communication protocol that solves it (see [26]).

▶ **Proposition 5.** *For* $0 < \delta < 1/2$*, any* δ*-error protocol for* Index_t *over* $\mathcal{D}_{\mathsf{Ind}}$*, ccmmunicates* $\Omega(t)$ *bits.*

3 2-Ruling Sets in Random-Order Streams

In this section, we state our main contribution: We give a simple streaming algorithm that computes a 2-ruling set of a given graph in **random-order stream** in $\tilde{O}(n)$ space. We now state our main theorem formally.

▶ **Theorem 6.** *There is an* $\tilde{O}(n)$*-space streaming algorithm that computes a* 2*-ruling set of any graph* G *with high probability when the edges of* G *arrive in a random order in the stream.*

In Section 3.1 we will show a structural theorem that we will use to compute 2-ruling sets in random order stream. In the subsequent section, we will state our algorithm and argue its correctness and space complexity.

3.1 Peeling Decomposition

In order to compute 2-Ruling Sets in random order streams, we will use a structural decomposition of the graph G. We formally state this decomposition. A similar decomposition was used first by [25] and [10] to compute β-ruling sets in the distributed setting, and then subsequently by [24] to get streaming algorithms for β-ruling sets. Since this decomposition was given in a different language in the previous works, we re-state it here for completeness. Our main contribution is the observation that this decomposition can be computed in the random-order setting with high probability and using a small amount of space.

▶ **Definition 7.** *Let* $G = (V, E)$ *be a graph, let* $r = \log n - \log\log n - 7$ *and let* $(d_0, d_1, \cdots, d_r)$ *be a sequence of integers such that* $d_i = n/2^i$ *for* $i \in [r]$*. We define the* ***Peeling Decomposition*** $(G_0, G_1, \cdots, G_r)$ *of* G *as follows.*

1. *Each* G_i *consists of vertices* V_i *and edges* E_i*.*
2. *We let* $G_0 = G$*, and* $G_i \subseteq G_{i-1}$ *for all* $i \geq 0$*.*
3. *For* $i \geq 1$*, we define* $V_i = \left\{v \in V_{i-1} \mid deg_{G_{i-1}}(v) \leq d_i\right\}$*,* $G_i = G[V_i]$ *and* $E_i = E(G_i)$*.*

Depending on the decomposition, with each vertex $v \in V$, we associate a level.

▶ **Definition 8.** *For every vertex* v*, we define the level of a vertex to be the unique index* l *such that* $v \in V_l$ *but* $v \notin V_{l+1}$*. If* $v \in V_r$ *then the level of* v *is* r*.*

We now describe a process to create an induced subgraph H of G whose MIS with high probability will be a 2-ruling set for G. Moreover, the total number of edges in H will be small.

Process 2 A Process for Sampling a 2-Ruling Set.

Input: Graph G, $(d_0, d_1, \cdots, d_r)$, where $r = \log n - \log\log n - 7$, $d_i = n/2^i$ and peeling decomposition $(G_0, \cdots, G_r)$.

1 For each $i \in \{1, 2, \cdots, r-1\}$, we sample S_i of size $\frac{10 \cdot |V_i| \cdot \log n}{d_{i+1}}$ from V_i uniformly at random.

2 Let H be the induced subgraph $G[\cup_{i=0}^{r-1} S_i \cup V_r]$.

3 Return $\mathcal{M}$, an arbitrary MIS of H.

We now prove the following guarantees on H and $\mathcal{M}$.

▶ **Lemma 9.** *The set $\mathcal{M}$ returned by Process 2 is a 2-ruling set of G with high probability.*

Proof. We first note that since H is an induced subgraph of G, $\mathcal{M}$ is an independent set in G. We want to show that for all $v \notin \mathcal{M}$, there is a vertex $u \in \mathcal{M}$ such that u is at most distance two from v.

Towards this, we first consider a vertex $v \in V_r$. Observe that since $V_r \subseteq V(H)$, this implies that either $v \in \mathcal{M}$ or there is a vertex $u \in \mathcal{M}$ such that $\text{dist}(v, u) = 1$. We now prove the claim for vertices on level $i < r$. Consider a vertex $v \in V_{i-1} \setminus V_i$. By definition, we know that $\deg_{G_{i-1}}(v) \geq d_i$. For a vertex $w \in N_{G_{i-1}}(v)$, we define X_w to be the indicator random variable that takes value 1 if $w \in S_{i-1}$ and value 0 if $w \notin S_{i-1}$. Since S_{i-1} is sampled uniformly at random from V_{i-1} and $|S_{i-1}| = \frac{10 \cdot |V_{i-1}| \cdot \log n}{d_i}$, we have the following bound for a vertex $w \in N_{G_{i-1}}(v)$.

$$\Pr(X_w = 1) = \frac{|S_{i-1}|}{|V_{i-1}|} = \frac{10 \cdot \log n}{d_i}$$

Note that the above bound is at most 1, since $d_r \geq 100 \cdot \log n$, by our choice of r and d_i's. Observe that the random variables $\{X_w \mid w \in N_{G_{i-1}}(v)\}$ are negatively associated (since they correspond to sampling without replacement). This gives us the following bounds:

$$\Pr\left(\bigwedge_{w \in N_{G_{i-1}}(v)} X_w = 0\right) \leq \prod_{w \in N_{G_{i-1}}(v)} \Pr(X_w = 0) = \left(1 - \frac{10 \cdot \log n}{d_i}\right)^{\deg_{G_{i-1}}(v)}$$
$$\leq \left(1 - \frac{10 \cdot \log n}{d_i}\right)^{d_i} = O\left(\frac{1}{n^{10}}\right).$$
$$(\text{Since } \deg_{G_{i-1}}(v) \geq d_i)$$

Taking a union bound over all vertices we conclude that with probability $1 - o(1)$ for $i \in [r]$, for all $v \in V_{i-1} \setminus V_i$, there is a vertex $w \in N_{G_{i-1}}(v)$ such that $w \in S_{i-1}$. This proves our claim. ◀

We now bound the total number of edges in H.

▶ **Lemma 10.** *The total number of edges in H, the induced subgraph of Process 2, is at most $O(\sum_{i=0}^{r-1} \frac{|V_i| \cdot d_i \cdot \log n}{d_{i+1}} + n \cdot d_r)$. By our choice of d_i's and r, this is at most $\tilde{O}(n)$.*

Proof. To bound the total number of edges in H, we give a charging argument. For every edge (u, v) we charge this edge to the endpoint with the lower level. We consider the following two cases.

1. **Total charge on $v \in V_r$.** We begin by observing, that a vertex $v \in V_r$ is charged an edge (u, v) then level of u is also r (in other words, $u \in V_r$). Since $\deg_{G_{r-1}}(v) \leq d_r$, by definition of V_r, and $V_r \subseteq V_{r-1}$, we conclude that $\deg_{G_r}(v) \leq d_r$. From this discussion, we know that the total charge on any $v \in V_r$ is at most d_r. Thus, the total charge on V_r is at most $|V_r| \cdot d_r$.
2. **Total charge on $v \in S_i$.** Consider a vertex v at level i, and suppose it is charged an edge (u, v). Let j be the level of u. By our charging scheme, $j \geq i$. Consequently, $u \in V_j \subseteq V_i \subseteq V_{i-1}$. Additionally, we know by definition of V_i that $\deg_{G_{i-1}}(v) \leq d_i$. By the above discussion, the total charge on v is at most d_i. Therefore, the total charge on S_i is at most $\frac{100 \cdot d_i \cdot |V_i| \cdot \log n}{d_{i+1}}$.

This proves our claim. ◀

3.2 The Algorithm

We briefly describe our algorithm. To give our algorithm, we would ideally like to sample the subgraph H as described. If we were able to achieve this, then at the end of the stream we could compute an MIS $\mathcal{M}$ of H, and using the Lemma 9, we would know this is a 2-ruling set of G with high probability. Moreover using Lemma 10, we would be able to conclude that we have a semi-streaming algorithm for this problem.

However, determining V_i even in random order streams seems impossible. Therefore, instead of computing V_i, we will compute sets $\tilde{V}_i$ and $\tilde{G}_i = G[\tilde{V}_i]$, which will have the following relaxed property: for all vertices $v \in \tilde{V}_i$, $\deg_{\tilde{G}_{i-1}}(v) \leq d_i$, and for all $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, $\deg_{\tilde{G}_{i-1}}(v) \geq d_i/2$. We can then show that the main property still holds: for $i \in \{1, \cdots, r\}$, suppose we sample $\tilde{S}_{i-1}$ of size $\frac{100 \cdot |\tilde{V}_{i-1}| \cdot \log n}{d_i}$ uniformly at random from $\tilde{V}_{i-1}$ then with high probability, for all $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, there is $u \in N_{\tilde{G}_{i-1}}(v)$ such that $u \in \tilde{S}_{i-1}$. Finally, if $H = \cup_{i=1}^{r-1} \tilde{S}_i \cup \tilde{V}_r$, then an argument similar to the one in Lemma 10 will give us a similar bound on $|E(H)|$.

We give some intuition about our algorithm. Our algorithm starts by first guessing $\tilde{V}_1$ described above as follows: given parameter d_1, we sample a set $\tilde{S}_0$ of size $\frac{10 \cdot n \cdot \log n}{d_1}$ uniformly at random from V. We determine the set $\tilde{V}_1$ by filtering out vertices that have a lot of edges to $\tilde{S}_0$. To do this we only look at a small portion of the stream (the first $\frac{100 \cdot m \cdot \log n}{d_1}$ edges), and remove vertices that have a lot of edges incident on them in this portion of the stream. We will argue via a Chernoff bound argument that $\tilde{V}_1$ has the above property with high probability: all vertices in $\tilde{V}_1$ have few edges to $\tilde{S}_0$, and all vertices in $V \setminus \tilde{V}_1$ have a lot of edges to $\tilde{S}_0$. For any $i \geq 2$, we repeat this process inductively: suppose we have $\tilde{V}_{i-1}$. We sample $\tilde{S}_{i-1}$ from it, and we obtain $\tilde{V}_i$ by looking at a small portion of the stream and removing vertices that have a lot of edges to $\tilde{S}_{i-1}$.

We formally describe our algorithm in Algorithm 3, and we now show its correctness and space complexity.

▷ Claim 11. Let $\tilde{V}_1, \cdots \tilde{V}_r$ be the sets computed by Algorithm 3. Then, with high probability the following facts hold for all $i \in [r]$.
1. For any $v \in \tilde{V}_i$, $\deg_{\tilde{G}_{i-1}}(v) \leq d_i$.
2. For any $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, $\deg_{\tilde{G}_{i-1}}(v) \geq d_i/2$.

Proof. Consider $v \in \tilde{V}_i$, we want to show $\deg_{\tilde{G}_{i-1}}(v) \leq d_i$. Recall how $\tilde{V}_i$ is created: we add $v \in \tilde{V}_{i-1}$ to $\tilde{V}_i$ if we see fewer than $70 \log n$ edges from $E_{\tilde{G}_{i-1}}(v)$ among the first $\frac{100 \cdot m \log n}{d_i}$ edges that have arrived in the stream. Suppose $\deg_{\tilde{G}_{i-1}}(v) > d_i$. Let Y_v denote the random variable $|E_{\tilde{G}_{i-1}}(v) \cap \mathsf{Stream}_{i-1}|$. Observe that Y_v is a sum of negatively associated 0-1 random

Algorithm 3 Computing a 2-Ruling Set in Random-Order Streams.

Input: Integers $r = \log n - \log\log n - 7$, $(d_0, d_1, \cdots, d_r)$, where $d_i = n/2^i$.

Phase 0**:**

$\tilde{V}_1 \leftarrow V$

Let $\tilde{S}_0$ be chosen uniformly at random from V, where $|\tilde{S}_0| = \frac{10 \cdot n \cdot \log n}{d_1}$

Store the first $\frac{100 \cdot m \cdot \log n}{d_1}$ edges of G that arrive in the stream (denoted by Stream_0).

For all vertices v with $|E_G(v) \cap \mathsf{Stream}_0| > 70 \log n$, let $\tilde{V}_1 \leftarrow \tilde{V}_1 \setminus \{v\}$.

Discard all edges (u, v) such that either $u \notin \tilde{S}_0 \cup \tilde{V}_1$ or $v \notin \tilde{S}_0 \cup \tilde{V}_1$.

Phase i **for** $i \in \{1, 2, \cdots, r-1\}$**:**

Initialize $\tilde{V}_{i+1} \leftarrow \tilde{V}_i$.

Sample $\tilde{S}_i$ uniformly at random from $\tilde{V}_i$, where $|\tilde{S}_i| = \frac{10 \cdot |\tilde{V}_i| \cdot \log n}{d_{i+1}}$.

Let $\tilde{G}_i = G[\tilde{V}_i]$. Let $\tilde{H}_i = G[\cup_{j=0}^{i-1} \tilde{S}_j \cup \tilde{V}_i]$. Process the stream, storing only edges of $\tilde{H}_i$. After the first $\frac{100 \cdot m \cdot \log n}{d_{i+1}}$ edges of G (denoted Stream_i) have been seen, do the following: for all vertices $v \in \tilde{V}_i$ with $|E_{\tilde{G}_i}(v) \cap \mathsf{Stream}_i| > 70 \log n$, let $\tilde{V}_{i+1} \leftarrow \tilde{V}_{i+1} \setminus \{v\}$.

Discard all edges (u, v) with $u \notin \cup_{j=0}^{i} \tilde{S}_j \cup \tilde{V}_{i+1}$ or $v \notin \cup_{j=0}^{i} \tilde{S}_j \cup \tilde{V}_{i+1}$.

After Phase $r-1$**:**

Process the stream, while only retaining edges that have both endpoints in $\cup_{j=0}^{r-1} \tilde{S}_j \cup \tilde{V}_r$.

At the end of the stream return $\mathcal{M} = \mathsf{MIS}(G[\cup_{j=0}^{r-1} \tilde{S}_j \cup \tilde{V}_r])$.

variables. We briefly explain why: for an edge $e \in E_{\tilde{G}_{i-1}}(v)$, define X_e to be the indicator random variable that takes value 1 when $e \in \mathsf{Stream}_{i-1}$ and 0 otherwise. Note that X_e are negatively associated since they correspond to sampling $\frac{100 \cdot m \cdot \log n}{d_i}$ edges without replacement. We additionally have the following bound for $e \in E_{\tilde{G}_{i-1}}(v)$.

$$\Pr(X_e = 1) = \left(\frac{1}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_i}\right)$$

The above bound is less than 1 due to the fact that $r = \log n - \log\log n - 7$ and $d_r > 100 \log n$. This gives us the following bound on $\mathbb{E}[X_v]$:

$$\mathbb{E}[Y_v] = \left(\frac{\deg_{\tilde{G}_{i-1}}(v)}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_i}\right) > \left(\frac{d_i}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_i}\right) = 100 \log n$$

We now apply Proposition 3 with $\delta = 3/10$ and $\mu_{\min} = 100 \log n$, to get:

$$\Pr(Y_v \le 70 \log n) \le \exp\left(-(3/10)^2 (100 \log n)(1/2)\right) = O\left(\frac{1}{n^4}\right)$$

$$\Pr(Y_v > 70 \log n) = 1 - O\left(\frac{1}{n^4}\right)$$

This implies that with high probability, v would be omitted from $\tilde{V}_i$, which is a contradiction.

We now prove that with high probability, for all $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, $\deg_{\tilde{G}_{i-1}}(v) \ge d_i/2$. The proof strategy will be same as before. Assume that $\deg_{\tilde{G}_{i-1}}(v) < d_i/2$. For $e \in E_{\tilde{G}_{i-1}}(v)$, we let X_e be a random variable that takes value 1 if $e \in \mathsf{Stream}_{i-1}$ and 0 otherwise. Let $Y_v = |E_{\tilde{G}_{i-1}}(v) \cap \mathsf{Stream}_{i-1}|$. As discussed before, we know that Y_v is a sum of 0-1 negatively correlated random variables $\left\{X_e \mid e \in E_{\tilde{G}_{i-1}}(v)\right\}$.

$$\Pr(X_e = 1) = \left(\frac{1}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_i}\right)$$

This gives us the following bound on $\mathbb{E}[Y_v]$.

$$\mathbb{E}[Y_v] = \left(\frac{\deg_{\tilde{G}_{i-1}}(v)}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_i}\right) < \left(\frac{d_i}{2m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_i}\right) = 50 \log n$$

We now apply Proposition 3 with $\delta = 2/5$ and $\mu_{\max} = 50 \log n$.

$$\Pr(Y_v \geq 70 \log n) \leq \exp(-(2/5)^2 (50 \log n)(1/3)) = O\left(\frac{1}{n^2}\right)$$

$$\Pr(Y_v < 70 \log n) > 1 - O\left(\frac{1}{n^2}\right).$$

This implies that with high probability, v would be included in $\tilde{V}_i$ which is a contradiction. Taking a union bound over all vertices, we conclude the proof of the lemma. ◁

▷ Claim 12. For $i \in [r]$, consider any $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, there is a neighbour of v in $\tilde{S}_{i-1}$ with high probability.

Proof. For the proof of this lemma, we condition on Claim 11. The set $\tilde{S}_{i-1}$ is sampled uniformly at random from $\tilde{V}_{i-1}$, and $|\tilde{S}_{i-1}| = \frac{10 \cdot |\tilde{V}_{i-1}| \cdot \log n}{d_i}$. Consider any $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, from Claim 11, we know that $\deg_{\tilde{G}_{i-1}}(v) \geq \frac{d_i}{2}$. For $w \in N_{\tilde{G}_{i-1}}(v)$, consider 0-1 random variable X_w, that takes value 1 if $w \in \tilde{S}_{i-1}$ and 0 otherwise. The random variables $\left\{X_w = 1 \mid w \in N_{\tilde{G}_{i-1}}(v)\right\}$ are negatively associated. Let Y_v denote $|N_{\tilde{G}_{i-1}}(v) \cap \tilde{S}_{i-1}|$. We know that Y_v is a sum of negatively associated 0-1 random variables. Having established this notation, for $w \in N_{\tilde{G}_{i-1}}(v)$, we have the following fact.

$$\Pr(X_w = 1) = \frac{|\tilde{S}_{i-1}|}{|\tilde{V}_{i-1}|} = \left(\frac{10 \cdot |\tilde{V}_{i-1}| \cdot \log n}{d_i}\right)\left(\frac{1}{|\tilde{V}_{i-1}|}\right) = \frac{10 \log n}{d_i}.$$

The above bound is at most 1, since $d_r > 100 \cdot \log n$ by our choice of r and d_i's. We now bound the probability that none of the vertices in $N_{\tilde{G}_{i-1}}(v)$ are included in $\tilde{S}_{i-1}$.

$$\Pr\left(\bigwedge_{w \in N_{\tilde{G}_{i-1}}(v)} \{X_w = 0\}\right) \leq \prod_{w \in N_{\tilde{G}_{i-1}}(v)} \Pr(X_w = 0) = \left(1 - \frac{10 \cdot \log n}{d_i}\right)^{\deg_{\tilde{G}_{i-1}}(v)}$$

(Negatively associated r.v.)

$$\leq \left(1 - \frac{10 \cdot \log n}{d_i}\right)^{d_i/2} = O\left(\frac{1}{n^5}\right)$$

$(\deg_{\tilde{G}_{i-1}}(v) \geq d_i/2)$.

Taking a union bound over all $v \in V$, we conclude that for all $i \in [r]$ and for all $v \in \tilde{V}_{i-1} \setminus \tilde{V}_i$, there is $w \in N_{\tilde{G}_{i-1}}(v)$ such that $w \in \tilde{S}_{i-1}$ with high probability. ◁

Correctness. To show correctness, note that Algorithm 3 retains all edges of $H = G\left[\cup_{j=0}^{r-1}\tilde{S}_j \cup \tilde{V}_r\right]$ till the end of the stream. This implies that the set of vertices $\mathcal{M}$ output by the algorithm, indeed forms an MIS in H. Since H is an induced subgraph of G, $\mathcal{M}$ is also an independent set in G. We know that all $v \in V(H)$ are 1-ruled. Additionally, from Claim 12 and Claim 11 we conclude that for all $i \in [r-1]$, for all $v \in \tilde{V}_i \setminus \tilde{V}_{i+1}$, there is a neighbour of v in $\tilde{S}_i$. Let u be this neighbour. Since $\mathcal{M}$ is an MIS, either $u \in \mathcal{M}$ or there is $x \in N_G(u)$ such that $x \in \mathcal{M}$. This implies that v is 2-ruled. We conclude that with probability at least $1 - o(1)$, all vertices in V are 2-ruled.

Space Complexity. We show that the space complexity of this algorithm. Towards this, we show that the total number of edges stored by the algorithm is at most $O(\sum_{i=0}^{r-1} \frac{|\tilde{V}_i| \cdot d_i \cdot \log n}{d_{i+1}} + n \cdot d_r) = \tilde{O}(n)$ since $r = \log n - \log\log n - 7$ and $d_i = n/2^i$.

▶ **Lemma 13.** *The total number of edges stored by Algorithm 3 is* $O(\sum_{i=0}^{r-1} \frac{150 \cdot |\tilde{V}_i| \cdot d_i \cdot \log n}{d_{i+1}} + n \cdot d_r)$ *with high probability.*

Proof. We start with showing the total number of edges in the memory at the end of Phase 0. Observe that Phase 0 ends when we have seen first $\frac{100 \cdot m \cdot \log n}{d_1}$ edges of G. Therefore, by the end of this phase we have at most $\frac{100 \cdot m \cdot \log n}{d_1}$ edges in the memory, which is $\frac{100 \cdot n^2 \cdot \log n}{d_1}$, the first term in the sum given in the lemma. During Phase i, we only store edges of $\tilde{H}_i = G\left[\cup_{j=0}^{i-1}\tilde{S}_j \cup \tilde{V}_i\right]$. We will now bound the number of such edges.

As before, we will make a charging argument. Recall the definition of the level of a vertex in Definition 8. Consider an edge $(u, v) \in \tilde{H}_i$ that appears in Stream_i. We charge (u, v) to the vertex with the lower level, breaking ties arbitrarily. Suppose v is at level $j < i$ and is charged the edge (u, v). This implies that level of u is at least j as well (since we charge the edge to a lower level). In particular, $u \in \tilde{G}_j$. Since $\deg_{\tilde{G}_{j-1}}(v) \le d_j$, this implies the total charge on v is at most d_j. Consider the set $\tilde{S}_j$, which is sampled from $\tilde{G}_j$. The total charge on $\tilde{S}_j$ is at most $\frac{100 \cdot d_j \cdot |\tilde{V}_j| \cdot \log n}{d_{j+1}}$.

Finally, consider $v \in \tilde{V}_i$, if v is charged for an edge (u, v), then we conclude that $u \in \tilde{V}_i$ as well. So, the total charge on $\tilde{V}_i$ is at most the total number of edges in $\tilde{G}_i \cap \mathsf{Stream}_i$. Observe that $\deg_{\tilde{G}_{i-1}}(v) \le d_i$ for all $v \in \tilde{V}_i$. This implies that $\deg_{\tilde{G}_i}(v) \le d_i$ for all $v \in \tilde{V}_i$ (from Claim 11) with high probability. In the rest of the argument, we condition on this event. Let Y_v be a random variable denoting $|E_{\tilde{G}_i}(v) \cap \mathsf{Stream}_i|$. Note that Y_v is a sum of negatively associated 0-1 indicator random variables $\left\{X_e \mid e \in E_{\tilde{G}_i}(v)\right\}$, which takes value 1 if $e \in \mathsf{Stream}_i$ and 0 otherwise. We have the following bounds on probabilities for $e \in E_{\tilde{G}_i}(v)$

$$\Pr(X_e = 1) = \left(\frac{1}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_{i+1}}\right)$$

The above bound is valid since $d_r > 100 \log n$ by our choice of parameters. Therefore, the bound on expectation of Y_v is as follows.

$$\mathbb{E}[Y_v] = \left(\frac{\deg_{\tilde{G}_i}(v)}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_{i+1}}\right) \le \left(\frac{d_i}{m}\right)\left(\frac{100 \cdot m \cdot \log n}{d_{i+1}}\right) = \frac{100 \cdot d_i \cdot \log n}{d_{i+1}}$$

We now apply Proposition 3 with $\mu_{\max} = \frac{100 \cdot d_i \cdot \log n}{d_{i+1}}$ and $\delta = 1/2$ to get the bound:

$$\Pr\left(Y_v \geq \frac{150 \cdot d_i \cdot \log n}{d_{i+1}}\right) \leq \exp\left(-(5/10)^2\left(\frac{100 \cdot d_i \cdot \log n}{d_{i+1}}\right)(1/3)\right)$$
$$= O\left(\frac{1}{n^8}\right)$$
$$\text{(We use the fact that } d_i > d_{i+1}\text{)}$$

Therefore, the total charge on $\tilde{V}_i$ is at most $\frac{150 \cdot d_i |\tilde{V}_i| \cdot \log n}{d_{i+1}}$. So from the above discussion it follows that the total number of edges stored in the memory till the end of Phase i is at most $\frac{100 \cdot n^2 \cdot \log n}{d_1} + \sum_{j=1}^{i-1} \frac{100 \cdot |\tilde{V}_j| \cdot d_j \cdot \log n}{d_{j+1}} + \frac{150 \cdot |\tilde{V}_i| \cdot d_i \cdot \log n}{d_{i+1}}$. From this we conclude that the total number of edges stored till the end of the stream $O(\frac{n^2 \cdot \log n}{d_1} + \sum_{i=1}^{r-1} \frac{n \cdot d_i \cdot \log n}{d_{i+1}} + n \cdot d_r)$, which proves our claim. ◀

▶ **Lemma 14.** *The total space complexity of Algorithm 3 is at most $\tilde{O}(n)$.*

Proof. From Lemma 13, the total number of edges stored in the memory at any time is $O(\frac{n^2 \cdot \log n}{d_1} + \sum_{i=1}^{r-1} \frac{n \cdot d_i \cdot \log n}{d_{i+1}} + n \cdot d_r)$. Since we chose $r = \log n - \log\log n - 7$ and $d_i = n/2^i$, we get the stated bound. ◀

(α, β)-Ruling sets in Random Order Streams. Algorithm 3 allows us to compute (α, β)-ruling sets for all $\beta \geq \alpha \geq 2$ in $\tilde{O}(n)$ space: in the final step, we just compute an$(\alpha, \alpha - 1)$ ruling set of $G[\cup_{j=0}^{r-1} \tilde{S}_j \cup \tilde{V}_r]$. We formally state the algorithm, and prove its correctness and space complexity in the full version of the paper.

4 Ruling Sets in Adversarial Streams

We also give an improved streaming algorithm that computes the β-ruling set of a graph in adversarial streams in $\tilde{O}(\beta \cdot n^{1+\frac{1}{2^{\beta}-1}})$ space. This is significant for $\beta = 2$, since the best known algorithm for 2-ruling sets had space complexity $\tilde{O}(n^{3/2})$, while our analysis improves it to $\tilde{O}(n^{4/3})$. We state our main result for this section.

▶ **Theorem 15.** *There is an $\tilde{O}(\beta \cdot n^{1+\frac{1}{2^{\beta}-1}})$-space streaming algorithm that computes a β-ruling set of any graph G with high probability. This implies an $\tilde{O}(n^{4/3})$-space streaming algorithm for computing a 2-ruling set of a graph G.*

4.1 A Slightly Improved Algorithm for Ruling Sets

Our algorithm does hierarchical sampling as described in the papers of [24, 25, 10]. However, we get a better bound on the space complexity. We illustrate the difference between their technique and ours for the simpler case of $\beta = 3$.

Comparison with Previous Work. Similar to [24, 25, 10], sample a set S_1 uniformly at random from V, and a set S_2 uniformly at random from S_1. During the stream, we collect edges incident on $G[S_1 \cup S_2]$. These sets are not modified during the stream in the earlier algorithms. On the other hand, we remove from S_1 all vertices that have a large degree in $G[S_1 \cup S_2]$, because we hope that for such vertices, one of its neighbours in $G[S_1 \cup S_2]$ will be included in S_2. This enables us to get a better bound on the degrees of vertices in $G[S_1 \cup S_2]$, and in turn a better bound on the space complexity. We note that [10, 25] state their peeling decomposition in a different way, however, their algorithm, too, peels off vertices that have higher "overall" degree (this is made explicit in Lemma 2.1 and Lemma 2.2 of [25]). Instead we remove vertices that have higher degree in the graph remaining after previous peeling steps. For completeness we now describe the sampling scheme.

▶ **Definition 16** (Sampling Scheme). *Given integers $s_1, s_2, \cdots, s_{\beta-1}$ such that $s_1 > s_2 > \cdots > s_{\beta-1} > 0$, we sample sets $S_1, \cdots, S_{\beta-1}$ as follows.*

1. *Let $S_0 = V$ and for $i \geq 1$, sample S_i uniformly at random from S_{i-1} and let $|S_i| = s_i$.*

For a vertex v we define the level of v, denoted $l(v)$ to be the unique index l such that $v \in S_l$, but $v \notin S_{l+1}$. If $v \in S_{\beta-1}$, then we say that level of v, denoted $l(v)$ is $\beta - 1$.

We now describe our algorithm.

Algorithm 4 Computing a β-Ruling Set in Adversarial Streams.

Input: Take as input $(s_1, s_2, \cdots, s_{\beta-1})$, where $s_1 > s_2 > \cdots > s_{\beta-1}$

1 $S_0 \leftarrow V$ and $s_0 \leftarrow n$.
2 Sample $S_1, S_2, \cdots, S_{\beta-1}$ with parameters $s_1, \cdots, s_{\beta-1}$ according to Definition 16.
3 $\tilde{S}_i \leftarrow S_i$ for $i \in \{0, 1, \cdots, \beta-1\}$. We store all edges in $G[\cup_{j=0}^{\beta-1} \tilde{S}_j]$ in the stream.
4 For a vertex $v \in \cup_{j=0}^{\beta-1} \tilde{S}_j$, with $l(v) = l$ such that $l \leq \beta - 2$, if $\deg_{S_l}(v) \geq \frac{100 \cdot s_l \cdot \log n}{s_{l+1}}$, then let $\tilde{S}_j \leftarrow \tilde{S}_j \setminus \{v\}$ for all $j \leq l$. Delete any edges not in $G[\cup_{j=0}^{\beta-1} \tilde{S}_j]$.
5 Output MIS $\mathcal{M}$ of $G[\cup_{j=0}^{\beta-1} \tilde{S}_j]$.

We first claim that the algorithm indeed outputs a β-ruling set of G with high probability. Towards this, we prove the following claim.

▶ **Lemma 17.** *Consider any $v \in V$, then v is β-ruled by $\mathcal{M}$.*

Proof. We prove a stronger claim: We show that for all $i \in \{0, 1, \cdots, \beta-1\}$ each vertex $v \in S_i$ is $\beta - i$ ruled by $\mathcal{M}$. We prove this by induction. We first consider $S_{\beta-1}$ and start with the observation that $S_{\beta-1} \subseteq \cup_{j=1}^{\beta-1} \tilde{S}_j$ (this is because in Algorithm 4 Step 4 we only delete vertices of level less than $\beta - 1$ from $\cup_{j=1}^{\beta-1} \tilde{S}_j$). Since $\mathcal{M}$ is an MIS, this implies that for every $v \in S_{\beta-1}$, either v itself is in $\mathcal{M}$ or there is a neighbour u of v such that $u \in \mathcal{M}$. So, these vertices are 1-ruled.

We assume the claim holds for all $v \in S_i$ for some $i < \beta$. Under this inductive hypothesis, we want to prove this claim for all $v \in S_{i-1}$. Observe that if $l(v) \geq i$, then we know that $v \in S_i$ as well, and statement of the lemma holds by inductive hypothesis. So, we assume that $l(v) = i - 1$. Additionally, if $v \in \cup_{j=1}^{\beta-1} \tilde{S}_j$, then we know that v is 2-ruled. So, we assume that $v \notin \tilde{S}_{i-1}$. We therefore conclude that $\deg_{S_{i-1}}(v) \geq \frac{100 \cdot s_{i-1} \cdot \log n}{s_i}$. We now show that with high probability there is $u \in N_{S_{i-1}}(v)$ such that $u \in S_i$. Let Y_v be a random variable denoting $|N_{S_{i-1}}(v) \cap S_i|$. This random variable is the sum of negatively associated indicator random variables X_w for $w \in N_{S_{i-1}}(v)$, which takes value 1 if $w \in S_i$ and 0 otherwise. We have the following probability bound.

$$\Pr(X_w = 1) = \left(\frac{s_i}{s_{i-1}}\right)$$

$$\mathbb{E}[Y_v] = \deg_{S_{i-1}}(v) \cdot \left(\frac{s_i}{s_{i-1}}\right) \geq 100 \log n$$

Applying Proposition 3 with $\mu_{\min} = 100 \log n$ and $\delta = \frac{1}{2}$, we have:

$$\Pr\left(X_v \leq 50 \log n\right) = \exp\left(-(1/2)^2 (100 \log n)(1/2)\right) = O\left(\frac{1}{n^{12}}\right)$$
$$\Pr\left(X_v > 50 \log n\right) = 1 - O\left(\frac{1}{n^{12}}\right).$$

This implies that with high probability, v has a neighbour in S_i. By inductive hypothesis, all vertices in S_i are at most $\beta - i$ ruled by $\mathcal{M}$, this implies that v is $\beta - i + 1$ ruled by $\mathcal{M}$. ◀

We now bound the total number of edges stored in the memory at any time.

▶ **Lemma 18.** *The total number of edges stored in the memory at any point in time is at most* $\tilde{O}(n^2/s_1 + \sum_{j=2}^{\beta-1} s_{j-1}^2/s_j + s_{\beta-1}^2)$.

Proof. To prove this claim, we do a charging argument. We charge every edge (u, v) to the vertex which is at a lower level, breaking ties arbitrarily. Consider a vertex $v \in \cup_{j=0}^{\beta-1} \tilde{S}_j$ and suppose $l(v) = i$. This implies that $\deg_{S_i}(v) \leq \frac{100 \cdot s_i \cdot \log n}{s_{i+1}}$. So, the total charge on vertices $v \in \cup_{j=0}^{\beta-1} \tilde{S}_j$ with level i is at most $\tilde{O}(s_i^2/s_{i+1})$. If a vertex $v \in S_{\beta-1}$ is charged for an edge (u, v), then $u \in S_{\beta-1}$ as well. So, the total bound on the number of edges is $\tilde{O}(n^2/s_1 + \sum_{j=2}^{\beta-1} s_{j-1}^2/s_j + s_{\beta-1}^2)$. ◀

▶ **Lemma 19.** *The space complexity of Algorithm 4 is at most* $\tilde{O}(\beta \cdot n^{1+1/2^{\beta}-1})$.

Proof. Consider the bound on the edges in Lemma 18, we let $s_i = n^{1-\frac{2^i-1}{2^\beta-1}}$ for $i \in \{0, 1, \cdots, \beta - 1\}$. Consider the term $\frac{s_i^2}{s_{i+1}}$ in the sum, we have the following bound on it.

$$\frac{s_i^2}{s_{i+1}} = \frac{n^{2-\frac{2^{i+1}-2}{2^\beta-1}}}{n^{1-\frac{2^{i+1}-1}{2^\beta-1}}} = n^{1+\frac{1}{2^\beta-1}}.$$

Similarly, $s_{\beta-1}^2 = n^{2-\frac{2^\beta-2}{2^\beta-1}} = n^{1+\frac{1}{2^\beta-1}}$.This proves our claim. ◀

Our analysis gives an improved bound of $\tilde{O}(n^{4/3})$ for 2-ruling set. The previous best known algorithm had a space bound of $\tilde{O}(n^{3/2})$.

(α, β)-Ruling sets in a Adversarial Order Streams. A modification of Algorithm 4 allows us to compute (α, β)-ruling sets for all $\beta \geq \alpha \geq 2$ in $\tilde{O}(n^{1+\frac{1}{2^{\beta-\alpha+2}-1}})$ space: in the final step, we just compute an $(\alpha, \alpha - 1)$ ruling set instead of MIS. We give the algorithm, along with a proof of its space bound and correctness in the full version of the paper

4.2 Lower Bound for Ruling Sets

In this section, our goal is to prove the following theorem.

▶ **Theorem 20.** *Every randomized constant error one-pass streaming algorithm in the adversarial edge arrival model that computes an $(\alpha, \alpha - 1)$-ruling set in an n-vertex graph for any <u>even</u> $\alpha > 2$, requires $\Omega(n^2/\alpha^2)$ space.*

We now give a hard distribution $\mathcal{D}_{\mathsf{Rule}}$ for our algorithm (refer to Figure 1).

Distribution 5 Distribution $\mathcal{D}_{\mathsf{Rule}}$.

Output: An instance $(G_{1/2}, a, b)$.

1 Let $\alpha := 2c$ for $c \geq 2$ be an even number and let $n = 2c \cdot N + 4c \cdot (c-1)$. Let $G_{1/2}$ be a random graph on N vertices with each edge being included with probability $\frac{1}{2}$

2 Create α disjoint copies $G_1, \cdots, G_\alpha$ of $G_{1/2}$.

3 Create α paths P_x^i with vertices $x_1^i, \cdots, x_{c-1}^i$ and P_y^i with vertices $y_1^i, \cdots, y_{c-1}^i$ for $i \in [\alpha]$.

4 Pick a pair of vertices (a, b) of $G_{1/2}$ uniformly at random. Let (a_i, b_i) be the copy of (a, b) in G_i for $i \in [\alpha]$.

5 For all $i \in [\alpha]$, add an edge between a_i and x_1^i, and between b_i and y_1^i.

6 For all $i \in [\alpha - 1]$, add an edge between each $w \in V(G_i) \setminus \{a_i, b_i\}$ and $z \in V(G_{i+1}) \setminus \{a_{i+1}, b_{i+1}\}$. Similarly, add an edge between each $w \in V(G_1) \setminus \{a_1, b_1\}$ and $z \in V(G_\alpha) \setminus \{a_\alpha, b_\alpha\}$.

To relate the ruling set problem with Index_t, we will condition on the following event.

Event $\mathcal{E}_{\text{DIST}}$. For all $u, v \in G_{1/2}$, $\text{dist}(u, v) \leq 2$.

We state the following claim.

▷ Claim 21. $\Pr(\mathcal{E}_{\text{DIST}}) \geq 1 - o(1)$.

Proof sketch. To bound the probability of $\mathcal{E}_{\text{DIST}}$, it is sufficient to bound the probability that there is a pair of vertices which don't share a common neighbour. Consider two vertices u and v, the probability that a fixed vertex $w \notin N(u) \cap N(v)$ is $3/4$. Therefore, the probability that $N(u) \cap N(v) = \emptyset$ is at most $(3/4)^N$. Thus, taking a union bound over all pairs of vertices, we have the desired claim. ◁

From now on, while discussing the properties of G, we condition on the event $\mathcal{E}_{\text{DIST}}$.

Before moving on to the proof of Theorem 20, we first clarify some notation. In what follows, $G := (G_{1/2}, a, b)$ will denote a graph sampled from $\mathcal{D}_{\mathsf{Rule}}$, and $V(G)$ will denote the vertices of G (This includes vertices of G_i, P_x^i and P_y^i for all $i \in [\alpha]$). The graph G_i for $i \in [\alpha]$ will denote the ith copy of $G_{1/2}$ in G, and $V(G_i)$ denotes its vertex set. The set $V(G_i)$ does not include vertices of P_x^i and P_y^i. We use $V(G_i \cup P_x^i \cup P_y^i)$ to denote the vertices of G_i, P_x^i, and P_y^i for all $i \in [\alpha]$.

Towards proving our main theorem, we state a few properties of $(G_{1/2}, a, b)$. It might help to refer to Figure 1 for the following claim.

▷ Claim 22. Let $G := (G_{1/2}, a, b)$ be drawn from $\mathcal{D}_{\mathsf{Rule}}$. For any $j \in [\alpha]$, consider any $u \in V(G_j) \setminus \{a_j, b_j\}$ and $v \in V(G) \setminus \{x_k^{c-1}, y_k^{c-1}\}$ for $k \equiv c + j \mod \alpha$; then $\text{dist}(u, v) \leq 2c - 1 = \alpha - 1$. Moreover $\text{dist}(u, x_k^{c-1}) = 2c = \alpha$ and $\text{dist}(u, y_k^{c-1}) = 2c = \alpha$.

Proof. We prove the claim for $j = 1$, and the proof is identical for all other values of j. We consider $u \in V(G_1) \setminus \{a_1, b_1\}$; we want to argue that u is at distance at most $2c - 1$ from all $v \in V(G) \setminus \{x_{c+1}^{c-1}, y_{c+1}^{c-1}\}$. We consider the following cases.

1. Consider any $v \in V(G_i) \setminus \{a_i, b_i\}$ for $i \leq c + 1$, then $\text{dist}(u, v) \leq c$. If $v \in V(G_1)$ then $\text{dist}(u, v) \leq 2$ since we condition on $\mathcal{E}_{\text{DIST}}$. If $v \in V(G_i) \setminus \{a_i, b_i\}$ for $i \neq 1$, u and v are connected by a path of length $i - 1$. For $i \geq c + 1$, the argument is identical

2. We can reach the vertices x_c^{c-1} and y_c^{c-1} in $2c-1$ hops from u: since we condition on $\mathcal{E}_{\text{DIST}}$, there is a $w \in V(G_c) \setminus \{a_c, b_c\}$ such that $\text{dist}(x_c^{c-1}, w) = c$. Moreover, $d(u, w) = c - 1$. This implies that all vertices of $V(G_i \cup P_x^i \cup P_y^i)$ for $i \le c$ are $2c-1$ ruled by u. The proof for all $i \ge c+2$ is identical.
3. Consider the vertices x_{c+1}^{c-1} and y_{c+1}^{c-1}. For these vertices, $\text{dist}(u, x_{c-1}^{c+1}) = 2c$ and $\text{dist}(u, y_{c-1}^{c+1}) = 2c$. Additionally, for all other vertices in the set $v \in V(G_{c+1} \cup P_x^{c+1} \cup P_y^{c+1}) \setminus \{x_{c+1}^{c-1}, y_{c+1}^{c-1}\}$, $\text{dist}(u, v) \le 2c - 1$.

◁

▶ **Lemma 23.** *Let I be a $(\alpha, \alpha-1)$-ruling set of $(G_{1/2}, a, b)$. Suppose for some $i \in [2c]$, $x_i^{c-1} \in I$; if $(a, b) \notin E(G_{1/2})$, then $y_i^{c-1} \in I$ as well.*

Proof. We show the claim for $i = 1$, and for other values, the proof is identical. To prove this claim, we show that the vertices that $(2c-1)$-rule x_1^{c-1} and y_1^{c-1} are the same. Let $A_1 = \{v \notin \{x_1^{c-1}, y_1^{c-1}\} \mid \text{dist}(v, x_1^{c-1}) = 2c-1\}$. These are the vertices that $2c-1$ rule x_1^{c-1}, not including y_1^{c-1}. Similarly we define $A_2 = \{v \notin \{x_1^{c-1}, y_1^{c-1}\} \mid \text{dist}(v, y_1^{c-1}) = 2c-1\}$. We claim that $A_1 = A_2$. We consider the following cases.

1. For any x_1^k for $k \le c-2$, $\text{dist}(x_1^k, y_1^{c-1}) \le 2c-1$ and similarly, $\text{dist}(x_1^k, y_1^{c-1}) \le 2c-1$. This shows that $x_1^k \in A_1$ and $x_1^k \in A_2$ as well. Similarly, we can show that for $k \le c-2$, $y_1^k \in A_1$ and $y_1^k \in A_2$.
2. For any vertex $v \in V(G_1)$, $\text{dist}(v, x_1^{c-1}) \le c+1$ and similarly, $\text{dist}(v, y_1^{c-1}) \le c+1$ (since we condition on $\mathcal{E}_{\text{DIST}}$), and therefore, $v \in A_1$ and $v \in A_2$.
3. We now want to consider all vertices $v \notin V(G_1 \cup P_x^1 \cup P_y^1)$. Since we condition on $\mathcal{E}_{\text{DIST}}$, we know that there exists a vertex $u \in V(G_1) \setminus \{a_1, b_1\}$, such that $\text{dist}(u, x_1^{c-1}) = \text{dist}(u, y_1^{c-1}) = c$. Therefore, for any $v \notin V(G_1 \cup P_x^1 \cup P_y^1)$, $\text{dist}(v, x_1^{c-1}) = \text{dist}(v, y_1^{c-1})$.

From the above three cases, we conclude that $A_1 = A_2$. So, if $x_1^{c-1} \in I$, then for all $w \in A_2$, $w \notin I$. Since $(a, b) \notin E(G_{1/2})$, this implies that $\text{dist}(x_1^{c-1}, y_1^{c-1}) = 2c$. So, $y_1^{c-1} \in I$. ◀

▷ Claim 24. For all $i \in [\alpha]$, we denote a_i and b_i by x_i^0 and y_i^0. Let I be a $(\alpha, \alpha-1)$-ruling set of $(G_{1/2}, a, b)$. Suppose $x_i^k \in I$ for some $k \le c-2$. If $(a, b) \notin E(G_{1/2})$, then $\{x_j^{c-1}, y_j^{c-1}\} \subseteq I$ for $j \equiv (i + c - k - 1) \mod \alpha$.

Proof. Consider x_j^{c-1}; there is a vertex $w \in V(G_j) \setminus \{a_j, b_j\}$ such that $\text{dist}(w, x_j^{c-1}) = c$ and there is a vertex $z \in V(G_i) \setminus \{a_i, b_i\}$ such that $\text{dist}(z, x_i^k) = k+1$. Additionally, by our construction, $\text{dist}(w, z) = j - i$. This implies that: $\text{dist}(x_i^k, x_j^{c-1}) = c + (k+1) + (j-i) = 2c$. We conclude that both vertices x_j^{c-1} and y_j^{c-1} are not $(2c-1)$-ruled by x_i^k.

Additionally, for all $k < j$, all vertices in $V(G_k \cup P_x^k \cup P_y^k)$ are excluded from being added in I, since all these vertices are $(2c-1)$-ruled by x_i^k. Similarly, all vertices in $V(G_j \cup P_x^j \cup P_y^j) \setminus \{x_j^{c-1}, y_j^{c-1}\}$ are $(2c-1)$-ruled by x_i^k, so these vertices are excluded from I as well. Finally, we argue that all vertices in $V(G_k \cup P_x^k \cup P_y^k)$ for $k \ge j+1$ that could $(2c-1)$-rule x_j^{c-1} and y_j^{c-1} are excluded from I. We prove this claim for $k = j+1$, but it is identical for all other values of k.

In $V(G_{j+1} \cup P_x^{j+1} \cup P_y^{j+1})$, the vertices that can be included in I are $\{x_{j+1}^{c-2}, y_{j+1}^{c-2}, x_{j+1}^{c-1}, y_{j+1}^{c-1}\}$, however $\text{dist}(x_{j+1}^{c-2}, x_j^{c-1}) = (c-1) + 1 + c = 2c$. Therefore, we conclude that all the vertices other than x_j^{c-1} and y_j^{c-1} that are still candidates for I cannot $(2c-1)$-rule x_j^{c-1} and y_j^{c-1}. Additionally, since $\text{dist}(x_j^{c-1}, y_j^{c-1}) = 2c$, this implies that $\{x_j^{c-1}, y_j^{c-1}\} \subseteq I$. ◁

▶ **Lemma 25.** *Let I be any $(\alpha, \alpha-1)$-ruling set of $(G_{1/2}, a, b)$. Then, there is an $i \in [\alpha]$ such that $\{x_i^{c-1}, y_i^{c-1}\} \subseteq I$ if and only if $(a, b) \notin G$.*

Proof. Suppose $(a, b) \in G_{1/2}$, then for any $i \in [\alpha]$, $\text{dist}(x_i^{c-1}, y_i^{c-1}) = 2c - 1$, so, both cannot be included in I. Consider the other direction when $(a, b) \notin G_{1/2}$. Let I be any $(\alpha, \alpha - 1)$-ruling set of $(G_{1/2}, a, b)$ and let $v \in I$. We consider the following cases for v.

1. $\boldsymbol{v \in V(G_i) \setminus \{a_i, b_i\}}$ **for some** $\boldsymbol{i \in [\alpha]}$. In this case, from Claim 22, we conclude that x_k^{c-1} and y_k^{c-1} for $k \equiv i + c \mod \alpha$ are the only two vertices that are not ruled. Additionally, since $\text{dist}(x_k^{c-1}, y_k^{c-1}) = \alpha$, we conclude that $\{x_k^{c-1}, y_k^{c-1}\} \subseteq I$.
2. $\boldsymbol{v = x_i^{c-1}}$ **or** $\boldsymbol{v = y_i^{c-1}}$ **for some** $\boldsymbol{i \in [\alpha]}$**.** In this case, Lemma 23, implies that $\{x_i^{c-1}, y_i^{c-1}\} \subseteq I$.
3. $\boldsymbol{v = x_i^k}$ **or** $\boldsymbol{v = y_i^k}$ **for some** $\boldsymbol{k \leq c - 2}$**.** In this case, from Claim 24, we conclude that $\{x_j^{c-1}, y_j^{c-1}\} \subseteq I$ for $j \equiv (i + c - k - 1) \mod 2c$.

This proves our lemma. ◀

We now state the next lemma, which will imply Theorem 20. Throughout this section, $\mathcal{A}_{\mathsf{Rule}}$ is an algorithm that for an even $\alpha > 2$ computes an $(\alpha, \alpha - 1)$ ruling set with error at most δ when the input is sampled from $\mathcal{D}_{\mathsf{Rule}}$.

▶ **Lemma 26.** *There exists a $(\delta + o(1))$-error protocol π_{Ind} for Index_t on $\mathcal{D}_{\mathsf{Ind}}$ such that the communication cost of π_{Ind} is $s(\mathcal{A}_{\mathsf{Rule}})$, where $s(\mathcal{A}_{\mathsf{Rule}})$ is the space complexity of $\mathcal{A}_{\mathsf{Rule}}$.*

Proof. We begin the proof by designing the protocol π_{Ind} (see Protocol 6). From now on, $t = \binom{N}{2}$ and $n = 2c \cdot N + 4c(c - 1)$.

Protocol 6 Protocol π_{Ind}.

Input: An instance $(X, \sigma) \sim \mathcal{D}_{\mathsf{Ind}}$
Output: Yes if $X_\sigma = 1$ and No if $X_\sigma = 0$.

1 Alice creates an N-vertex graph H whose adjacency matrix is given by X.
2 She then creates α disjoint copies of H: $G_1, G_2, \cdots, G_\alpha$.
3 Alice then creates 4α paths on $c - 1$ vertices: $\{P_x^i\}_{1 \leq i \leq \alpha}$ and $\{P_y^i\}_{1 \leq i \leq \alpha}$.
4 Bob treats $\sigma \in [t]$ as an edge (a, b) of an n-vertex graph. He adds edges (a_i, x_i^1), and (b_i, y_i^1) for all $i \in [\alpha]$. Additionally, for all $i \in [\alpha - 1]$, he adds edges (x, y) for all $x \in V(G_i) \setminus \{a_i, b_i\}$ and for all $y \in V(G_{i+1}) \setminus \{a_{i+1}, b_{i+1}\}$. Edges (x, y) are also added for all $x \in V(G_1) \setminus \{a_1, b_1\}$ and $y \in V(G_{2c}) \setminus \{a_{2c}, b_{2c}\}$.
5 The players compute an $(\alpha, \alpha - 1)$-ruling set on their graph using $\mathcal{A}_{\mathsf{Rule}}$ and output No if there is an $i \in [\alpha]$ such that $\{x_i^{c-1}, y_i^{c-1}\} \subseteq I$, where I is an $(\alpha, \alpha - 1)$-ruling set. Output Yes if there is no such $i \in [\alpha]$.

The distribution of (G, a, b) created by π_{Ind} matches the distribution $\mathcal{D}_{\mathsf{Rule}}$ exactly. From Lemma 25 and using the fact that $X_\sigma = 1$ if and only if $(a, b) \in E(G)$, we conclude that π_{Ind} outputs Yes if $X_\sigma = 1$ and No if $X_\sigma = 0$. Since the event $\neg\mathcal{E}_{\text{DIST}}$ happens with probability $o(1)$, we conclude that: $\Pr(\pi_{\mathsf{Ind}} \text{ errs}) \leq \Pr(\mathcal{A}_{\mathsf{Rule}} \text{ errs}) + \Pr(\neg\mathcal{E}_{\text{DIST}}) = \delta + o(1)$. Since Alice sends the memory contents of $\mathcal{A}_{\mathsf{Rule}}$, we know that the communication cost of π_{Ind} is $s(\mathcal{A}_{\mathsf{Rule}})$. ◀

Proof of Theorem 20. If there was an $o(n^2/\alpha^2)$-space δ-error one-pass streaming algorithm for the problem of finding a $(\alpha, \alpha - 1)$-ruling set of a graph G, then Alice and Bob would be able to solve Index_t using this algorithm. Alice would run it on her input, and send the contents of the memory to Bob, who would run it on his input and give the ruling set output by the algorithm. Since the contents of the memory are $o(n^2/\alpha^2)$, this would give a protocol for Index_t with communication complexity $o(n^2/\alpha^2) = o(t)$ and $(\delta + o(1))$-error, thus contradicting Proposition 5. ◀

References

1. Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Ami Paz. Smaller cuts, higher lower bounds. *arXiv preprint*, 2019. arXiv:1901.01630.
2. Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2237–2246. JMLR.org, 2015.
3. Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. *Analyzing Graph Structure via Linear Measurements*, pages 459–467. SIAM, 2012. doi:10.1137/1.9781611973099.40.
4. Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1616–1635. SIAM, 2019. doi:10.1137/1.9781611975482.98.
5. Sepehr Assadi and Soheil Behnezhad. Beating two-thirds for random-order streaming matching. *CoRR*, abs/2102.07011, 2021. arXiv:2102.07011.
6. Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786. SIAM, 2019. doi:10.1137/1.9781611975482.48.
7. B. Awerbuch, M. Luby, A.V. Goldberg, and S.A. Plotkin. Network decomposition and locality in distributed computation. In *30th Annual Symposium on Foundations of Computer Science*, pages 364–369, 1989. doi:10.1109/SFCS.1989.63504.
8. Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. *J. ACM*, 63(3), 2016. doi:10.1145/2903137.
9. Aaron Bernstein. Improved bounds for matching in random-order streams. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 12:1–12:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.12.
10. Tushar Bisht, Kishore Kothapalli, and Sriram Pemmaraju. Brief announcement: Super-fast t-ruling sets. In *PODC*, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2611462.2611512.
11. Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 641–650, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374470.
12. Yi-Jun Chang, Wenzheng Li, and Seth Pettie. An optimal distributed $(\delta + 1)$-coloring algorithm? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 445–456, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188964.
13. Graham Cormode, Jacques Dark, and Christian Konrad. Independent sets in vertex-arrival streams. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 45:1–45:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.45.
14. Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, 1998. doi:10.1002/(SICI)1098-2418(199809)13:2\%3C99::AID-RSA1\%3E3.0.CO;2-M.
15. Alireza Farhadi, MohammadTaghi Hajiaghayi, Tung Mai, Anup Rao, and Ryan A. Rossi. Approximate maximum matching in random streams. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, page 1773–1785, USA, 2020. Society for Industrial and Applied Mathematics.

16 Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, page 491–500, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3293611.3331603.

17 Beat Gfeller and Elias Vicari. A randomized distributed algorithm for the maximal independent set problem in growth-bounded graphs. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '07, page 53–60, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1281100.1281111.

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

19 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. A deterministic almost-tight distributed algorithm for approximating single-source shortest paths. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 489–498, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897638.

20 H. Jowhari, M. Saglam, and G. Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. *arXiv*, abs/1012.4889, 2011. arXiv:1012.4889.

21 Tomasz Jurdziński and Krzysztof Nowicki. Mst in o(1) rounds of congested clique. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, page 2620–2632, USA, 2018. Society for Industrial and Applied Mathematics.

22 Christian Konrad. MIS in the congested clique model in $o(\log\log\Delta)$ rounds. *CoRR*, abs/1802.07647, 2018. arXiv:1802.07647.

23 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, volume 7408 of *Lecture Notes in Computer Science*, pages 231–242. Springer, 2012. doi:10.1007/978-3-642-32512-0_20.

24 Christian Konrad, Sriram V. Pemmaraju, Talal Riaz, and Peter Robinson. The Complexity of Symmetry Breaking in Massive Graphs. In Jukka Suomela, editor, *33rd International Symposium on Distributed Computing (DISC 2019)*, volume 146 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:18, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.DISC.2019.26.

25 Kishore Kothapalli and Sriram V. Pemmaraju. Super-fast 3-ruling sets. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 136–147. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.136.

26 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In Frank Thomson Leighton and Allan Borodin, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 596–605. ACM, 1995. doi:10.1145/225058.225277.

27 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *J. ACM*, 63(2):17:1–17:44, 2016. doi:10.1145/2742012.

28 Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Rec.*, 43(1):9–20, 2014. doi:10.1145/2627692.2627694.

29 Morteza Monemizadeh, S. Muthukrishnan, Pan Peng, and Christian Sohler. Testable bounded degree graph properties are random order streamable. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 131:1–131:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.131.

30 Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. Fast distributed algorithms for connectivity and mst in large graphs. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '16, page 429–438, New York, NY, USA, 2016. Association for Computing Machinery. `doi:10.1145/2935764.2935785`.

31 Johannes Schneider and Roger Wattenhofer. A new technique for distributed symmetry breaking. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '10, page 257–266, New York, NY, USA, 2010. Association for Computing Machinery. `doi:10.1145/1835698.1835760`.

32 David Wajc. Negative association-definition, properties, and applications, 2017. URL: `http://www.cs.cmu.edu/~dwajc/notes/Negative%20Association.pdf`.

A Missing Figures

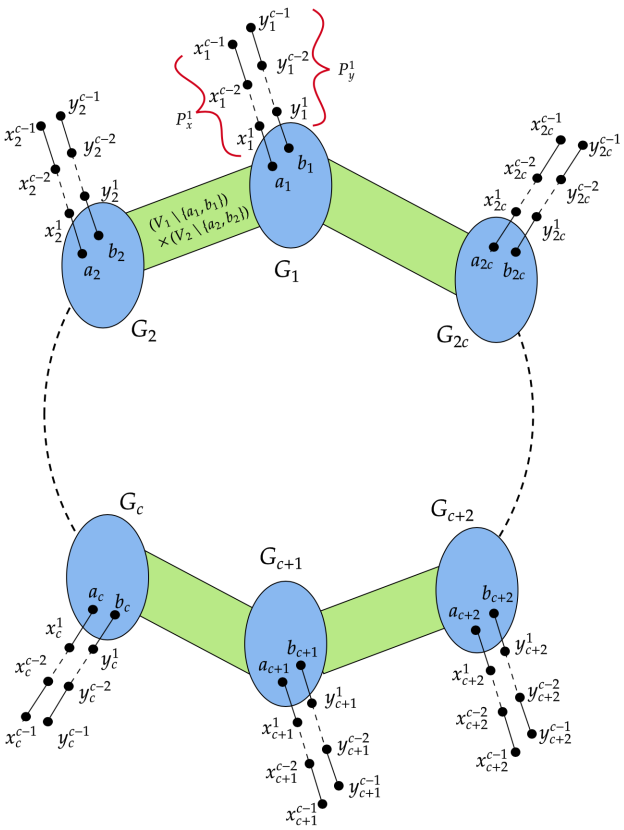

Figure 1 To create an instance $(G_{1/2}, a, b)$, graph $G_{1/2}$ is sampled by including every edge on N-vertices with probability $1/2$. Then $\alpha := 2c$ disjoint copies of $G_{1/2}$ are created. Additionally, $4c$ paths on $c-1$ vertices, $\left\{P_x^i\right\}_{1 \le i \le 2c}$ and $\left\{P_y^i\right\}_{1 \le i \le 2c}$ are created. A random edge (a, b) is sampled and $4c$ edges, $\left\{(a_i, x_i^1)\right\}_{1 \le i \le 2c}$ and $\left\{(b_i, y_i^1)\right\}_{1 \le i \le 2c}$ are added. Edges (x, y) for all $x \in V(G_i) \setminus \{a_i, b_i\}$ and $y \in V(G_{i+1}) \setminus \{a_{i+1}, b_{i+1}\}$ for all $i \in [2c-1]$ are added. Finally, edges (w, z) for all $w \in V(G_1) \setminus \{a_1, b_1\}$ and $z \in V(G_{2c}) \setminus \{a_{2c}, b_{2c}\}$ are also added.