Estimating Weighted Matchings in $o(n)$ Space

Elena Grigorescu∗ Morteza Monemizadeh† Samson Zhou‡
April 27, 2016

Abstract

We consider the problem of estimating the weight of a maximum weighted matching of a weighted graph $G(V,E)$ whose edges are revealed in a streaming fashion. We develop a reduction from the maximum weighted matching problem to the maximum cardinality matching problem that only doubles the approximation factor of a streaming algorithm developed for the maximum cardinality matching problem. Our results hold for the insertion-only and the dynamic (i.e., insertion and deletion) edge-arrival streaming models. The previous best-known reduction is due to Bury and Schwiegelshohn (ESA 2015) who develop an algorithm whose approximation guarantee scales by a polynomial factor.

As an application, we obtain improved estimators for weighted planar graphs and, more generally, for weighted bounded-arboricity graphs, by feeding into our reduction the recent estimators due to Esfandiari et al. (SODA 2015) and to Chitnis et al. (SODA 2016). In particular, we obtain a $(48 + \epsilon)$-approximation estimator for the weight of a maximum weighted matching in planar graphs.

1 Introduction

We study the problem of estimating the weight of a maximum weighted matching in a weighted graph $G(V,E)$ whose edges arrive in a streaming fashion. Computing a maximum cardinality matching (MCM) in an unweighted graph and a maximum weighted matching (MWM) of a weighted graph are fundamental problems in computational graph theory (e.g., [25], [13]).

Recently, the MCM and MWM problems have attracted a lot of attention in modern big data models such as streaming (e.g., [12, 24, 23, 11, 16, 2, 17, 3]), online (e.g., [5, 21, 6]), MapReduce (e.g., [22]) and sublinear-time (e.g., [4, 26]) models.

Formally, the Maximum Weighted Matching problem is defined as follows.

Definition 1 (Maximum Weighted Matching (MWM)) Let $G(V,E)$ be an undirected weighted graph with edge weights $w : E \rightarrow \mathbb{R}^+$. A matching $M$ in $G$ is a set of pairwise non-adjacent edges; that is, no two edges share a common vertex. A matching $M$ is called a maximum weighted matching of graph $G$ if its weight $w(M) = \sum_{e \in M} w(e)$ is maximum.

∗Department of Computer Science, Purdue University, West Lafayette, IN. Email: elena-g@purdue.edu.
†Rutgers University, Piscataway, NJ 08854, USA. Supported by NSF CCF 1535878, IIS 1447793 and CCF 1161151. Email: mortezam@dimacs.rutgers.edu.
‡Department of Computer Science, Purdue University, West Lafayette, IN. Email: samsonzhou@gmail.com.
If the graph $G$ is unweighted (i.e., $w : E \rightarrow \{1\}$), the maximum weighted matching problem becomes the Maximum Cardinality Matching (MCM) problem.

In streaming models, the input graph is massive and the algorithm can only use a small amount of working space to solve a computational task. In particular, the algorithm cannot store the entire graph $G = (V, E)$ in memory, but can only operate with a sublinear amount of space, preferably $o(n)$, where $|V| = n$. However, many tasks are not solvable in this amount of space, and in order to deal with such problems, the semi-streaming model [12] was proposed, which allows $O(n \text{ polylog}(n))$ amount of working space. Both these settings have been studied in the adversarial model, where the edge order may be worst-case, and in the random order model, where the order of the edges is a uniformly random permutation of the set of edges.

For matching problems, if the goal is to output a set of edges that approximates the optimum matching, algorithms that maintain only $O(n)$ edges cannot achieve better than $(e/e - 1)$-approximation ratio ([14], [19]). Showing upper bounds has drawn a lot of recent interest (e.g., [12], [20], [23], [27], [10]), including a recent result [15] showing a 3.5-approximation, which improves upon the previous 4-approximation of [9].

If, on the other hand, the goal is to output only an estimate of the size of the matching, and not a matching itself, algorithms that use only $o(n)$ space are both desirable and possible. Surprisingly, very little is known about MWM/MCM in this model. Recent work by Kapralov et al. [18] shows the first polylog($n$) approximate estimator using only polylog($n$) space for the MCM problem. Further, if $\tilde{O}(n^{2/3})$ space is allowed, then constant factor approximation algorithms are possible [11].

In a recent work, Bury and Schwiegelshohn [7] consider the MWM problem in $o(n)$ space, showing a reduction to the MCM problem, that scales the approximation factor polynomially. In particular, they are the first to show a constant factor estimator for weighted graphs with bounded arboricity. Their results hold in the adversarial insertion-only model (where the updates are only edge insertion), and in the dynamic models (where the updates are both edge insertion and deletion). They also provide an $\Omega(n^{1-\epsilon})$ space lower bound to estimate the matching within $1 + O(\epsilon)$. Our results significantly improve the current best-known upper bounds of [7], as detailed in the next section.

## 2 Our Contribution

We show a reduction from MWM to MCM that preserves the approximation within a factor of $2(1 + \epsilon)$. Specifically, given a $\lambda$-approximation estimation for the size of a maximum cardinality matching, the reduction provides a $(2(1 + \epsilon) \cdot \lambda)$-approximation estimation of the weight of a maximum weighted matching. Our algorithm works both in the insertion-only streaming model, and in the dynamic setting. In both these models the edges appear in adversarial order.

We next state our main theorem. As it is typical for sublinear space algorithms, we assume that the edge-weights of $G = (V, E)$ are bounded by $\text{poly}(n)$.

### Theorem 2

Suppose there exists a streaming algorithm (in insertion-only, or dynamic streaming model) that estimates the size of a maximum cardinality matching of an unweighted graph within a factor of $\lambda$, with probability at least $(1 - \delta)$, using $S(n, \delta)$ space. Then, for every $\epsilon > 0$, there exists a streaming algorithm that estimates the weight of a maximum weighted matching of a weighted graph within a factor of $2\lambda(1 + \epsilon)$, with probability at least $(1 - \delta)$, using $O \left( S \left( n, \frac{\delta}{\epsilon \cdot \log n} \right) \log n \right)$ space.
We remark that if the estimator for MCM is specific to a monotone graph property (a property of graphs that is closed under edge removal), then our algorithm can use it as a subroutine to obtain an estimator for MWM in the weighted versions of the graphs with such properties (instead of using a subroutine for general graphs, which may require more space, or provide worse approximation guarantees).

Our result improves the result of [7], who show a reduction from MWM to MCM that achieves a $O(\lambda^4)$-approximation estimator for MWM, given a $\lambda$-approximation estimator for MCM. Their reduction also allows extending MCM estimators to MWM estimators in monotone graph properties.

In particular, using specialized estimators for graphs of bounded arboricity, we obtain improved approximation guarantees compared with the previous best results of [7], as explained in Section 2.1, e.g., Table 2.1. In addition, our algorithm is natural and allows for a clean analysis.

2.1 Applications

Theorem 2 has immediate consequences for computing MWM in graphs with bounded arboricity. A graph $G = (V, E)$ has arboricity $\nu$ if

$$\nu = \max_{U \subseteq V} \frac{|E(U)|}{|U| - 1},$$

where $E(U)$ is the subset of edges with both endpoints in $U$. The class of graphs with bounded arboricity includes several important families of graphs, such as planar graphs, or more generally, graphs with bounded degree, genus, or treewidth. Note that these families of graphs are monotone.

**Theorem 3** Let $G$ be a weighted graph with arboricity $\nu$ and $n = \omega(\nu^2)$ vertices. Let $\epsilon, \delta \in (0, 1)$. Then, there exists an algorithm that estimates the weight of a MWM in $G$ within a $2\lambda$-approximation factor, where $\lambda = (5\nu + 9)(1 + \epsilon)$, in the insertion-only streaming model, with probability at least $1 - \delta$, using $\tilde{O}(\nu \epsilon^{-2} \log(\delta^{-1}) n^{2/3})$ space. Both the update time and final processing time are $\tilde{O}(\log(\delta^{-1}) \log n)$.

In particular, for planar graphs, $\nu = 3$ and by choosing $\delta = n^{-1}$ in Theorem 3 and $\epsilon$ as a small constant, the output of our algorithm is within $(48 + \epsilon)$-approximation factor of a MWM, with probability at least $1 - \frac{1}{n}$, using $\tilde{O}(n^{2/3})$ space. The previous result of [7] gave an approximation factor of $> 3 \cdot 10^6$ for planar graphs.

Table 2.1 summarizes the state of the art for MWM.

**Graphs with Bounded Arboricity in the Dynamic Model** Our results also apply to the dynamic model. Here we make use of the recent result of Chitnis *et al.* [8] that provides an estimator for MCM in the dynamic model (See Theorem 6 in the Preliminaries).

Again, Theorem 6 satisfies the conditions of Theorem 2 with $\lambda = (5\nu + 9)(1 + \epsilon)$, and consequently, we have the following application.

\[1 \tilde{O}(f) = \tilde{O}(f \cdot (\log n)^c)\] for a large enough constant $c$. 

3
Approximation for Planar Graphs  Approximation for Graphs with Arboricity ν

| [7] | > 3 \cdot 10^9 | 12(5ν + 9)^3 |
|-----|---------------|---------------|
| Here | 48 + ε | 2(5ν + 9) + ε |

Table 2.1: The insertion-only streaming model requires $\tilde{O}(νe^{-2} \log(δ^{-1})n^{2/3})$ space for all graph classes, while the dynamic streaming model requires $\tilde{O}(νe^{-2} \log(δ^{-1})n^{1/3})$ space for all graph classes.

**Theorem 4** Let $G$ be a weighted graph with arboricity $ν$ and $n = \omega(ν^2)$ vertices. Let $ε, δ \in (0, 1)$. Then, there exists an algorithm that estimates the weight of a maximum weighted matching in $G$ within a $2(5ν + 9)(1+ε)$-factor in the dynamic streaming model with probability at least $(1−δ)$, using $\tilde{O}(νe^{-2} \log(δ^{-1})n^{4/5})$ space. Both the update time and final processing time are $\tilde{O}(\log(δ^{-1}) \log n)$.

In particular, for planar graphs, $ν = 3$, and by choosing $δ = n^{-1}$ and $ε$ as a small constant, the output of our algorithm is a $(48 + ε)$-approximation of the weight of a maximum weighted matching with probability at least $1 − \frac{1}{n}$ using at most $\tilde{O}(n^{4/5})$ space.

We further remark that if 2-passes over the stream are allowed, then we may use the recent results of [8] to obtain a $(2(5ν + 9)(1+ε))$-approximation algorithm for MWM using only $\tilde{O}(\sqrt{n})$ space.

### 2.2 Overview

We start by splitting the input stream into $O(\log n)$ substreams $S_1, S_2, \ldots$, such that substream $S_i$ contains every edge $e \in E$ whose weight is at least $(1+ε)^i$, that is, $w(e) \geq (1+ε)^i$. Splitting the stream into sets of edges of weight only bounded below was used in [9], leading to better approximation algorithms for MWM in the semi-streaming model.

For each substream $S_i$, we treat its edges as unweighted edges and apply a MCM estimator. We then implicitly apply a greedy strategy, where we iteratively add as many edges possible from the remaining substreams of highest weight, tracking an estimate for both the weight of a maximum weighted matching, and the number of edges in the corresponding matching. The details of the algorithm appear in Section 4.

Our key observation is that at any point, any edge in our MWM estimator can conflict with at most two edges in the MCM estimator. Therefore, if the MCM estimator for a certain substream is greater than double the number of edges in the associated matching, we add the remaining edges to our estimator, as shown below in Figure 2.2.

More formally, for each $i$, let $U_i^*$ be a maximum cardinality matching for $S_i$. Then each edge of $U_i^*$ intersects with either one, or two edges of $U_j^*$, for all $j < i$. Thus, if $|U_{i-1}^*| > 2|U_i^*|$, then at least $|U_{i-1}^*| - 2|U_i^*|$ edges from $U_{i-1}^*$ can be added to $U_i^*$ while remaining a matching. We use a variable $B_i$ to serve as an estimator for this lower bound on the number of edges in a maximum weighted matching, including edges from $U_i^*$, for $j \geq i$. We then use the estimator for MCM in each substream $i$ as a proxy for $U_i^*$.

Our algorithm differs from the algorithm of [7] in several points. They consider substreams $S_i$ containing the edges with weight $[2^i, 2^{i+1})$, and their algorithm estimates the number of each edges in each stream, and chooses to include the edges if both the number of the edges and their combined weight exceed certain thresholds, deemed to contribute a significant value to the estimate. However, this approach may not capture a small number of edges which nonetheless contribute a significant weight.

Our greedy approach is able to handle both these facets of a MWM problem. Namely, by greedily taking as many edges as possible from the heavier substreams, and then accounting for edges that may be
edges which are inserted (revealed) up to time $i \in \mathbb{N}$. The graph $G$ be an unweighted graph with arboricity $\omega$. Let $\nu \in \mathbb{N}$ be two arbitrary positive values less than one. There exists an algorithm that estimates the size of a maximum matching in $G$ within a $O(\log |S| + |S|)$-factor in the insertion-only streaming model with probability at least $1 - \delta$, using $O(\log |S|)$ space. Both the update time and final processing time are $O(\log |S|)$. In particular, for planar graphs, we can $(2 + \epsilon)$-approximate the size of a maximum matching with probability at least $1 - \delta$ using $O(n^{2/3})$ space.

Theorem 5 [11] Let $G$ be an unweighted graph with arboricity $\nu$ and $n = \omega(\nu^2)$ vertices. Let $\epsilon, \delta \in (0, 1)$ be two arbitrary positive values less than one. There exists an algorithm that estimates the size of a maximum matching in $G$ within a $O(\log |S| + |S|)$-factor in the dynamic streaming model with probability at least $1 - \delta$, using $O(\log |S|)$ space. Both the update time and final processing time are $O(\log |S|)$. In particular, for planar graphs, we can $(2 + \epsilon)$-approximate the size of a maximum matching with probability at least $1 - \delta$ using $O(n^{2/3})$ space.

Theorem 6 [8] Let $G$ be an unweighted graph with arboricity $\nu$ and $n = \omega(\nu^2)$ vertices. Let $\epsilon, \delta \in (0, 1)$ be two arbitrary positive values less than one. There exists an algorithm that estimates the size of a maximum matching in $G$ within a $O(\log |S| + |S|)$-factor in the dynamic streaming model with probability at least $1 - \delta$, using $O(\log |S|)$ space. Both the update time and final processing time are $O(\log |S|)$. In particular, for planar graphs, we can $(2 + \epsilon)$-approximate the size of a maximum matching with probability at least $1 - \delta$ using $O(n^{2/3})$ space.

3 Preliminaries

Let $S$ be a stream of insertions of edges of an underlying undirected weighted graph $G(V, E)$ with weights $w : E \rightarrow \mathbb{R}$. We assume that vertex set $V$ is fixed and given, and the size of $V$ is $|V| = n$. Observe that the size of stream $S$ is $|S| \leq \binom{n}{2} = \frac{n(n-1)}{2} \leq n^2$, so that we may assume that $O(\log |S|) = O(\log n)$. Without loss of generality we assume that at time $i$ of stream $S$, edge $e_i$ arrives (or is revealed). Let $E_i$ denote those edges which are inserted (revealed) up to time $i$, i.e., $E_i = \{e_1, e_2, e_3, \ldots, e_i\}$. Observe that at every time $i \in |S|$ we have $|E_i| \leq \binom{i}{2} \leq n^2$, where $[x] = \{1, 2, 3, \ldots, x\}$ for some natural number $x$. We assume that at the end of stream $S$ all edges of graph $G(V, E)$ arrived, that is, $E = E_{|S|}$.

We assume that there is a unique numbering for the vertices in $V$ so that we can treat $v \in V$ as a unique number $\nu$ for $1 \leq \nu \leq n = |V|$. We denote an undirected edge in $E$ with two endpoints $u, v \in V$ by $(u, v)$. The graph $G$ can have at most $\binom{n}{2} = n(n-1)/2$ edges. Thus, each edge can also be thought of as referring to a unique number between 1 and $\binom{n}{2}$.

The next theorems imply our results for graphs with bounded arboricity in the insert-only and dynamic models.

Figure 2.2: If $|U^*_i| > 2|U^*_{i-1}|$, then some edge(s) from $U^*_{i-1}$ can be added while maintaining a matching.
4 Algorithm

For a weighted graph $G(V, E)$ with weights $w : E \rightarrow \mathbb{R}$ such that the minimum weight of an edge is at least 1 and the maximum weight $W$ of an edge is polynomially bounded in $n$, i.e., $W = n^c$ for some constant $c$, for $T = \lceil \log_{1+\epsilon} W \rceil$, we create $T + 1$ substreams such that substream $S_i = \{e \in S : w(e) \geq (1 + \epsilon)^i\}$.

Given access to a streaming algorithm MCM Estimator which estimates the size of a maximum cardinality matching of an unweighted graph $G$ within a factor of $\lambda$ with probability at least $(1 - \delta)$, we use MCM Estimator as a black box algorithm on each $S_i$ and record the estimates. In general, for a substream $S_i$, we track an estimate $A_i$, of the weight of a maximum weighted matching of the subgraph whose edges are in the substream $S_i$, along with an estimate, $B_i$, which represents the number of edges in our estimate $A_i$. The estimator $B_i$ also serves as a running lower bound estimator for the number of edges in a maximum matching. We greedily add edges to our estimation of the weight of a maximum weighted matching of graph $G$. Therefore, if the estimator $\hat{M}_{i-1}$ for the maximum cardinality matching of the substream $S_{i-1}$ is more than double the number of edges in $B_i$ represented by our estimate $A_i$ of the substream $S_i$, we let $B_{i-1}$ be $B_i$ plus the difference $\hat{M}_{i-1} - 2B_i$, and let $A_{i-1}$ be $A_i$ plus $(\hat{M}_{i-1} - 2B_i) \cdot (1 + \epsilon)^{i-1}$. We iterate through the substream estimators, starting from the substream $S_T$ of largest weight, and proceeding downward to substreams of lower weight. We initialize our greedy approach by setting $B_T = \hat{M}_T$, equivalent to taking all edges in $\hat{M}_T$.

**Algorithm 1 Estimating Weighted Matching in Data Streams**

**Input:** A stream $S$ of edges of an underlying graph $G(V, E)$ with weights $w : E \rightarrow \mathbb{R}^+$ such that the maximum weight $W$ of an edge is polynomially bounded in $n$, i.e., $W = n^c$ for some constant $c$.

**Output:** An estimator $\hat{A}$ of $w(M^*)$, the weight of a maximum weighted matching $M^*$, in $G$.

1. Let $A_1$ be a running estimate for the weight of a maximum weighted matching.
2. Let $B_1$ be a running lower bound estimate for the number of edges in a maximum weighted matching.
3. Initialize $A_{T+1} = 0$, $B_{T+1} = 0$, and $\hat{M}_{T+1} = 0$.
4. for $i = T$ to $i = 0$ do
5. Let $S_i = \{e \in S : w(e) \geq (1 + \epsilon)^i\}$ be a substream of $S$ of edges whose weights are at least $(1 + \epsilon)^i$.
6. Let $S'_i$ be unweighted versions of edges in $S_i$.
7. Let $\hat{S}_i'$ be the output of MCM Estimator for each $S''_i$ with parameter $\delta' = \frac{\delta}{T}$.
8. Let $\hat{M}_i = \max(\hat{M}_{i+1}, S'_i)$.
9. Set $\Delta_i = \max(0, [\hat{M}_i - 2B_{i+1}])$.
10. Update $B_i = B_{i+1} + \Delta_i$.
11. Update $A_i = A_{i+1} + (1 + \epsilon)^i \Delta_i$.
12. Output estimate $\hat{A} = A_0$.

We note that the quantities $A_i$ and $B_i$ satisfy the following properties, which will be useful in the analysis.

**Observation 7** $A_i = \sum_{i=0}^{T} (1 + \epsilon)^i \Delta_i$

**Observation 8** $B_j = \sum_{i=j}^{T} \Delta_i$

5 Analysis

**Lemma 9** For all $i$, $B_i \leq \hat{M}_i \leq 2B_i$. 

6
Proof: We prove the statement by induction on $i$, starting from $i = T$ down to $i = 0$. For the base case $i = T$, we initialize $B_{i+1} = 0$. In particular, $\Delta_i = \hat{M}_i$, so $B_i = B_{i+1} + \Delta_i = \hat{M}_i$, and the desired inequality follows.

Now, we suppose the claim is true for $B_{i+1} \leq \hat{M}_{i+1} \leq 2B_{i+1}$. Next, we prove it for $B_i \leq \hat{M}_i \leq 2B_i$. To prove the claim for $i$ we consider two cases. The first case is when $2B_{i+1} < \hat{M}_i$. Then

$$B_i = B_{i+1} + \Delta_i \quad \text{(By definition)}$$

$$= B_{i+1} + \hat{M}_i - 2B_{i+1} \quad \text{(} \Delta_i = \hat{M}_i - 2B_{i+1} \text{)}$$

$$= \hat{M}_i - B_{i+1}$$

$$\leq \hat{M}_i$$

Additionally,

$$\hat{M}_i < \hat{M}_i + (\hat{M}_i - 2B_{i+1}) \quad \text{(} 2B_{i+1} < \hat{M}_i \text{)}$$

$$= 2(B_{i+1} + (\hat{M}_i - 2B_{i+1}))$$

$$= 2(B_{i+1} + \Delta_i) \quad \text{(} \Delta_i = \hat{M}_i - 2B_{i+1} \text{)}$$

$$= 2B_i \quad \text{(By definition)}$$

and so $B_i \leq \hat{M}_i \leq 2B_i$.

The second case is when $\hat{M}_i \leq 2B_{i+1}$. Then, by definition, $B_i = B_{i+1}$. Since $S_{i+1}'$ is a subset of $S_i'$, then

$$B_i = B_{i+1} \leq \hat{M}_{i+1} \quad \text{(Inductive hypothesis)}$$

$$\leq \hat{M}_i \quad \text{(} \hat{M}_i = \max(\hat{M}_{i+1}, S_i') \text{)}$$

$$\leq 2B_{i+1} = 2B_i \quad \text{(} \hat{M}_i \leq 2B_{i+1} \text{)}$$

and again $B_i \leq \hat{M}_i \leq 2B_i$, which completes the proof. $\Box$

Corollary 10 Suppose for all $i$, the estimator $\hat{M}_i$ satisfies $\hat{M}_i \leq |U_i^*| \leq \lambda \hat{M}_i$, where $U_i^*$ is the size of a maximum cardinality matching of $S_i^*$. Then $B_i \leq |U_i^*| \leq 2\lambda B_i$.

Proof: By Lemma 9, $\hat{M}_i \leq 2B_i$, so then $\lambda \hat{M}_i \leq 2\lambda B_i$. Similarly, by Lemma 9, $B_i \leq \hat{M}_i$. But by assumption, $\hat{M}_i \leq |U_i^*| \leq \lambda \hat{M}_i$, and so

$$B_i \leq \hat{M}_i \leq |U_i^*| \leq \lambda \hat{M}_i \leq 2\lambda B_i.$$  

$\Box$

Lemma 11 Suppose for all $i$, the estimator $\hat{M}_i$ satisfies $\hat{M}_i \leq |U_i^*| \leq \lambda \hat{M}_i$, where $U_i^*$ is the size of a maximum cardinality matching of $S_i^*$. Then, for all $j$,

$$\sum_{i=j}^{T} \Delta_i \leq \sum_{i=j}^{T} |M^* \cap (S_j - S_{j+1})| \leq \sum_{i=j}^{T} 2\lambda \Delta_i,$$

where $M^*$ is a maximum weighted matching.
Proof: Since $M^*$ is a matching, then the number of edges in $M^*$ with weight at least $(1+\epsilon)^j$ is at most $|U_j^*|$. Thus,

$$\sum_{i=j}^{T} |M^* \cap (S_j - S_{j+1})| \leq |U_j^*|.$$

Note that by Observation 8, $\sum_{i=j}^{T} \Delta_i = B_j$, so then by Corollary 10

$$\sum_{i=j}^{T} |M^* \cap (S_j - S_{j+1})| \leq 2\lambda \sum_{i=j}^{T} \Delta_i.$$

On the other hand, $B_i$ is a running estimate of the lower bound on the number of edges in $M^* \cap S_i$, so

$$\sum_{i=j}^{T} \Delta_i = B_j \leq \sum_{i=j}^{T} |M^* \cap (S_j - S_{j+1})|,$$

as desired. $\square$

Lemma 12 With probability at least $1 - \delta$, the estimator $\hat{M}_i$ satisfies $|\hat{M}_i| \leq |U_i^*| \leq \lambda \hat{M}_i$ for all $i$, where $U_i^*$ is the maximum cardinality matching of $S_i$.

Proof: Since $\hat{M}_i \leq |U_i^*| \leq \lambda \hat{M}_i$ succeeds with probability at least $1 - \delta$, then the probability $\hat{M}_i$ succeeds for $i = 1, 2, \ldots, T$ is at least $1 - \delta$ by a union bound. $\square$

We now prove our main theorem.

Proof of Theorem 2: We complete the proof of Theorem 2 by considering the edges in a maximum weighted matching $M^*$. We partition these edges by weight and bound the number of edges in each partition. We will show that $A_0 \leq w(M^*) \leq 2\lambda(1+\epsilon)A_0$. First, we have

$$w(M^*) = \sum_{e \in M^*} w(e)$$

$$= \sum_{i=0}^{T} \sum_{e \in M^* \cap (S_i - S_{i+1})} w(e)$$

$$\leq \sum_{i=0}^{T} \sum_{e \in M^* \cap (S_i - S_{i+1})} (1+\epsilon)^{i+1}$$

$$\leq \sum_{i=0}^{T} |M^* \cap (S_i - S_{i+1})|(1+\epsilon)^{i+1}$$

$$\leq \sum_{i=0}^{T} 2\lambda \Delta_i(1+\epsilon)^{i+1}$$

$$\leq 2\lambda(1+\epsilon) \sum_{i=0}^{T} \Delta_i(1+\epsilon)^i = 2\lambda(1+\epsilon)A_0,$$
where the identity in line (2) results from partitioning the edges by weight, so that \( e \in M^* \) appears in \( S_i - S_{i+1} \) if \( (1 + e)^i \leq w(e) < (1 + e)^{i+1} \). The inequality in line (3) results from each edge \( e \) in \( S_i - S_{i+1} \) having weight less than \( (1 + e)^{i+1} \), so an upper bound on the sum of the weights of edges in \( M^* \cap (S_i - S_{i+1}) \) is \( (1 + e)^{i+1} \) times the number of edges in \( |M^* \cap (S_i - S_{i+1})| \), as shown in line (4). By Lemma 11 the partial sums of \( 2\lambda \Delta_i \) dominates the partial sums of \( |M^* \cap (S_i - S_{i+1})| \), resulting in the inequality in line (5). The final identity in line (6) results from Observation 7.

\[
\begin{align*}
    w(M^*) &= \sum_{e \in M^*} w(e) \\
    &= \sum_{i=0}^{T} \sum_{e \in M^* \cap (S_i - S_{i+1})} w(e) \\
    &\geq \sum_{i=0}^{T} \sum_{e \in M^* \cap (S_i - S_{i+1})} (1 + e)^i \\
    &\geq \sum_{i=0}^{T} |M^* \cap (S_i - S_{i+1})|(1 + e)^i \\
    &\geq \sum_{i=0}^{T} \Delta_i (1 + e)^i \\
    &\geq \sum_{i=0}^{T} A_i = A_0, \\
\end{align*}
\]

where the identity in line (2) again results from partitioning the edges by weight, so that \( e \in M^* \) appears in \( S_i - S_{i+1} \) if \( (1 + e)^i \leq w(e) < (1 + e)^{i+1} \). The inequality in line (3) results from each edge \( e \) in \( S_i - S_{i+1} \) having weight at least \( (1 + e)^i \), so a lower bound on the sum of the weights of edges in \( M^* \cap (S_i - S_{i+1}) \) is \( (1 + e)^i \) times the number of edges in \( |M^* \cap (S_i - S_{i+1})| \), as shown in line (4). By Lemma 11 the partial sums of \( |M^* \cap (S_i - S_{i+1})| \) dominates the partial sums of \( \Delta_i \), resulting in the inequality in line (5). The final identity in line (6) results from Observation 7.

Thus, \( \hat{A} = A_0 \) is a \( 2\lambda (1 + e) \)-approximation for \( w(M^*) \).

Note that the assumption of Lemma 11 holds with probability at least \( 1 - \delta \) by Lemma 12. Since we require \( \tilde{M}_i \leq |U_i^*| \leq \lambda \tilde{M}_i \) with probability at least \( 1 - \frac{\delta}{T} \), then \( \mathcal{S} \left( n, \frac{\delta}{c \log n} \right) \) space is required for each estimator. Since \( T = \log W \) substreams are used and \( W \leq n^c \) for some constant \( c \), then the overall space necessary is \( \mathcal{S} \left( n, \frac{\delta}{c \log n} \right) (c \log n) \). This completes the proof. \( \square \)

References

[1] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms; 9
[2] Abhash Anand, Surender Baswana, Manoj Gupta, and Sandeep Sen. Maintaining approximate maximum weighted matching in fully dynamic graphs. In IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India, pages 257–266, 2012.

[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] S. Baswana, M. Gupta, and S. Sen. Fully dynamic maximal matching in $O(\log n)$ update time. In Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS), pages 383–392, 2011.

[5] Benjamin E. Birnbaum and Claire Mathieu. On-line bipartite matching made simple. SIGACT News, 39(1):80–87, 2008.

[6] Bartlomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Shortest augmenting paths for online matchings on trees. In Approximation and Online Algorithms - 13th International Workshop, WAOA 2015, Patras, Greece, September 17-18, 2015. Revised Selected Papers, pages 59–71, 2015.

[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] M. Crouch and D. S. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In Proceedings of the 17th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM), pages 96–104, 2014.

[10] L. Epstein, A. Levin, J. Mestre, and D. Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. SIAM J. Discrete Math, 25(3):1251–1265, 2011.

[11] H. Esfandiari, M. T. Hajiaghayi, V. Liaghat, M. Monemizadeh, and K. Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In SODA, pages 1217–1233, 2015.

[12] 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.

[13] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 434–443, 1990.
[14] 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.

[15] Elena Grigorescu, Morteza Monemizadeh, and Samson Zhou. A 3.5-approximation for maximum weighted matching in data streams. Manuscript, 2016.

[16] S. Guha and A. McGregor. Graph synopses, sketches, and streams: A survey. PVLDB, 5(12):2030–2031, 2012.

[17] Manoj Gupta and Richard Peng. Fully dynamic (1+ ε)-approximate matchings. In 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA, pages 548–557, 2013.

[18] M. Kapralov, S. Khanna, and M. 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.

[19] 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.

[20] C. Konrad, F. Magniez, and C. Mathieu. Maximum matching in semi-streaming with few passes. In Proceedings of the 15th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), pages 231–242, 2012.

[21] Nitish Korula, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Bicriteria online matching: Maximizing weight and cardinality. In Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings, pages 305–318, 2013.

[22] Silvio Lattanzi, Benjamin Moseley, Siddharth Suri, and Sergei Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011), pages 85–94, 2011.

[23] A. McGregor. Finding graph matchings in data streams. In Proceedings of the of 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), pages 170–181, 2005.

[24] Andrew McGregor. Graph mining on streams. In Encyclopedia of Database Systems, pages 1271–1275. Springer, 2009.

[25] S. Micali and V. V. Vazirani. An o(√|V||e|) algorithm for finding maximum matching in general graphs. Proceedings of the 21st IEEE Symposium on Foundations of Computer Science (FOCS), pages 17–27, 1980.

[26] O. Neiman and S. Solomon. Simple deterministic algorithms for fully dynamic maximal matching. Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC), 2013.

[27] M. Zelke. Weighted matching in the semi-streaming model. Algorithmica, 62(1):1–12, 2012.