Optimal Lower Bounds for Matching and Vertex Cover in Dynamic Graph Streams

Jacques Dark
Department of Computer Science, University of Warwick, Coventry, UK
j.dark@warwick.ac.uk

Christian Konrad
Department of Computer Science, University of Bristol, UK
http://people.cs.bris.ac.uk/~konrad/
christian.konrad@bristol.ac.uk

Abstract

In this paper, we give simple optimal lower bounds on the one-way two-party communication complexity of approximate Maximum Matching and Minimum Vertex Cover with deletions. In our model, Alice holds a set of edges and sends a single message to Bob. Bob holds a set of edge deletions, which form a subset of Alice's edges, and needs to report a large matching or a small vertex cover in the graph spanned by the edges that are not deleted. Our results imply optimal space lower bounds for insertion-deletion streaming algorithms for Maximum Matching and Minimum Vertex Cover.

Previously, Assadi et al. [SODA 2016] gave an optimal space lower bound for insertion-deletion streaming algorithms for Maximum Matching via the simultaneous model of communication. Our lower bound is simpler and stronger in several aspects: The lower bound of Assadi et al. only holds for algorithms that (1) are able to process streams that contain a triple exponential number of deletions in n, the number of vertices of the input graph; (2) are able to process multi-graphs; and (3) never output edges that do not exist in the input graph when the randomized algorithm errs. In contrast, our lower bound even holds for algorithms that (1) rely on short ($\mathrm{O}(n^2)$-length) input streams; (2) are only able to process simple graphs; and (3) may output non-existing edges when the algorithm errs.

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

Keywords and phrases Maximum matching, Minimum vertex cover, Dynamic graph streams, Communication complexity, Lower bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2020.30

Funding *Jacques Dark*: Jacques Dark was supported by an EMEA Microsoft Research studentship and European Research Council grant ERC-2014-CoG 647557.

Acknowledgements The authors are grateful for the numerous excellent comments and suggestions from an anonymous reviewer.

1 Introduction

Streaming algorithms for processing massive graphs have been studied for two decades [16]. In the most traditional setting, the *insertion-only model*, an algorithm receives a sequence of the edges of the input graph in arbitrary order, and the objective is to solve a graph problem using as little space as possible. The insertion-only model has received significant attention, and many problems, such as matchings (e.g. [26, 13, 21, 27, 18, 24, 31, 12]), independent sets (e.g. [15, 14, 9, 10]), and subgraph counting (e.g. [20, 11, 7]), have since been studied in this model. See [29] for an excellent survey.

35th Computational Complexity Conference (CCC 2020).
Editor: Shubhangi Saraf; Article No. 30; pp. 30:1–30:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In 2012, Ahn et al. [1] introduced the first techniques for addressing *insertion-deletion* graph streams, where the input stream consists of a sequence of edge insertions and deletions. They showed that many problems, such as Connectivity and Bipartiteness, can be solved using the same amount of space as in insertion-only streams up to poly-logarithmic factors. Various other works subsequently gave results of a similar flavor and presented insertion-deletion streaming algorithms with similar space complexity as their insertion-only counterparts for problems including Spectral Sparsification [22] and $\Delta + 1$-coloring [3]. Konrad [23] and Assadi et al. [5] were the first to give a separation result between the insertion-only graph stream model and the insertion-deletion graph stream model: While it is known that a 2-approximation to Maximum Matching can be computed using space $\mathrm{O}(n \log n)$ in insertion-only streams, Konrad showed that space $\Omega(n^{\frac{3}{2}-4\epsilon})$ is required for an n^{ϵ}-approximation in insertion-deletion streams, and Assadi et al. gave a lower bound of $n^{2-3\epsilon-o(1)}$ for such an approximation. Assadi et al. also presented an $\tilde{\mathrm{O}}(n^{2-3\epsilon})$ space algorithm that matches their lower bound up to lower order terms, which establishes that their lower bound is optimal (a different algorithm that matches this lower bound is given by Chitnis et al. [8]).

Both Konrad and Assadi et al. exploit an elegant connection between insertion-deletion streaming algorithms and linear sketches. Ai et al. [2], building on the work of Yi et al. [28], showed that insertion-deletion graph streaming algorithms can be characterized as algorithms that essentially solely rely on the computation of linear sketches of the input stream. A consequence of this result is that space lower bounds for insertion-deletion streaming algorithms can also be proved in the *simultaneous model of communication*, since linear sketches can be implemented in this model. This provides an alternative to the more common approach of proving streaming lower bounds in the one-way model of communication. In particular, the lower bounds by Konrad and Assadi et al. are proved in the simultaneous model of communication.

From a technical perspective, this model has various attractive features, however, it comes with a major disadvantage: The characterization of Ai et al. only holds for insertion-deletion streaming algorithms that (1) are able to process "very long" input streams, i.e., input streams of triple exponential length in n, the number of vertices of the input graph, and (2) are able to process multi-graphs. In particular, this characterization does not hold for insertion-deletion streaming algorithms that rely on the assumption that input streams are short and the graph described by the input stream is always simple. Consequently, the lower bounds of Konrad and Assadi et al. do not hold for such algorithms.

Our Results. In this work, we prove an optimal space lower bound for Maximum Matching in insertion-deletion streams via the one-way two-party model of communication. Our lower bound construction yields insertion-deletion streams of length $\mathrm{O}(n^2)$ and does not involve multi-edges. Our lower bound therefore also holds for streaming algorithms that are designed for short input streams and simple graphs for which the characterization by Ai et al. does not hold. Furthermore, the optimal lower bound by Assadi et al. [5] only holds for streaming algorithms that never output non-existing edges when the (randomized) algorithms fail. We do not require this restriction.

Our lower bound method is simple and more widely applicable. Using the same method, we also give an optimal lower bound for Minimum Vertex Cover, showing that computing a n^{ϵ}-approximation requires $\Omega(n^{2-2\epsilon})$ space. Assadi and Khanna mention in [4] that the $n^{2-3\epsilon-o(1)}$ space lower bound for Maximum Matching given in [5] also applies to Minimum Vertex Cover. Our lower bound therefore improves on this result by a factor of $n^{\epsilon+o(1)}$. Furthermore, we show that our lower bound is optimal up to a factor of $\log n$: We give a very simple deterministic insertion-deletion streaming algorithm for Minimum Vertex Cover that uses space $\mathrm{O}(n^{2-2\epsilon} \log n)$.

While the main application of our lower bounds in the one-way two-party communication model are lower bounds for insertion-deletion graph streaming algorithms, we believe that our lower bounds are of independent interest. Indeed, the one-way two-party communication complexity of Maximum Matching without deletions has been addressed in [13], and our result can therefore also be understood as a generalization of their model to incorporate deletions.

The Simultaneous Model of Communication. The lower bounds by Konrad [23] and Assadi et al. [5] are proved in the simultaneous model of communication. In this model, a typically large number of parties k hold not necessarily disjoint subsets of the edges of the input graph. Each party P_i sends a message M_i to a referee, who then outputs the result of the protocol. The connection between insertion-deletion streaming algorithms and linear sketches by Ai et al. [2] then implies that a lower bound on the size of any message M_i yields a lower bound on the space requirements of any insertion-deletion streaming algorithm.

In the lower bound of Assadi et al. [5] for Maximum Matching, each party P_i holds the edges E_i of a dense subgraph, which itself constitutes a *Ruzsa-Szemerédi graph*, i.e., a graph whose edge set can be partitioned into large disjoint induced matchings. All previous streaming lower bounds for approximate Maximum Matching rely on realizations of Ruzsa-Szemerédi graphs [13, 23, 5]. Their construction is so that only a single induced matching of every party P_i is useful for the construction of a global large matching. Due to symmetry of the construction, the parties are unable to identify the important induced matching and therefore need to send large messages that contain information about most of the induced matchings to the referee for them to be able to compute a large global matching. Interestingly, none of the parties hold edge deletions in their construction.

The One-way Model of Communication. In this paper, we give a lower bound in the one-way two-party model of communication. In this model, Alice holds a set of edges E of the input graph and sends a message M to Bob. Bob holds a set of edge deletions $D \subseteq E$ and outputs a large matching in the graph spanned by the edges $E \setminus D$. A standard reduction shows that a lower bound on the size of message M also constitutes a lower bound on the space requirements of an insertion-deletion streaming algorithm. The two models are illustrated in Figure 1.

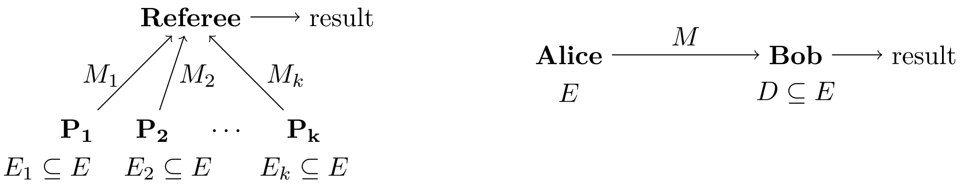


Figure 1 The simultaneous (left) and the one-way two-party (right) models of communication.

Our Techniques. To prove our lower bound, we identify that an insertion-deletion streaming algorithm for Maximum Matching or Minimum Vertex Cover can be used to obtain a one-way two-party communication protocol for a two-dimensional variant of the well-known Augmented Index problem that we denote by Augmented Bi-Index, or BInd in short. In an instance of BInd, Alice holds an n-by-n binary matrix $A \in \{0,1\}^{n \times n}$. Bob is given a position $(x, y) \in [n-k]^2$ and needs to output the bit $A_{x,y}$. Besides (x, y), he also knows the k-by-k submatrix of A with upper left corner at position (x, y), however with the bit at position (x, y) missing – we will denote this k-by-k submatrix with (x, y) missing by $A_{S(x,y)}$. We show that this problem has a one-way communication complexity of $\Omega((n-k)^2)$ by giving a reduction from the Augmented Index problem.

To obtain a lower bound for Maximum Matching, we show that Alice and Bob can construct a protocol for BInd given an insertion-deletion streaming algorithm for Maximum Matching. In our reduction, we will consider instances with $k = n - \Theta(n^{1-\epsilon})$, for some $\epsilon > 0$. Consider the following attempt: Suppose that the input matrix A is a uniform random binary matrix and that $A_{x,y} = 1$ (we will get rid of these assumptions later). Alice and Bob interpret the matrix A as the incidence matrix of a bipartite graph G. Bob interprets the "1" entries in the submatrix $A_{S(x,y)}$ outside the diagonal, i.e., all "1" entries except those in positions $\{(x+j, y+j) \ : \ 0 \le j < k\}$, as edge deletions F. The graph $G-F$ has a large matching: Since the diagonal of $A_{S(x,y)}$ is not deleted, and each entry in the diagonal is 1 with probability $1/2$, we expect that half of all potential edges in the diagonal of $S(x, y)$ are contained in $G-F$ and thus form a matching of size $\Theta(k) = \Theta(n - n^{1-\epsilon})$. An n^{ϵ}-approximation algorithm for Maximum Matching would therefore report $\Omega(n^{1-\epsilon})$ of these edges. Suppose that the algorithm reported $\Omega(n^{1-\epsilon})$ uniform random edges from the diagonal in $A_{S(x,y)}$ (we will also get rid of this assumption). Then, by repeating this scheme $\Theta(n^{\epsilon})$ times in parallel, with large constant probability the edge corresponding to $A_{x,y}$ is reported at least once, which allows us to solve BInd. This reduction yields an optimal $\Omega(n^{2-3\epsilon})$ space lower bound for insertion-deletion streaming algorithm for Maximum Matching, since $\Theta(n^{\epsilon})$ parallel executions are used to solve a problem that has a lower bound of $\Omega((n-k)^2) = \Omega(n^{2-2\epsilon})$.

In the description above, we assumed that (1) A is a uniform random binary matrix; (2) $A_{x,y} = 1$; and (3) the algorithm outputs uniform random positions from the diagonal of $A_{S(x,y)}$. To eliminate (1) and (2), Alice and Bob first sample a uniform random binary matrix $X \in \{0,1\}^{n \times n}$ from public randomness and consider the matrix obtained by computing the entry-wise XOR between A and X, i.e., matrix $A \oplus X$, instead. Observe that $A \oplus X$ is a uniform random binary matrix (independently of A), and with probability $\frac{1}{2}$, property (2), i.e., $(A \oplus X)_{x,y} = 1$, holds. Regarding assumption (3), besides computing the XOR $A \oplus X$, Alice and Bob also sample two random permutations $\sigma_1, \sigma_2 : [n] \to [n]$ from public randomness. Alice and Bob permute the rows and columns of $A \oplus X$ using σ_1 and σ_2, respectively. Then, no matter which elements from the permuted relevant diagonal of $A \oplus X$ are reported by the algorithm, due to the random permutations, these elements could have originated from any other position in this diagonal. This in turn makes every element along the diagonal equally likely to be reported, including the position (x, y) (in the unpermuted) matrix that we are interested in.

Our reduction for Minimum Vertex Cover is similar but simpler. We show that only a constant number of parallel executions of an insertion-deletion streaming are required.

Further Related Work. Hosseini et al. [17] were able to improve on the "triple exponential length" requirement of the input streams for a characterization of insertion-deletion streaming algorithms in terms of linear sketches by Li et al. [28] and Ai et al. [2]. They showed that in the case of XOR-streams and 0/1-output functions, input streams of length $O(n^2)$ are enough.

Very recently, Kallaugher and Price [19] showed that if either the stream length or the maximum value of the stream (e.g. the maximum multiplicity of an edge in a graph stream) are substantially restricted, then the characterization of turnstile streams as linear sketches cannot hold. For these situations they discuss problems where linear sketching is exponentially harder than turnstile streaming.

Besides the Maximum Matching problem, the only other separation result between the insertion-only and the insertion-deletion graph stream models that we are aware of is a recent result by Konrad [25], who showed that approximating large stars is significantly harder in insertion-deletion streams.

Outline. We give a lower bound on the communication complexity of Augmented Bi-Index in Section 2. Then, in Section 3, we show that a one-way two-party communication protocol for Maximum Matching can be used to solve Augmented Bi-Index, which yield an optimal space lower bound for Maximum Matching in insertion-deletion streams. We conclude with a similar reduction for Minimum Vertex Cover in Section 4, which also implies an optimal space lower bound for Minimum Vertex Cover in insertion-deletion streams.

2 Augmented Bi-Index

In this section, we define the one-way two-party communication problem Augmented Bi-Index and prove a lower bound on its communication complexity.

▶ **Problem 1** (Augmented Bi-Index). *In an instance of* Augmented Bi-Index $\mathsf{BInd}_\delta^{n,k}$ *we have two players denoted Alice and Bob. Alice holds a binary matrix $A \in \{0,1\}^{n \times n}$. Bob holds indices $x, y \in [n-k]$ and the incomplete*[1] *binary matrix $A_{S(x,y)}$ where*

$$S(x,y) = \{(i,j) \in [n]^2 \,|\, (x \leq i < x+k) \text{ and } (y \leq j < y+k)\} \setminus \{(x,y)\} \ .$$

Alice sends a single message M to Bob who must output $A_{x,y}$ with probability at least $1-\delta$.

Our lower bound proof consists of a reduction from the well-known Augmented Index problem, which is known to have large communication complexity.

▶ **Problem 2** (Augmented Index). *In an instance of* Augmented Index Ind_δ^n *we have two players denoted Alice and Bob. Alice holds a binary vector $V \in \{0,1\}^n$. Bob holds an index $\ell \in [n]$ and the vector suffix $V_{>\ell} = (V_{\ell+1}, V_{\ell+2}, \cdots, V_n)$. Alice sends a single message M to Bob who must output V_ℓ with probability at least $1-\delta$.*

As a consequence of Lemma 13 in [30], we can see that this problem has linear communication complexity (see also Lemma 2 in [6] for a more direct proof technique).

▶ **Theorem 3** (e.g. [30]). *For $\delta < 1/3$, any randomised one-way communication protocol which solves Ind_δ^n must communicate $\Omega(n)$ bits.*

We are now ready to prove our lower bound for Augmented Bi-Index.

▶ **Theorem 4.** *For $\delta < 1/3$, any randomised one-way communication protocol which solves $\mathsf{BInd}_\delta^{n,k}$ must communicate $\Omega((n-k)^2)$ bits.*

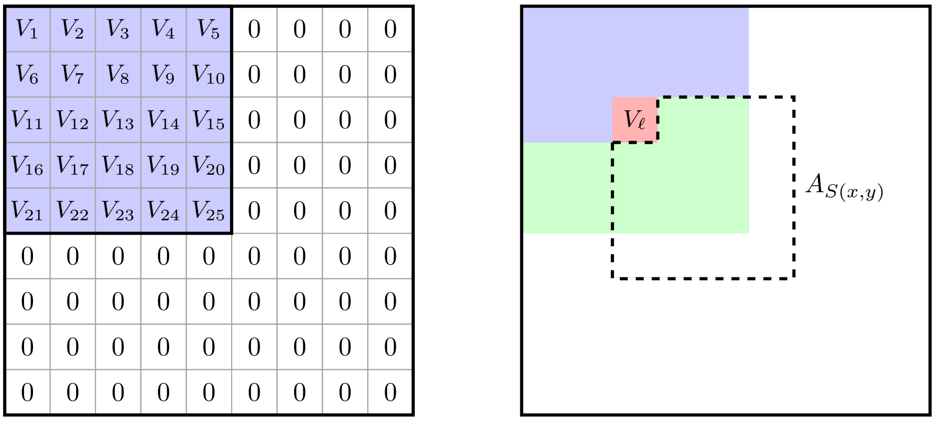
Proof. Let $\mathcal{P}$ be a communication protocol for $\mathsf{BInd}_\delta^{n,k}$ that uses messages of length at most $S(n,k)$ bits. We will show how $\mathcal{P}$ can be used to solve $\mathsf{Ind}_\delta^{(n-k)^2}$ with the same message size.

Let V, ℓ be any instance of $\mathsf{Ind}_\delta^{(n-k)^2}$. Alice builds the matrix $A \in \{0,1\}^{n \times n}$ by placing the bits of V in lexicographical order in the top-left $(n-k)$-by-$(n-k)$ region:

$$A_{i,j} = \begin{cases} V_{j+(n-k)(i-1)} & \text{for } i,j \in [n-k] \\ 0 & \text{otherwise} \end{cases} .$$

This packing is illustrated in Figure 2(a).

[1] We use A_S to refer to the collection of entries indexed by the set S, so $A_S = (A_{i,j})_{(i,j)\in S}$.



V_1	V_2	V_3	V_4	V_5	0	0	0	0
V_6	V_7	V_8	V_9	V_{10}	0	0	0	0
V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	0	0	0	0
V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	0	0	0	0
V_{21}	V_{22}	V_{23}	V_{24}	V_{25}	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(a) Example packing of the bits of V into matrix A with $n = 9$ and $k = 4$.

(b) Bob can construct the area $A_{S(x,y)}$ given $V_{>\ell}$, which is part of his input.

Figure 2 The construction of A and $A_{S(x,y)}$ in Theorem 4.

Alice runs protocol $\mathcal{P}$ on A and sends the resulting message M to Bob. Now, Bob has the message M, the index $\ell \in [(n-k)^2]$ and the suffix $V_{>\ell}$. Let $x, y \in [n-k]$ be the unique pair of integers such that $\ell = y + (n-k)(x-1)$. Observe that $A_{x,y} = V_\ell$.

For Bob to be able to complete protocol $\mathcal{P}$ he needs to provide $A_{S(x,y)}$. Because of the way we packed the entries of V onto A, the overlap between V and $A_{S(x,y)}$ is a subset of the entries of $V_{>\ell}$ (see Figure 2(b) for an illustration). Therefore Bob can complete the protocol and determine $A_{x,y} = V_\ell$ with probability at least $1-\delta$. By Theorem 3, it must be that $S(n,k) = \Omega((n-k)^2)$. ◀

3 Maximum Matching

Let $\mathbf{A}$ be a C-approximation insertion-deletion streaming algorithm for Maximum Matching that errs with probability at most 1/10. We will now show that $\mathbf{A}$ can be used to solve $\mathsf{BInd}_\delta^{n,k}$.

3.1 Reduction

Let $A \in \{0,1\}^{n \times n}, x \in [n-k]$ and $y \in [n-k]$ be an instance of $\mathsf{BInd}_\delta^{n,k}$. Alice and Bob first sample a uniform random binary matrix $X \in \{0,1\}^{n \times n}$ and random permutations $\sigma_1, \sigma_2 : [n] \to [n]$ from public randomness. Alice then computes matrix A' which is obtained by first computing the entry-wise XOR of A and X, denoted by $A \oplus X$, and then by permuting the rows and columns of the resulting matrix by σ_1 and σ_2, respectively. Next, Alice interprets A' as the incidence matrix of a bipartite graph $G(A')$. Alice runs algorithm $\mathbf{A}$ on a random ordering of the edges of $G(A')$ and sends the resulting memory state to Bob.

Next, Bob also computes the entry-wise XOR between the part of the matrix A that he knows about, $A_{S(x,y)}$, and X, followed by applying the permutations σ_1 and σ_2. In doing so, Bob knows the matrix entries of A' at positions $(\sigma_1(i), \sigma_2(j))$ for every $(i,j) \in S(x,y)$. He can therefore compute the subset E_S of the edges of $G(A')$ with

$$E_S = \{(\sigma_1(i), \sigma_2(j)) \in [n]^2 \mid (i,j) \in S(x,y) \text{ and } A'(\sigma_1(i), \sigma_2(j)) = 1\} .$$

Furthermore, let $E_{diag} \subseteq E_S$ be the set of edges $(\sigma_1(i), \sigma_2(j))$ so that (i,j) lies on the same diagonal in A as (x,y), or, in other words, there exists an integer $1 \le q \le k-1$ such that $(x+q, y+q) = (i,j)$. Then, let $E_{del} = E_S \setminus E_{diag}$. Bob continues the execution of algorithm $\mathbf{A}$, as follows: for every edge $e \in E_{del}$, Bob introduces an edge deletion of e, in random order.

Let M' be the matching returned by $\mathbf{A}$. From M' Bob computes the matching M as follows: If $|M' \le 0.99\frac{k}{2C}|$ then Bob sets $M = \varnothing$. Otherwise, Bob sets M to be a uniform random subset of M' of size exactly $0.99\frac{k}{2C}$.

Parallel Executions. Alice and Bob execute the previous process $\ell = 100 \cdot C$ times in parallel. Let M^i, X^i, σ_1^i and σ_2^i be M, X, σ_1 and σ_2 that are used in run i, respectively. Let Q_i be the indicator random variable that is 1 iff M^i contains the edge $(\sigma_1^i(x), \sigma_2^i(y))$. We also define $p = \sum_i Q_i$ to be the total number of times the edges $(\sigma_1^i(x), \sigma_2^i(y))$ are reported. Whenever the edge $(\sigma_1^i(x), \sigma_2^i(y))$ is reported, we interpret this to be a claim that $A_{x,y} = \neg X^i_{x,y}$. So depending on the value of $X^i_{x,y}$, this acts as a claim that $A_{x,y} = 0$ or $A_{x,y} = 1$. We define $p_0 = \sum_{i:Q_i=1} X^i_{x,y}$ (which counts how often $A_{x,y} = 0$ was claimed) and let $p_1 = p - p_0$ (the number of times $A_{x,y} = 1$ was claimed). Bob outputs 1 as his estimator for $A_{x,y}$ if $p_1 \ge p_0$ and 0 otherwise.

3.2 Analysis

Let G be the bipartite graph with incidence matrix $A \oplus X$, and let

$$F = \{(i,j) \in S(x,y) \mid (A \oplus X)_{i,j} = 1 \text{ and } \nexists\, q \text{ s.t. } (i,j) = (x+q, y+q)\} \ .$$

Then the graph $G - F$ is isomorphic to the graph $G(A') - E_{del}$. In particular, $G(A') - E_{del}$ is obtained from $G - F$ by relabeling the vertex sets of the two bipartitions using the permutations σ_1 and σ_2.

We will first bound the maximum matching size in $G(A') - E_{del}$. To this end, we will bound the maximum matching size in $G - F$, which is easier to do:

▶ **Lemma 5.** *With probability $1 - \frac{1}{k^{10}}$, the graph $G(A') - E_{del}$ is such that:*

$$0.99\frac{k}{2} \le \mu(G(A') - E_{del}) \le 1.01\frac{k}{2} + 2(n-k) \ ,$$

where $\mu(G)$ denotes the matching number *of G, i.e., the size of a maximum matching.*

Proof. We will consider the graph $G - F$ instead, since it is isomorphic to $G(A') - E_{del}$ and has the same maximum matching size.

First, observe that G is a random bipartite graph where every edge is included with probability $\frac{1}{2}$. Let U and V denote the bipartitions in G, and consider the subsets $U' = [x, x+k)$ and $V' = [y, y+k)$. Observe that in the vertex induced subgraph $G[U' \cup V']$ all edges are deleted in F except those that connect the vertices $x+i$ and $y+i$, for every $0 \le i \le k-1$. By a Chernoff bound, the number of edges and thus the maximum matching size in $G[U' \cup V']$ is bounded by:

$$0.99 \cdot \frac{k}{2} \le \mu(G[U' \cup V']) \le 1.01 \cdot \frac{k}{2} \ ,$$

with probability $1 - \frac{1}{k^{10}}$.

Observe that, with probability $1 - \frac{1}{k^{10}}$, the neighborhood $\Gamma(U')$ is such that

$$0.99 \cdot \frac{k}{2} \leq |\Gamma(U')| \leq 1.01 \cdot \frac{k}{2} + (n-k) \ .$$

The set U' can therefore be matched to at most $1.01 \cdot \frac{k}{2} + (n-k)$ vertices in V. We thus obtain

$$\mu(G - F) \leq 1.01 \cdot \frac{k}{2} + 2(n-k) \ ,$$

since we may also be able to match all $n-k$ vertices of $U \setminus U'$. ◀

▶ **Lemma 6.** *Suppose that $M_i \neq \varnothing$. Then:*

$$\frac{0.99}{2C} - \frac{2(n-k)}{k} \quad \leq \quad \mathbb{P}\left[Q_i = 1\right] \leq \frac{0.99}{2C} \ .$$

Proof. First, by construction of our reduction, since $M_i \neq \varnothing$ we have $|M_i| = 0.99\frac{k}{2C}$. Let

$$U_i' = \sigma_1^i([x, x+k)) \text{ and } V_i' = \sigma_2^i([y, y+k)) \ .$$

Let $\tilde{M_i}$ be the set of edges of M_i connecting vertices in U_i' to V_i'. Observe that there are $2(n-k)$ vertices in the graph outside the set $U_i' \cup V_i'$. We thus have

$$|M_i| - 2(n-k) \leq |\tilde{M_i}| \leq |M_i| \ .$$

Next, since the permutations σ_1^i, σ_2^i are chosen uniformly at random, any edge of $\tilde{M_i}$ may have originated from any of the diagonal entries in $A_{S(x,y)}$. Hence, $\tilde{M_i}$ claims the bits of at least $|M_i| - 2(n-k)$ and at most $|M_i|$ uniform random positions in the diagonal of $A_{S(x,y)}$. Every entry in the diagonal of $A_{S(x,y)}$ is thus claimed with the same probability. Since the diagonal of $A_{S(x,y)}$ is of length k, this probability is at least

$$\frac{|M_i| - 2(n-k)}{k} = \frac{0.99\frac{k}{2C} - 2(n-k)}{k} = \frac{0.99}{2C} - \frac{2(n-k)}{k} \ ,$$

and at most

$$\frac{|M_i|}{k} = \frac{0.99\frac{k}{2C}}{k} = \frac{0.99}{2C} \ .$$ ◀

▶ **Theorem 7.** *Let $\mathbf{A}$ be a n^ϵ-approximation insertion-deletion streaming algorithm for* Maximum Matching *that errs with probability at most $1/10$ and uses space s. Then there exists a communication protocol for* $\mathsf{BInd}_{0.05}^{n,n-\frac{1}{40}n^{1-\epsilon}}$ *that communication $\mathrm{O}(n^\epsilon \cdot s)$ bits.*

Proof. Let $C = n^\epsilon$ and let $k = n - \frac{1}{40}n^{1-\epsilon}$. First, by Lemma 5, with probability $1 - \frac{1}{k^{10}}$, the graph $G(A') - E_{del}$ contains a matching of size at least $0.99k/2$. By a union bound, the probability that this graph is of at least this size in each of the ℓ iterations is at least $1 - \frac{\ell}{k^{10}}$. Suppose from now on that this event happens.

Let ℓ_1 be the number of times the algorithm $\mathbf{A}$ succeeds, and let ℓ_0 be the number of times $\mathbf{A}$ errs. Then, $\ell = \ell_0 + \ell_1$. Whenever $\mathbf{A}$ succeeds, since $\mathbf{A}$ is a C-approximation algorithm, the matching M_i' is of size $0.99\frac{k}{2C}$, which further implies that M_i is of size exactly $0.99\frac{k}{2C}$. Since the algorithm must return a correct matching, every time we have a claim (i.e. $Q_i = 1$), the claimed bit value must be correct. Thus, by Lemma 6, we get a correct claim on $A_{x,y}$ with probability at least

$$\frac{0.99}{2C} - \frac{2(n-k)}{k} \quad = \quad \frac{0.99}{2n^\epsilon} - \frac{2(\frac{1}{40}n^{1-\epsilon})}{n - \frac{1}{40}n^{1-\epsilon}} \geq \frac{0.99}{2n^\epsilon} - \frac{\frac{1}{40}n^{1-\epsilon}}{n} = \frac{0.99}{2n^\epsilon} - \frac{1}{40n^\epsilon} \geq \frac{2}{5n^\epsilon} \ ,$$

where we used the inequality $\frac{2x}{y-x} \geq \frac{x}{y}$, which holds for every $y > x$. We thus expect to see the correct bit claimed at least $\ell_1 \cdot \frac{2}{5n^\epsilon}$ times in total. On the other hand, incorrect claims of the bit value can only occur when the algorithm errs. In the worst case, $\mathbf{A}$ will make as many false claims as possible - so we assume the algorithm never results in $M_i = \emptyset$ when it errs. Lemma 6 also allows us to bound the probability of an incorrect claim for this bad algorithm by $\frac{0.99}{2n^\epsilon}$. We thus expect to see the wrong bit value claimed at most $\ell_0 \cdot \frac{0.99}{2C} \leq \frac{\ell_0}{2n^\epsilon}$ times.

Recall that $\ell = 100n^\epsilon$. Then, by standard concentration bounds, the probability that $\ell_0 \geq 2 \cdot \frac{\ell}{10}$ is at most $\frac{1}{100}$ (recall that the error probability of $\mathbf{A}$ is at most $\frac{1}{10}$). Suppose now that $\ell_0 \leq \frac{1}{5}\ell$ holds, which also implies that $\ell_1 \geq \frac{4}{5}\ell$. We thus expect to learn the correct bit at least

$$\frac{4}{5}100n^\epsilon \cdot \frac{2}{5n^\epsilon} = 32$$

times, and using a Chernoff bound, it can be seen that the probability that we learn the correct bit less than 21 times is at most 0.02. Similarly, we expect to learn the incorrect bit at most

$$\frac{1}{5}100n^\epsilon \cdot \frac{1}{2n^\epsilon} = 10$$

times, and by a Chernoff bound, it can be seen that the probability that we learn the incorrect bit at least 20 times is at most 0.01. Our algorithm therefore succeeds if all these events happen. Taking a union bound over all failure probabilities that occurred in this proof, we see that our algorithm succeeds with probability

$$1 - \frac{100n^\epsilon}{k^{10}} - 0.01 - 0.02 - 0.01 \geq 0.95 \ . \qquad ◀$$

Since by Theorem 4, $\mathsf{BInd}_{0.05}^{n,n-\frac{1}{40}n^{1-\epsilon}}$ has randomized one-way communication complexity $\Omega(n^{2-2\epsilon})$, by Theorem 7 we obtain our main result of this section:

▶ **Corollary 8.** *Every insertion-deletion n^ϵ-approximation streaming algorithm for* Maximum Matching *that errs with probability at most $\frac{1}{10}$ requires space $\Omega(n^{2-3\epsilon})$.*

4 Minimum Vertex Cover

Let $\mathbf{B}$ be a C-approximation insertion-deletion streaming algorithm for Minimum Vertex Cover that succeeds with probability $1 - 1/400$. Similar to the previous section, we will now show how $\mathbf{B}$ can be used to solve $\mathsf{BInd}_\delta^{n,k}$.

4.1 Reduction

Let $A \in \{0,1\}^{n \times n}, x \in [n-k]$ and $y \in [n-k]$ be an instance of $\mathsf{BInd}_\delta^{n,k}$. The reduction for Minimum Vertex Cover is very similar to the reduction for Maximum Matching presented in the previous section. Alice's behaviour is in fact identical:

First, Alice and Bob sample a uniform random binary matrix $X \in \{0,1\}^{n \times n}$ and random permutations $\sigma_1, \sigma_2 : [n] \to [n]$ from public randomness. Alice then computes matrix A' which is obtained by first computing $A \oplus X$ and then permuting the rows and then the columns of the resulting matrix by σ_1 and σ_2, respectively. Alice interprets A' as the incidence matrix of a bipartite graph $G(A')$. Alice then runs algorithm $\mathbf{B}$ on a random ordering of the edges of $G(A')$ and sends the resulting memory state to Bob.

Next, Bob also computes the entry-wise XOR between the part of the matrix A that he knows about and X, followed by applying the permutations σ_1 and σ_2. In doing so, Bob knows the matrix entries of A' at positions $(\sigma_1(i), \sigma_2(j))$ for every $(i, j) \in S(x, y)$. He can therefore compute the subset E_S of the edges of $G(A')$ with

$$E_S = \{(\sigma_1(i), \sigma_2(j)) \in [n]^2 \mid (i, j) \in S(x, y) \text{ and } A'(\sigma_1(i), \sigma_2(j)) = 1\} .$$

Next, Bob continues the execution of $\mathbf{B}$ and introduces deletions *for all edges in E_S* in random order. Observe that this step is different to the reduction for Maximum Matching. Let I be the vertex cover produced by $\mathbf{B}$.

Parallel Executions. Alice and Bob run the procedure above 40 times in parallel. Denote by I^i, X^i, E^i_S, A'^i, σ^i_1, and σ^i_2 the variables I, X, E_S, A', σ_1 and σ_2 used in iteration i. Furthermore, let Q_i be the indicator variable that is 1 iff $\{\sigma^i_1(x), \sigma^i_2(y)\} \cap I_i \neq \varnothing$, i.e., the potential edge $(\sigma^i_1(x), \sigma^i_2(y))$ is covered by the vertex cover.

If there exists a run j with $Q_j = 0$, then Bob predicts $A_{x,y} = X_{x,y}$ (if there are multiple such runs then Bob breaks ties arbitrarily). Otherwise, Bob returns `fail` and the algorithm errs.

4.2 Analysis

The first lemma applies to every parallel run j. For simplicity of notation, we will omit the superscripts that indicate the parallel run in our random variables.

We first show an upper bound on the size of a minimum vertex cover in $G(A') - E_S$.

▶ **Lemma 9.** *The size of a minimum vertex cover in $G(A') - E_S$ is at most $2(n - k) + 1$.*

Proof. Let U, V be the bipartitions of the graph $G(A') - E_S$, let $U' = \{\sigma_1(a) : a \in [x, x+k)\}$ and let $V' = \{\sigma_2(b) : b \in [y, y + k)\}$. Observe that $(G(A') - E_S)[U' \cup V']$ contains at most one edge: The potential edge between $\sigma_1(x)$ and $\sigma_2(y)$. A valid vertex cover of $G(A') - E_S$ is therefore $(U \setminus U') \cup (V \setminus V') + \sigma_1(x)$, which is of size $2(n - k) + 1$. ◀

Next, we prove the key property of our reduction: We show that if $A'_{\sigma_1(x),\sigma_2(y)} = 0$ (or equivalently, $A_{x,y} \oplus X_{x,y} = 0$) then neither $\sigma_1(x)$ nor $\sigma_2(y)$ is in the output vertex cover with large probability.

▶ **Lemma 10.** *Assume that algorithm $\mathbf{B}$ does not err in run j. Suppose that $A'^j_{\sigma^j_1(x),\sigma^j_2(y)} = 0$. Then the probability that $Q_j = 1$ is at most*

$$\frac{3C \cdot (2(n - k) + 1)}{k} .$$

Proof. Consider the set $D = \{(\sigma^j_1(x + i), \sigma^j_2(y + i)) \mid 0 \leq i \leq k - 1\}$, i.e., the positions of the diagonal of $S(x, y) \cup \{x, y\}$ permuted by σ^j_1 and σ^j_2. Then, since A'^j is a uniform random matrix, with probability at least $1 - \frac{1}{k^{10}}$, the "permuted diagonal" A'^j_D contains at least $0.99k/2$ entries with value 0, or, in other words, graph $G(A'^j) - E^j_S$ contains at least $0.99k/2$ non-edges in the positions of the permuted diagonal D. By Lemma 9, the size of a minimum vertex cover in $G(A'^j) - E^j_S$ is at most $2(n - k) + 1$, and since $\mathbf{B}$ has an approximation factor of C, the vertex cover I_j is of size at most $C \cdot (2(n - k) + 1)$. Hence, at most $C \cdot (2(n - k) + 1)$ non-edges in D can be covered in I_j. However, since the permutations are random, the probability that the non-edge $(\sigma^j_1(x), \sigma^j_2(y))$ is covered, which is identical to the event $Q_j = 1$, is therefore at most

$$\frac{C \cdot (2(n - k) + 1)}{0.99k/2} \leq \frac{3C \cdot (2(n - k) + 1)}{k} .$$ ◀

▶ **Theorem 11.** *Let* $\mathbf{B}$ *be a* n^{ϵ}*-approximation insertion-deletion streaming algorithm for Minimum Vertex Cover that uses space* s *and errs with probability at most* $1/400$*. Then, there exists a communication protocol for* $\mathsf{BInd}_{\frac{1}{3}}^{n,n-\frac{1}{20}n^{1-\epsilon}}$ *that communicates* $\mathrm{O}(s)$ *bits.*

Proof. Let $k = n - \frac{1}{40}n^{1-\epsilon}$ and let $C = n^{\epsilon}$. Consider the reduction given in the previous subsection. First, observe that since $\mathbf{B}$ errs with probability at most $1/400$, by the union bound the probability that $\mathbf{B}$ errs at least once in the 40 parallel executions of our reduction is at most $\frac{1}{10}$. We assume from now on that the algorithm never errs.

Observe that the matrices A'^{j} are random matrices. Hence, the probability that there exists at least one run i with $A'^{i}_{\sigma_1^i(x),\sigma_2^i(y)} = 0$ is at least $1 - (\frac{1}{2})^{40}$. Suppose that this event happens. Let run i be so that $A'^{i}_{\sigma_1^i(x),\sigma_2^i(y)} = 0$. Then, by Lemma 10, the probability that the non-edge $(\sigma_1^i(x), \sigma_2^i(y))$ is covered by I_i, or in other words, the probability that $Q_i = 1$, is at most

$$\frac{3C \cdot (2(n-k)+1)}{k} = \frac{3n^{\epsilon} \cdot (\frac{1}{20}n^{1-\epsilon}+1)}{n - \frac{1}{40}n^{1-\epsilon}} = \frac{\frac{3}{20}n + 3n^{\epsilon}}{n - \frac{1}{40}n^{1-\epsilon}} = \frac{3}{20} + o(1) \ .$$

Observe that whenever $Q_i = 0$, the algorithm outputs $X^i_{x,y}$ as a predictor for $A_{x,y}$. Since the algorithm $\mathbf{B}$ does not err, we have $A_{x,y} \oplus X^i_{x,y} = 0$. This implies that $A_{x,y} = X^i_{x,y}$, which establishes correctness.

Last, we need to bound the error probability of our algorithm. First, the probability that at least one of the 40 runs fails is at most $\frac{1}{10}$. Next, the probability that none of the runs are such that $A'^{j}_{\sigma_1^j(x),\sigma_2^j(y)} = 0$ is at most $(\frac{1}{2})^{40}$. Furthermore, the probability that $Q_i = 1$ when $A'^{i}_{\sigma_1^i(x),\sigma_2^i(y)} = 0$ is at most $\frac{3}{20} + o(1)$. Applying the union bound, we see that the overall error probability of our algorithm is at most

$$\frac{1}{10} + (\frac{1}{2})^{40} + \frac{3}{20} + o(1) \leq \frac{1}{3} \ ,$$

for large enough n. ◀

Since by Theorem 4, $\mathsf{BInd}_{\frac{1}{3}}^{n,n-\frac{1}{40}n^{1-\epsilon}}$ has a communication complexity of $\Omega(n^{2-2\epsilon})$, we obtain the following result:

▶ **Corollary 12.** *Every insertion-deletion* n^{ϵ}*-approximation streaming algorithm for Minimum Vertex Cover with error probability at most* $\frac{1}{400}$ *requires space* $\Omega(n^{2-2\epsilon})$*.*

4.3 Insertion-deletion Streaming Algorithm for Minimum Vertex Cover

We now sketch a simple deterministic n^{ϵ}-approximation insertion-deletion streaming algorithm for Minimum Vertex Cover on general graphs that uses space $\mathrm{O}(n^{2-2\epsilon}\log n)$. Let $G = (V, E)$ be the graph described by the input stream. The algorithm proceeds as follows:

Algorithm 1 A simple deterministic n^{ϵ}-approximation insertion-deletion streaming algorithm for Minimum Vertex Cover.

```
1. Arbitrarily partition $V$ into subsets $V_1, V_2, \dots, V_{n^{1-\epsilon}}$, each of size $n^{\epsilon}$.
2. Consider the multi-graph $G'$ obtained from $G$ by contracting the sets $V_i$ into vertices.
3. While processing the stream: For each pair of vertices $V_i, V_j$ in $G'$ deterministically
   maintain the number of edges connecting $V_i$ to $V_j$.
4. Post-processing: Compute a minimum vertex cover $I'$ in the multi-graph $G'$.
5. Return $I = \cup_{V_j \in I'} V_j$ as the vertex cover in $G$.
```

Analysis: Regarding space, the dominating space requirement is the maintenance of the number of edges between every pair V_i, V_j. Since there are $n^{2-2\epsilon}$ such pairs, this requires space $\mathrm{O}(n^{2-2\epsilon} \cdot \log n)$.

Concerning the approximation factor, let I^* be a minimum vertex cover in G. Recall that I' is an optimal cover in G' and hence $|I'| \leq |I^*|$ (edge contractions cannot increase the size of a minimum vertex cover). Since every set V_j is of size n^ϵ, the computed vertex cover I is of size at most $|I'| \cdot n^\epsilon \leq |I^*| n^\epsilon$, which proves the approximation factor. By construction of the algorithm, every edge is covered.

▶ **Theorem 13.** *There is a deterministic n^ϵ-approximation insertion-deletion streaming algorithm for* Minimum Vertex Cover *that uses space $\mathrm{O}(n^{2-2\epsilon} \log n)$.*

References

1 Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 459–467, USA, 2012. Society for Industrial and Applied Mathematics.

2 Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 20:1–20:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.20.

3 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.

4 Sepehr Assadi and Sanjeev Khanna. Randomized composable coresets for matching and vertex cover. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '17, page 3–12, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3087556.3087581.

5 Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364. SIAM, 2016. doi:10.1137/1.9781611974331.ch93.

6 Ziv Bar-Yossef, Thathachar S Jayram, Robert Krauthgamer, and Ravi Kumar. The sketching complexity of pattern matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 261–272. Springer, 2004.

7 Suman K. Bera and Amit Chakrabarti. Towards tighter space bounds for counting triangles and other substructures in graph streams. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.11.

8 Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1326–1344. SIAM, 2016. doi:10.1137/1.9781611974331.ch92.

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 Ali Ridha Mahjoub, editors, *Combinatorial Optimization - 5th International Symposium, ISCO 2018, Marrakesh, Morocco, April 11-13, 2018, Revised Selected Papers*, volume 10856 of *Lecture Notes in Computer Science*, pages 101–114. Springer, 2018. doi:10.1007/978-3-319-96151-4_9.

10 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.

11 Graham Cormode and Hossein Jowhari. A second look at counting triangles in graph streams (corrected). *Theor. Comput. Sci.*, 683:22–30, 2017. doi:10.1016/j.tcs.2016.06.020.

12 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.

13 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485. SIAM, 2012. doi:10.1137/1.9781611973099.41.

14 Bjarni V. Halldórsson, Magnús M. Halldórsson, Elena Losievskaja, and Mario Szegedy. Streaming algorithms for independent sets in sparse hypergraphs. *Algorithmica*, 76(2):490–501, 2016. doi:10.1007/s00453-015-0051-5.

15 Magnús M. Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. Streaming and communication complexity of clique approximation. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2012. doi:10.1007/978-3-642-31594-7_38.

16 Monika Rauch Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. In James M. Abello and Jeffrey Scott Vitter, editors, *External Memory Algorithms, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, May 20-22, 1998*, volume 50 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 107–118. DIMACS/AMS, 1998. doi:10.1090/dimacs/050/05.

17 Kaave Hosseini, Shachar Lovett, and Grigory Yaroslavtsev. Optimality of Linear Sketching Under Modular Updates. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2019.13.

18 Sagar Kale and Sumedh Tirodkar. Maximum matching in two, three, and a few more passes over graph streams. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 15:1–15:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.15.

19 John Kallaugher and Eric Price. Separations and equivalences between turnstile streaming and linear sketching. In *Symposium on Theory of Computing, STOC 2020*, 2020. to appear.

20 Daniel M. Kane, Kurt Mehlhorn, Thomas Sauerwald, and He Sun. Counting arbitrary subgraphs in data streams. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 598–609. Springer, 2012. doi:10.1007/978-3-642-31585-5_53.

21 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751. SIAM, 2014. doi:10.1137/1.9781611973402.55.

22 Michael Kapralov, Aida Mousavifar, Cameron Musco, Christopher Musco, Navid Nouri, Aaron Sidford, and Jakab Tardos. Fast and space efficient spectral sparsification in dynamic streams. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1814–1833. SIAM, 2020. doi:10.1137/1.9781611975994.111.

23 Christian Konrad. Maximum matching in turnstile streams. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 840–852. Springer, 2015. doi:10.1007/978-3-662-48350-3_70.

24 Christian Konrad. A simple augmentation method for matchings with applications to streaming algorithms. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPIcs*, pages 74:1–74:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.74.

25 Christian Konrad. Streaming frequent items with timestamps and detecting large neighborhoods in graph streams. *CoRR*, abs/1911.08832, 2019. arXiv:1911.08832.

26 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.

27 Christian Konrad and Adi Rosén. Approximating semi-matchings in streaming and in two-party communication. *ACM Trans. Algorithms*, 12(3):32:1–32:21, 2016. doi:10.1145/2898960.

28 Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183. ACM, 2014. doi:10.1145/2591796.2591812.

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

30 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

31 Ami Paz and Gregory Schwartzman. A $(2+\epsilon)$-approximation for maximum weight matching in the semi-streaming model. *ACM Trans. Algorithms*, 15(2):18:1–18:15, 2019. doi:10.1145/3274668.