Matchings in Low-Arboricity Graphs in the Dynamic Graph Stream Model

Christian Konrad
University of Bristol, UK

Andrew McGregor
University of Massachusetts Amherst, MA, USA

Rik Sengupta
IBM Research, Cambridge, MA, USA
University of Massachusetts Amherst, MA, USA

Cuong Than
University of Massachusetts Amherst, MA, USA

Abstract

We consider the problem of estimating the size of a maximum matching in low-arboricity graphs in the dynamic graph stream model. In this setting, an algorithm with limited memory makes multiple passes over a stream of edge insertions and deletions, resulting in a low-arboricity graph. Let n be the number of vertices of the input graph, and α be its arboricity. We give the following results.

1. As our main result, we give a three-pass streaming algorithm that produces an $(\alpha+2)(1+\epsilon)$-approximation and uses space $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{1/2} \cdot \log n)$. This result should be contrasted with the $\Omega(\alpha^{-5/2} \cdot n^{1/2})$ space lower bound established by [Assadi et al., SODA'17] for one-pass algorithms, showing that, for graphs of constant arboricity, the one-pass space lower bound can be achieved in three passes (up to poly-logarithmic factors). Furthermore, we obtain a two-pass algorithm that uses space $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{3/5} \cdot \log n)$.
2. We also give a $(1+\epsilon)$-approximation multi-pass algorithm, where the space used is parameterized by an upper bound on the size of a largest matching. For example, using $O(\log\log n)$ passes, the space required is $O(\epsilon^{-1} \cdot \alpha^2 \cdot k \cdot \log n)$, where k denotes an upper bound on the size of a largest matching.

Finally, we define a notion of arboricity in the context of matrices. This is a natural measure of the sparsity of a matrix that is more nuanced than simply bounding the total number of nonzero entries, but less restrictive than bounding the number of nonzero entries in each row and column. For such matrices, we exploit our results on estimating matching size to present upper bounds for the problem of rank estimation in the dynamic data stream model.

2012 ACM Subject Classification Theory of computation → Sketching and sampling

Keywords and phrases Data Streams, Graph Matching, Graph Arboricity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.29

Funding *Christian Konrad*: Supported by EPSRC New Investigator Award EP/V010611/1.
Andrew McGregor: Funded in part by NSF Award CCF 1934846.
Rik Sengupta: Funded in part by NSF Award CCF 1934846.
Cuong Than: Funded by NSF CAREER Award CCF 2237288, NSF Award CCF 2121952, and a Google Research Scholar Award.

Acknowledgements We wish to thank Cameron Musco, Hung Le, Rajarshi Bhattacharjee, and David Woodruff for a lot of preliminary discussions about this work.

1 Introduction

Streaming algorithms for graph problems have been studied for more than 25 years [25]. In this setting, an algorithm performs one or more passes over the input graph and produces a solution using as little space as possible.

Much of the development of the literature on graph streams has been driven by the study of the Maximum Matching problem (see, e.g., [2–6, 10–12, 17–19, 22–24, 28–34, 36, 41] for a non-comprehensive list of recent and early works). This problem was first addressed by Feigenbaum et al. [21], and a large number of algorithms and impossibility results that cover various aspects of the problem are known today, such as one-pass/multi-pass algorithms, adversarial/random order streams, insertion-only/insertion-deletion streams, and dense/sparse input graphs.

In this paper, we consider the size estimating variant of the problem, which we denote by Matching Size Estimation (MSE). This is in contrast to the much better understood objective of outputting the actual edges of a large matching. We address MSE in *dynamic* or *insertion-deletion* streams, i.e., streams consisting of a sequence of edge insertions and deletions. Our focus lies on sparse graphs as parameterized by the *arboricity* of the input graph. The arboricity of a graph is the smallest[1] integer α such that the edges of the graph can be partitioned into α forests. Nash-Williams [39] showed that an equivalent definition of the arboricity is $\alpha = \max_{S \subseteq V} |e(S)|/(|S|-1)$ where $e(S)$ is the number of edges in the induced subgraph $G[S]$.

Chitnis et al. [14] were the first to study MSE in graphs of bounded arboricity in the dynamic graph stream setting. They showed that there is a one-pass $\tilde{O}(\alpha \cdot n^{4/5})$-space algorithm[2] with approximation factor $O(\alpha)$. Cormode et al. [15] subsequently gave an $\tilde{O}(\alpha^2)$-approximation algorithm using space $\tilde{O}(\alpha^{10/3} \cdot n^{2/3})$, albeit under the restriction that the length of the input stream is $O(\alpha \cdot n)$. On the lower bound side, Assadi et al. [7] showed that computing an α-approximation in a single pass requires space $\Omega(n^{1/2} \cdot \alpha^{-5/2})$. The problem is thus wide open, even in the one-pass setting, and even for constant-arboricity graphs.

1.1 Our Results

In this paper, we give the first multi-pass algorithms for MSE in the dynamic graph stream setting, for graphs of arboricity α. We assume throughout that n, m, and α are known in advance. All our algorithms succeed with high probability, i.e., they output the correct matching size with probability $1 - 1/\text{poly}(n)$. We observe at this juncture that none of our algorithms require the assumption that the input stream length is bounded. We reiterate that the one-pass algorithm by Cormode et al. [15], which uses space $O(\alpha^{10/3} \cdot n^{2/3})$, requires this assumption.

Our main result is a three-pass $O(\alpha)$-approximation algorithm that uses roughly $\sqrt{n}$ space[3].

▶ **Theorem 1.** *There exists a three-pass algorithm using $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{1/2} \cdot \log^3 n)$ space that returns an $(\alpha + 2)(1 + \epsilon)$-approximation for* MSE *with high probability.*

[1] We shall abuse terminology slightly and say the arboricity of a graph G is α as long as the smallest integer is at most α.
[2] We write $\tilde{O}(.)$ to mean $O(.)$ with poly-log dependencies on n suppressed.
[3] Henceforth, we say specify the space use of the algorithms in terms of the number of words of memory where a word may store $O(\log n)$ bits.

This result should be contrasted with the "$\sqrt{n}$-barrier" result established by Assadi et al. [7], who showed that one-pass α-approximation algorithms for MSE in dynamic graph streams on graphs of arboricity α require space $\Omega(\sqrt{n}/\alpha^{2.5})$. While our algorithm uses three passes, and, consequently, the lower bound from [7] does not apply in this setting, we nevertheless show that the $\sqrt{n}$-barrier can be achieved at the expense of just two additional passes. Interestingly, we note that no multi-pass $O(\alpha)$-approximation dynamic streaming algorithms are known for MSE that break the $\sqrt{n}$-barrier, even if significantly more passes are allowed.

Next, we also give a new two-pass algorithm that requires less space than the best one-pass algorithms known (e.g., [15]).

▶ **Theorem 2.** *There exists a two-pass algorithm using $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{3/5} \cdot \log^3 n)$ space that returns an $(\alpha + 2)(1 + \epsilon)$-approximation for* MSE *with high probability.*

We also show a $(1+\epsilon)$-approximation multi-pass algorithm, in the case when the maximum matching size is bounded by a given parameter k.

▶ **Theorem 3.** *If the maximum matching size is upper bounded by k, there exists a $O(\epsilon^{-1} \cdot \alpha^2 \cdot n^{1/(2^p-1)} \cdot k^{1-1/(2^p-1)} \cdot \log n)$ space, p-pass dynamic graph streaming algorithm that returns a $(1 + \epsilon)$-approximation for* MSE *with high probability. In particular, there exists a $O(\log \log n)$-pass algorithm that uses space $O(\epsilon^{-1} \cdot \alpha^2 \cdot k \cdot \log n)$.*

This result is similar in spirit to a result by Chitnis et al. [14], who showed that, in general graphs, a matching of size k (if there is one) can be computed in the one-pass dynamic setting using space $\tilde{O}(k^2)$; this result can also be obtained from the algorithm given by Assadi et al. [8]. Phrased differently, given an upper bound k on the size of a largest matching, we can compute one using space $\tilde{O}(k^2)$.

Lastly, as a more conceptual contribution, we introduce the notion of *low-arboricity* matrices. We say that a matrix A has arboricity α if every $t \times t$ submatrix of A has at most $\alpha \cdot t$ nonzero entries. This generalizes many natural subclasses of sparse matrices (including the adjacency matrices of low-arboricity graphs). We show that, given such a matrix A, we can associate a bipartite graph G_A of arboricity at most α with A, such that the rank of A is within an α-factor of $\mu(G_A)$, where $\mu(G_A)$ denotes the size of a largest matching in G_A. Hence, using any β-approximation algorithm for estimating the size of a maximum matching in graphs of arboricity α (where β might depend on α), we obtain an $(\alpha \cdot \beta)$-approximation to its rank. In particular, our two-pass and three-pass $O(\alpha)$-approximation algorithms immediately yield $O(\alpha^2)$-approximation algorithms for rank approximation in matrices of arboricity α.

1.2 Our Techniques

We will first discuss the ideas behind our $(1 + \epsilon)$-approximation algorithm and our two-pass algorithm. Our three-pass algorithm, which constitutes our main result, combines ideas from these two algorithms.

$(1+\epsilon)$-approximation Algorithm. Our $(1+\epsilon)$-approximation algorithm works by iteratively identifying all "high" degree vertices with Count-Sketch. In the ith pass (for $i \leq p - 1$) we identify a set of vertices V_i in the induced subgraph $G[V - V_1 - \ldots - V_{i-1}]$ using estimates of their degrees. Removing the high-degree vertices discovered in previous passes and exploiting properties of the degree sequence of low-arboricity graphs enable us to get increasingly accurate estimates of the degrees in the remaining graph. In the final pass, we collect all edges in $G[V - V_1 - \ldots - V_{p-1}]$, along with a few edges incident to every vertex in $V_1 \cup \ldots \cup V_{p-1}$. We are then able to argue that, by carefully setting the parameters of the algorithm and appealing to a sparsification result by Solomon [40], this approach can be used to obtain a $(1+\epsilon)$-approximation. The space used by the algorithm (in terms of the number of edges) decreases at a rate which is doubly-exponential in p. This space/passes trade-off is somewhat unusual (but not unprecedented, e.g., [1, 9]); it is more typical in the data streams literature to see a singly-exponential or even just a polynomial trade-off.

2-pass Algorithm. Our 2-pass algorithm uses a result about the fractional matching due to McGregor and Vorotnikova [37]. They showed that it is possible to set a weight for each edge that just depends on the degrees of the endpoints of the edge, such that the total weight of all edges yields an $(\alpha + 2)$-approximation to the size of the maximum matching. This yields a simple 2-pass algorithm: we sample edges uniformly (first pass), compute their weights (second pass), and return an estimator based on these weights. This approach works well when the matching is large. By combining this approach with ideas from the previous result, we get our final algorithm, which uses small space regardless of the size of the matching.

3-pass Algorithm. Our 3-pass algorithm combines ideas from the other two algorithms. In the first pass, we identify all vertices with degree roughly $\sqrt{n}$ or higher. Let E_H be the edges that share an endpoint with these vertices, and let E_L be the remaining edges. The approach is to sparsify E_H to produce E'_H in such a way that $\mu(E'_H \cup E_L) \geq \mu(E_H \cup E_L)(1-\epsilon)$. We then use uniform sampling to construct a multiset E'_L of edges, and use this, along with the weights of these edges, to estimate $\mu(E'_H \cup E_L)$ via the fractional matching approach.

1.3 Further Related Work: Matching Size Estimation in Graph Streams

Esfandiari et al. [20] were the first to consider the MSE problem in low-arboricity graphs in the streaming model. They focused on the insertion-only setting, where edges can only be inserted but not deleted, and gave a one-pass $O(\alpha)$-approximation algorithm that uses $\tilde{O}(\alpha \cdot n^{2/3})$ space. This result was significantly improved by Cormode et al. [15], who gave an algorithm with the same approximation guarantee, that uses only $O(\alpha \cdot \log^2 n)$ space. This was further improved, in terms of both the approximation factor and space, by McGregor and Vorotnikova [38].

At the heart of many of the algorithms for approximating the matching size in low-arboricity graphs lie structural lemmas that relate the maximum matching size to a function of the degree sequence of the graph in question. McGregor and Vorotnikova [37] gave such a characterization which gave rise to improved approximation guarantees over those established by Esfandiari et al. See also [26] for a different structural result. In particular, they obtained a $(5+\epsilon)$-approximation for planar graphs, improving over a $(24+\epsilon)$-approximation guarantee established in [20].

Assadi et al. [7] gave various upper and lower bounds for approximating the matching size in general graphs. They showed that there is a one-pass $\tilde{O}(n^2/\alpha^4)$-space algorithm that approximates the size of a matching within a factor of α in the dynamic graph stream setting. They also gave a lower bound, showing that space $n^{2-O(\epsilon)}$ is needed to obtain a $(1+\epsilon)$-approximation factor.

1.4 Outline

In Section 2, we present known results, including a result due to McGregor and Vorotnikova [37] that links the matching size in low-arboricity graphs to the degree sequence of the graph, as well as a matching-size preserving sparsification result by Solomon [40]. We will need both of these in this paper. Subsequently, in Section 3, we give all our algorithmic results. In Section 4, we introduce the notion of low-arboricity matrices and show how their ranks can be approximated using algorithms for approximating the matching size in low-arboricity graphs. Finally, we conclude in Section 5 with some open problems.

2 Preliminaries

Notation and Definitions. Given a graph $G = (V, E)$, for each vertex $u \in V$, we denote by $\deg(u)$ the degree of u. The size of the maximum matching of G is denoted by $\mu(G)$. We also use $\mu(E')$ to denote the maximum matching among any set E' of edges. Throughout this paper, we let n and m denote the sizes of the sets V and E respectively, and α denote the arboricity of G. It is well-known that $m \leq \alpha \cdot n$.

Algorithmic Primitives. We will use the following results throughout the remaining sections, to simplify our proofs.

▶ **Theorem 4** (Algorithmic Primitives [13, 16, 27]). *There exist single-pass dynamic graph stream algorithms for:*

1. *Uniformly sampling edges: The algorithm uses $O(\log^2 n)$ space. This is a special case of ℓ_0-sampling [27].*
2. *Estimating degrees: We can compute estimates $\widetilde{\deg}$ such that with probability at least $1 - 1/n$:*
   a. *For all $u \in V$: $\deg(u) \leq \widetilde{\deg}(u) \leq \deg(u) + \|\mathbf{d}_{w\text{-tail}}\|_1/w$*
   b. *For all $u \in V$: $|\widetilde{\deg}(u) - \deg(u)| \leq \|\mathbf{d}_{w\text{-tail}}\|_2/\sqrt{w}$*
   where $\mathbf{d}_{w\text{-tail}}$ is the vector of degrees with the largest w entries replaced by 0. The first guarantee is achieved by CountMin sketch [16] and the second by CountSketch [13]. Both algorithms use $O(w \log n)$ space.

Structural Results. We will make repeated use of the following structural results for low-arboricity graphs. The first theorem effectively shows that there is a fractional matching where (a) the weight of each edge is just a function of the degrees of the endpoints of that edge, and (b) the total weight of the fractional matching is still a good approximation to the maximum-cardinality matching.

▶ **Theorem 5** (McGregor and Vorotnikova [37]). *Given a graph $G = (V, E)$ with arboricity at least α, let:*

$$w(G) = \sum_{(u,v) \in E} w_{u,v}, \text{ where } w_{u,v} = \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha + 1}\right) .$$

Then, we have:

$$\mu(G) \leq (\alpha + 1) w(G) \leq \mu(G) \cdot \gamma, \text{ where } \gamma = \begin{cases} \alpha + 2 & \text{if } \alpha \text{ is odd} \\ \frac{(\alpha+3)(\alpha+1)}{\alpha+2} & \text{if } \alpha \text{ is even} \\ \alpha + 1 & \text{if } G \text{ is bipartite.} \end{cases}$$

Algorithm 1 A p-pass algorithm for approximating matching size.

1. $G_1 \leftarrow G$ and $s = m^{1/(2^p-1)} \cdot (2k)^{1-1/(2^p-1)}$, where k is an upper bound on $\mu(G)$.
2. For $i = 1$ to $p-1$:
   a. *Pass i:* Use CountSketch to compute an estimate $\widetilde{\deg}(u)$ of the degree of each vertex $u \in G_i$. Let $V_i = \{u \in G_i : \widetilde{\deg}(u) \geq 0.75 \cdot \tau_{i+1}\}$, where $\tau_2 = \sqrt{m/s}$ and for $i \geq 2$, we have $\tau_{i+1} = m^{1/2^i} \cdot (2k/s)^{1-1/2^{i-1}}$. Also let:
   $$G_{i+1} := G[V - V_1 - V_2 - \ldots - V_i]$$
3. *Pass p:* Store all edges in G_p, and call this set E_L. For each $u \in V_1 \cup \ldots \cup V_{p-1}$, store $\Theta(\alpha/\epsilon)$ incident edges to u. Call these edges E'_H.
4. *Output:* $\mu(E_L \cap E'_H)$.

Note that, in all cases, $\gamma \leq \alpha + 2$. Furthermore, if h is the number of vertices of degree at least $\alpha + 2$, and s is the number of edges whose endpoints have degree strictly less than $\alpha + 2$, then $\mu(G) \leq h + s \leq (\alpha + 2) \cdot \mu(G)$.

The next theorem demonstrates that, in a low-arboricity graph, it is possible to remove most of the edges incident to high degree vertices without significantly reducing the size of the maximum-cardinality matching.

▶ **Theorem 6** (Solomon [40]). *Fix a graph G with arboricity α, and a positive number $\epsilon > 0$. For each vertex $u \in V(G)$, mark $\Theta(\alpha/\epsilon)$ arbitrary edges incident to u. Let G' be the graph containing edges marked by both ends. Then, $\mu(G') \leq \mu(G) \leq (1+\epsilon) \cdot \mu(G')$.*

3 Graph Results

3.1 $(1+\epsilon)$-Approximation

Consider Algorithm 1. The main idea behind the algorithm is to exploit the fact that, if G has low arboricity and $\mu(G)$ is "small", then G has only a "small" number of "high"-degree vertices. If we can remove the high-degree vertices, then the remaining graph has significantly fewer edges, by the following lemma.

▶ **Lemma 7.** *Any graph G with maximum degree Δ has at most $(2\Delta - 1) \cdot \mu(G) < 2\Delta \cdot \mu(G)$ edges.*

Proof. This follows because the endpoints of a maximal matching is a vertex cover. ◀

If we can decrease the total number of edges, we can estimate the degrees in the remaining graph with greater accuracy. The next lemma quantifies this, where the proof exploits properties of the degree sequence for low-arboricity graphs. Repeating this process for $p-1$ passes allows us to iteratively "peel off" high-degree vertices, until we are left with a graph that is sufficiently sparse such that it can be stored explicitly.

▶ **Lemma 8.** *Let $s \geq (\alpha + 2) \cdot \mu(G)$. Using $O(\alpha \cdot s \cdot \log n)$ space in the dynamic graph streaming model, given a graph with m edges, with high probability we can identify a subset of vertices that includes all vertices of degree $\geq \sqrt{m/s}$, and no vertex of degree $< 0.5\sqrt{m/s}$.*

Proof. We use CountSketch to find all vertices of degree more than $\sqrt{m/s}$. Let $d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n$ be the degree sequence of G. Observe that since $s \geq (\alpha+2) \cdot \mu(G)$, Theorem 5 implies s is greater than the total number of vertices with degree at least $\alpha + 2$. Thus, all degrees d_i with $i > s$ are at most $\alpha + 1$.

Hence, the ℓ_2^2 of the tail $\mathbf{t} = (d_{s+1}, d_{s+2}, \ldots, d_n)$ is at most $\|\mathbf{t}\|_\infty \cdot \|\mathbf{t}\|_1 \leq m(\alpha+1)$. Using CountSketch with space $O(\alpha \cdot s \cdot \log n)$, we can compute the degree of each vertex with additive error:

$$\frac{\ell_2((d_{s+1}, d_{s+2}, \ldots, d_n))}{\sqrt{\alpha \cdot s}} = O\left(\sqrt{m/s}\right) \ .$$

With the proper choice of the suppressed constant, we obtain an additive error of $0.25\sqrt{m/s}$. For each vertex u, let $\widetilde{\deg}(u)$ be the degree of u estimated by CountSketch. Let $V' = \{u \ : \ \widetilde{\deg}(u) \geq 0.75\sqrt{m/s}\}$. For any $u \in V'$, we have $\deg(u) \geq 0.75\sqrt{m/s} - 0.25\sqrt{m/s} = 0.5\sqrt{m/s}$. Furthermore, for every vertex u such that $\deg(u) \geq \sqrt{m/s}$, we have $\widetilde{\deg}(u) \geq \sqrt{m/s} - 0.25\sqrt{m/s} = 0.75\sqrt{m/s}$, implying that $u \in V'$. ◀

This leads to the following theorem.

▶ **Theorem 3.** *If the maximum matching size is upper bounded by k, there exists a $O(\epsilon^{-1} \cdot \alpha^2 \cdot n^{1/(2^p-1)} \cdot k^{1-1/(2^p-1)} \cdot \log n)$ space, p-pass dynamic graph streaming algorithm that returns a $(1+\epsilon)$-approximation for* MSE *with high probability. In particular, there exists a $O(\log\log n)$-pass algorithm that uses space $O(\epsilon^{-1} \cdot \alpha^2 \cdot k \cdot \log n)$.*

Proof. Consider the multi-pass algorithm where in pass i, we find a subset V_i of vertices in $G_i = G[V - V_1 - V_2 - \ldots - V_{i-1}]$ that includes all vertices of degree at least $\Delta_{i+1} = \sqrt{m_i/s}$, where m_i is the number of edges in G_i (see Algorithm 1). By Lemma 7, we have $m_i \leq 2\Delta_i \cdot \mu(G_i) \leq 2\Delta_i \cdot k$, since the maximum degree of G_i is less than Δ_i. Hence:

$$\begin{aligned}
\Delta_{i+1} = \sqrt{m_i/s} \quad &\leq \quad \Delta_i^{1/2}(2k/s)^{1/2} \\
&\leq \quad \Delta_{i-1}^{1/4}(2k/s)^{1/2+1/4} \\
&\leq \quad \ldots \\
&\leq \quad \Delta_2^{1/2^{i-1}}(2k/s)^{1-1/2^{i-1}} \\
&\leq \quad m^{1/2^i}(2k)^{1-1/2^{i-1}}/s^{1-1/2^i} \ ,
\end{aligned}$$

since $\Delta_2 \leq \sqrt{m/s}$. In particular, if $s = m^{1/(2^p-1)}(2k)^{1-1/(2^p-1)}$, then $\Delta_p \leq s/(2k)$.

Hence, G_p has at most $2(s/(2k)) \cdot k = s$ edges, which can be stored in memory. We collect $O(\alpha/\epsilon)$ edges incident to all vertices in $V_1 \cup \ldots \cup V_{p-1}$, and all edges between the remaining vertices. Let G_S be the new graph. The total number of edges in G_S is at most the number of edges in G_p, plus $(\alpha/\epsilon) \cdot \sum_{i=1}^{p-1} |V_i|$, which is at most $\alpha(\alpha+2) \cdot \mu(G)/\epsilon$, since by Theorem 5, the total number of vertices with degree higher than $\alpha + 2$ is at most $(\alpha+2) \cdot \mu(G)$.

We then have that $\mu(G_S) \leq \mu(G) \leq (1+\epsilon) \cdot \mu(G_S)$. It follows from Theorem 6 that G_S is a supergraph of some graph G' satisfying $\mu(G') \geq \mu(G)/(1+\epsilon)$. Thus, $\mu(G_S) \geq \mu(G)/(1+\epsilon)$. On the other hand, since G_S is a subgraph of G, we obtain $\mu(G_S) \leq \mu(G)$, as desired. ◀

3.2 $O(\alpha)$-Approximation, Two Passes, and $\tilde{O}(m^{3/5})$ Space

Consider Algorithm 2. The analysis to establish the following theorem is a relatively straightforward application of the Chernoff bound.

Algorithm 2 A 2-pass algorithm for approximating matching size.

1. *First Pass:* Sample $t := 3m\epsilon^{-2}(\alpha+1)\log(2n)/k$ edges with replacement. Let E' be the (multi-)set of edges sampled.
2. *Second Pass:* For each $(u, v) \in E'$, compute $w_{u,v} = \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right)$.
3. *Output:* $W = (m/t) \cdot X$, where $X = \sum_{e \in E'} w_e$.

▶ **Theorem 9.** *If $\mu(G) \geq k$, there exists a $O(\epsilon^{-2} \cdot \alpha \cdot mk^{-1} \cdot \log^3 n)$ space, 2-pass dynamic graph streaming algorithm that returns a $(\alpha+2)(1+\epsilon)$-approximation of $\mu(G)$ with probability at least $1 - 1/n$.*

Proof. Note that $\mathbb{E}[X] = t \cdot w(G)/m$. Since each edge sampled is drawn independently and each $0 \leq w_{u,v} \leq 1$, we can apply the Chernoff bound to conclude:

$$\Pr\left[|W - w(G)| \geq \epsilon \cdot w(G)\right] = \Pr\left[\left|X - \frac{w(G) \cdot t}{m}\right| \geq \epsilon \cdot \frac{w(G) \cdot t}{m}\right]$$
$$\leq 2 \cdot \exp\left(-\epsilon^2 \cdot \frac{w(G) \cdot t}{3m}\right) .$$

Theorem 5 implies $w(G) \geq k/(\alpha + 1)$. Hence, setting $t = 3m\epsilon^{-2}(\alpha + 1) \cdot \log(2n)/k$ ensures $\Pr[|W - w(G)| \geq \epsilon \cdot w(G)] \leq 1/n$. The result follows from Theorem 5. The space bound follows from the space complexity of edge sampling (Theorem 4). ◀

We can then combine the approach above with the result in Theorem 3 to yield a two-pass algorithm whose space complexity does not depend on upper or lower bounds on $\mu(G)$. Specifically:

1. We run the algorithm in Theorem 3 with $p = 2$ passes, and $k = m^{2/5}$, $\epsilon = 1$. This uses $O(\alpha^2 \cdot n^{3/5} \cdot \log n)$ space and returns a 2-approximation if $\mu(G) \leq m^{2/5}$.
2. In parallel, we run Algorithm 2 with $t = 12m\epsilon^{-2}(\alpha + 1)\log(2n)/m^{2/5}$. This uses $O(\epsilon^{-2} \cdot \alpha \cdot m^{3/5} \cdot \log^3 n)$ space and returns an $(\alpha + 2)(1 + \epsilon)$-approximation if $w(G) \geq m^{2/5}/(\alpha + 1) \cdot 1/4$.
3. To determine whether to output the result from the first algorithm or the second algorithm, we consider the variable $X = \sum_{e \in E'} w_e$ defined in Algorithm 2. Note $\mathbb{E}[X] = w(G) \cdot t/m$ and by an application of Chernoff bounds, if $w(G) \geq m^{2/5}/(\alpha + 1)$, then we have:

   $$\Pr[X \leq \theta] \leq \exp(-m^{2/5}/(\alpha+1) \cdot t/m \cdot (1/3)) \leq 1/n \text{ , where } \theta = m^{2/5}/(\alpha+1) \cdot (1/2) \cdot t/m$$

   whereas if $w(G) \leq m^{2/5}/(\alpha + 1) \cdot (1/4)$, then we have:

   $$\Pr[X \geq \theta] \leq \exp(-m^{2/5}/(\alpha + 1) \cdot t/m \cdot (1/12)) \leq 1/n \ .$$

   Consider returning the result from the first algorithm if $X < \theta$, and the result from the second algorithm otherwise. If $w(G) \leq m^{2/5}/(\alpha + 1) \cdot 1/4$ then (a) with probability $1 - 1/n$ we return the output of the first algorithm, and (b) $\mu(G) \leq m^{2/5}$ by appealing to Theorem 5. Hence, we achieve a 2-approximation with high probability. If $w(G) \geq m^{2/5}/(\alpha+1)$, then with probability $1-1/n$ we return the output of the second algorithm and hence we achieve a $(\alpha + 2)(1 + \epsilon)$-approximation. If $(1/4) \cdot m^{2/5}/(\alpha + 1) < w(G) < m^{2/5}/(\alpha+1)$, then the approximation factor from either algorithm is at most $(\alpha+2)(1+\epsilon)$.

▶ **Theorem 2.** *There exists a two-pass algorithm using $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{3/5} \cdot \log^3 n)$ space that returns an $(\alpha + 2)(1 + \epsilon)$-approximation for* MSE *with high probability.*

Algorithm 3 A 3-pass algorithm for approximating matching size.

1. *First Pass:* Use CountMin Sketch with $O(\alpha \cdot n^{1/2} \cdot \log n)$ space to find approximations of all degrees, such that with high probability, for all $u \in V$:

$$\deg(u) \le \widetilde{\deg}(u) \le \deg(u) + \sqrt{n}/2 .$$

Let $H = \{v : \widetilde{\deg}(u) \ge \sqrt{n}\}$, and note that H contains all vertices with degree at least $\sqrt{n}$ and no vertices with degree strictly less than $\sqrt{n}/2$. Let E_H be the set of edges incident to a vertex in H.

2. *Second Pass:* Let $G_L = (V \setminus H, E_L)$ be the graph formed by removing all vertices in H.
   - Compute m_L, the number of edges in G_L.
   - Sample $t = 3\epsilon^{-2} \cdot n^{1/2} \cdot \ln(2n)$ edges E'_L from G_L with replacement via ℓ_0-sampling. Note that E'_L could be a multiset.
   - For each $v \in H$, pick $O(\alpha/\epsilon)$ incident edges. Let E'_H be the chosen edges.
3. *Third Pass:* For each edge $(u, v) \in E'_L \cup E'_H$, compute

$$w'_{u,v} = \min(1/\deg'(u), 1/\deg'(v), 1/(\alpha + 1)) ,$$

where $\deg'(u)$ is the number of incident edges in $E'_H \cup E_L$

4. *Output:* $w_1 + w_2$ where $w_1 = \sum_{e \in E'_H} w'_e$ and $w_2 = \frac{m_L}{t} \sum_{e \in E'_L} w'_e$.

3.3 $O(\alpha)$-Approximation, Three Passes, and $\tilde{O}(m^{1/2})$ Space

Consider Algorithm 3. The analysis proceeds as follows. After the first pass, we have partitioned the vertices into H and $V \setminus H$, such that all vertices in H have degree at least $\sqrt{n}/2$, and all vertices in $V \setminus H$ have degree at most $\sqrt{n}$. We will argue via Theorem 6 that maintaining a few edges incident to each vertex in H (these edges are called E'_H in the algorithm) decreases the size of the maximum matching by at most a factor of $(1 - \epsilon)$. We then approximate the matching in the resulting graph via fractional matchings. Let A be the weight of the fractional matching on edges incident to H, and B be the weight of the other edges. We can compute A exactly (this will be returned as w_1), and we can estimate B by sampling edges that are not incident to vertices in H. The next lemma shows that our estimate for B is sufficiently accurate. The proof exploits that the weights of the edges contributing to B are all at least $1/\sqrt{n}$.

▶ **Lemma 10.** $\Pr[|w_2 - B| \ge \epsilon B] \le 1/n$, *where* $B = \sum_{e \in E_L} w'_e$.

Proof. Let Δ_L be the maximum degree of a vertex in $V \setminus H$. The definition of H ensures that $\Delta_L < \sqrt{n}$. The weight of each edge in E_L is between $1/\Delta_L$ and 1, and the average is B/m_L. Let X be the sum of w'_e for each $e \in E'_L$. Hence:

$$\mathbb{E}[X] = tB/m_L \ge t/\Delta_L .$$

By an application of the Chernoff bound, we have:

$$\Pr[|X - \mathbb{E}[X]| \ge \epsilon \cdot \mathbb{E}[X]] \le 2 \cdot \exp\left(-\epsilon^2 \cdot \frac{\mathbb{E}[X]}{3}\right) \le 2 \cdot \exp\left(-\epsilon^2 \cdot \frac{t}{3\Delta_L}\right) .$$

Hence, setting $t = 3\epsilon^{-2} \cdot n^{1/2} \cdot \ln(2n)$ makes the failure probability $1/n$. ◀

▶ **Theorem 1.** *There exists a three-pass algorithm using* $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{1/2} \cdot \log^3 n)$ *space that returns an* $(\alpha + 2)(1 + \epsilon)$*-approximation for* MSE *with high probability.*

Proof. By Theorem 6, we have $\mu(E)/(1+\epsilon) \le \mu(E'_H \cup E_L) \le \mu(E)$. Hence, by Theorem 5:

$$\mu(E)/(1+\epsilon) \le (\alpha+1) \cdot w'(E'_H \cup E_L) \le (\alpha+2) \cdot \mu(E) \ .$$

Note that $w'(E'_H \cup E'_L) = w_1 + w_2$, and $w(E'_H \cup E_L) = w_1 + B$. But, by Lemma 10, with probability at least $1 - 1/n$, we have $(1-\epsilon) \cdot B \le w_2 \le (1+\epsilon) \cdot B$. Therefore:

$$\mu(E) \cdot \frac{1-\epsilon}{1+\epsilon} \le w_1 + w_2 \le (\alpha+2) \cdot (1+\epsilon) \cdot \mu(E) \ .$$

Reparameterizing $\epsilon \leftarrow \epsilon/4$ gives the claimed result. The space used by the algorithm is $O(|H| \cdot \alpha/\epsilon + \alpha \cdot n^{1/2} \log n + t \log^2 n)$. Note that $|H| \le 2m/\sqrt{n}$, and so the space is as claimed. ◀

4 Estimating Rank of Sparse Matrices

In this section, we consider the problem of estimating the rank of a matrix. It is well-known (e.g., see [35]) that the rank of the Tutte matrix[4] of a graph $G = (V, E)$ is exactly $2 \cdot \mu(G)$. It is therefore natural to look for other connections between rank and matching size.

Given an arbitrary matrix A, we next define a bipartite graph G_A that captures the structure of the nonzero entries of A.

▶ **Definition 11.** *Given an arbitrary $n_1 \times n_2$ matrix A define a bipartite graph $G_A = (L, R, E)$ where $L = [n_1], R = [n_2]$ and $(i, j) \in E$ if $A[i, j] \neq 0$. Note that a matching in G_A corresponds to a set of nonzero entries in M such that no two of these entries fall in the same column or row of A.*

Unfortunately it is too much to hope that $\mathsf{rank}(A)$ and $\mu(G_A)$ are always closely related. To see this, suppose that A is an $n \times n$ matrix of all 1s. Then, $\mathsf{rank}(A) = 1$, but $\mu(G_A) = n$. However, we show a significantly closer relationship for a certain family of sparse matrices which, by analogy to graph terminology, we call α-arboricity matrices.

▶ **Definition 12.** *A matrix is said to have arboricity α if every $t \times t$ submatrix has at most $\alpha \cdot t$ nonzero entries.*

Note that the class of matrices of arboricity α is much larger than the class of matrices which have a bounded number of nonzero entries in every row and column. However, it is more restrictive than bounding the number of nonzero elements; an $n \times n$ matrix with at most $\alpha \cdot n$ nonzero entries does not necessarily have arboricity α. For example, let A be an $n \times n$ matrix which has all zeros, except for a $\sqrt{\alpha \cdot n} \times \sqrt{\alpha \cdot n}$ submatrix of 1s; note that A has only $\alpha \cdot n$ nonzero entries, but does not have arboricity β for any $\beta < \sqrt{\alpha \cdot n}$.

▷ Claim 13. If G is a graph with arboricity α, its adjacency matrix A_G has arboricity at most 2α.

Proof. Consider an arbitrary $t \times t$ submatrix A'_G of A_G, consisting of t rows and t columns from A_G. Suppose the indices of the *common* rows and columns are $I = \{i_1, \ldots, i_s\}$. In addition, A'_G has rows $r_{j_1}, \ldots, r_{j_{s'}}$ and columns $c_{j'_1}, \ldots, c_{j'_{s'}}$ from A_G, such that $J = \{j_1, \ldots, j_{s'}\}$ and $J' = \{j'_1, \ldots, j'_{s'}\}$ are disjoint. Note that I, J, and J' correspond to disjoint subsets of

[4] Recall that the *Tutte matrix* T_G of a graph $G = (V, E)$ is the skew-symmetric matrix where $T[i, j] = 0$ if $(i, j) \notin E$; $T[i, j] = x_{i,j}$ if $i < j$ and $(i, j) \in E$; and $T[i, j] = -x_{i,j}$ if $i > j$ and $(i, j) \in E$.

vertices of G, and so A'_G is a submatrix of the adjacency matrix for the induced subgraph $G[I \cup J \cup J']$, which has at most $2\alpha \cdot (|I|+|J|+|J'|)$ nonzero entries. Each edge between I and J (or between I and J') corresponds to exactly one nonzero entry in the submatrix. None of the edges in $G[J \cup J']$ shows up. Finally, each edge in $G[I]$ corresponds to two nonzero entries (corresponding to the standard adjacency submatrix of A_G). Altogether, counting the weights of each of these relevant pairwise disjoint submatrices separately, this gives us $\alpha \cdot (2|I| + |J| + |J'|)$ nonzero entries in A'_G, which is at most $\alpha(|I| + |J| + |I| + |J'|) = 2\alpha \cdot t$, as claimed. ◁

▶ **Theorem 14.** *For any matrix A, we have $\mu(G_A)/\alpha \leq \mathsf{rank}(A) \leq \mu(G_A)$.*

Proof. To prove $\mu(G_A) \geq \mathsf{rank}(A)$, let T be the Tutte matrix of G_A. Then, from [35], we know that $\mathsf{rank}(T) = 2 \cdot \mu(G_A)$. Furthermore, $\mathsf{rank}(T) \geq 2 \cdot \mathsf{rank}(A)$. Hence, $\mathsf{rank}(A) \leq \mu(G_A)$, as claimed.

To prove $\mu(G_A)/\alpha \leq \mathsf{rank}(A)$, note that we may permute the rows and columns of A such that the first $\mu(G_A)$ diagonal entries of A are all nonzero. Let F be the top left $k \times k$ submatrix where $k = \mu(G_A)$. Note $\mathsf{rank}(F) \leq \mathsf{rank}(A)$. To lower bound the rank of F, first note that the total number of non-diagonal entries that are nonzero is at most $(\alpha - 1) \cdot k$, and so, at least one of F or F^T has $(\alpha - 1) \cdot k/2$ or fewer nonzero entries below the diagonal. Assume this is the case for F (if not, we can apply the rest of the argument to F^T rather than F). We will show that F contains a $(k/\alpha) \times (k/\alpha)$ principal submatrix where all the diagonal entries are nonzero and all entries below the diagonal are nonzero.

The process for finding this submatrix is as follows. We maintain a set $D \subseteq [k]$, where $j \in D$ means that the jth row and jth column will *not* be included in the submatrix. D is initially empty. We say that the jth column and row are *marked* if $j \in D$. Let nz_i be the total number of nonzero entries in the ith row and column of this submatrix that are below the diagonal and are not in marked rows/columns , i.e.:

$$nz_i = |S_i|, \text{ where } S_i = \{j < i : A[i,j] \neq 0, j \notin D\} \cup \{j > i : A[j,i] \neq 0, j \notin D\} \ .$$

Note that $\sum_{i \notin D} nz_i \leq (\alpha - 1) \cdot (k - |D|)$, because F has arboricity α, and the sum counts each element twice. Hence, $\min(nz_i) \leq (\alpha - 1)$. Let $i^* = \mathrm{argmin}_{i:nz_i>0} nz_i$ and update $D \leftarrow D \cup S_{i^*}$. We repeat this process until all nonzero entries under the diagonal are in marked rows/columns. In each iteration, $|D|$ increases by at most $\alpha - 1$, but there is at least one more value i such that $nz_i = 0$. Hence, at the end of the algorithm, we have $k - |D| \geq k/\alpha$. ◀

Therefore, all our matching algorithms (and all previous results on MSE) also give algorithms for estimating the rank of low-arboricity matrices, with an additional multiplicative factor of α. Note that for rank approximation, we assume the dynamic model where at each time step, some entry of the matrix is set to a nonzero value (if it was currently zero) or set to zero (if it was currently nonzero).

The following example shows that the above bound is tight up to constants. And furthermore, any approximation via $w(G)$ loses an $O(\alpha^2)$ factor.

▶ **Example 15.** Let A and B be an $n \times n$ binary matrices, where

$$A[i,j] = \begin{cases} 1 & \text{if } i \leq \alpha \text{ or } j \leq \alpha \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad B[i,j] = \begin{cases} 1 & \text{if } i \leq \alpha \text{ or } j \leq \alpha \text{ or } i = j \\ 0 & \text{otherwise.} \end{cases}$$

First, note that $\mathsf{rank}(A) = 2, \mathsf{rank}(B) = n - \alpha + 1$, $\mu(G_A) = 2\alpha$, and $\mu(G_B) = n$. This establishes that the quantity $\mathsf{rank}(M)/\mu(G_M)$ can vary by an $\Omega(\alpha)$ factor.

$$w(G_A) = \frac{\alpha^2}{n} + \frac{2(n-\alpha)\alpha}{n} \approx 2\alpha \qquad \text{and} \qquad w(G_B) = \frac{\alpha^2}{n} + \frac{2(n-\alpha)\alpha}{n} + \frac{(n-\alpha)}{\alpha+1} \approx \frac{n}{\alpha} .$$

Hence, $\mathsf{rank}(M)/w(G_M)$ can vary by an $\Omega(\alpha^2)$ factor.

We end by noting that for a matrix A, the value of $\mu(G_A)$ does not depend on the the values of the nonzero entries in A. This immediately implies the following curious corollary.

▶ **Corollary 16.** *Changing the nonzero values in a matrix of arboricity α can change its rank by at most a factor of α.*

5 Conclusion

In this paper, we gave new multi-pass streaming algorithms for MSE in dynamic graph streams on graphs of arboricity α. As our main result, we showed that an $O(\alpha)$-approximation can be achieved in three passes with space $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{1/2} \cdot \log n)$, and we also gave a two-pass algorithm with a similar approximation guarantee that uses space $O(\epsilon^{-2} \cdot \alpha^2 \cdot n^{3/5} \cdot \log n)$. Furthermore, we designed a multi-pass algorithm with approximation factor $(1 + \epsilon)$ that operates based on an upper bound k on the maximum matching size. For example, it can give an $O(\log \log n)$-pass algorithm that uses space $O(\epsilon^{-1} \cdot \alpha^2 \cdot k \cdot \log n)$. Lastly, we introduced the notion of low-arboricity matrices and argued that matching algorithms for low-arboricity graphs can be used to approximate the rank of low-arboricity matrices with an $O(\alpha)$ loss in the approximation factor.

We conclude with two open problems. First, we are particularly intrigued by whether the $\sqrt{n}$-barrier established by Assadi et al. [7] for one-pass algorithms persists when multiple passes over the input are allowed. For instance, is there a constant pass algorithm with approximation factor $O(\alpha)$, whose space dependency on n is $o(\sqrt{n})$? Second, can we tighten the bounds in the one-pass setting?

References

1 Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. *Algorithmica*, 83(7):1980–2017, 2021. doi:10.1007/S00453-021-00816-9.

2 Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, 2013. doi:10.1016/j.ic.2012.10.006.

3 Sepehr Assadi. A two-pass (conditional) lower bound for semi-streaming maximum matching. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 708–742. SIAM, 2022. doi:10.1137/1.9781611977073.32.

4 Sepehr Assadi. A simple (1 - ϵ)-approximation semi-streaming algorithm for maximum (weighted) matching. In Merav Parter and Seth Pettie, editors, *2024 Symposium on Simplicity in Algorithms, SOSA 2024, Alexandria, VA, USA, January 8-10, 2024*, pages 337–354. SIAM, 2024. doi:10.1137/1.9781611977936.31.

5 Sepehr Assadi, Soheil Behnezhad, Sanjeev Khanna, and Huan Li. On regularity lemma and barriers in streaming and dynamic matching. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 131–144. ACM, 2023. doi:10.1145/3564246.3585110.

6 Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 627–669. SIAM, 2022. doi:10.1137/1.9781611977073.29.

7 Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1723–1742. SIAM, 2017. doi:10.1137/1.9781611974782.113.

8 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 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016. doi:10.1137/1.9781611974331.ch93.

9 Sepehr Assadi, Christian Konrad, Kheeran K. Naidu, and Janani Sundaresan. O(log log n) passes is optimal for semi-streaming maximal independent set. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 847–858. ACM, 2024. doi:10.1145/3618260.3649763.

10 Sepehr Assadi and Vihan Shah. An asymptotically optimal algorithm for maximum matching in dynamic streams. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 9:1–9:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ITCS.2022.9.

11 Sepehr Assadi and Janani Sundaresan. Hidden permutations to the rescue: Multi-pass streaming lower bounds for approximate matchings. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 909–932. IEEE, 2023. doi:10.1109/FOCS57990.2023.00058.

12 Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 263–274, 2015. doi:10.1007/978-3-662-48350-3_23.

13 M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *ICALP*, pages 693–703, 2002.

14 Rajesh Hemant Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to dynamic graph streams. *CoRR*, abs/1505.01731, 2015. URL: http://arxiv.org/abs/1505.01731, arXiv:1505.01731.

15 Graham Cormode, Hossein Jowhari, Morteza Monemizadeh, and S. Muthukrishnan. The sparse awakens: Streaming algorithms for matching size estimation in sparse graphs. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, volume 87 of *LIPIcs*, pages 29:1–29:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.ESA.2017.29.

16 Graham Cormode and S. Muthukrishnan. Space efficient mining of multigraph streams. In *ACM Principles of Database Systems*, pages 271–282, 2005. doi:10.1145/1065167.1065201.

17 Michael Crouch and Daniel S. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 96–104, 2014. doi:10.4230/LIPIcs.APPROX-RANDOM.2014.96.

18 Michael S. Crouch, Andrew McGregor, and Daniel Stubbs. Dynamic graphs in the sliding-window model. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 337–348, 2013. doi:10.1007/978-3-642-40450-4_29.

19 Leah Epstein, Asaf Levin, Julián Mestre, and Danny Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math.*, 25(3):1251–1265, 2011. doi:10.1137/100801901.

20 Hossein Esfandiari, Mohammad Taghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1217–1233, 2015. doi:10.1137/1.9781611973730.81.

21 J. Feigenbaum, S. Kannan, McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Theoretical Computer Science*, 348(2-3):207–216, 2005. doi:10.1016/J.TCS.2005.09.013.

22 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2):207–216, 2005. doi:10.1016/j.tcs.2005.09.013.

23 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. URL: http://portal.acm.org/citation.cfm?id=2095157&CFID=63838676&CFTOKEN=79617016, doi:10.1137/1.9781611973099.41.

24 Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, Palo Alto, California, USA, 5-7 June, 2013*, pages 287–298, 2013. doi:10.1109/CCC.2013.37.

25 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.

26 Hossein Jowhari. An estimator for matching size in low arboricity graphs with two applications. *J. Comb. Optim.*, 45(1):21, 2023. doi:10.1007/S10878-022-00929-Z.

27 Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '11, pages 49–58, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1989284.1989289.

28 Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013. doi:10.1137/1.9781611973105.121.

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

30 Christian Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015. doi:10.1007/978-3-662-48350-3_70.

31 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *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*, pages 231–242, 2012. doi:10.1007/978-3-642-32512-0_20.

32 Christian Konrad and Kheeran K. Naidu. On two-pass streaming algorithms for maximum bipartite matching. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 19:1–19:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.APPROX/RANDOM.2021.19`.

33 Christian Konrad and Kheeran K. Naidu. An unconditional lower bound for two-pass streaming algorithms for maximum matching approximation. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 2881–2899. SIAM, 2024. `doi:10.1137/1.9781611977912.102`.

34 Christian Konrad, Kheeran K. Naidu, and Arun Steward. Maximum matching via maximal matching queries. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 41:1–41:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPICS.STACS.2023.41`.

35 László Lovász. On determinants, matchings, and random algorithms. In Lothar Budach, editor, *Fundamentals of Computation Theory, FCT 1979, Proceedings of the Conference on Algebraic, Arthmetic, and Categorial Methods in Computation Theory, Berlin/Wendisch-Rietz, Germany, September 17-21, 1979*, pages 565–574. Akademie-Verlag, Berlin, 1979.

36 Andrew McGregor. Finding graph matchings in data streams. *APPROX-RANDOM*, pages 170–181, 2005. `doi:10.1007/11538462_15`.

37 Andrew McGregor and Sofya Vorotnikova. Planar matching in streams revisited. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, volume 60 of *LIPIcs*, pages 17:1–17:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPICS.APPROX-RANDOM.2016.17`.

38 Andrew McGregor and Sofya Vorotnikova. A simple, space-efficient, streaming algorithm for matchings in low arboricity graphs. In Raimund Seidel, editor, *1st Symposium on Simplicity in Algorithms, SOSA 2018, January 7-10, 2018, New Orleans, LA, USA*, volume 61 of *OASIcs*, pages 14:1–14:4. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/OASICS.SOSA.2018.14`.

39 C. St.J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, s1-36(1):445–450, 1961. `doi:10.1112/jlms/s1-36.1.445`.

40 Shay Solomon. Local algorithms for bounded degree sparsifiers in sparse graphs. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 52:1–52:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPICS.ITCS.2018.52`.

41 Mariano Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, 2012. `doi:10.1007/s00453-010-9438-5`.