Independent Sets in Vertex-Arrival Streams

Graham Cormode
University of Warwick, UK
g.cormode@warwick.ac.uk

Jacques Dark
University of Warwick, UK
j.dark@warwick.ac.uk

Christian Konrad
University of Bristol, UK
christian.konrad@bristol.ac.uk

Abstract

We consider the maximal and maximum independent set problems in three models of graph streams:

- In the edge model we see a stream of edges which collectively define a graph; this model is well-studied for a variety of problems. We show that the space complexity for a one-pass streaming algorithm to find a maximal independent set is quadratic (i.e. we must store all edges). We further show that it is not much easier if we only require approximate maximality. This contrasts strongly with the other two vertex-based models, where one can greedily find an exact solution in only the space needed to store the independent set.
- In the "explicit" vertex model, the input stream is a sequence of vertices making up the graph. Every vertex arrives along with its incident edges that connect to previously arrived vertices. Various graph problems require substantially less space to solve in this setting than in edge-arrival streams. We show that every one-pass c-approximation streaming algorithm for maximum independent set (MIS) on explicit vertex streams requires $\Omega(\frac{n^2}{c^6})$ bits of space, where n is the number of vertices of the input graph. It is already known that $\tilde{\Theta}(\frac{n^2}{c^2})$ bits of space are necessary and sufficient in the edge arrival model (Halldórsson *et al.* 2012), thus the MIS problem is not significantly easier to solve under the explicit vertex arrival order assumption. Our result is proved via a reduction from a new multi-party communication problem closely related to pointer jumping.
- In the "implicit" vertex model, the input stream consists of a sequence of objects, one per vertex. The algorithm is equipped with a function that maps pairs of objects to the presence or absence of edges, thus defining the graph. This model captures, for example, geometric intersection graphs such as unit disc graphs. Our final set of results consists of several improved upper and lower bounds for interval and square intersection graphs, in both explicit and implicit streams. In particular, we show a gap between the hardness of the explicit and implicit vertex models for interval graphs.

2012 ACM Subject Classification Theory of computation → Lower bounds and information complexity; Theory of computation → Streaming models

Keywords and phrases streaming algorithms, independent set size, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.45

Category Track A: Algorithms, Complexity and Games

Related Version The full version of this paper is available at: `https://arxiv.org/abs/1807.08331`.

Funding *Graham Cormode*: Supported by European Research Council grant ERC-2014-CoG 647557.
Jacques Dark: Supported by an EMEA Microsoft Research scholarship. Part of the work was done while J.D. was at the Alan Turing Institute, under EPSRC grant EP/N510129/1.
Christian Konrad: C.K. carried out most work on this paper while being at the University of Warwick. He was supported by the Centre for Discrete Mathematics and its Applications (DIMAP) at Warwick University and by EPSRC award EP/N011163/1.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 45; pp. 45:1–45:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The streaming model supposes that, rather than being loaded into memory all at once, the input is received piece-by-piece over a period of time. Only a sublinear amount of memory (in the input size) is made available, preventing any algorithm from "seeing" even a constant fraction of the whole input at once.

In graph streams (see [19] for an excellent survey), we distinguish between the "edge-arrival" model, where the stream consists of individual edges arriving in any order, and the "vertex-arrival" model, where the stream consists of batches of edges incident to a particular vertex – as each vertex "arrives" we are given all the edges from the new vertex to previously arrived vertices. We will shorten the names to edge streams and vertex streams, respectively. Problems are always at least as hard on edge streams as on vertex streams (as any vertex stream is also a valid edge stream).

There is a further variant which we will call "implicit" vertex streams (as opposed to the normal explicit representation). In this model, the stream consists of a series of small (polylog(n)-sized) identifiers – one per vertex. We are additionally provided with some symmetric function or oracle which maps a pair of identifiers to a Boolean output indicating whether the two vertices are connected or not. This implicitly defines a graph over the list of identifiers received. Geometric intersection graphs, received as a stream of geometric objects, are the most natural members of this class. For example, a unit interval intersection graph can be given by a set of points in $\mathbb{R}$. Then a pair of vertices x, y are adjacent if and only if $|x - y| \leq 1$.

Explicit and implicit vertex streams are closely related but distinct, with neither being strictly "harder" than the other. For example: it is easy to count exactly the number of edges in $\tilde{O}(1)$ space[1] for an explicit vertex stream, however, doing so for an implicit stream requires linear space – otherwise we cannot hope to know how many edges are incident to the final vertex. On the other hand: implicit vertex streams can be stored entirely in $\tilde{O}(n)$ space, whereas explicit vertex streams require $\Omega(n^2)$ space to store the full structure.

Maximum Independent Set (MIS) is an important problem on graphs. The task is to find a largest subset of vertices which have no edges between them. The size of a MIS in a graph G is denoted $\alpha(G)$, the independence number of G. Unfortunately, it is NP-hard to find a maximum independent set in a general graph [16], and even hard to approximate within a factor of $n^{1-\epsilon}$, for any $\epsilon > 0$ [20]. It is also known to be hard in the edge-arrival streaming model: Halldórsson *et al.* [15] showed that space $\tilde{\Theta}(\frac{n^2}{c^2})$ is necessary (and sufficient) for computing a c-approximation on an n-vertex graph, despite being allowed unlimited computation.

Our Results. In this paper, we study the hardness of approximate MIS in the explicit and implicit vertex streaming models. Since many problems are significantly easier to solve in vertex streams than in edge streams, we ask whether this is also the case for the MIS problem. As our main result, we answer this question in the negative:

▶ **Theorem 1.** *Any constant error one-pass c-approximation streaming algorithm for* MIS *(or the size of a* MIS*) in the explicit vertex stream model requires $\Omega\left(\frac{n^2}{c^6}\right)$ bits of space.*

[1] All space bounds in this paper are given as number of bits. We use $\tilde{O}$, $\tilde{\Theta}$, and $\tilde{\Omega}$ to mean O, Θ, and Ω (respectively) with log factors suppressed.

Space Bound	Approx. MIS $\tilde{O}(\alpha(G))$	Approx. $\alpha(\mathbf{G})$ poly$(\log n)$	$\Omega(n)$
Unit Interval	2 (Greedy alg.)	$O\left(\frac{\log^2 n}{\log\log n}\right)$ [9]	$< 5/3$

Figure 1 Approximation factors for explicit vertex streams.

Space Bound	Approx. MIS $\tilde{O}(\alpha(G))$	Approx. $\alpha(\mathbf{G})$ poly$(\log n, \epsilon^{-1})$	$\Omega(n)$
Unit Interval	3/2 [11]	$3/2+\epsilon$ [7]	$< 3/2$ [11]
Interval	2 [11]	$2+\epsilon$ [7]	< 2 [11]
Unit Square	3	$3+\epsilon$	$< 5/2$

Figure 2 Approximation factors for implicit vertex streams. The first column concerns algorithms that output independent sets themselves, while the second columns concerns algorithms that output estimations of the maximum independent set size. Results from this paper are highlighted.

Our lower bound also holds for the Minimum Vertex Coloring (MVC) problem, where the objective is to color the vertices of the input graph such that adjacent vertices have different colors, using the fewest colors possible. This quantity is the chromatic number and denoted by $\chi(G)$. Our result is the first lower bound known for this problem, even for edge streams (in particular, the work by Halldórsson *et al.* [15] does not imply such a result).

Next, we show that the situation is very different for the related *maximal* independent set problem, where we need to find a subset of non-adjacent vertices that cannot be enlarged. While it is easy to maintain a maximal independent set in vertex arrival streams (both explicit and implicit) using space $\tilde{O}(\alpha(G)) = \tilde{O}(n)$, we prove that $\Omega(n^2)$ space is required in the edge-arrival model. We further show that even if we relax the maximality constraint to *approximate maximality* and allow for a slightly sublinear number of vertices that are not adjacent to vertices of the independent set then space $\Omega(n^{2-o(1)})$ is still required.

Finally, we show various improved upper and lower bounds for certain geometric intersection graph classes in both vertex streaming models: unit interval intersection graphs given as explicit vertex streams require $\Omega(n)$ space to get a better than $\frac{5}{3}$-approximation to $\alpha(G)$, making them harder than their implicit vertex stream equivalents; and we can 3-approximate MIS for a stream of unit squares in the plane using $\tilde{O}(\alpha(G))$ space, but achieving better than a $\frac{5}{2}$-approximation to $\alpha(G)$ requires $\Omega(n)$ space. Figures 2 and 1 shows these results in the context of previously known bounds.

Techniques. Halldórsson *et al.* [15] proved their space lower bound for MIS in edge streams via the *one-way two-party communication framework*. Two parties, denoted Alice and Bob, each hold a subset of the edges of the input graph. Alice sends a single message to Bob, who, upon receipt, outputs a large independent set. Via a reduction from a well-known communication problem, they showed that if Bob outputs a c-approximate MIS then Alice must send a message of size $\tilde{\Omega}(\frac{n^2}{c^2})$ to Bob. A common reduction then implies that the same lower bound holds for the space complexity of one-pass streaming algorithms.

Proving a similar result in vertex streams is significantly harder since the two-party communication abstraction cannot yield the desired result. When partitioning the vertices of a vertex stream between Alice and Bob both parties hold *vertex-induced subgraphs*, as opposed to the *spanning subgraphs* obtained when partitioning the edges of an edge stream. Since an independent set in an induced subgraph is also an independent set in the whole

graph, and since either Alice or Bob holds at least half the vertices of any MIS, one of them must already know a 2-approximation to the MIS. Using the same reasoning, it is trivial to compute a p-approximation in the one-way p-party communication setting. To obtain our lower bound result, we therefore need to consider multi-party communication with $\Omega(c)$ parties.

To this end, we define a new k-party communication problem denoted $\textsc{Chain}_k$ – which can be seen as chaining together multiple two-party instances of the well-known Index problem (see Definition 2) that are guaranteed to have the same answer. We first give a $\Omega(\frac{n}{k^2})$ lower bound for $\textsc{Chain}_k$ by showing a reduction from a multi-party pointer jumping problem [8]. We then improve this lower bound to $\Omega(\frac{n}{k})$ for k up to $\tilde{O}\left(\sqrt[4]{n}\right)$ using the same party elimination techniques described in [8]. The actual reduction from $\textsc{Chain}_k$ to MIS relies on an involved graph construction using erasure codes based on affine planes.

Our lower bound for the computation of a maximal independent set in edge streams is obtained via a reduction from the Index problem in the two-party communication framework. This construction is then extended to yield results for approximate maximality via a construction involving Ruzsa-Szemerédi graphs.

Our upper bound results on 2D geometric intersection graphs are obtained by generalizing 1D bounds, with more work to cover the increased number of cases that occur in 2D. The lower bounds involve intricate packing arguments to show that knowledge of $\alpha(G)$ can be used to recover encoded information, which is used in conjunction with our multiparty $\textsc{Chain}_k$ problem to demonstrate approximation hardness.

Further Related Work. Grouping the three streaming models:

- *Edge Streams.* As previously mentioned, Halldórsson *et al.* [15] showed that for general graphs in the edge-arrival model $\tilde{\Omega}\left(\frac{n^2}{c^2}\right)$ space is required to obtain a c-approximation to the maximum independent set size (or maximum clique size). A corresponding $\tilde{O}\left(\frac{n^2}{c^2}\right)$ space random sampling algorithm shows that this is tight up to logarithmic factors. Braverman *et al.* [6] showed that space $\Omega(\frac{m}{c^2})$ is needed, even if $c = o(\log n)$, where m is the number of edges of the input graph, though this bound only holds for small m.
- *Explicit Vertex Streams.* The work of Halldórsson *et al.* [13] gives an $O(n \log n)$ space streaming algorithm which can find an independent set of expected size at least $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$. On general graphs, this only gives a $\Theta(n)$-approximation, but for polynomially bounded independence graphs, this gives a polylog(n)-approximation [14]. In our prior work, we showed how to return an estimate $\gamma \in \Omega\left(\frac{\beta(G)}{\log n}\right)$ with $\gamma \leq \alpha(G)$ from an explicit vertex arrival stream using only $O(\log^3 n)$ space [9]. This result, for example, gives a $O(\frac{\log^2 n}{\log \log n})$-approximation on unit interval graphs (see Figure 1). However, the technique samples vertices based on their degree and does not extend to implicit vertex streams.

  Braverman *et al.* [6] showed that in a variant of the vertex arrival model, where every vertex arrives together with *all* its incident edges (as opposed to only the edges incident to previously arrived vertices), space $\Omega(\frac{m}{c^3})$ is required for computing a c-approximate MIS. In their construction the input graph has $\Theta(nc)$ edges, which thus yields a lower bound of $\Omega(\frac{n}{c^2})$. Observe that our lower bound for explicit vertex streams is $\Omega(\frac{n^2}{c^6})$, a quadratic improvement for constant c.
- *Implicit Vertex Streams.* In [11], it was shown that it is possible to $\frac{3}{2}$-approximate MIS for the intersection graph of a unit interval stream using $\tilde{O}(\alpha(G))$ space. In the same space, a 2-approximation is possible for arbitrary interval streams. Both are shown to

be tight: any $(\frac{3}{2}-\epsilon)$-approximation for unit intervals, or $(2-\epsilon)$ for general intervals, requires $\Omega(n)$ space. By clever use of sampling, the result can be adapted to provide an approximation of $\alpha(G)$ of $\frac{3}{2}+\epsilon$ for unit intervals and $2+\epsilon$ for general intervals with only polylog(n, ϵ^{-1}) space [7].

Concurrent Work. Independently of, and concurrently with, an earlier version of this paper [10, v1], Assadi *et al.* [3] also gave an $\Omega(n^2)$ lower bound for maximal independent set in edge streams using a similar construction.

Outline. We present our main result, the lower bound for MIS in vertex streams, in Section 2. Our lower bounds for maximal and approximately maximal independent sets in edge streams are given in Section 3. Section 4 covers our results on interval and square graphs, and we give a brief conclusion in Section 5.

2 Maximum Independent Set in Explicit Vertex Streams

We first introduce and show the hardness of a "chained index" problem, which we then use to show the hardness of approximating the size $\alpha(G)$ – and hence also for finding an approximate MIS.

2.1 Chained Index Communication Problem

We define a multi-party communication problem $\textsc{Chain}_k$, which allows us to prove new lower bounds on several streaming problems. The problem is closely related to pointer jumping and generalizes the classic two-party Index communication problem to more parties by "chaining" together multiple instances which have the same answer but are otherwise independent. Index is defined as follows:

▶ **Definition 2.** *In the two-party communication problem* Index*, Alice holds an n-bit string $X \in \{0,1\}^n$ and Bob holds an index $\sigma \in [n]$. Alice sends a single message to Bob who, upon receipt, outputs X_σ.*

It is well known that Alice essentially needs to send all n bits to Bob (see [18]):

▶ **Theorem 3.** *The randomized constant error communication complexity of* Index *is $\Omega(n)$.*

In $\textsc{Chain}_k$, each party (except the last) holds a binary vector that contains a special bit which is the answer to the instance. Each party (except the first) knows where the answer bit is located in the previous party's vector. Communication is one-way and private, with each player receiving a message from the previous player and then sending a message to the next player. Formally:

▶ **Definition 4.** *The k-party chained index problem $\textsc{Chain}_k$ consists of $(k-1)$ n-bit binary vectors $\{X^{(i)}\}_{i=1}^{k-1}$, along with corresponding indices $\{\sigma_i\}_{i=1}^{k-1}$ from the range $[n]$. We have the promise that the entries $\{X^{(i)}_{\sigma_i}\}_{i=1}^{k-1}$ are all equal to the desired answer bit $z \in \{0,1\}$. The input is initially allocated as follows:*

- *The first party P_1 knows $X^{(1)}$*
- *Each intermediate party P_p for $1 < p < k$ knows $X^{(p)}$ and σ_{p-1}*
- *The final party P_k knows just σ_{k-1}*

Communication proceeds as follows: P_1 sends a single message to P_2, then P_2 communicates to P_3, and so on, with each party sending exactly one message to its immediate successor. After all messages are sent, P_k must correctly output z, succeeding with probability at least $2/3$. If the promise condition is violated, any output is considered correct.

There is a trivial communication upper bound of $O(n)$ bits: for instance, simply have the penultimate party send $X^{(k-1)}$ to the final party who can then return $X^{(k-1)}_{\sigma_{k-1}}$.

We claim two bounds on the communication complexity of this problem.

▶ **Theorem 5.** *Any communication scheme $\mathcal{B}$ which solves $\textsc{Chain}_k$ must communicate at least $\Omega(\frac{n}{k^2})$ bits in total.*

This first bound is shown by reducing instances of another problem (conservative pointer jumping [8]) to instances of our problem.

▶ **Theorem 6.** *There is a constant $C > 0$ such that any communication scheme $\mathcal{B}$ which solves $\textsc{Chain}_k$ for $k \leq C\left(\frac{n}{\log n}\right)^{\frac{1}{4}}$ must communicate at least $\Omega(\frac{n}{k})$ bits in total.*

This second bound is shown by a lengthy and technical proof based on the structure of the pointer jumping bound given in [8]. Due to space restrictions we omit both proofs here – they can be found in the full paper.

In particular, for constant k, we have a tight bound on the communication complexity of the k-party chained index problem of $\Theta(n)$. We conjecture that a dependence on k is not necessary.

▶ **Conjecture 1.** *Any communication scheme for $\textsc{Chain}_k$ requires $\Omega(n)$ communication.*

2.2 MIS Hardness in Explicit Vertex Streams

We show a new lower bound for the vertex streaming space complexity of approximate MIS.

▶ **Theorem 1** (restated)**.** *Any algorithm for the explicit vertex stream model which finds a c-approximation to $\alpha(G)$ with probability at least $2/3$ requires $\Omega\left(\frac{n^2}{c^6}\right)$ space.*

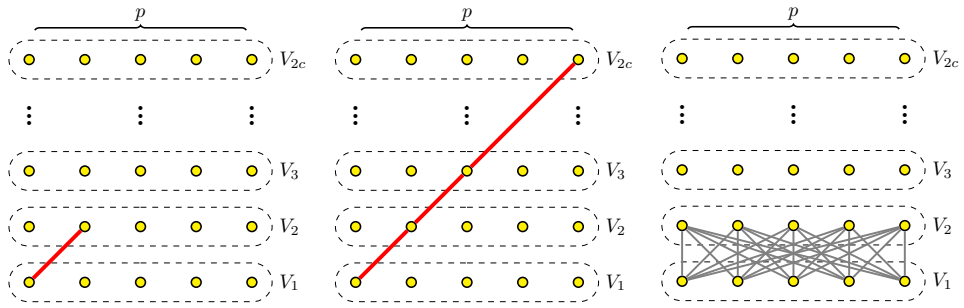
For ease of argument, we will actually prove an equivalent result for the problem of clique number approximation, and then note that the complement of the constructed graph can be used with the same arguments to prove Theorem 1. To see this equivalence, note that an MIS of a graph is a maximum clique in its complement.

▶ **Theorem 7.** *Any algorithm for the explicit vertex stream model which finds a c-approximation to the size of the largest clique $\omega(G)$ with probability at least $2/3$ requires $\Omega\left(\frac{n^2}{c^6}\right)$ space.*

The heart of our construction is to use an erasure code to encode a length $\Theta(\frac{n^2}{c^4})$ binary vector on $\Theta(\frac{n}{c})$ vertices, with each bit corresponding to the presence or absence of a clique of size $2c$. The use of the erasure code is to ensure that no pair of these cliques can share an edge. We can then chain together $2c$ such gadgets to encode an instance of $\textsc{Chain}_{2c}$ such that if the correct answer is 1, the resulting graph has an independent set of size $4c^2$, while if the correct answer is 0 the graph has no independent set larger than $4c - 1$. Any (one-sided) c-approximation algorithm could distinguish these two cases, which proves the result.

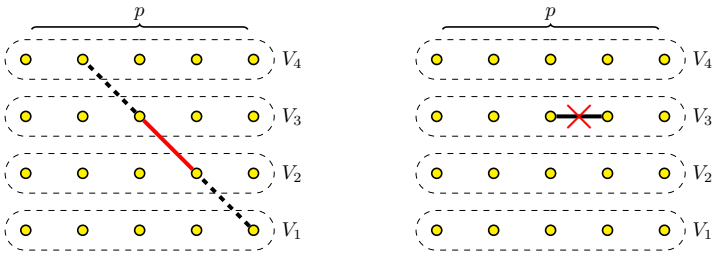
First we define our clique gadget.

▶ **Lemma 8.** *For any positive integers n and $c^2 < \frac{n}{8}$, there exists a graph on n vertices containing $\frac{n^2}{16c^2}$ edge-disjoint cliques of size $2c$ and no cliques of size larger than $2c$.*



(a) Select an edge. **(b)** Extend linearly to size $2c$. **(c)** Can make p^2 such cliques.

Figure 3 Clique gadget construction in Lemma 8.



(a) No edge belongs to two different lines (cliques).

(b) No edges within layers, so no cliques larger than $2c$.

Figure 4 Clique gadget proof sketch for Theorem 7.

Proof. We construct the sets from an erasure code with block size $2c$ and message size 2. Choose a prime p such that $\frac{n}{4c} \leq p \leq \frac{n}{2c}$ (which is guaranteed to exist). Now take $2c < p$ groups of vertices, each of size p. Label the groups V_i (for $i \in [2c]$) and label the items in each group V_i as v_j^i (for $j \in [p]$). Leftover vertices are added to the final graph as isolated vertices.

For each polynomial $\mathcal{P} \in \mathrm{GF}(p^2)$ we define $K_{\mathcal{P}}$ to be the clique over vertices $\{v^i_{\mathcal{P}(i)} | i \in [2c]\}$. This can be viewed as taking each of the p^2 possible edges between V_1 and V_2 and extending them "linearly" to the other layers (see Figure 3). In other words, the cliques correspond to non-horizontal lines in the affine plane of order p. Clearly $\mathcal{K} = \{K_{\mathcal{P}} | \mathcal{P} \in \mathrm{GF}(p^2)\}$ consists of $p^2 > \frac{n^2}{16c^2}$ cliques, each of size $2c$. We next show that they are pairwise edge-disjoint and that their union contains no larger cliques.

Each clique contains exactly one vertex from each group V_i, so for two cliques to share an edge there must be distinct polynomials $\mathcal{P}, \mathcal{Q} \in GF(p^2)$ that have the same value at two different points: $\mathcal{P}(i) = \mathcal{Q}(i)$ and $\mathcal{P}(j) = \mathcal{Q}(j)$ for $i \neq j$ – a contradiction. Finally, because no clique contains a pair of vertices from a single V_i, their union can contain no internal edges on any V_i. So any clique can contain at most 1 vertex from each V_i, giving a maximum size of $2c$. Hence, $\bigcup_{\mathcal{P} \in \mathrm{GF}(p^2)} K_{\mathcal{P}}$ is a graph with the required properties. ◀

Proof of Theorem 7. Suppose we have an algorithm $\mathcal{C}$ for explicit vertex streams which can, with probability at least $\frac{2}{3}$, produce a c-approximation to $\omega(G)$, the size of the largest clique. We will show that such an algorithm can be used to solve $\textsc{Chain}_{2c}$, by communicating its state $2c - 1$ times.

Fix an instance of $\textsc{Chain}_{2c}$ with vectors of length $b = \frac{n^2}{64c^4}$. Our lower bound in Theorem 6 implies that any algorithm that can solve this must send at least one message of size $\Omega\left(\frac{b}{c^2}\right) = \Omega\left(\frac{n^2}{c^6}\right)$ bits. Take n vertices and partition the nodes into $2c$ groups of size $\frac{n}{2c}$. Each group will be added to the stream by one of the parties.

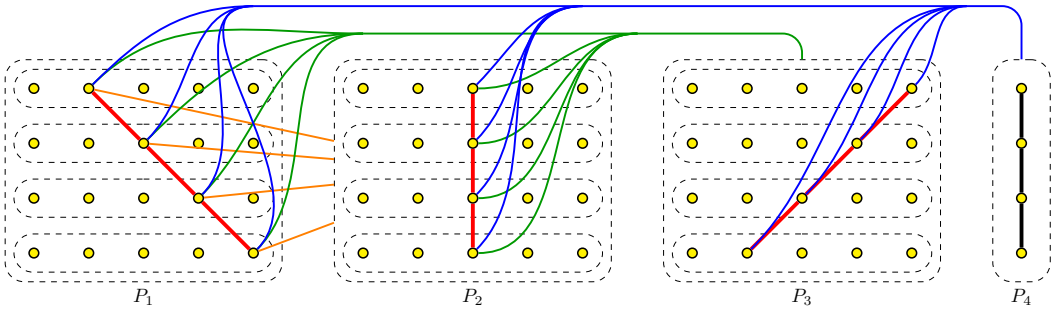


Figure 5 Example lower bound instance with 4 players for Theorem 7. Cliques corresponding to σ_1, σ_2, and σ_3 are shown in bold red – other cliques are omitted.

Intra-party edges. First, consider the group of nodes associated with party P_i. We will encode the bits of $X^{(i)}$ onto the internal edges of this group using the construction from Lemma 8. The size $\frac{n}{2c}$ sub-graph can fit b cliques of size $2c$. We include the edges of clique j if and only if $X^{(i)}_j = 1$. This is well defined as the cliques are edge-disjoint. Label the clique in party P_i corresponding to bit j of $X^{(i)}$ as $\mathcal{K}^i_j$. The final party P_{2c} has no associated vector. Instead, it constructs a single clique of size $2c$ and leaves the other vertices isolated.

Inter-party edges. We also need edges between the sub-graphs associated with different parties. Each party P_i will connect all its vertices to some of the vertices belonging to previous parties (P_j for $j < i$). These edges are considered to belong to party P_i, as they will be added by this party in the vertex streaming model. For each $j < i$ the party P_i connects every one of its vertices to all of $\mathcal{K}^j_{\sigma_j}$ (the clique corresponding to index σ_j). For this to happen, P_i must know all σ_j for $j < i$. This information is not known initially, but can be appended to the communications between players with only $O(c)$ overhead.

Now that we have our construction, we need to show bounds on $\omega(G)$ for the two cases. First, consider when every $X^{(i)}_{\sigma_i} = 1$. In this case we have each of the cliques $\mathcal{K}^i_{\sigma_i}$ present and connected together, forming a clique of size $4c^2$. Now consider the case when every $X^{(i)}_{\sigma_i} = 0$. Consider a clique $\mathcal{K}$ in the graph. If $\mathcal{K}$ contains multiple vertices belonging to one party P_i, then it can contain none from any subsequent party P_j ($j > i$), and at most one from each preceding party P_l ($l < i$). Hence the size of any clique is bounded by $4c - 1$. To see why this holds, observe that for any $i < 2c$, our clique can contain only one vertex from $\mathcal{K}^i_{\sigma_i}$, as none of its edges are included in the graph. So to contain multiple vertices from party P_i, the clique $\mathcal{K}$ must contain a vertex v from some $\mathcal{K}^i_j$ with $j \neq \sigma_i$. But then all subsequent parties P_j ($j > i$) will have no vertices adjacent to v, so cannot contribute anything to $\mathcal{K}$. So the best we can do is include one vertex from each $\mathcal{K}^i_{\sigma_i}$ and then $2c$ from party P_{2c} giving a clique of size $4c - 1$.

To complete the proof, observe that this gap in clique sizes can be distinguished by a c-approximation algorithm, and any streaming algorithm gives a communication protocol by having each party update the algorithm state with their information and then pass it to the next party. ◀

Interestingly, the same construction gives us hardness for approximating the chromatic number of a graph. This is notably not possible in the 2-party edge stream construction in [15], as the random graphs used as gadgets have large chromatic number w.h.p. (see [5]).

▶ **Corollary 9.** *Any explicit vertex streaming algorithm to find a c-approximation to $\chi(G)$ (the chromatic number), succeeding with probability at least 2/3 requires $\Omega\left(\frac{n^2}{c^6}\right)$ space.*

Proof. Consider the construction in the proof of Theorem 1. In the case of all $X_{\sigma_i}^{(i)} = 1$, the graph contains a clique of size $4c^2$, so it requires at least as many colours.

Conversely, in the case of every $X_{\sigma_i}^{(i)} = 0$, we can construct a $4c$-coloring of the graph. First color each of the nodes in each $\mathcal{K}_{\sigma_i}^i$ with the i^{th} color (this is allowed, as they have no internal edges). The remaining vertices in each party are then not adjacent to any uncolored vertices from other parties, so we simply need to be able to complete the coloring of each party in isolation with $2c$ new colors and we are finished. This is easily done, as each party's sub-graph is $2c$-partite by construction. ◀

3 Maximal Independent Set in Edge Streams

In this section, we consider streaming algorithms for the *maximal* independent set problem. Vertex streams (both explicit and implicit) are well-suited to the maximal independent set problem, since they allow the implementation of the Greedy algorithm for independent sets, which greedily adds every incoming vertex v to an initially empty independent set I if this is possible, i.e., if $I \cup \{v\}$ is an independent set. The algorithm only stores the computed independent set. This yields the following result:

▶ **Fact 1.** *The* Greedy *algorithm for independent sets is a one-pass $\tilde{O}(\alpha(G)) = \tilde{O}(n)$ space maximal independent set algorithm in vertex streams (both implicit and explicit).*

This raises the question of how well we can solve the maximal independent set problem in edge streams. We show that computing a maximal independent set in one pass in the edge-arrival model is not possible using sublinear space, i.e., space $\Omega(n^2)$ is required. This result is obtained through a reduction from the Index problem in two-party communication complexity. This proof is available in the full version of the paper.

▶ **Theorem 10.** *Every randomized constant error one-pass streaming algorithm in the edge-arrival model that computes a maximal independent set requires $\Omega(n^2)$ space.*

Since computing a maximal independent set with sublinear space is impossible in edge streams, we ask whether we can compute an *approximately maximal* independent set instead:

▶ **Definition 11** (Approximate Maximality)**.** *Let $G = (V, E)$ be an n-vertex graph, and let $I \subseteq V$ be an independent set. Then I is δ-maximal, if $|I \cup \Gamma_G[I]| \geq \delta n$.*

A δ-maximal independent set I covers a δ-fraction of the vertices, or, in other words, when removing I and its neighbors $\Gamma_G[I]$ from the graph, then at most $(1-\delta)n$ vertices are remaining. We will next show that establishing approximate maximality in edge streams requires strictly more space than computing a maximal independent set in vertex streams (i.e., $\omega(n)$ space), even if $\delta = \frac{24}{25}$. Regarding stronger approximate maximality, our lower bound yields that computing a $(1 - \frac{1}{n^\epsilon})$-maximal independent set requires space $\Omega(n^{2-o(1)})$, for every $\epsilon > 0$.

Central to our construction are *Ruzsa-Szemerédi graphs*, which have previously been used for the construction of streaming space lower bounds for maximum matching [12, 17, 4]:

▶ **Definition 12** (Ruzsa-Szemerédi graph)**.** *A bipartite graph G is an (r, s)*-Ruzsa-Szemerédi graph *if its edge set can be partitioned into r induced matchings each of size s.*

Recall that a matching $M \subseteq E$ in a graph $G = (V, E)$ is induced, if the edge set of the vertex-induced subgraph $G[V(M)]$ equals M, i.e., there are no other edges interconnecting $V(M)$ different from M.

Our lower bound for approximate maximality is obtained by a reduction from the two-party communication problem RS-Index, defined as follows:

▶ **Definition 13** (RS-Index). *Let H be an (r, s)-Ruzsa-Szemerédi graph with induced matchings $M_1, M_2, \ldots, M_r$. For each induced matching M_i, let $M_i' \subseteq M_i$ be a uniform random subset of size $s/2$ (we assume that s is even). The* RS-Index *problem is a one-way two-party communication problem, where H, and, in particular, $M_1, M_2, \ldots, M_r$ are known by both parties. In addition, Alice holds the graph $G = H[\cup_i M_i']$, and Bob holds a uniform random index $i \in \{1, 2, \ldots, r\}$. Alice sends a single message to Bob, who, upon receipt, outputs at least $C \cdot s$ edges of M_i', for an arbitrary small constant C.*

Observe that this problem is similar in spirit to Index: In Index, Bob needs to learn one uniform random bit, while in RS-Index, Bob needs to learn the presence of many edges of M_i'. A lower bound on the communication complexity of RS-Index is implicit in [12] [2]:

▶ **Theorem 14** ([12]). *The randomized constant error communication complexity of* RS-Index *is $\Omega(r \cdot s)$.*

Equipped with the RS-Index problem, we now give a reduction to approximate maximality from RS-Index, which yields our lower bound for streaming algorithms:

▶ **Lemma 15.** *Let r, s, n be integers such that there is an n-vertex (r, s)-Ruzsa-Szemerédi graph. Then, every randomized constant error one-pass streaming algorithm in the edge-arrival model that computes a $(1 - \frac{s}{6n})$-maximal independent set requires $\Omega(r \cdot s)$ space.*

Proof. Let H be an n-vertex (r, s)-Ruzsa-Szemerédi graph, and let G be Alice's input graph for the RS-Index problem derived from H. Let $M_1, M_2, \ldots, M_r$ denote the induced matchings in H, let $V_i = V(M_i)$, and let $M_i' \subseteq M_i$ denote the subset of edges of matching M_i that is included in G. Let i be Bob's input. Furthermore, let $\mathcal{A}$ be a constant error randomized one-pass streaming algorithm for the edge-arrival model that computes a $(1-\frac{s}{6N})$-maximal independent set on a graph on N vertices. We now show how $\mathcal{A}$ can be used to solve RS-Index:

Given G, let $\tilde{G}$ be the graph obtained from G, where every induced matching M_i' in G is replaced by edges $\tilde{M}_i' := M_i \setminus M_i'$ (observe that $E(G) \cup E(\tilde{G}) = E(H)$). Alice now constructs two disjoint copies G_1 and G_2 of $\tilde{G}$, runs algorithm $\mathcal{A}$ on $G_1 \dot{\cup} G_2$ (on an arbitrary ordering of their edges), and sends the memory state to Bob. Bob constructs the edge set F that connects every vertex $v_1 \in V(G_1) \setminus V_{i1}$ with every vertex $v_2 \in V(G_2) \setminus V_{i2}$, where V_{i1} and V_{i2} are the copies of the vertices V_i in graphs G_1 and G_2, respectively, and continues the execution of $\mathcal{A}$ on F. Let I be the independent set produced by algorithm $\mathcal{A}$.

Observe that the graph processed by algorithm $\mathcal{A}$ contains $N = 2n$ vertices. Since I is $(1 - \frac{s}{6N})$-maximal, we have $|V \setminus \Gamma[I]| \leq N - (1 - \frac{s}{6N})N = s/6$. This allows us to identify $\Omega(s)$ edges of M_i' as follows:

Let a, b be the incident vertices to an arbitrary edge of M_i', let a_1, b_1 be the copies of a, b in G_1, and let a_2, b_2 be the copies of a, b in G_2. Observe that a_1 and b_1 are not connected in G_1, and a_2 and b_2 are not connected in G_2. We now claim that if all vertices a_1, b_1, a_2, b_2

[2] In [12] a lower bound is given for the task of computing a maximum matching. Their hardness stems from the fact that it is hard to learn many edges of M_i' under the distribution described in the definition of RS-Index.

are covered by I, i.e., $\{a_1, b_1, a_2, b_2\} \subseteq \Gamma[I]$, then either $\{a_1, b_1\} \subseteq I$ or $\{a_2, b_2\} \subseteq I$ (or both). Indeed, suppose that this is not the case. Then there are vertices $x_1 \in \{a_1, b_1\}$ and $x_2 \in \{a_2, b_2\}$ with $x_1, x_2 \notin I$. Let $y_1 \in I$ be a vertex incident to x_1, and let $y_2 \in I$ be a vertex incident to x_2. By the construction of the input graph, $y_1 \in V(G_1) \setminus V_{i1}$, and $y_2 \in V(G_2) \setminus V_{i2}$. Observe, however, that the edge y_1y_2 was included by Bob, which implies that y_1, y_2 are not independent: a contradiction. Hence, either $\{a_1, b_1\} \subseteq I$ or $\{a_2, b_2\} \subseteq I$ (or both) hold. This implies that the algorithm identified that there is no edge between a_1, b_1, which in turn implies that we learned one edge of M'_i. Hence, for every pair of vertices a, b of M'_i, either at least one vertex among $\{a_1, b_1, a_2, b_2\}$ is not covered by I, or we learn one edge of M'_i. Since there are $s/2$ edges in M'_i, and at most $s/6$ vertices of the input graph are not covered by I, we learn at least $s/2 - s/6 = \Omega(s)$ edges of M'_i, which thus solves RS-Index. By Theorem 14, algorithm $\mathcal{A}$ therefore requires space $\Omega(r \cdot s)$. ◀

In [12] it is shown that there are n-vertex $(n^{\Theta(\frac{1}{\log\log n})}, (\frac{1}{4} - \epsilon)n)$ Ruzsa-Szemerédi graphs, for every $\epsilon > 0$, and in [2], it is shown that there are such graphs with $\Theta(n^{2-o(1)})$ edges such that each matching is of size $n^{1-o(1)}$. Combined with Lemma 15, we obtain:

▶ **Theorem 16.** *Every randomized constant error one-pass streaming algorithm that computes a $\frac{24}{25}$-maximal independent set requires space $n^{1+\Omega(\frac{1}{\log\log n})}$, and every such algorithm computing a $(1 - \frac{1}{n^\epsilon})$-maximal independent set requires space $\Omega(n^{2-o(1)})$, for every $\epsilon > 0$.*

Last, interestingly, if we allow an algorithm to perform multiple passes, then sublinear space algorithms can be obtained. Such algorithms are in fact immediately implied by the correlation clustering algorithms given in [1]. Their result yields the following theorem:

▶ **Theorem 17.** *There is a $O(\log\log n)$-pass streaming algorithm for maximal independent set that uses space $\tilde{O}(n)$.*

4 Maximum Independent Set in Geometric Intersection Graphs

We now present a collection of results around geometric intersection graphs, in one and two dimensions, given as explicit or implicit vertex streams. We consider intervals and squares.

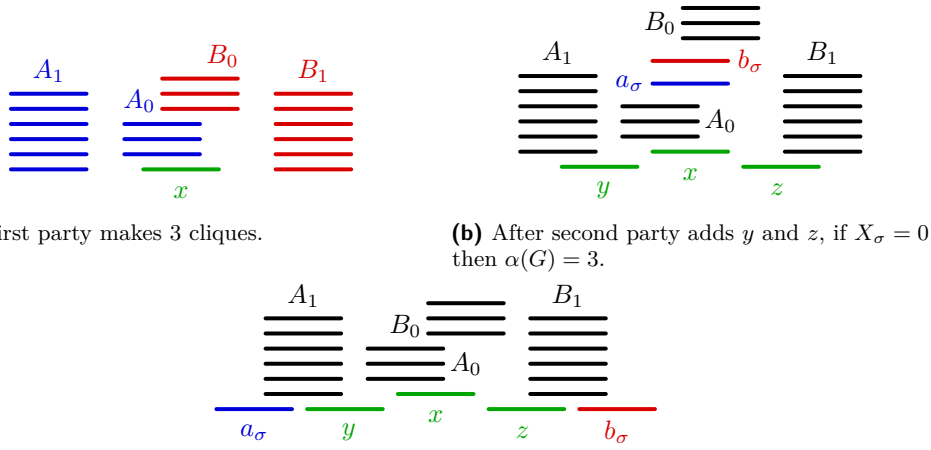
A geometric intersection graph is a graph where nodes correspond to geometric objects, and edges indicate whether or not a particular pair of objects intersect. These graphs can be described implicitly as the collection of geometric objects, or explicitly as a collection of vertices and edges under the promise that some geometric representation exists.

For implicit representations, we assume that intervals and squares are presented by their centers and their lengths. We assume that the center is a value in $[M]^d$ ($d = 1$ for intervals, and $d = 2$ for squares), and the length is in $[M]$, for some $M \in$ poly n.

4.1 Unit Interval Graphs: $\mathrm{d} = 1$

As discussed in Section 1, given a stream of unit intervals we can compute a $\frac{3}{2}$-approximation to MIS in $\tilde{O}(\alpha(G))$ space, and any better approximation requires $\Omega(n)$ space. A natural question is how this compares with the space complexity for an interval intersection graph given as an explicit vertex stream:

▶ **Theorem 18.** *Any algorithm with constant error probability that returns a $(\frac{5}{3} - \epsilon)$-approximation of $\alpha(G)$ for a unit interval intersection graph given as an explicit vertex stream requires $\Omega(n)$ space.*



(a) First party makes 3 cliques.

(b) After second party adds y and z, if $X_\sigma = 0$ then $\alpha(G) = 3$.

(c) Conversely, if $X_\sigma = 1$ then $\alpha(G) = 5$.

Figure 6 Interval representations for the construction in theorem 18. Horizontal positioning represents the location of the intervals in $\mathbb{R}$, vertical positioning is for clarity only.

Proof. We will show this bound by a reduction from the 2-party INDEX communication problem. Consider an instance of INDEX with bit vector $X \in \{0,1\}^n$ and index to be queried $\sigma \in [n]$. We will construct a $2n+3$ vertex graph as an explicit vertex stream.

Label the vertices x, y, z and a_i, b_i for $i \in [n]$. Split the a_i's into two sets based on the bit vector X: $A_1 = \{a_i\}_{X_i=1}$ and $A_0 = \{a_i\}_{X_i=0}$. Similarly let $B_1 = \{b_i\}_{X_i=1}$ and $B_0 = \{b_i\}_{X_i=0}$. Now the first party creates the following subgraph in the stream: a clique consisting of all the vertices in A_1, a second clique made from B_1, and a third clique containing $A_0 \cup B_0 \cup \{x\}$.

So far this represents a valid interval graph, which can be interpreted as three adjacent "stacks" of intervals. Now, the second player adds y with edges to every a_i except a_σ and then adds z with edges to every b_i except b_σ. This can still be viewed as a valid interval graph, but we now require some intervals from each stack to be "shifted" to overlap with the two new intervals.

In the case of $X_\sigma = 0$, the resulting graph has $\alpha(G) = 3$. Otherwise, $\alpha(G) = 5$. Hence, any algorithm giving a better than $\frac{5}{3}$-approximation factor could distinguish them and solve INDEX. ◀

This shows that MIS for interval graphs is strictly more difficult in explicit vertex streams than implicit ones.

4.2 Square Graphs: $\mathrm{d} = 2$

We obtain several improved bounds for the 2D case. Full details can be found in the extended paper, but we briefly summarise here.

Our first result for 2D is a 3-approximation algorithm for MIS on a unit square stream. This is a generalization of the algorithm of [7] for unit interval streams – we perform a decomposition of the plane into 2-by-3 strips, similar to their decomposition of the line into length 3 segments.

▶ **Theorem 19.** *There is a 3-approximation streaming algorithm for* MIS *on a stream of unit squares (implicit vertex stream) using $\tilde{O}(\alpha(G))$ space.*

As in [7] for unit intervals, this immediately leads to a sublinear space algorithm for estimating $\alpha(G)$ with only a $(1+\epsilon)$ factor loss in approximation factor, through a combination of counting distinct elements and clever sampling.

▶ **Corollary 20.** *We can $(3+\epsilon)$-approximate $\alpha(G)$ with constant probability in a stream of unit squares using $O(\epsilon^{-2}\log\epsilon^{-1}+\log n)$ space.*

One might speculate whether this decomposition approach could afford a better approximation factor based on some different partitioning of the plane. We give evidence for the negative, since any larger strip size results in the fixed-size sub-problems not being solvable exactly, as the following result shows.

▶ **Theorem 21.** *Given a stream of w-by-w squares contained in a $(2+\delta)w$-by-$(2+\delta)w$ region, achieving a $(\frac{3}{2}-\epsilon)$-approximation to $\alpha(G)$, with constant probability of success for any $\epsilon,\delta>0$ requires $\Omega(n)$ space.*

Our next result for two dimensions is a stronger lower bound for approximating $\alpha(G)$ of a stream of unit squares in an unrestricted region, based on a reduction from the chained index communication problem used in our main result in Section 2.

▶ **Theorem 22.** *Achieving a $(\frac{5}{2}-\epsilon)$-approximation of $\alpha(G)$, with constant probability of success, on a unit square stream requires $\Omega(n)$ space for any $\epsilon>0$.*

If we are allowed a combination of large and small balls, we can slightly improve the lower bound up to the maximum possible for a 3-party construction.

▶ **Theorem 23.** *Achieving a $(3-\epsilon)$-approximation of $\alpha(G)$, with constant probability of success, on a stream of squares or arbitrary side lengths requires $\Omega(n)$ space for any $\epsilon>0$.*

5 Conclusion

We have looked at the complexity of Maximal and Maximum Independent Set (and various relaxations and related problems) under three natural models of graph streams: edge-arrival, explicit vertex-arrival, and implicit vertex-arrival.

By making use of a new communication problem $\textsc{Chain}_k$, we showed that MIS is not significantly easier on explicit vertex streams than edge streams. However, the question of whether they have exactly the same complexity is left open. Improving the communication bound on $\textsc{Chain}_k$ to $\Omega(n)$, as we conjectured, would improve our MIS lower bound to $\Omega\left(\frac{n^2}{c^5}\right)$, but we do not know of any vertex stream upper bounds better than the $\tilde{O}\left(\frac{n^2}{c^2}\right)$ algorithm for general edge streams.

There are a number of other open questions that naturally follow from our study:

- Is there a multi-pass lower bound for *maximal* independent set in edge streams?
- Are there $o(\alpha(G))$ space algorithms for achieving constant factor approximations to $\alpha(G)$ for classes of geometric intersection graphs given as *explicit* vertex streams?
- Can we close the gap between the 3 and 5/2 factors of the upper and lower bounds for approximating MIS in a unit square stream?
- Is there an $O(\alpha(G))$ space constant factor approximation algorithm for MIS on streams of arbitrary sized squares?
- Can $\textsc{Chain}_k$ be used to form novel lower bounds for other kinds of problems?

References

1 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, pages 2237–2246. JMLR.org, 2015. URL: `http://dl.acm.org/citation.cfm?id=3045118.3045356`.

2 Noga Alon, Ankur Moitra, and Benny Sudakov. Nearly Complete Graphs Decomposable into Large Induced Matchings and Their Applications. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1079–1090, New York, NY, USA, 2012. ACM. doi:10.1145/2213977.2214074.

3 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear Algorithms for $(\Delta + 1)$ Vertex Coloring. In *SODA*, 2019.

4 Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum Matchings in Dynamic Graph Streams and the Simultaneous Communication Model. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1345–1364, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2884435.2884528.

5 Béla Bollobás. The chromatic number of random graphs. *Combinatorica*, 8(1):49–55, 1988.

6 Vladimir Braverman, Zaoxing Liu, Tejasvam Singh, N. V. Vinodchandran, and Lin F. Yang. New Bounds for the CLIQUE-GAP Problem Using Graph Decomposition Theory. *Algorithmica*, 80(2):652–667, February 2018. doi:10.1007/s00453-017-0277-5.

7 Sergio Cabello and Pablo Pérez-Lantero. Interval selection in the streaming model. *Theoretical Computer Science*, 702:77–96, 2017.

8 Amit Chakrabarti. Lower bounds for multi-player pointer jumping. In *Computational Complexity, 2007. CCC'07. Twenty-Second Annual IEEE Conference on*, pages 33–45. IEEE, 2007.

9 Graham Cormode, Jacques Dark, and Christian Konrad. Approximating the Caro-Wei Bound for Independent Sets in Graph Streams. In Jon Lee, Giovanni Rinaldi, and A. Ridha Mahjoub, editors, *Combinatorial Optimization*, pages 101–114, Cham, 2018. Springer International Publishing.

10 Graham Cormode, Jacques Dark, and Christian Konrad. Independent Sets in Vertex-Arrival Streams. *CoRR*, abs/1807.08331, 2018. arXiv:1807.08331.

11 Yuval Emek, Magnús M Halldórsson, and Adi Rosén. Space-constrained interval selection. *ACM Transactions on Algorithms (TALG)*, 12(4):51, 2016.

12 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485, 2012.

13 Bjarni V Halldórsson, Magnús M Halldórsson, Elena Losievskaja, and Mario Szegedy. Streaming algorithms for independent sets. In *International Colloquium on Automata, Languages, and Programming*, pages 641–652. Springer, 2010.

14 Magnús M. Halldórsson and Christian Konrad. Computing Large Independent Sets in a Single Round. *Distrib. Comput.*, 31(1):69–82, February 2018. doi:10.1007/s00446-017-0298-y.

15 Magnús M Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. Streaming and communication complexity of clique approximation. In *International Colloquium on Automata, Languages, and Programming*, pages 449–460. Springer, 2012.

16 R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

17 Christian Konrad. Maximum Matching in Turnstile Streams. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015*, pages 840–852, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

18 I. Kremer, N. Nisan, and D. Ron. On Randomized One-round Communication Complexity. *computational complexity*, 8(1):21–49, 1999. doi:10.1007/s000370050018.

19 Andrew McGregor. Graph Stream Algorithms: A Survey. *SIGMOD Rec.*, 43(1):9–20, May 2014. doi:10.1145/2627692.2627694.

20 David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.