Structural Results on Matching Estimation with Applications to Streaming*

Marc Bury, Elena Grigorescu, Andrew McGregor, Morteza Monemizadeh,

Chris Schwiegelshohn, Sofya Vorotnikova, Samson Zhou

We study the problem of estimating the size of a matching when the graph is revealed in a streaming fashion. Our results are multifold:

1. We give a tight structural result relating the size of a maximum matching to the *arboricity* of a graph, which has been one of the most studied graph parameters for matching algorithms in data streams.
2. We further show that the weight of a maximum weighted matching can be efficiently estimated by augmenting any routine for estimating the size of an unweighted matching. Namely, given an algorithm for computing a λ-approximation in the unweighted case, we obtain a $2(1+\varepsilon)\cdot\lambda$ approximation for the weighted case, while only incurring a multiplicative logarithmic factor in the space bounds. The algorithm is implementable in any streaming model, including *dynamic* streams.
3. We also investigate algebraic aspects of computing matchings in data streams, by proposing new algorithms and lower bounds based on analyzing the rank of the *Tutte-matrix* of the graph. In particular, we present an algorithm determining whether there exists a matching of size k using k^2polylog n space, where n is the number of nodes in the graph. We also show a lower bound of $\Omega(n^{1-\varepsilon})$ space for small approximation factors to the rank of a matrix in *insertion-only* streams.

1 Introduction

In the graph streaming model, introduced by Henzinger et al. [24], we are given a sequence of updates to the adjacency matrix, and we aim to solve a graph problem, using as little space as possible. Much of the recent work has typically focused on the semi-streaming model [18], where we are allowed to use a single pass over the data, while using only $O(n\cdot\text{polylog}\ n)$ space, where n is the number of nodes in the graph.

In particular, computing matchings is arguably the most studied problem in graph streaming models. However, while a lot is already well-understood about the complexity of this problem, many intriguing questions remain still open.

* This work is composed of results appearing in [7, 22, 36].

M. B. and C. S. were supported by Deutsche Forschungsgemeinschaft within the Collaborative Research Center SFB 876, project A2

A. M. and S. V. were supported by NSF CAREER Award CCF-0953754 and CCF-1320719 and a Google Faculty Research Award.

E. G. and S. Z were supported in part by NSF CCF-1649515.

To bypass the $\Omega(n)$ lower bound required to store a matching, recent research has begun to focus on only approximating the size of matchings, resulting in several algorithms with sublinear-space bounds, even with respect to the number of nodes. We continue this line of work, and present several results for estimating matchings in weighted and unweighted graphs, using only $o(n)$ space.

1.1 Our Contribution

Table 1 contains a succinct overview of our specific results and the most relevant previous work.

Reference	Graph class	Streaming model	Approx. factor	Space
[27]	General	Random	$\text{polylog}(n)$	$\text{polylog}(n)$
[17]	Trees	Insert	$2+\varepsilon$	$\tilde{O}(\sqrt{n})$
[17]	Constant arboricity	Insert	$5\alpha+9$	$\tilde{O}(\alpha n^{2/3})$
here	Constant arboricity	Insert	$\alpha+2$	$\tilde{O}(\alpha n^{2/3})$
[10]	Constant arboricity	Insert	$22.5\alpha+6$	$\tilde{O}(\alpha \cdot \text{polylog } n)$
here	Constant arboricity	Vertex-Arrival	$(\alpha+2)^2/2$	$O(\log n)$
here	Trees	Dynamic	$2+\varepsilon$	$O(\log^2 n)$
[8]	Constant arboricity	Dynamic	$22.5\alpha+6$	$\tilde{O}(\alpha n^{4/5})$
here	Constant arboricity	Dynamic	$\alpha+2$	$\tilde{O}(\alpha n^{4/5})$
[17]	Forests	Insert	$\frac{3}{2}-\varepsilon$	$\Omega(\sqrt{n})$
here	General	Insert	$1+\frac{2\varepsilon}{3-2\varepsilon}$	$\Omega\left(n^{1-\varepsilon}\right)$

Table 1: Results for estimating the size of a maximum matching in data streams. α is the arboricity of a graph, e.g., 3 for planar graphs. $\tilde{O}(f(n))$ hides factors polylogarithmic in $f(n)$. We also suppressed $(1+\varepsilon)$ multiplicative factors in some approximation ratios and polynomial dependencies on ε^{-1} in the space bounds. All upper bounds can be extended to weighted matching at an additional loss of a multiplicative factor of 2 in the approximation ratio.

In what follows, we discuss these results in more detail.

Structural Results. Most previous papers on estimating matching sizes in streams focus on classes of sparse graphs, either by limiting the degree of each node [27], or more generally by assuming bounded arboricity [13, 17]. The *arboricity* of a graph $G(V,E)$ is defined as $\alpha := \max_{U \subseteq V} \left\lceil \frac{|E(U)|}{|U|-1} \right\rceil$. Equivalently, the arboricity can be defined as the minimum number of forests into which the edges of the graph can be decomposed. Our main result is a structural theorem relating the matching size to the arboricity:

Theorem 1. *Let* $\text{match}(G)$ *be the size of the maximum cardinality matching in* $G(V,E)$*. For an edge* $e=\{u,v\} \in E$ *define* $x_e = \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right)$*. Then,*

$$\text{match}(G) \leq (\alpha+1)\sum_{e \in E} x_e \leq (\alpha+2)\,\text{match}(G)$$

Therefore, estimating $\sum_{e\in E} x_e$ allows us to estimate the matching size while losing an $(\alpha+2)$ factor. We also show how to estimate this sum, provided it is large enough. If the sum is too small to be efficiently estimated, the matching is also small. In insertion-only streams, we can greedily maintain a maximal matching, which was also done by Esfandiari et al. [17], resulting in a $(\alpha+2)(1+\varepsilon)$ approximation using $\tilde{O}(\varepsilon^{-2}\alpha n^{2/3})$ bits of space.

Reduction of Weighted to Unweighted Matching Estimation. Building on similar approaches for parallel algorithms [40], and on approximate matching computation in streams [12], we give a reduction from unweighted to weighted matching estimation as follows:

Theorem 2. *Given a λ-approximate estimation using S space with failure probability δ, there exists an $2(1+\varepsilon)\lambda$-approximate estimation algorithm for the weighted matching problem with weight range $[1, W]$ using $O(S \cdot \log W \cdot \varepsilon^{-1})$ space with failure probability $\delta / \log_{1+\varepsilon} W$.*

This reduction applies to any streaming model including the dynamic stream model.

Algebraic Techniques for Computing Matchings in Streams. Finally, we introduce algebraic techniques for analyzing matching problems in data streams. Like most algebraic matching algorithms, we focus on properties of the *Tutte-matrix* of a graph. Tutte [39] famously showed that a graph has a perfect matching if and only if the maximum rank of the Tutte-matrix is n. This was later generalized by Lovász [33], who showed that the maximum possible rank of the Tutte-matrix is exactly twice the maximum matching size. Moreover, he showed that a suitable random assignment results in the maximum rank with high probability.

This has two consequences for streaming algorithms, namely that (1) any approximation algorithm for the rank of a matrix can be used to approximate the matching size of a graph, and that (2) lower bounds for estimation tasks of the matching size can be used to derive lower bounds for the rank of a matrix.

Using rank-preserving sketches applied to the Tutte matrix, we then obtain the following upper bound:

Theorem 3. *Let $G(V, E)$ be an arbitrary graph. Then there exists a dynamic streaming algorithm that either (1) outputs k if $\operatorname{match}(G) \geq k$ or (2) outputs $\operatorname{match}(G)$ otherwise. The algorithm uses $O(k^2 \log k \log n)$ bits of space.*

Theorem 3 was also reproved by Chitnis et al. [8] using sampling-based approaches. Combining this result with our estimation techniques for bounded arboricity matchings, we obtain a $(\alpha+2)(1+\varepsilon)$ factor estimation of the matching using $\tilde{O}(\varepsilon^{-2}\alpha n^{4/5})$ space, which is one of the few non-trivial results for dynamic graph streams using sublinear space.

For lower bounds, we show that by approximating the matching to a sufficiently good factor, we can solve hard instances of the Boolean Hidden Hypermatching problem. Specifically,

Theorem 4 (informal version)**.** *Any 1-pass streaming algorithm approximating the size of the maximum matching matching up to an $(1 + O(\varepsilon))$ factor requires $\Omega(n^{1-\varepsilon})$ bits of space.*

Using the aforementioned connection between matching size and rank of a matrix established by the Tutte matrix, we also obtain an $\Omega(n^{1-\varepsilon})$ space bound for $1 + O(\varepsilon)$ approximating the rank of a matrix in data streams which also improves the $\Omega(\sqrt{n})$ bound by Li, Nguyen, and Woodruff [30] for linear sketches. It also gives an exponential separation between estimating the rank of a diagonal matrix which admits a $O(\log n)$ space algorithm by estimating the ℓ_0 norm of a vector, see for instance Kane et al. [25]

1.2 Related Work

Matching Computation Maintaining a 2-approximation to the maximum matching in an insertion-only stream can be straightforwardly done by greedily maintaining a maximal matching [18]. This remains the best algorithm discovered thus far and no single pass semi-streaming algorithm using $O(n \cdot \operatorname{polylog} n)$ space can do better than $e/(e-1)$, see Goel et al. [21] and Kapralov [26]. If the edges of an insertion only stream are assumed to arrive in random order as opposed to an adversarial order, Konrad et al. [29] were able to obtain a semi-streaming algorithm with an approximation

factor of 1.989. In sliding window streams, Crouch et al. [11] gave a $(3+\varepsilon)$-approximation. For dynamic streams, Assadi et al. [3] showed that any sketching algorithm computing a n^{ε} approximate matching requires $\Omega(n^{2-3\varepsilon})$ space, see also the earlier work by Konrad [28]. Since linear sketches can be used to obtain lower bounds of dynamic graphs [1], this result gives a lower bound for maintaining approximate matchings in dynamic graphs.

For weighted matching, a trickle of results sequentially improving on the approximation ratio have been recently published for insertion streams [18, 35, 15, 16, 41, 43, 12, 23].

Matching Estimation To bypass the natural $\Omega(n)$ bound required by any algorithm maintaining an approximate matching, recent research has begun to focus on only estimating the size of the maximum matching. In one of the few non-trivial graph streaming results using polylog n space, Kapralov et al. [27] obtained a polylogarithmic approximate estimate for randomly ordered streams. The remaining algorithms in this line of research focus on approximating matching sizes in graphs of bounded arboricity. Estimators relating the matching size to arboricity were first considered in the field of distributed computing by Czygrinow et al. [13] In a streaming setting this task was first addressed by Esfandiari et al. [17], who obtained a $(5\alpha+9)(1+\varepsilon)$ approximation using $\tilde{O}(\varepsilon^{-2}\alpha n^{2/3})$ bits of space in insertion only streams and $\tilde{O}(\varepsilon^{-2}\alpha n^{1/2})$ bits of space in random-order streams. The authors also gave a lower bound of $\Omega(\sqrt{n})$ for any approximation better than $\frac{3}{2}$. This was subsequently extended to a $\tilde{O}(\varepsilon^{-2}\alpha n^{4/5})$ space algorithm in dynamic streams indepedently by Chitnis et al. [8] and the initial publication of Bury and Schwiegelshohn [7]. Recently, Cormode et al. [10] gave an insertion-only algorithm with an approximation factor of $22.5\alpha+6$ and using only $\tilde{O}(\varepsilon^{-2}\alpha\log^2 n)$ space.

Schatten Norm Estimation The p Schatten norm of a matrix A is the ℓ_p norm of the vector containing the singular values of A. Taking the limit of $p\to 0$, the 0 Schatten norm corresponds to the rank. Most previous work focused on lower bounds for dynamic streams. Clarkson and Woodruff [9] obtained a $\Omega(k^2)$ lower bound for determining whether a matrix has rank at least k. Li et al. [30] showed a lower bound of $\Omega(\sqrt{n})$ for the target dimension of any linear sketch-based constant factor approximation of the rank and an $\Omega(n^2)$ target dimension for bi-linear sketching, see also later extensions and improvements for other Schatten norms in [32]. The related question of finding the largest eigenvalues of a matrix was investigated by Andoni and Nguyen [2], whose algorithm can also be used to solve the rank decision problem.

The only other result pertaining to insertion-only streams we are aware of is due to Li and Woodruff [31]. Their result extends upon our construction initially published in [7] by showing that any algorithm estimating certain classes of functions of singular values well enough can be used to solve a hard instance of Boolean Hidden Hypermatching. Specifically, they characterized functions for whose evaluation on a Tutte-matrix with randomly chosen entries is affected enough to be detected by a sufficiently small approximation factor.

1.3 Preliminaries

We use $\tilde{O}(f(n))$ to hide factors polylogarithmic in $f(n)$. We require that any randomized algorithm succeeds with at least probability $2/3$. Graphs are denoted by $G(V,E,w)$ where V is the set of n nodes, E is the set of edges and $w: E\to[1,W]$ is a weight function with maximum weight W. Our estimated value $\widehat{M}$ is a λ-approximation to the size of the maximum matching $\mathrm{match}(G)$ if $\widehat{M}\leq \mathrm{match}(G)\leq \lambda\widehat{M}$.

The Hamming norm of a vector x is defined as $\ell_0(x)=|\{i : x_i\neq 0\}|$. The singular value decomposition of a matrix by $A=U\Sigma V^T$ where $U,V\in\mathbb{R}^{n\times n}$ are orthogonal and Σ is diagonal. The rank of a matrix is the number of non-zero entries of Σ, or alternatively the Hamming norm

of the vector containing the singular values of A. The spectral norm $||A||_2$ is the largest entry of Σ.

There exists a rich body of work on algebraic aspects of matching, which is particularly relevant for this paper, usually based around the Tutte-matrix T of a graph $G(V, E)$ defined as

$$T_{i,j} = \begin{cases} x_{i,j} & \text{if } i > j \text{ and } (i,j) \in E \\ -x_{i,j} & \text{if } j > i \text{ and } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E, \end{cases}$$

where $x_{i,j}$ are indeterminates. In his seminal paper, Tutte [39] showed that a graph contains a *perfect matching*, i.e., a matching of size $n/2$ if and only if for some choice of indeterminates the determinant of T is nonzero. This was later generalized by Lovász [33] to arbitrary matching size as follows (see also Rabin and Vazirani [37] for an alternative proof).

Theorem 5 (Lovász [33])**.** *Let $G = (V, E)$ be a graph with a maximum matching M and Tutte matrix T_G. For an assignment $w \in \mathbb{R}^{|E|}$ to the indeterminates of T_G we denote the matrix by $T_G(w)$ where the indeterminates are replaced by the corresponding assignment in w. Then we have*

$$\max_w \{rank(T_G(w))\} = 2 \cdot |M|.$$

In order to calculate the maximum of the rank, Lovász [33] also showed that the rank of the matrix where the indeterminates are replaced by random numbers uniformly drawn from $\{1, \ldots, R\}$ is equal to $\max_w \{rank(T_G(w))\}$ with probability at least $1 - |E|/R$.

Theorem 6 (Lovász [33])**.** *Let $G = (V, E)$ be a graph and $r \in \mathbb{R}^{|E|}$ be a random vector where each coordinate is uniformly chosen from $\{1, \ldots, R\}$ with $R \geq |E|$. Then we have*

$$rank(T_G(r)) = \max_w \{rank(T_G(w))\}$$

with probability at least $1 - |E|/R$.

If we are interested in matchings of some smaller size k, it is sufficient to sample the coordinates using $O(k^2 \log k)$-wise random bits. This can be seen by considering the $2k$ by $2k$ submatrix containing the maximum matching and applying the aforementioned theorem by sampling integers uniformly from $\{1, \ldots, O(k^2)\}$, and then noting that linear independent vectors remain linear independent when further adding arbitrary coordinates. Since we require a random value for every edge, we can draw the random bits from a seed of length $O(k^2 \log k \log n)$.

2 Structural Results on Matching in Bounded Arboricity Graphs

2.1 Warm Up: Matching Sizes of Trees

Let $T = (V, E)$ be a tree with at least 3 nodes and let h_T be the number of internal nodes, i.e., nodes with degree greater than 1. It is easy to see that the matching size is bounded by h_T since every edge in a matching has to be incident to at least 1 internal node. A maximum matching is also lower bounded by $h_T/2$ which is a consequence of Hall's Theorem. Therefore, it suffices to estimate the number of internal nodes of a tree to approximate the maximum matching within $2 + \varepsilon$ factor which was also observed in [17]. Consider the degree vector $d \in \mathbb{R}^n$, such that $d_i = \deg(v_i) - 1$. Then v_i is an internal node if and only if $d_i \neq 0$. Thus

Lemma 1. *Let $T = (V, E)$ be a tree with at least 3 nodes. Then* $\text{match}(T)/\ell_0(d) \leq 2$.

2.2 Proof of Theorem 1

Define the fractional matching polytope for a graph G as:

$$\mathsf{FM}(G) = \{\mathbf{x} \in \mathbb{R}^E : x_e \geq 0 \text{ for all } e \in E, \sum_{e \in E: u \in e} x_e \leq 1 \text{ for all } u \in V\} .$$

We say any $\mathbf{x} \in \mathsf{FM}(G)$ is a fractional matching. The size of this fractional matching is $\sum_{e \in E} x_e$ and for a graph where edge e has weight w_e, the weight of the matching is $\sum_{e \in E} w_e x_e$. A standard result on fractional matching is that the maximum size of a fractional matching is at most a factor $3/2$ larger than the maximum size of an (integral) matching. We will also make use of the following lemma which is a simple corollary of Edmonds Matching Polytope theorem [14].

Lemma 2. *For $U \subseteq V$, let $G[U]$ denote the induced subgraph on U. Let $\mathbf{x} \in \mathsf{FM}(G)$ and suppose there exist $\lambda_3, \lambda_5, \lambda_7 \ldots$ such that*

$$\forall U \subseteq V \text{ where } |U| \in \{3, 5, 7, \ldots\} , \quad \sum_{e \in G[U]} x_e \leq \lambda_{|U|} \left(\frac{|U| - 1}{2} \right) .$$

Then for any edge weights $\{w_e\}_{e \in E}$,

$$\sum_{e \in E} w_e x_e \leq \max(1, \lambda_3, \lambda_5, \ldots) \operatorname{match}(G)$$

where $\operatorname{match}(G)$ *is the weight of the maximum weighted (integral) matching.*

Proof. By Edmonds theorem, $\operatorname{match}(G) = \max_{\mathbf{z} \in \mathsf{IM}(G)} \sum_e w_e z_e$ where

$$\mathsf{IM}(G) = \left\{ \mathbf{x} \in \mathbb{R}^E : x_e \geq 0 \text{ for all } e \in E, \sum_{e \in E: u \in e} x_e \leq 1 \text{ for all } u \in V, \right.$$
$$\left. \sum_{e \in G[U]} x_e \leq \left(\frac{|U| - 1}{2} \right) \text{ for all } U \subset V \text{ of odd size} \right\} .$$

But $\frac{\mathbf{x}}{\max(1, \lambda_3, \lambda_5, \ldots)} \in \mathsf{IM}(G)$ and so $\sum_{e \in E} w_e x_e \leq \max(1, \lambda_3, \lambda_5, \ldots) \operatorname{match}(G)$ as required. □

For the streaming applications we will be interested in fractional matchings that can be computed locally.

Definition 1. *For a given graph G, we say a fractional matching $\mathbf{x} \in \mathsf{FM}(G)$ is* local *if every x_e is only a function of the edges (and their weights in the case of a weighted graph) that share an end point with e.*

Now define $\mathbf{x} \in \mathbb{R}^E$ where for $e = \{u, v\} \in E$, we set

$$x_e = \min \left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha + 1} \right) .$$

The next two theorems show that $\mathbf{x}$ is a local fractional matching and

$$\frac{1}{\alpha + 1} \cdot \operatorname{match}(G) \leq \operatorname{score}(\mathbf{x}) \leq \frac{\alpha + 2}{\alpha + 1} \cdot \operatorname{match}(G)$$

where $\operatorname{score}(\mathbf{x}) = \sum_e x_e$. This proves Theorem 1 and we note that the upper bound can be improved slightly if α is even. In Section 3, we show that it is possible to efficiently estimate $\operatorname{score}(\mathbf{x})$ in the data stream model.

Theorem 7. $\mathbf{x} \in \mathsf{FM}$ *and*

$$\frac{\text{score}(\mathbf{x})}{\text{match}(G)} \leq \begin{cases} \frac{\alpha+2}{\alpha+1} & \text{if } \alpha \text{ odd} \\ \frac{\alpha+3}{\alpha+2} & \text{if } \alpha \text{ even} \end{cases} .$$

Furthermore, if G is bipartite then $\text{score}(\mathbf{x}) \leq \text{match}(G)$.

Proof. First note that $x_e \geq 0$ for each $e \in E$ and for any $u \in V$,

$$\sum_{e \in E: u \in e} x_e \leq \sum_{e \in E: u \in e} 1/\deg(u) = 1 .$$

and hence $\mathbf{x} \in \mathsf{FM}$. The bound for bipartite graphs follows because the maximum size of a fractional matching in a bipartite graph equals the maximum size of an integral matching. For the rest of the result, we appeal to Lemma 2. Since $\mathbf{x} \in \mathsf{FM}$, it is simple to show that $\mathbf{x}$ satisfies the conditions of the lemma with $\lambda_t \leq t/(t-1)$; this follows because $\sum_{e \in G[U]} x_e \leq |U|/2$ for any $\mathbf{x} \in \mathsf{FM}$. Furthermore, since there are at most $\binom{|U|}{2}$ edges in $G[U]$ and $x_e \leq 1/(\alpha+1)$ for all e,

$$\sum_{e \in G[U]} x_e \leq \binom{|U|}{2} \frac{1}{\alpha+1} = \frac{|U|-1}{2} \cdot \frac{|U|}{\alpha+1} .$$

Therefore, $\lambda_t \leq \min\left(t/(t-1), t/(\alpha+1)\right)$. Consequently,

$$\max_{t \text{ odd}} \lambda_t = \begin{cases} \frac{\alpha+2}{\alpha+1} & \text{if } \alpha \text{ odd} \\ \frac{\alpha+3}{\alpha+2} & \text{if } \alpha \text{ even} \end{cases} .$$

□

We next bound $\text{score}(\mathbf{x})$ in terms of the number of high degree nodes and edges that are not incident to high degree nodes. As observed in previous work, these two quantities can then easily be related the size of the maximum matching.

Theorem 8. *Let h be the number of "heavy" nodes with degree at least $\alpha+2$ and s be the number of "shallow" edges whose endpoints are both not heavy. Then,*

$$\text{score}(\mathbf{x}) \geq 2h/(\alpha+2) + s/(\alpha+1) .$$

Furthermore, $\text{match}(G) \leq (\alpha+1)\,\text{score}(\mathbf{x})$.

Proof. Let d_i be the degree of node i and assume $d_1 \geq d_2 \geq d_3 \geq \ldots$. Let $b_i = |\{j < i : \{i,j\} \in E\}|$ and $c_i = |\{i < j : \{i,j\} \in E\}|$, i.e., the number of neighbors of node i that have higher or lower degree respectively than node i where ties are broken by the ordering supposed in the above line. Consider labeling an edge e with weight x_e where we first label edges incident to node 1, then the (remaining unlabeled) edges incident to node 2, etc. Then $c_1 = d_1$ edges get labeled with $\min(1/d_1, 1/(\alpha+1))$, c_2 edges get labeled with $\min(1/d_2, 1/(\alpha+1))$, c_3 edges get labeled with $\min(1/d_3, 1/(\alpha+1))$ etc. Let $\theta = \alpha+2$, then

$$\begin{aligned} \text{score}(\mathbf{x}) &= \sum_i c_i \min(1/d_i, 1/(\alpha+1)) \\ &= \sum_{i: d_i \geq \theta} c_i/d_i + \sum_{i: d_i \leq \theta-1} c_i/(\alpha+1) \\ &= h - \sum_{i: d_i \geq \theta} b_i/d_i + \sum_{i: d_i \leq \theta-1} c_i/(\alpha+1) \\ &\geq h - (\sum_{i: d_i \geq \theta} b_i)/\theta + (\sum_{i: d_i \leq \theta-1} c_i)/(\alpha+1) \end{aligned}$$

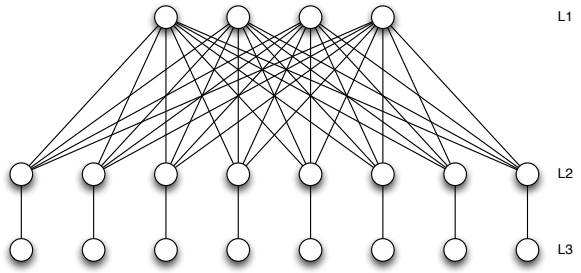

Figure 1: A tight example for Theorem 8. Let L_1 consist of α nodes whereas L_2 and L_2 consist of $n \gg \alpha$ nodes. The edges are a complete bipartite graph of L_1 and L_2 and a matching between L_2 and L_3. Then $\text{score}(\mathbf{x}) = \alpha n \times 1/n + n \times 1/(\alpha+1)$ and $\text{match}(G) = n$. Hence $\text{match}(G)/\text{score}(\mathbf{x})$ tends to $\alpha + 1$ as n tends to infinity.

Note that $\sum_{i:d_i \geq \theta} b_i$ is the number of edges in the induced subgraph on heavy nodes. This is at most αh because these edges in this induced subgraph can be partitioned into at most α forests. Similarly, $\sum_{i:d_i \leq \theta-1} c_i$ is the number of shallow edges. Therefore

$$\text{score}(\mathbf{x}) \geq h(1 - \alpha/\theta) + s/(\alpha+1) = 2h/(\alpha+2) + s/(\alpha+1)$$

as required. Note that $h + s \geq \text{match}(G)$ because every edge in a matching is either shallow or has at least one heavy node as an endpoint. Therefore

$$\text{score}(\mathbf{x}) \geq (h+s)/(\alpha+1) \geq \text{match}(G)/(\alpha+1) \ .$$

□

See Figure 1 for an example that shows that the above theorem is tight.

2.3 Structural Result for Weighted Graphs

In this section we show how to find a good local fractional matching for weighted graphs. It does not improve upon the bounds given by the unweighted to weighted reduction of Section 4. However, we think the structural result is interesting and could be useful in other computational models.

Define $\mathbf{y} \in \mathbb{R}^E$ where for $e = \{u, v\} \in E$, we set

$$y_e = \min\left(\frac{1}{\deg^e(u) \cdot H(\deg(u))}, \frac{1}{\deg^e(v) \cdot H(\deg(v))}, \frac{1}{\alpha+1}\right)$$

where $\deg^e(u)$ and $\deg^e(v)$ are the number of edges at least as heavy as e that are incident to u and v respectively and $H(r) = 1/1 + 1/2 + \ldots + 1/r$ is the harmonic function.

The next two theorems show that $\mathbf{y}$ is a local fractional matching and

$$\frac{1}{H(D) \cdot (\alpha+1)} \text{match}(G) \leq \text{score}(\mathbf{y}) \leq \frac{\alpha+2}{\alpha+1} \text{match}(G)$$

where $\text{score}(\mathbf{y}) = \sum_e w_e y_e$ and D is the maximum degree of the graph. Note that D can be as large as $n-1$ even for a low arboricity graph. However, since the average degree of G is at most 2α, we expect D to typically be much smaller for many low arboricity graphs of interest.

Theorem 9. $\mathbf{y} \in \mathsf{FM}$ *and*

$$\frac{\text{score}(\mathbf{y})}{\text{match}(G)} \leq \begin{cases} \frac{\alpha+2}{\alpha+1} & \text{if } \alpha \text{ odd} \\ \frac{\alpha+3}{\alpha+2} & \text{if } \alpha \text{ even} \end{cases} .$$

Furthermore, if G is bipartite then $\text{score}(\mathbf{y}) \leq \text{match}(G)$.

Proof. For all $u \in V$,

$$\sum_{e \in E: u \in e} x_e \leq \frac{1}{H(\deg(u))} \sum_{e \in E: u \in e} \frac{1}{\deg^e(u)} \leq \frac{1}{H(\deg(u))}(1/1 + 1/2 + \ldots + 1/\deg(u)) = 1 \ ,$$

and hence $\mathbf{y} \in \mathsf{FM}$. The result of the proof follows as in the proof of Theorem 7 since $y_e \leq 1/(\alpha+1)$ for all e. □

Theorem 10. $\text{match}(G) \leq H(D)(\alpha+1)\,\text{score}(\mathbf{y})$ *where D is the maximum degree.*

Proof. Let z_e be the optimum weighted integral matching. Let $0 < w_1 < w_2 < w_3 < \ldots$ be the distinct weights in the graph and let $w_0 = 0$. Let G_k be the unweighted graph formed from the original weighted graph where all edges whose weight is $< w_k$ are deleted and the other edges are given weight 1. Let z_e^k be the optimum unweighted integral matching for G_k and let $\deg_k(u)$ be the degree of node u in G_k.

Then,

$$\begin{aligned} \text{score}(\mathbf{z}) = \sum_e z_e w_e &\leq \sum_k (w_k - w_{k-1}) \sum_{e \in G_k} z_e^k \\ &\leq (\alpha+1) \sum_k (w_k - w_{k-1}) \sum_{e \in G_k} \min\left(\frac{1}{\deg_k(u)}, \frac{1}{\deg_k(v)}, \frac{1}{\alpha+1}\right) \end{aligned}$$

where the last inequality follows by our result for the unweighted case.

But for any $e \in E$,

$$\begin{aligned} & \sum_{k: e \in G_k} (w_k - w_{k-1}) \min\left(\frac{1}{\deg_k(u)}, \frac{1}{\deg_k(v)}, \frac{1}{\alpha+1}\right) \\ \leq & \sum_{k: e \in G_k} (w_k - w_{k-1}) \min\left(\frac{1}{\deg^e(u)}, \frac{1}{\deg^e(v)}, \frac{1}{\alpha+1}\right) \\ \leq & \ w_e \min\left(\frac{1}{\deg^e(u)}, \frac{1}{\deg^e(v)}, \frac{1}{\alpha+1}\right) \\ \leq & \ H(D) w_e y_e \end{aligned}$$

where the first inequality follows because $\deg_k(u) \geq \deg^e(u)$ for all k such that $e \in G_k$. Therefore $\text{match}(G) \leq H(D)(\alpha+1)\,\text{score}(\mathbf{y})$ as claimed. □

2.4 Exact Degree Distribution

Using ideas from the previous sections, we now show that the size of the maximum matching can be approximated up to a $O(\alpha^2)$ factor given just the degree distribution of G. Specifically, consider the following estimate:

$$\tilde{M} = \sum_{u \in V} \min(\alpha + 1 - \deg(u)/2, \deg(u)/2) \ .$$

The next theorem shows that $\tilde{M}$ is a $O(\alpha^2)$ approximation for $\text{match}(G)$.

Theorem 11. $\text{match}(G) \leq \tilde{M} \leq \frac{(\alpha+2)^2}{2} \cdot \text{match}(G)$.

Proof. Let h be the number of "heavy" nodes with degree at least $\alpha + 2$. Partition the edges E into E_0, E_1, and E_2 depending on whether the edge has zero, one, or two heavy endpoints. Note that E_0 is just the set of shallow edges. Then,

$$\begin{aligned}
& \sum_{u \in V} \min(\alpha + 1 - \deg(u)/2, \deg(u)/2) \\
= & \sum_{u \in V} \deg(u)/2 - \max(\deg(u) - \alpha - 1, 0) \\
= & |E_0| + |E_1| + |E_2| - \left(\sum_{u:\deg(u) \geq \alpha+2} \max(\deg(u) - \alpha - 1, 0) \right) \\
= & |E_0| + |E_1| + |E_2| - \left(\sum_{u:\deg(u) \geq \alpha+2} \deg(u) \right) + h(\alpha + 1) \\
= & |E_0| + |E_1| + |E_2| - |E_1| - 2|E_2| + h(\alpha + 1) \\
= & |E_0| - |E_2| + h(\alpha + 1)
\end{aligned}$$

First note that $|E_2| \leq \alpha h$ because the number of edges in any induced subgraph is at most α times the number of nodes in that subgraph. Hence,

$$|E_0| - |E_2| + h(\alpha + 1) \geq |E_0| + h \geq \text{match}(G) \ .$$

By appealing to Theorem 8 and Theorem 7

$$\begin{aligned}
|E_0| - |E_2| + h(\alpha + 1) & \leq |E_0| + h(\alpha + 1) \\
& \leq \frac{(\alpha + 2)(\alpha + 1)}{2} \cdot (|E_0|/(\alpha + 1) + 2h/(\alpha + 2)) \\
& \leq \frac{(\alpha + 2)(\alpha + 1)}{2} \cdot \frac{\alpha + 2}{\alpha + 1} \cdot \text{match}(G) \\
& \leq \frac{(\alpha + 2)^2}{2} \cdot \text{match}(G) \ .
\end{aligned}$$

□

3 Algorithmic Applications in Streaming Models

3.1 Adversarial Insertion-Only Streams

In this section we briefly describe a streaming estimation based on the results from Section 2.

From Theorem 1, we know we can estimate the size of the maximum cardinality via the following quantity,

$$A := \sum_{\{u,v\} \in E} \min \left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha + 1} \right) \ .$$

To do this we first show that A can be estimated via the quantity,

$$A_S := \sum_{\{u,v\} \in E : u,v \in S} \min \left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha + 1} \right) \ .$$

where S is a subset of V formed by sampling each node independently with probability p. The next lemma shows that A_S is within a $1 + \epsilon$ factor of Ap^2 with probability at least $3/4$ assuming

p is sufficiently large. Note that a similar approach is taken in Esfandiari et al. [17] and Chitnis et al. [8] in the context of their algorithm to estimate the number of high degree nodes and edges that are not incident to high degree nodes.

Lemma 3. *If $p \geq \sqrt{12\epsilon^{-2}A^{-1}}$, then $\mathbb{P}\left[|A_S - Ap^2| \leq \epsilon \cdot Ap^2\right] \geq 3/4$.*

Proof. For each edge $e = \{u, v\} \in E$, let $x_e = \min(1/\deg(u), 1/\deg(v), 1/(\alpha+1))$ and define a random variable X_e where $X_e = x_e$ if $u, v \in S$ and $X_e = 0$ otherwise. Note that $A_S = \sum_{e \in E} X_e$. Then, the expectation and variance of A_S are $\mathbb{E}[A_S] = Ap^2$ and

$$\mathbb{V}[A_S] = \sum_{e \in E} \sum_{e' \in E} \mathbb{E}[X_e X_{e'}] - \mathbb{E}[X_e]\,\mathbb{E}[X_{e'}] \ .$$

Note that

$$\sum_{e' \in E} \mathbb{E}[X_e X_{e'}] - \mathbb{E}[X_e]\,\mathbb{E}[X_{e'}] = \begin{cases} x_e^2(p^2 - p^4) & \text{if } e = e' \\ x_e x_{e'}(p^3 - p^4) & \text{if } e \text{ and } e' \text{ share exactly one endpoint} \\ 0 & \text{if } e \text{ and } e' \text{ share no endpoints} \end{cases} .$$

Since the sum of all $x_{e'}$ that share an endpoint with e is at most 2 because $\mathbf{x} \in \mathsf{FM}$,

$$\mathbb{V}[A_S] \leq \left(\sum_{e \in E} x_e^2(p^2 - p^4)\right) + 2A(p^3 - p^4) \leq 3Ap^2 \ .$$

We then use Chebyshev's inequality to obtain

$$\mathbb{P}\left[|A_S - Ap^2| \leq \epsilon Ap^2\right] \leq \frac{3Ap^2}{\epsilon^2 A^2 p^4} = \frac{3}{\epsilon^2 Ap^2} \leq 3/4 \ .$$

□

Given this key lemma, the algorithm and analysis proceed similarly to that of Esfandiari et al. [17]. Specifically, two algorithms are run in parallel: a greedy matching algorithm and a sampling-based algorithm. The greedy matching algorithm uses $O(n^{2/3} \log n)$ space to find a maximal matching of size at least $\min(n^{2/3}, \text{match}(G)/2)$. The sampling-based algorithm uses $O(\alpha n^{2/3} \log n)$ space to sample each node with probability $p = \Theta(\epsilon^{-1}/n^{2/3})$ and then find all edges whose endpoints are both sampled along with the degrees of the sampled edges. If the greedy matching has size less than $n^{2/3}$ then it is necessarily a 2 approximation of match(G). If not, we can use the estimate of A based on the nodes sampled since in this case $A = \Omega(n^{2/3})$.

Theorem 12. *There exists a single pass data stream algorithm using $O(\alpha\epsilon^{-1}n^{2/3}\log\delta^{-1})$ space that returns a $(\alpha+2)(1+\epsilon)$ approximation of the maximum matching with probability at least $1-\delta$.*

3.2 Adjacency List Graph Streams

In the *adjacency list model*[1] the edges incident to each node v appear consecutively in the stream [34, 6, 5]. Thus, every edge $\{u, v\}$ will appear twice: once when we view the adjacency list of u and once for v. Aside from that constraint, the stream is ordered arbitrarily. For example, for the graph consisting of a cycle on three nodes $V = \{v_1, v_2, v_3\}$, a possible ordering of the stream could be $\langle v_3v_1, v_3v_2, v_2v_3, v_2v_1, v_1v_2, v_1v_3\rangle$. Note that in this model it is trivial to compute

$$\tilde{M} = \sum_{u \in V} \min(\alpha + 1 - \deg(u)/2, \deg(u)/2) \ .$$

[1] The adjacency list order model is closely related to the vertex arrival model [21, 26] and row-order arrival model considered in the context of linear algebra problems [9, 20].

in $O(\log n)$ space since the degree of a node can be calculated exactly when the adjacency list of that node appears. The next theorem immediately follows from Theorem 11.

Theorem 13. *An $(\alpha+2)^2/2$-approximation of the size of maximum matching can be computed using $O(\log n)$ in the adjacency list model. In particular, this yields a 12.5-approximation for planar graphs.*

3.3 Dynamic Streams

One application is a combination of Lemma 1 with sketching algorithms for the Hamming norm. In order to estimate the matching size of a tree T, we maintain a ℓ_0-Estimator for the degree vector $d \in \mathbb{R}^n$ such that $d_i = \deg(v_i) - 1$ holds at the end of the stream and with it $\ell_0(d) = h_T$. In other words, we initialize the vector by adding -1 to each entry and update the two corresponding entries when we get an edge deletion or insertion. Using Theorem 10 from [25] we can maintain the ℓ_0-Estimator for d in $O(\varepsilon^{-2}\log^2 n)$ space.

Theorem 14. *Let $T=(V,E)$ be a tree with at least 3 nodes and let $\varepsilon \in (0,1)$. Then there is an algorithm that estimates the size of a maximum matching in T within a $(2+\varepsilon)$-factor in the dynamic streaming model with high constant probability using 1-pass over the data and $O(\varepsilon^{-2}\log n)$ space.*

The other application is a combination of Theorems 12 and a Tutte-matrix based estimation which we now describe in detail[2]. Our aim is to randomly choose entries of a Tutte matrix and update this matrix with the corresponding value whenever an edge is inserted or deleted.

One crucial ingredient are the following two results due to Clarkson and Woodruff [9], see Sarlos [38] for similar, slightly weaker statements.

Lemma 4 (Lemma 3.4 of [9])**.** *Given integer k and $\varepsilon, \delta > 0$, there is $m = O(k\log(1/\delta)/\varepsilon)$ and an absolute constant η such that if S is an $n \times m$ sign matrix with $\eta(k+\log(1/\delta))$- wise independent entries, then for $n \times k$ matrix U with orthonormal columns, with probability at least $1-\delta$, the spectral norm $||U^T SS^T U - U^T U||_2 \leq \varepsilon$.*

Since U is orthogonal, all singular values are 1. If we choose ε to be some constant, the singular values of of $S^T U$ and U differ only by multiplicative constant factors, which also implies that $S^T U$ and U have the same rank. For the purpose of this paper, $\varepsilon = 1/3$ will be sufficient.

Our algorithm now proceeds as follows. We initialize the Tutte-matrix T of the input graph G with randomly chosen entries drawn from a $4k^2$-independent hash function. Whenever we process an edge operation, the appropriate random value of the corresponding entry in T is queried and added or subtracted. We then independently sample two sign matrices S_1 and S_2 where $S_1, S_2 \in \{-1,1\}^{n \times c\cdot k}$, where c is some sufficiently large absolute constant, and maintain $S_1^T T S_2$. The correctness of this algorithm is an almost direct application of Lemma 4:

Proof of Theorem 3. We randomly chose the weights of the Tutte-matrix from 1 to $O(k^2)$. By Theorem 6, Theorem 5 holds when we query the size of the matching with constant probability. Let $r \leq k$ be the rank of T. Let $U_1 \Sigma U_2^T$ be the singular value truncated decomposition of T such that $U_1, U_2 \in \mathbb{R}^{n\times r}$ are orthogonal and $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal. Lemma 4 guarantees us that any rank up to k of $S_1^T U_1$ and $U_2^T S_2$ is preserved with constant probability. Since $U_1\Sigma$ has full rank, $\operatorname{rank}(U_1\Sigma U_2^T S_2) = \operatorname{rank}(U_2^T S_2) = \operatorname{rank}(U_2) = r$. By the same argument and independence of S_1 and S_2, $\operatorname{rank}(S_1^T T S_2) = r$. The overall probability of success can be amplified in the standard

[2] We note that a sampling strategy from [8] could replace the Tutte-matrix based estimation. In fact their result is somewhat stronger, as the show that using roughly the same space, they can recover any matching up to size k. Nevertheless, we believe that our technique may be of independent interest.

Algorithm 1 Approximation of Weighted Matching from [40]

Require: Graph $G = (V, E = \bigcup_{i=1}^{t} E_i)$
Ensure: Matching M

$M \leftarrow \emptyset$
for $i = t$ to 1 **do**
  Find a maximal matching M_i in $G_i = (V, E_i)$.
  Add M_i to M.
  Remove all edges e from E such that $e \in M_i$ or e shares a node with an edge in M_i.
return M

way by independent repetition and outputting the maximum. The space bound of each $S_1^T T S_2$ is in $O(k^2 \log n)$ due to the dimension of the sign matrices via Lemma 4 and by observing that the magnitude of entries of $S_1^T T S_2$ is polynomial in n. The total space bound is therefore dominated by the seed length $O(k^2 \log k \log n)$ of the pseudorandom generator. □

We use Theorem 3 to determine the matching size up to $n^{2/5}$. For larger matchings, we apply Lemma 3 with $p = \Theta(\varepsilon^{-1}/n^{4/5})$.

Theorem 15. *There exists a single pass data stream algorithm using $\tilde{O}(\alpha\epsilon^{-1} n^{4/5} \log \delta^{-1})$ space that returns a $(\alpha+2)(1+\epsilon)$ approximation of the maximum matching with probability at least $1-\delta$.*

4 Weighted Matching

We start by describing the parallel algorithm by Uehara and Chen [40], see Algorithm 1. Let $\gamma > 1$ and $k > 0$ be constant. We partition the edge set by t ranks where all edges e in rank $i \in \{1, \ldots, t\}$ have a weight $w(e) \in \left(\gamma^{i-1} \cdot \frac{w_{max}}{kN}, \gamma^i \cdot \frac{w_{max}}{kN}\right]$ where w_{max} is the maximal weight in G. For simplicity, assume $\frac{w_{max}}{kN}$ to be scaled to 1. Let $G' = (V, E, w')$ be equal to G but each edge e in rank i has weight γ^i for all $i = 1, \ldots, t$. Starting with $i = t$, we compute an unweighted maximal matching M_i considering only edges in rank i (in G') and remove all edges incident to a matched node. Continue with $i - 1$. The weight of the matching $M = \bigcup M_i$ is $w(M) = \sum_{i=1}^{t} \gamma^i \cdot |M_i|$ and satisfies $w_G(M^*) \geq w_{G'}(M) \geq \frac{1}{2\gamma} \cdot w_G(M^*)$ where M^* is an optimal weighted matching in G.

The previous algorithms [12, 18, 35, 15, 16, 43] for insertion-only streams use a similar partitioning of edge weights. Since these algorithms store no more than one maximal matching per rank, they cannot compute residual maximal matchings, but by charging the smaller edge weights into the higher ones the resulting approximation factor can be made reasonably close to that of Uehara and Chen.

In order to adapt this idea to our setting, we need to work out the key properties of the partitioning and how we can implement it in a stream. Recalling the partitioning of Uehara and Chen, we disregard all edges with weight smaller than $\frac{w_{max}}{kN}$ which is possible because the contribution of these edges is at most $\frac{N}{2} \cdot \frac{w_{max}}{kN} = \frac{w_{max}}{2k} \leq \frac{OPT}{2k}$ where OPT is the weight of an optimal weighted matching. Thus, we can only consider edges with larger weight and it is also possible to partition the set of edges in a logarithmic number of sets. Here, we use the properties that edge weights within a single partition set are similar and that $\frac{1}{\gamma} \leq \frac{w(e)}{w(e')} \leq \gamma$ for two edges $e \in E_i$ and $e' \in E_{i-1}$ with $i \in \{2, \ldots, t\}$. These properties are sufficient to get a good approximation on the optimal weighted matching which we show in the next lemma. The proof is essentially the same as in [40].

Algorithm 2 Estimation of Weighted Matching

Require: Graph $G = (V, E = \bigcup_{i=1}^{t} E_i)$, unweighted estimation routine A
Ensure: Estimated weight $\widehat{W}$

for $i = t$ to 2 **do**
  Use λ-estimator A to estimate the size of a maximum matching in $G_i = (V, \bigcup_{j=i}^{t} E_j)$.
  $W' \leftarrow W' + (w'(i) - w'(i-1)) \cdot A(G_i)$.
Use A to estimate the size of a maximum matching G_1 in $G = (V, E)$.
$W' \leftarrow W' + S_1$.
return $\widehat{W} = \lambda \cdot W'$.

Lemma 5. *Let $G = (V, E, w)$ be a weighted graph and $\varepsilon > 0$ be an approximation parameter. If a partitioning $E_1, \ldots, E_t$ of E and a weight function $w' : E \to \mathbb{R}$ satisfies*

$$\frac{1}{1+\varepsilon} \leq \frac{w'(e)}{w(e)} \leq 1 \text{ for all } e \in E \quad \text{and} \quad \frac{w(e_1)}{w(e_2)} \leq 1+\varepsilon \quad \text{and} \quad w(e) < w(e')$$

for all choices of edges $e_1, e_2 \in E_i$ and $e \in E_i, e' \in E_j$ with $i < j$ and $i, j \in \{1, \ldots, t\}$ then Algorithm 1 returns a matching $M = \bigcup_{i=1}^{t} M_i$ with

$$\frac{1}{2(1+\varepsilon)} \cdot w(M^*) \leq w'(M) \leq w(M^*)$$

where M^* is an optimal weighted matching in G.

Proof. The first property $\frac{1}{1+\varepsilon} \leq \frac{w'(e)}{w(e)} \leq 1$ for all $e \in E$ implies that $\frac{w(S)}{1+\varepsilon} \leq w'(S) \leq w(S)$ for every set of edges $S \subseteq E$. Thus, it remains to show that $\frac{1}{2(1+\varepsilon)} \cdot w(M^*) \leq w(M) \leq w(M^*)$. Since M^* is an optimal weighted matching, it is clear that $w(M) \leq w(M^*)$. For the lower bound, we distribute the weight of the edges from the optimal solution to edges in M. Let $e \in M^*$ and $i \in \{1, \ldots, t\}$ such that $e \in E_i$. We consider the following cases:

1. $e \in M_i$: We charge the weight $w(e)$ to the edge itself.
2. $e \notin M_i$ but at least one node incident to e is matched by an edge in M_i: Let $e' \in M_i$ be an edge sharing a node with e. Distribute the weight $w(e)$ to e'.
3. $e \notin M_i$ and there is no edge in M_i sharing a node with e: By Algorithm 1, there has to be an edge $e' \in M_j$ with $j > i$ which shares a node with e. We distribute the weight $w(e)$ to e'.

Since M^* is a matching, there can only be at most two edges from M^* distributing their weights to an edge in M. We know that $\frac{w(e)}{w(e')} \leq 1+\varepsilon$ for all choices of two edges $e, e' \in E_i$ with $i \in \{1, \ldots, t\}$ which means that in the case 2. we have $w(e) \leq (1+\varepsilon) \cdot w(e')$. In case 3. it holds $w(e) < w(e')$. Thus, the weight distributed to an edge e' in M is at most $2(1+\varepsilon)w(e')$. This implies that $w(M^*) = \sum_{e \in M^*} w(e) \leq \sum_{e' \in M} 2(1+\varepsilon) \cdot w(e') = 2(1+\varepsilon) \cdot w(M)$ which concludes the proof. □

Using Lemma 5, we can partition the edge set in a stream in an almost oblivious manner: Let $(e, w(e))$ be the first inserted edge. Then an edge e' belongs to E_i iff $(1+\varepsilon)^{i-1} \cdot w(e) < w(e') \leq (1+\varepsilon)^i \cdot w(e)$ for some $i \in \mathbb{Z}$. For the sake of simplicity, we assume that the edge weights are in $[1, W]$. Then the number of sets is $O(\log W)$. We would typically expect $W \in \text{poly } n$ as otherwise storing weights becomes infeasible. In the following, denote by $w'(i)$ the weight of edges in rank E_i. We now are able to give our weighted estimation algorithm and state our main theorem.

Theorem 2. *Let $G = (V, E, w)$ be a weighted graph where the weights are from $[1, W]$. Let A be an algorithm that returns an λ-estimator $\widehat{M}$ for the size of a maximum matching M of a graph with $1/\lambda \cdot |M| \leq \widehat{M} \leq |M|$ with failure probability at most δ and needs space S. If we partition the edge set into sets $E_1, \ldots, E_t$ with $t = \lfloor \log_{1+\varepsilon} W \rfloor \in O(\varepsilon^{-1} \log W)$ where E_i consists of all edges with weight in $[(1+\varepsilon)^{i-1}, (1+\varepsilon)^i)$, and use A as the unweighted matching estimator in Algorithm 2, then the algorithm returns an $2 \cdot (1+\varepsilon)\lambda$-estimator $\widehat{W}$ for the weight of the maximum weighted matching with failure probability at most $\delta \cdot (t+1)$ using $O(S \cdot t)$ space., i.e. $\frac{1}{2(1+\varepsilon)\lambda} \cdot w(M^*) \leq \widehat{W} \leq w(M^*)$ where M^* is an optimal weighted matching.*

Proof. Let M' be the matching computed by Algorithm 1 and denote by $S_i := M' \cap \left(\bigcup_{j \geq i}^{t} E_j\right)$ the partial matching of $G\left(V, \left(\bigcup_{j \geq i}^{t} E_j\right)\right)$ for any rank $i \in \{1, \ldots t\}$. Note that $S_1 := M$. We can then decompose the objective function as follows

$$\begin{aligned} w(M) &= \sum_{i=1}^{t} \sum_{e \in M \cap E_i} w(e) \\ &\geq \sum_{i=1}^{t} \sum_{e \in M \cap E_i} (1+\varepsilon)^{i-1} \\ &= \sum_{i=2}^{t} \sum_{e \in S_i} \left((1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2}\right) + \sum_{e \in M} 1 \\ &= \sum_{i=2}^{t} |S_i| \cdot \left((1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2}\right) + |S_1|. \end{aligned}$$

Assume now that each call to the unweighted λ-estimation algorithm for the maximum matching of $G\left(V, \left(\bigcup_{j \geq i}^{t} E_j\right)\right)$ succeeds, which happens with probability $1 - \delta(t+1)$. We have

$$\begin{aligned} W' &= \sum_{i=2}^{t} A(G_i) \cdot \left((1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2}\right) + A(G_1) \\ &\geq \sum_{i=2}^{t} \frac{1}{\lambda} \cdot \mathrm{match}(G_i) \left((1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2}\right) + \frac{1}{\lambda} \cdot \mathrm{match}(G_1) \\ &\geq \sum_{i=2}^{t} \frac{1}{\lambda} \cdot |M^* \cap E_i| \left((1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2}\right) + \frac{1}{\lambda} \cdot |M^* \cap E_1| \geq \frac{1}{\lambda} \cdot w(M^*) \end{aligned}$$

Since S_i is maximal w.r.t. $G\left(V, \left(\bigcup_{j \geq i}^{t} E_j\right)\right)$, we have $A(G_i) \leq 2|S_i|$. Then for the upper bound

$$\begin{aligned} W' &= \sum_{i=2}^{t} A(G_i) \cdot \left((1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2}\right) + A(G_1) \\ &\leq 2 \cdot \left(\sum_{i=2}^{t} |S_i| \cdot (1+\varepsilon)^{i-1} - (1+\varepsilon)^{i-2} + |S_1| \right) \leq 2(1+\varepsilon) \cdot w(M) \leq 2(1+\varepsilon) \cdot w(M^*). \end{aligned}$$

Combining these two bounds and setting $\widehat{W} = \lambda \cdot W'$, the theorem follows. □

5 Lower Bounds for Insertion-Only Streams

Esfandiari et al. [17] showed a space lower bound of $\Omega(\sqrt{n})$ for any estimation better than 3/2. Their reduction (see below) uses the Boolean Hidden Matching Problem introduced by Bar-Yossef

et al. [4], and further studied by Gavinsky et al. [19]. We will use the following generalization due to Verbin and Yu [42].

Definition 2 (Boolean Hidden Hypermatching Problem [42])**.** *In the* Boolean Hidden Hypermatching *Problem $BHH_{t,n}$ Alice gets a vector $x \in \{0,1\}^n$ with $n = 2kt$ and $k \in \mathbb{N}$ and Bob gets a perfect t-hypermatching M on the n coordinates of x, i.e., each edge has exactly t coordinates, and a string $w \in \{0,1\}^{n/t}$. We let Mx denote the following vector of length n/t,*

$$Mx := \left(\bigoplus_{1 \le i \le t} x_{M_{1,i}}, \ldots, \bigoplus_{1 \le i \le t} x_{M_{n/t,i}} \right)$$

where $(M_{1,1}, \ldots, M_{1,t}), \ldots, (M_{n/t,1}, \ldots, M_{n/t,t})$ are the edges of M and $\bigoplus$ denotes XOR. The problem is to return 1 if $Mx \oplus w = 1^{n/t}$ and 0 if $Mx \oplus w = 0^{n/t}$, otherwise the algorithm may answer arbitrarily.

Verbin and Yu [42] showed a lower bound of $\Omega(n^{1-1/t})$ for the randomized one-way communication complexity for $BHH_{t,n}$. For our reduction we require $w = 0^{n/t}$ and $x \in \{0,1\}^n$ has exactly $n/2$ bits set to 1. We denote this problem by $BHH^0_{t,n}$. We can show that this does not reduce the communication complexity.

Lemma 6. *The communication complexity of $BHH^0_{t,4n}$ is lower bounded by the communication complexity of $BHH_{t,n}$.*

Proof. First, let us assume that t is odd. Let $x \in \{0,1\}^n$ with $n = 2kt$ for some $k \in \mathbb{N}$ and M be a perfect t-hypermatching on the n coordinates of x and $w \in \{0,1\}^{n/t}$. We define $x' = [x^T x^T \overline{x}^T \overline{x}^T]^T$ to be the concatenation of two identical copies of x and two identical copies of the vector resulting from the bitwise negation of x. Without loss of generality, let $\{x_1, \ldots, x_t\} \in M$ be the l-th hyperedge of M. Then we add the following four hyperedges to M':

- $\{x_1, \overline{x_2}, \ldots, \overline{x_t}\}, \{\overline{x_1}, x_2, \overline{x_3}, \ldots, \overline{x_t}\}, \{\overline{x_1}, \overline{x_2}, x_3, \ldots, x_t\}$, and $\{x_1, \ldots, x_t\}$ if $w_l = 0$,
- $\{\overline{x_1}, x_2, \ldots, x_t\}, \{x_1, \overline{x_2}, \ldots, x_t\}, \{x_1, x_2, \overline{x_3}, \ldots, \overline{x_t}\}$, and $\{\overline{x_1}, \ldots, \overline{x_t}\}$ if $w_l = 1$.

The important observation here is that, since t is odd, we flip an even number of bits in the case $w_l = 0$ and an odd number of bits if $w_l = 1$. Since every bit flip results in a change of the parity of the set of bits, the parity does not change iff we flip an even number of bits. Therefore, $w_l \oplus x_1 \oplus \ldots \oplus x_t = 0$ iff the parity of each of the corresponding new hyperedges is 0. Applying the same reasoning to all hyperedges, we deduce that $M'x' = 0^{4n/t}$ if $Mx \oplus w = 0^{n/t}$ and $M'x' = 1^{4n/t}$ if $Mx \oplus w = 1^{n/t}$. The number of ones in $x' \in \{0,1\}^{4n}$ is exactly $2n$. If t is even, we can just change the cases for the added edges such that we flip an even number of bits in the case $w_l = 0$ and an odd number of bits if $w_l = 1$. Overall, this shows that a lower bound for $BHH_{t,n}$ implies a lower bound for $BHH^0_{t,4n}$. □

Theorem 4. *Any randomized streaming algorithm that approximates the maximum matching size within a $1 + \frac{1}{3t/2-1}$ factor for $t \ge 2$ needs $\Omega(n^{1-1/t})$ space.*

Proof. Let x, M be the input to the $BHH^0_{t,n}$ problem, i.e., M is a perfect t-hypermatching on the coordinates of x, x has exactly $n/2$ ones and it is promised that either $Mx = 0^{n/t}$ or $Mx = 1^{n/t}$. We construct the graph for the reduction as described above: For each bit x_i we have two nodes $v_{1,i}, v_{2,i}$ and Alice adds the edge $\{v_{1,i}, v_{2,i}\}$ iff $x_i = 1$. For each edge $\{x_{i_1}, \ldots, x_{i_t}\} \in M$. Bob adds a t-clique consisting of the nodes $v_{2,i_1}, \ldots, v_{2,i_t}$.

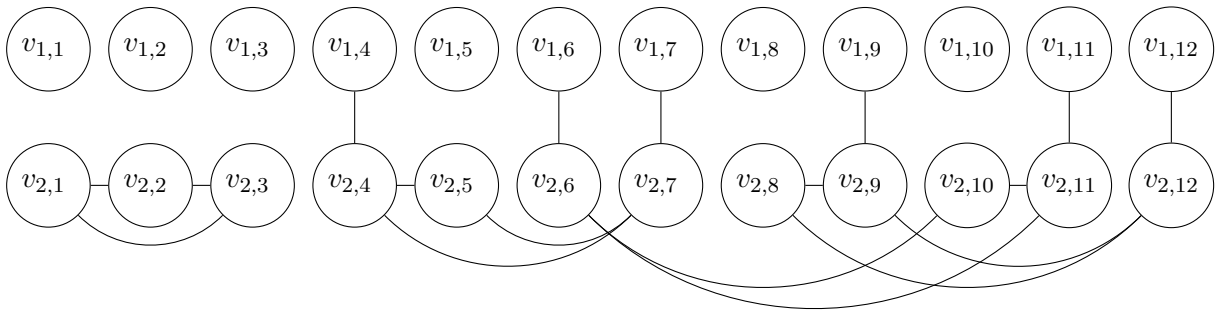

Figure 2: Worst case instance for $t = 3$. Bob's hypermatching corresponds to disjoint 3-cliques among the lower nodes and Alice's input vector corresponds to the edges between upper and lower nodes.

We first consider the case where t is odd. We know that the matching is at least $n/2$ because x has exactly $n/2$ ones. Since Bob adds a clique for every edge it is always possible to match all (or all but one) nodes of the clique whose corresponding bit is 0. In the case of $Mx = 0^{n/t}$ the parity of every edge is 0, i.e., the number of nodes whose corresponding bit is 1 is even. Let $M_{2i} \subseteq M$ be the hyperedges containing exactly $2i$ one bits and define $l_{2i} := |M_{2i}|$. Then we know $n/2 = \sum_{i=0}^{\lfloor t/2 \rfloor} 2i \cdot l_{2i}$ and $|M| = n/t = \sum_{i=0}^{\lfloor t/2 \rfloor} l_{2i}$. For every edge in M_{2i} the size of the maximum matching within the corresponding subgraph is exactly $2i + \lfloor (t-2i)/2 \rfloor = 2i + \lfloor t/2 \rfloor - i$ for every $i = 0, \ldots, \lfloor t/2 \rfloor$ (see Fig. 2). Thus, we have a matching of size

$$\sum_{i=0}^{\lfloor t/2 \rfloor} (2i + (\lfloor t/2 \rfloor - i)) l_{2i} = \frac{n}{2} + \frac{t-1}{2} \cdot \frac{n}{t} - \frac{n}{4} = \frac{3n}{4} - \frac{n}{2t}.$$

If we have $Mx = 1^{n/t}$ then let $M_{2i+1} \subseteq M$ be the hyperedges containing exactly $2i+1$ one bits and define $l_{2i+1} := |M_{2i+1}|$. Again, we know $n/2 = \sum_{i=0}^{\lfloor t/2 \rfloor} (2i+1) \cdot l_{2i+1}$ and $|M| = n/t = \sum_{i=0}^{\lfloor t/2 \rfloor} l_{2i+1}$. For every edge in M_{2i+1} the size of the maximum matching within the corresponding subgraph is exactly $2i + 1 + (t - 2i - 1)/2 = 2i + 1 + \lfloor t/2 \rfloor - i$ for every $i = 0, \ldots, \lfloor t/2 \rfloor$. Thus, the maximum matching has a size

$$\sum_{i=0}^{\lfloor t/2 \rfloor} (2i + 1 + (\lfloor t/2 \rfloor - i)) l_{2i+1} = \frac{n}{2} + \frac{t-1}{2} \cdot \frac{n}{t} - \frac{1}{2} \sum_{i=0}^{\lfloor t/2 \rfloor} (2i+1) \cdot l_{2i+1} + \frac{n}{2t} = \frac{3n}{4}.$$

For t even, the size of the matching is

$$\sum_{i=0}^{t/2} (2i + (t-2i)/2) l_{2i} = \frac{n}{2} + \frac{t}{2} \cdot \frac{n}{t} - \frac{n}{4} = \frac{3n}{4}$$

if $Mx = 0^{n/t}$. Otherwise, we have

$$\begin{aligned} \sum_{i=0}^{t/2} \left(2i + 1 + \left\lfloor \frac{t-2i-1}{2} \right\rfloor \right) l_{2i+1} &= \frac{n}{2} + \sum_{i=0}^{t/2} (t/2 - i - 1) l_{2i+1} \\ &= \frac{n}{2} - (t/2 - 1) \cdot \frac{n}{t} - \frac{n}{4} + \frac{n}{2t} = \frac{3n}{4} - \frac{n}{2t}. \end{aligned}$$

As a consequence, every streaming algorithm that computes an α-approximation on the size of a maximum matching with

$$\alpha < \frac{(3/4)n}{((3/4) - 1/(2t))n} = 1/(1 - 4/6t) = 1 + \frac{1}{3t/2 - 1}$$

can distinguish between $Mx = 0^{n/t}$ and $Mx = 1^{n/t}$ and, thus, needs $\Omega(n^{1-1/t})$ space. □

Finally, constructing the Tutte-matrix with randomly chosen entries from public randomness and applying Theorem 6 gives us

Corollary 1. *Any randomized streaming algorithm that approximates* $\text{rank}(A)$ *of* $A \in \mathbb{R}^{n\times n}$ *within a* $1 + \frac{1}{3t/2-1}$ *factor for* $t \geq 2$ *requires* $\Omega(n^{1-1/t})$ *space.*

References

[1] Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 20:1–20:22, 2016.

[2] Alexandr Andoni and Huy L. Nguyen. Eigenvalues of a matrix in the streaming model. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1729–1737, 2013.

[3] 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.

[4] Ziv Bar-Yossef, T. S. Jayram, and Iordanis Kerenidis. Exponential separation of quantum and classical one-way communication complexity. *SIAM Journal on Computing*, 38(1):366–384, 2008.

[5] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 623–632, 2002.

[6] Luciana S. Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. Counting triangles in data streams. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 253–262, 2006.

[7] 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.

[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 *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1326–1344, 2016.

[9] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2009.

[10] Graham Cormode, Hossein Jowhari, Morteza Monemizadeh, and S. Muthukrishnan. The sparse awakens: Streaming algorithms for matching size estimation in sparse graphs. *CoRR*, abs/1608.03118, 2016.

[11] Michael Crouch, Andrew McGregor, and Daniel Stubbs. Dynamic graphs in the sliding-window model. In *Proceedings of the 21st Annual European Symposium (ESA)*, pages 337–348, 2013.

[12] Michael Crouch and Daniel Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Proceedings of the 18th Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 96–104, 2014.

[13] Andrzej Czygrinow, Michal Hanckowiak, and Edyta Szymanska. Fast distributed approximation algorithm for the maximum matching problem in bounded arboricity graphs. In *Proceedings of the 20th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 668–678, 2009.

[14] Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards, 69:125-130*, 1965.

[15] Leah Epstein, Asaf Levin, Julián Mestre, and Danny Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM Journal on Discrete Mathematics*, 25(3):1251–1265, 2011.

[16] Leah Epstein, Asaf Levin, Danny Segev, and Oren Weimann. Improved bounds for online preemptive matching. In *Proceedings of the 30th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 389–399, 2013.

[17] Hossein Esfandiari, Mohammad T Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1217–1233, 2015.

[18] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theoretical Computer Science*, 348(2-3):207–216, 2005.

[19] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM Journal on Computing*, 38(5):1695–1708, 2008.

[20] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *CoRR*, abs/1501.01711, 2015.

[21] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 468–485, 2012.

[22] Elena Grigorescu, Morteza Monemizadeh, and Samson Zhou. Estimating weighted matchings in o(n) space. *CoRR*, abs/1604.07467, 2016.

[23] Elena Grigorescu, Morteza Monemizadeh, and Samson Zhou. Streaming weighted matchings: Optimal meets greedy. *CoRR*, abs/1608.01487, 2016.

[24] Monika R. Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. In *External Memory Algorithms: DIMACS Workshop External Memory and Visualization*, volume 50, pages 107–118. American Mathematical Society, 1999.

[25] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA*, pages 41–52, 2010.

[26] Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1679–1697, 2013.

[27] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 734–751, 2014.

[28] C. Konrad. Maximum matching in turnstile streams. *ArXiv e-prints*, 2015.

[29] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Proceedings of the 16th Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 231–242, 2012.

[30] Yi Li, Huy L. Nguyen, and David P. Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1562–1581, 2014.

[31] Yi Li and David P. Woodruff. On approximating functions of the singular values in a stream. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 726–739, 2016.

[32] Yi Li and David P. Woodruff. Tight bounds for sketching the operator norm, schatten norms, and subspace embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, pages 39:1–39:11, 2016.

[33] László Lovász. On determinants, matchings, and random algorithms. In *Proceedings of the 2nd Conference on Fundamentals of Computation Theory (FCT)*, pages 565–574, 1979.

[34] Madhusudan Manjunath, Kurt Mehlhorn, Konstantinos Panagiotou, and He Sun. Approximate counting of cycles in streams. In *Algorithms - ESA 2011 - 19th Annual European Symposium, Saarbrücken, Germany, September 5-9, 2011. Proceedings*, pages 677–688, 2011.

[35] Andrew McGregor. Finding graph matchings in data streams. In *Proceedings of the 9th Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 170–181, 2005.

[36] Andrew McGregor and Sofya Vorotnikova. Planar matching in streams revisited. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, pages 17:1–17:12, 2016.

[37] Michael O. Rabin and Vijay V. Vazirani. Maximum matchings in general graphs through randomization. *J. Algorithms*, 10(4):557–567, 1989.

[38] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[39] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 22:107–111, 1947.

[40] Ryuhei Uehara and Zhi-Zhong Chen. Parallel approximation algorithms for maximum weighted matching in general graphs. *Information Processing Letters*, 76(1-2):13–17, 2000.

[41] Ashwinkumar Badanidiyuru Varadaraja. Buyback problem - approximate matroid intersection with cancellation costs. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, pages 379–390, 2011.

[42] Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 11–25, 2011.

[43] Mariano Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, 2012.