Maximum Matching in Semi-Streaming with Few Passes *

Christian Konrad[†1,2], Frédéric Magniez[‡1], and Claire Mathieu[§3]

[1]LIAFA, Université Paris Diderot, Paris, France
[2]Université Paris-Sud, Orsay, France
[3]Computer Science Department, Brown University, Providence

Abstract

We present three semi-streaming algorithms for Maximum Bipartite Matching with one and two passes. Our one-pass semi-streaming algorithm is deterministic and returns a matching of size at least $1/2 + 0.005$ times the optimal matching size in expectation, assuming that edges arrive one by one in (uniform) random order. Our first two-pass algorithm is randomized and returns a matching of size at least $1/2 + 0.019$ times the optimal matching size in expectation (over its internal random coin flips) for any arrival order. These two algorithms apply the simple Greedy matching algorithm several times on carefully chosen subgraphs as a subroutine. Furthermore, we present a two-pass deterministic algorithm for any arrival order returning a matching of size at least $1/2 + 0.019$ times the optimal matching size. This algorithm is built on ideas from the computation of semi-matchings.

1 Introduction

Streaming. Classical algorithms assume random access to the input. This is a reasonable assumption until one is faced with massive data sets as in bioinformatics for genome decoding, Web databases for the search of documents, or network monitoring. The input may then be too large to fit into the computer's memory. A typical situation is a continuous flow of traffic logs sent to a router. Streaming algorithms sequentially scan the input piece by piece in one pass, while using sublinear memory space. The analysis of Internet traffic [2] was one of the first applications of such algorithms. A similar but slightly different situation arises when the input is recorded on an external storage device where only sequential access is possible, such as optical disks, or even hard drives. Then a small number of passes, ideally constant, can be performed.

By sublinear memory one ideally means memory that is polylogarithmic in the size of the input. Nonetheless, polylogarithmic memory is often too restrictive for graph problems: as shown in [7], deciding basic graph properties such as bipartiteness or connectivity already requires $\Omega(n)$ space. Muthukrishnan [16] initially mentioned massive graphs as typical examples where one assumes a

*Supported by the French ANR Defis program under contract ANR-08-EMER-012 (QRAC project). Christian Konrad is supported by a Fondation CFM-JP Aguilar grant. Claire Mathieu is supported by NSF grant CCF-0964037.

†christian.konrad@liafa.jussieu.fr
‡frederic.magniez@univ-paris-diderot.fr
§claire@cs.brown.edu

semi-external model, that is, not the entire graph but the vertex set can be stored in memory. In that model, an n-vertex graph is given by a stream of edges arriving in arbitrary order. A *semi-streaming* algorithm has memory space $O(n \operatorname{polylog} n)$, and the graph vertices are usually known before processing the stream of edges.

Matching. In this paper we focus on an iconic graph problem: finding large matchings. In the semi-streaming model, the problem was primarily addressed by Feigenbaum, Kannan, McGregor, Suri and Zhang [6]. In the meantime a variety of semi-streaming matching algorithms for particular settings exist (unweighted/weighted, bipartite/general graphs). Most works, however, consider the multipass scenario [1, 4] where the goal is to find a $(1-\epsilon)$ approximation while minimizing the number of passes. The techniques are based on finding augmenting paths, and, recently, linear programming was also applied. Ahn and Guha [1] provide an overview of the current best algorithms. In this paper, we also take the augmenting paths route.

In the one-pass setting, in the unweighted case, the greedy matching algorithm is still the best known algorithm as far as we know. (We note that in the weighted case, progress was made [17] [5], but when the edges are unweighted those algorithms are of no help.) The greedy matching constructs a matching in the following online fashion: starting with an empty matching M, upon arrival of edge e, it adds e to M if $M+e$ remains a matching. A *maximal matching* is a matching that can not be enlarged by adding another edge to it. It is well-known that the cardinality of maximal matchings is at least half of the cardinality of maximum matchings. By construction, since the greedy matching is maximal, M is a 1/2-approximation of any maximum matching M^*, that is $|M| \geq |M^*|/2$. The starting point of this paper was to address the following question: **Is the greedy algorithm best possible, or is it possible to get an approximation ratio better than 1/2?**

In fact, a very recent result [8] rules out the possibility of any one-pass semi streaming algorithm for Maximum Bipartite Matching (MBM) with approximation ratio better than 2/3, since that would require memory space $n^{1+\Omega(1/\log\log n)}$. Nevertheless, there is still room between 1/2 and 2/3.

To get an approximation ratio better than 1/2, prior semi-streaming algorithms require at least 3 passes, for instance the algorithm of [4] can be used to run in 3 passes providing a matching strictly better than a 1/2 approximation.

Random order of edge arrivals. The behavior of the greedy matching algorithm has been studied extensively in a variety of settings. The most relevant reference [3] considers a (uniform) random order of edge arrivals. In that setting, Dyer and Frieze showed that the expected approximation ratio is still 1/2 for some graphs (their example can be extended to bipartite graphs), but can be better for particular graph classes such as planar graphs and forests.

In the context of streaming and semi-streaming algorithms, the model of random order arrival has been first studied for the problems of sorting and selecting in limited space by Munro and Paterson [15]. Then Guha and McGregor [9] gave an exponential separation between random order and adversarial order models. One justification of the random order model is to understand why certain problems do not admit a memory efficient streaming algorithms in theory, while in practice, heuristics are often sufficient.

Other related work. MBM was also intensively studied in the online setting, where nodes from one side arrive in adversarial order together with all their incident edges. In this model, the decision to take or discard an edge has to be taken before accessing the edges of the next vertex. The well-known randomized algorithm by Karp and Vazirani [11] achieves an approximation ratio of $(1-1/e)$ for bipartite graphs where all nodes from one side are known in advance, the nodes from the other side arrive online. They prove that their bound is optimal in the worst case. This barrier was broken only recently by modifying the worst case assumption (worst input graph and worst arrival order) to assume that, although the graph itself is worst-case, the arrival order is according to some (known or unknown) distribution [10, 12].

Our results. In this paper we present algorithms for settings in which we can beat $1/2$. We design semi-streaming algorithms for MBM with one and two passes. Our one-pass semi-streaming algorithm is deterministic and achieves an expected approximation ratio $1/2 + 0.005$ for any graph (**Theorem 1**), but has to assume that the edges arrive one by one in (uniform) random order. Our two two-pass semi-streaming algorithm do not need the random order assumption. We present a randomized two-pass algorithm with expected approximation ratio $1/2 + 0.019$ against its internal random coin flips, for any graph and for any arrival order (**Theorem 3**). Furthermore, we present a deterministic counterpart with the same approximation ratio for any graph and any arrival order (**Theorem 4**).

Techniques. The one-pass algorithm as well as the randomized two-pass algorithm apply each three times the greedy matching algorithm on different and carefully chosen subgraphs. The deterministic two-pass algorithm is slightly more complicated as it uses besides the greedy algorithm a subroutine that computes a particular semi-matching.

General idea common to all our algorithms: If we had three passes at our disposal (see for instance Algorithm 2 in [6]), we could use one pass to build a maximal matching M_0 between the two sides A and B of the bipartition, a second pass to find a matching M_1 between the A vertices matched in M_0 and the B vertices that are free w.r.t. M_0 whose combination with edges of M_0 forms paths of length 2. Finally, a third pass to find a matching M_2 between B vertices vertices matched in M_0 and A vertices that are free w.r.t. M_0 whose combination with M_0 and M_1 forms paths of length 3 that can be used to augment matching M_0. All our algorithms simulate these 3 passes in less passes.

One-pass algorithm for random arrival order: To simulate this with a single pass, we split the sequence of arrivals $[1, m]$ into three phases $[1, \alpha m]$, $(\alpha m, \beta m]$, and $(\beta m, m]$ and build M_0 during the first phase, M_1 during the second phase, and M_2 during the third phase. Of course, we see only a subset of the edges for each phase, but thanks to the random order arrival, these subsets are random, and, intuitively, we loose only a constant fraction in the sizes of the constructed matchings. As it turns out, the intuition can be made rigorous, as long as the first matching M_0 is maximal or close to maximal. We observe that, if the greedy algorithm, executed on the entire sequence of edges, produces a matching that is not much better than a $1/2$ approximation of the optimal maximum matching, then that matching *is built early on.* More precisely (Lemma 4), if the greedy matching on the entire graphs is no better than a $1/2 + \epsilon$ approximation, then after seeing a mere one third of the edges of the graph, the greedy matching is already a $1/2 - \epsilon$ approximation, so it is already close to maximal.

Randomized two-pass algorithm for any arrival order: Assume a bipartite graph (A, B, E) comprising a perfect matching. If A' is a small random subset of A, then, regardless of the arrival order, the greedy algorithm that constructs a greedy matching between A' and B (that is, the greedy algorithm restricted to the edges that have an endpoint in A') will find a matching that is near-perfect, that is, almost every vertex of A' is matched (see Theorem 2 for a slightly more general version of this statement). This property of the greedy algorithm may be of independent interest. Then, in one pass we compute a greedy matching M_0 and also via the greedy algorithm independently and in parallel a matching M_1 between a subset $A' \subset A$ and the B vertices. It turns out that $M_0 \cup M_1$ comprise some length 2 paths that can be completed to 3-augmenting paths by a third matching M_2 that we compute in the second pass.

Deterministic two-pass algorithm for any arrival order: Again, assume a bipartite graph (A, B, E) comprising a perfect matching and some integer λ. Add now greedily edges ab to a set S if the degree of a in S is yet 0, and the degree of b is smaller than λ. This algorithm computes an *incomplete semi-matching* with a degree limitation λ on the B nodes. In the first pass, we run this algorithm in parallel to the greedy matching algorithm for constructing M_0. S replaces the computation of M_1, and we will that there are length 2 paths in $M_0 \cup S$ that can be completed to 3-augmenting paths in the second pass via a further greedy matching M_2.

Extension to general graphs. All algorithms presented in this paper generalize to non-bipartite graphs. When searching for augmenting paths in general graphs, algorithms have to cope with the fact that a candidate edge for an augmenting path may form an undesired triangle with the edge to augment and an optimal edge. In this case, the candidate edge can block the entire augmenting path. McGregor [13] overcomes this problem by repeatedly sampling bipartite graphs from the general graph. Such a strategy, however, is not necessary for our randomized algorithms. Since these make use of randomness (either over all input sequences, or over internal random coins), we show that undesired triangles simply do not appear *too often* allowing our techniques to still work. For our deterministic two-pass algorithm, a direct combinatorial argument can be used to bound the number of triangles.

2 Preliminaries

Notations and Definitions. Let $G = (A, B, E)$ be a bipartite graph with $V = A \cup B$, $n = |V|$ vertices and $m = |E|$ edges. For an edge $e \in E$ with end points $a \in A$ and $b \in B$, we denote e by ab. The input G is given as a sequence of edges arriving one by one in some order. Let $\Pi(G)$ be the set of all edge sequences of G.

Definition 1 (Semi-Streaming Algorithm)**.** *A k-pass* semi-streaming algorithm ***A*** *with processing time per letter t is an algorithm such that, for every input stream* $\pi \in \Pi(G)$ *encoding a graph G with n vertices: (1)* ***A*** *performs in total at most k passes on stream* π*, (2)* ***A*** *maintains a memory space of size* $O(n \operatorname{polylog} n)$*, (3)* ***A*** *has running time* $O(t)$ *per edge.*

For a subset of edges F, we denote by $\text{opt}(F)$ a matching of maximum size in the graph G restricted to edges F. We may write $\text{opt}(G)$ for $\text{opt}(E)$, and we use $M^* = \text{opt}(G)$. We say that an algorithm **A** computes a *c-approximation of the maximum matching* if **A** outputs a matching M such that $|M| \geq c \cdot |\text{opt}(G)|$. We consider two potential sources of randomness: from the algorithm and from the arrival order. Nevertheless, we will always consider worst case against the graph. For each situation, we relax the notion of c-approximation so that the expected approximation ratio is c, that is $\mathbb{E}\,|M| \geq c \cdot |\text{opt}(G)|$ where the expectation can be taken either over the internal random coins of the algorithm, or over all possible arrival orders.

For simplicity, we assume from now on that A, B and $m = |E|$ are given in advance the symmetric difference $(S_1 \setminus S_2) \cup (S_2 \setminus S_1)$ of the two sets.

For an input stream $\pi \in \Pi(G)$, we write $\pi[i]$ for the i-th edge of π, and $\pi[i, j]$ for the subsequence $\pi[i]\pi[i+1]\dots\pi[j]$. In this notation, a parenthesis excludes the smallest or respectively largest element: $\pi(i, j] = \pi[i+1, j]$, and $\pi[i, j) = \pi[i, j-1]$. If i, j are real, $\pi[i, j] := \pi[\lfloor i \rfloor, \lfloor j \rfloor]$, and

$\pi[i] := \pi[\lfloor i \rfloor]$. Given a subset $S \subseteq V$, $\pi|_S$ is the largest subsequence of π such that all edges in $\pi|_S$ are among vertices in S.

For a set of vertices S and a set of edges F, let $S(F)$ be the subset of vertices of S covered by F. Furthermore, we use the abbreviation $\overline{S(F)} := S \setminus S(F)$. For $S_A \subseteq A$ and $S_B \subseteq B$, we write $\text{opt}(S_A, S_B)$ for $\text{opt}(G|_{S_A \times S_B})$, that is a maximum matching in the subgraph of G induced by vertices $S_A \cup S_B$.

Maximal and Greedy Matchings. Formally, the *greedy matching algorithm* Greedy *on stream* π is defined as follows: Starting with an empty matching M, upon arrival of an edge $\pi[i]$, Greedy inserts Denote by $\text{Greedy}(\pi)$ the matching M after the stream π has been fully processed. By maximality, $|\text{Greedy}(\pi)| \geq \frac{1}{2}|\text{opt}(G)|$. Greedy can be seen as a semi-streaming algorithm for MBM with expected approximation ratio $\frac{1}{2}$ and $O(1)$ processing time per letter. We now state some preliminary properties. Lemma 1 shows that a maximal matching that is far from the optimal matching in value must also be far from the optimal matching in Hamming distance.

Lemma 1. *Let $M^* = \text{opt}(G)$, and let M be a maximal matching of G. Then $|M \cap M^*| \leq 2(|M| - \frac{1}{2}|M^*|)$.*

Proof. This is a piece of elementary combinatorics. Since M is a maximal matching, for every edge e of $M^* \setminus M$, at least one of the two endpoints of e is matched in $M \setminus M^*$, and so $|M \setminus M^*| \geq (1/2)|M^* \setminus M|$. We have $|M^* \setminus M| = |M^*| - |M^* \cap M|$. Combining gives

$$|M \cap M^*| = |M| - |M \setminus M^*| \leq |M| - \frac{1}{2}|M^* \setminus M| = |M| - \frac{1}{2}(|M^*| - |M^* \cap M|)$$

which implies the Lemma. □

Lemma 2 shows that maximal matchings that are small in size contain many edges that are 3-*augmentable*. Given a maximum matching $M^* = \text{opt}(G)$, and a maximal matching M, we say that an edge $e \in M$ is 3-augmentable if the removal of e from M allows the insertion of two edges f, g from $M^* \setminus M$ into M.

Lemma 2. *Let $\epsilon \geq 0$. Let $M^* = \text{opt}(G)$, let M be a maximal matching of G st. $|M| \leq (\frac{1}{2} + \epsilon)|M^*|$. Then M contains at least $(\frac{1}{2} - 3\epsilon)|M^*|$ 3-augmentable edges.*

Proof. The proof is folklore. Let k_i denote the number of paths of length i in $M \oplus M^*$. Since M^* is maximum, it has no augmenting path, so all odd length paths are augmenting paths of M. Since M is maximal, there are no augmenting paths of length 1, so $k_1 = 0$. Every even length path and every cycle has an equal number of edges from M and from M^*. A path of length $2i + 1$ has i edges from M and $i + 1$ edges from M^*.

$$|M^*| - |M| = \sum_{i \geq 1} k_{2i+1} \leq k_3 + \sum_{i \geq 2} \frac{1}{2} i k_{2i+1} = \frac{1}{2}k_3 + \frac{1}{2}\sum_{i \geq 1} i k_{2i+1} \leq \frac{1}{2}k_3 + \frac{1}{2}|M|.$$

Thus, using our assumption on $|M|$,

$$k_3 \geq 2|M^*| - 3|M| \geq 2|M^*| - (\frac{3}{2} + 3\epsilon)|M^*|,$$

implying the Lemma. □

3 One-pass algorithm on random order

We discuss now a one-pass semi-streaming algorithm for MBM with an expected approximation ratio strictly greater than $\frac{1}{2}$.

Algorithm. Here is a key observation in the random order setting: if Greedy performs badly on some input graph G, then most edges of Greedy appear within the first constant fraction of the stream, see Lemma 4. Our strategy is hence to run Greedy on a first part of the stream, and then, on the remaining part of the stream, we focus on searching for 3-augmenting paths.

Let M_0 denote the matching computed by Greedy on the first part of the stream. Assume that Greedy performs badly on the input graph G. Lemma 2 tells us that almost all of the edges of M_0 are 3-augmentable. To find 3-augmenting paths, in the next part of the stream we run Greedy to compute a matching M_1 between $B(M_0)$ and $\overline{A(M_0)}$. The edges in M_1 serve as one of the edges of 3-augmenting paths (from the B-side of M_0). In Lemma 5, we show that we find a constant fraction of those. In the last part of the stream, again by the help of Greedy, we compute a matching M_2 that completes the 3-augmenting paths. Lemma 8 shows that by this strategy we find many 3-augmenting paths. Then, either a simple Greedy matching performs well on G, or else we can find many 3-augmenting paths and use them to improve M_0: see the main theorem, Theorem 1 whose proof is deferred to the end of this section. An illustration is provided in Figure 1 in Appendix A.

Algorithm 1 Matching in one pass

$\alpha \leftarrow 0.4312, \beta \leftarrow 0.7595$
$M_G \leftarrow \text{Greedy}(\pi)$
$M_0 \leftarrow \text{Greedy}(\pi[1, \alpha m])$, matching obtained by running Greedy on the first $\lfloor \alpha m \rfloor$ edges
$F_1 \leftarrow$ complete bipartite graph between $B(M_0)$ and $\overline{A(M_0)}$
$M_1 \leftarrow \text{Greedy}(F_1 \cap \pi(\alpha m, \beta m])$, matching obtained by running Greedy on edges $\lfloor \alpha m \rfloor + 1$ through βm that intersect F_1
$A' \leftarrow \{a \in A \mid \exists b \in B(M_1) : ab \in M_0\}$
$F_2 \leftarrow$ complete bipartite graph between A' and $\overline{B(M_0)}$
$M_2 \leftarrow \text{Greedy}(F_2 \cap \pi(\beta m, m])$, matching obtained br running Greedy on edges $\lfloor \beta m \rfloor + 1$ through m that intersect F_2
$M \leftarrow$ matching obtained from M_0 augmented by $M_1 \cup M_2$
return larger of the two matchings M_G and M

Observe that our algorithm only uses memory space $O(n \log n)$. Indeed, each subsets F_1 and F_2 can be compactly represented by two n-bit arrays, and checking if an edge of π belongs to one of them can be done within time $O(1)$ from that compact representation.

Theorem 1. *Algorithm 1 is a deterministic one-pass semi-streaming algorithm for* MBM *with approximation ratio* $\frac{1}{2} + 0.005$ *against (uniform) random order for any graph, and can be implemented with* $O(1)$ *processing time per letter.*

Analysis. We use the notations of Algorithm 1. Consider α and β as variables with $0 \le \alpha \le \frac{1}{2} < \beta < 1$.

Lemma 3. $\forall e = ab \in E : \Pr[a \text{ and } b \notin V(M_0)] \le (\frac{1}{\alpha} - 1) \Pr[e \in M_0]$.

Proof. Observe: $\Pr[a \text{ and } b \notin V(M_0)] + \Pr[e \in M_0] = \Pr[a \text{ and } b \notin V(M_0 \setminus \{e\})]$, because the two events on the left hand side are disjoint and their union is the event on the right hand side.

Consider the following probabilistic argument. Take the execution for a particular ordering π. Assume that a and $b \notin V(M_0 \setminus \{e\})$ and let t be the arrival time of e. If we modify the ordering by changing the arrival time of e to some time $t' \leq t$, then we still have a and $b \notin V(M_0 \setminus \{e\})$. Thus[1]

$$\Pr[a \text{ and } b \notin V(M_0 \setminus \{e\})] \leq \Pr[a \text{ and } b \notin V(M_0 \setminus \{e\}) | e \in \pi[1, \alpha m]].$$

Now, the right-hand side equals $\Pr[e \in M_0 | e \in \pi[1, \alpha m]]$, which simplifies into $\Pr[e \in M_0] / \Pr[e \in \pi[1, \alpha m]]$ since e can only be in M_0 if it is one of the first αm arrivals. The we conclude the Lemma by the random order assumption $\Pr[e \in \pi[1, \alpha m]] = \alpha$.

□

Lemma 4. *If* $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2} + \epsilon)|M^*|$, *then* $\mathbb{E}_\pi |M_0| \geq |M^*|(\frac{1}{2} - (\frac{1}{\alpha} - 2)\epsilon)$.

Proof. Rather than directly analyzing the number of edges $|M_0|$, we analyze the number of vertices matched by M_0, which is equivalent since $|V(M_0)| = 2(|M_0|)$.

Fix an edge $e = ab$ of M^*. Either $e \in M_0$, or at least one of a, b is matched by M_0, or neither a nor b are matched. Summing over all $e \in M^*$ gives

$$|V(M_0)| \geq 2|M^* \cap M_0| + |M^* \setminus M_0| - \sum_{e=ab \in M^*} \chi[a \text{ and } b \notin V(M_0)],$$

where $\chi[X] = 1$ if the event X happens, otherwise $\chi[X] = 0$. Taking expectations and using Lemma 3,

$$\begin{aligned} \mathbb{E}_\pi(|V(M_0)|) &\geq 2\,\mathbb{E}_\pi |M^* \cap M_0| + \mathbb{E}_\pi |M^* \setminus M_0| - (\frac{1}{\alpha} - 1)\,\mathbb{E}_\pi |M^* \cap M_0| \\ &= |M^*| - (\frac{1}{\alpha} - 2)\,\mathbb{E}_\pi |M^* \cap M_0|. \end{aligned}$$

Since M_0 is just a subset of the edges of M_G, using Lemma 1 and linearity of expectation,

$$\mathbb{E}_\pi |M^* \cap M_0| \leq \mathbb{E}_\pi |M^* \cap M_G| \leq 2(\mathbb{E}_\pi |M_G| - \frac{1}{2}|M^*|) \leq 2\epsilon|M^*|.$$

Combining gives the Lemma. □

Lemma 5. *Assume that* $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2} + \epsilon)|M^*|$. *Then the expected size of the maximum matching between the vertices of A left unmatched by* M_0 *and the vertices of B matched by* M_0 *can be bounded below as follows:*

$$\mathbb{E}_\pi |\text{opt}(\overline{A(M_0)}, B(M_0))| \geq |M^*|(\frac{1}{2} - (\frac{1}{\alpha} + 2)\epsilon).$$

Proof. The size of a maximum matching between $\overline{A(M_0)}$ and $B(M_0)$ is at least the number of augmenting paths of length 3 in $M_0 \oplus M^*$. By Lemma 2, in expectation, the number of augmenting paths of length 3 in $M_G \oplus M^*$ is at least $(\frac{1}{2} - 3\epsilon)|M^*|$. All of those are augmenting paths of length 3 in $M_0 \oplus M^*$, except for at most $|M_G| - |M_0|$. Hence, in expectation, M_0 contains $(\frac{1}{2} - 3\epsilon)|M^*| - (\mathbb{E}_\pi |M_G| - \mathbb{E}_\pi |M_0|)$ 3-augmentable edges. Lemma 4 concludes the proof. □

[1] Formally, we define a map f from the uniform distribution on all orderings to the uniform distribution on all orderings such that $e \in \pi[1, \alpha m]$: if $e \in \pi[1, \alpha m]$ then $f(\pi) = \pi$ and otherwise $f(\pi)$ is the permutation obtained from π by removing e and re-inserting it at a position picked uniformly at random in $[1, \alpha m]$.

Lemma 6. $\mathbb{E}_\pi |M_1| \geq \frac{1}{2}(\beta-\alpha)(\mathbb{E}_\pi |\text{opt}(\overline{A(M_0)}, B(M_0))| - \frac{1}{1-\alpha})$.

Proof. Since Greedy computes a maximal matching which is at least half the size of a maximum matching, $\mathbb{E}_\pi |M_1| \geq \frac{1}{2}\mathbb{E}_\pi |\text{opt}(\overline{A(M_0)}, B(M_0)) \cap \pi(\alpha m, \beta m]|$.

By independence between M_0 and the ordering within $(\alpha m, m]$, we see that even if we condition on M_0, we still have that $\pi(\alpha m, \beta m]$ is a random uniform subset of $\pi(\alpha m, m]$. Thus:

$$\mathop{\mathbb{E}}_{\pi} |\text{opt}(\overline{A(M_0)}, B(M_0)) \cap \pi(\alpha m, \beta m]| = \tfrac{\beta-\alpha}{1-\alpha} \mathop{\mathbb{E}}_{\pi} |\text{opt}(\overline{A(M_0)}, B(M_0)) \cap \pi(\alpha m, m]|.$$

We use a probabilistic argument similar to but slightly more complicated than the proof of Lemma 3. We define a map f from the uniform distribution on all orderings to the uniform distribution on all orderings such that $e \in \pi(\alpha m, m]$: if $e \in \pi(\alpha m, m]$ then $f(\pi) = \pi$ and otherwise $f(\pi)$ is the permutation obtained from π by removing e and re-inserting it at a position picked uniformly at random in $(\alpha m, m]$; in the latter case, if this causes an edge $f = a'b'$, previously arriving at time $\lfloor \alpha m \rfloor + 1$, to now arrive at time $\lfloor \alpha m \rfloor$ and to be added to M_0, we define $M_0' = M_0 \setminus \{f\}$; in all other cases we define $M_0' = M_0$. Thus, if in π we have $e \in \text{opt}(\overline{A(M_0)}, B(M_0))$, then in $f(\pi)$ we have $e \in \text{opt}(\overline{A(M_0')}, B(M_0'))$. Since the distribution of $f(\pi)$ is uniform conditioned on $e \in \pi(\alpha m, m]$:

$$\frac{\Pr[e \in \text{opt}(\overline{A(M_0')}, B(M_0')) \text{ and } e \in \pi(\alpha m, m]]}{\Pr[e \in \pi(\alpha m, m]]} \geq \Pr[e \in \text{opt}(\overline{A(M_0)}, B(M_0))],$$

Using $\Pr[e \in \pi(\alpha m, m]] = 1 - \alpha$ and summing over e:

$$\mathbb{E}_\pi |\text{opt}(\overline{A(M_0')}, B(M_0')) \cap \pi(\alpha m, m]| \geq (1-\alpha)\,\mathbb{E}_\pi |\text{opt}(\overline{A(M_0)}, B(M_0))|.$$

Since M_0' and M_0 differ by at most one edge, $|\text{opt}(\overline{A(M_0)}, B(M_0))| \geq |\text{opt}(\overline{A(M_0')}, B(M_0'))| - 1$, and the Lemma follows. ☐

Lemma 7. *Assume that* $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2}+\epsilon)|M^*|$. *Then:*

$$\mathop{\mathbb{E}}_{\pi} |\text{opt}(A', \overline{B(M_0)}| \geq \mathop{\mathbb{E}}_{\pi} |M_1| - 4\epsilon|M^*|.$$

Proof. $|\text{opt}(A', \overline{B(M_0)}|$ is at least $|M_1|$ minus the number of edges of M_0 that are not 3-augmentable. Since M_0 is a subset of M_G, the latter term is bounded by the number of edges of M_G that are not 3-augmentable, which by Lemma 2 is in expectation at most $(\frac{1}{2}+\epsilon)|M^*| - (\frac{1}{2}-3\epsilon)|M^*| = 4\epsilon|M^*|$. ☐

Lemma 8. $\mathop{\mathbb{E}}_{\pi} |M_2| \geq \dfrac{1}{2}((1-\beta)\mathop{\mathbb{E}}_{\pi} |\text{opt}(A', \overline{B(M_0)})| - 1)$.

Proof. Since Greedy computes a maximal matching which is at least half the size of a maximum matching,

$$\mathop{\mathbb{E}}_{\pi} |M_2| \geq \frac{1}{2} \mathop{\mathbb{E}}_{\pi} |\text{opt}(A', \overline{B(M_0)}) \cap \pi(\beta m, m]|.$$

Formally, we define a map f from the uniform distribution on all orderings to the uniform distribution on all orderings such that $e \in \pi(\beta m, m]$: if $e \in \pi(\beta m, m]$ then $f(\pi) = \pi$ and otherwise $f(\pi)$ is the permutation obtained from π by removing e and re-inserting it at a position picked uniformly at random in $(\beta m, m]$; in the latter case, if this causes an edge $e' = a'b'$, previously arriving at time $\lfloor \beta m \rfloor + 1$, to now arrive at time $\lfloor \beta m \rfloor$ and to be added to M_1, we define $A'' = A' \setminus \{M_0(b')\}$; in all other cases we define $A'' = A'$. Thus, if in π we have $e \in \text{opt}(A', \overline{B(M_0)})$, then in $f(\pi)$ we have $e \in \text{opt}(A'', \overline{B(M_0)})$. Since the distribution of $f(\pi)$ is uniform conditioned on $e \in \pi(\beta m, m]$:

$$\frac{\Pr[e \in \text{opt}(A'', \overline{B(M_0)}) \text{ and } e \in \pi(\beta m, m]]}{\Pr[e \in \pi(\beta m, m]]} \geq \Pr[e \in \text{opt}(A', \overline{B(M_0)})],$$

Using $\Pr[e \in \pi(\beta m, m]] = 1 - \beta$ and summing over e:

$$\mathop{\mathbb{E}}_{\pi} |\text{opt}(A'', \overline{B(M_0)}) \cap \pi(\beta m, m]| \geq (1-\beta) \mathop{\mathbb{E}}_{\pi} |\text{opt}(A', \overline{B(M_0)})|.$$

Since A' and A'' differ by at most one vertex,
$|\text{opt}(A'', \overline{B(M_0)})| \geq |\text{opt}(A', \overline{B(M_0)})| - 1$, and the Lemma follows. □

We now present the proof of the main theorem, Theorem 1.

of Theorem 1. Assume that $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2} + \epsilon)|M^*|$. By construction, every $e \in M_2$ completes a 3−augmenting path, hence $|M| \geq |M_0| + |M_2|$. In Lemma 4 we show that $\mathbb{E}_\pi |M_0| \geq |M^*|(\frac{1}{2} - (\frac{1}{\alpha} - 2)\epsilon)$. By Lemmas 8 and 7, $|M_2|$ can be related to $|M_1|$:

$$\mathop{\mathbb{E}}_{\pi} |M_2| \geq \frac{1}{2}(1-\beta) \mathop{\mathbb{E}}_{\pi} |\text{opt}(A', \overline{B(M_0)})| - \frac{1}{2} \geq \frac{1}{2}(1-\beta)(\mathop{\mathbb{E}}_{\pi} |M_1| - 4\epsilon|M^*|) - \frac{1}{2}.$$

By Lemmas 6 and 5, $|M_1|$ can be related to $|M^*|$:

$$\begin{aligned} \mathop{\mathbb{E}}_{\pi} |M_1| &\geq \tfrac{1}{2}(\beta - \alpha) \mathop{\mathbb{E}}_{\pi} |\text{opt}(\overline{A(M_0)}, B(M_0)| - O(1) \\ &\geq \tfrac{1}{2}(\beta - \alpha)(|M^*|(\frac{1}{2} - (\frac{1}{\alpha} + 2)\epsilon)) - O(1). \end{aligned}$$

Combining,

$$\mathop{\mathbb{E}}_{\pi} |M| \geq |M^*|(\tfrac{1}{2} - (\tfrac{1}{\alpha} - 2)\epsilon + \tfrac{1}{2}(1-\beta)(\tfrac{1}{2}(\beta-\alpha)(\tfrac{1}{2} - (\tfrac{1}{\alpha} + 2)\epsilon) - 4\epsilon)) - O(1).$$

The expected value of the output of the Algorithm is at least $\min_\epsilon \max\{(\frac{1}{2} + \epsilon)|M^*|, \mathbb{E}_\pi |M|\}$. We set the right hand side of the above Equation equal to $(\frac{1}{2} + \epsilon)|M^*|$. By a numerical search we optimize parameters α, β. Setting $\alpha = 0.4312$ and $\beta = 0.7595$, we obtain $\epsilon \approx 0.005$ which proves the Theorem. □

4 Randomized two-pass algorithm on any order

We present now a randomized two-pass semi-streaming algorithm for MBM with approximation ratio strictly greater than $\frac{1}{2}$. The algorithm relies on a property of the Greedy algorithm that we discuss before the presentation of the algorithm. This property may be of independent interest.

Matching many vertices of a random subset of A. For a fixed parameter $0 < p \leq 1$, the following algorithm generates an independent random sample of vertices $A' \subseteq A$ such that $\Pr[a \in A'] = p$, for all $a \in A$. Theorem 2 shows that the greedy algorithm restricted to the edges with an endpoint in A' will output a matching of expected approximation ratio $p/(1+p)$, compared to a maximum matching $\text{opt}(G)$ over the full graph G. Since, in expectation, the size of A' is $p|A|$, one can roughly say that a fraction of $1/(1+p)$ of vertices in $|A'|$ has been matched.

Algorithm 2 Matching a random subset

```
1: Take independent random sample A' ⊆ A st. Pr[a ∈ A'] = p, for all a ∈ A
2: Let F be the complete bipartite graph between A' and B
3: return M' = Greedy(F ∩ π)
```

The proof of Theorem 2 will use Wald's equation for super-martingales, see [14], Wald's Equation, p.300, section 12.3.[2]

Lemma 9 (Wald's equation)**.** *Consider a process described by a sequence of random states* $(S_i)_{i\geq 0}$ *and let* D *be a random stopping time for the process, such that* $\mathbb{E}\, D < \infty$. *Let* $(\Phi(S_i))_{i\geq 0}$ *be a sequence of random variables for which there exist* c, μ *such that*

1. $\Phi(S_0) = 0$;
2. $\Phi(S_{i+1}) - \Phi(S_i) < c$ *for all* $i < D$; *and*
3. $\mathbb{E}[\Phi(S_{i+1}) - \Phi(S_i) \,|\, S_i] \leq \mu$ *for all* $i < D$.

Then:

$$\mathbb{E}\, \Phi(S_D) \leq \mu\, \mathbb{E}\, D.$$

Theorem 2. *Let* $0 < p \leq 1$, *let* $G = (A, B, E)$ *be a bipartite graph. Let* A' *be an independent random sample* $A' \subset A$ *such that* $\Pr[a \in A'] = p$, *for all* $a \in A$. *Let* F *be the complete bipartite graph between* A' *and* B *Then for any input stream* $\pi \in \Pi(G)$: $\underset{A'}{\mathbb{E}} |\text{Greedy}(F \cap \pi)| \geq \dfrac{p}{1+p} |\text{opt}(G)|$.

Proof. Let $M' = \text{Greedy}(F \cap \pi)$. For $i \leq |M'|$, denote by M'_i the first i edges of M', in the order in which they were added to M' during the execution of Greedy.

Let M^* be a fixed maximum matching in G and let M_F denote the edges of M^* that are in F. Let $A'' = A(M_F)$ denote the vertices of A' matched by M_F. Consider a vertex $a \in A''$ and its match b in matching M_F. We say that a is *live* with respect to M'_i if both a and b are unmatched in M'_i. A vertex that is not live is *dead*. Furthermore, we say that an edge of $M'_{i+1} \setminus M'_i$ *kills* a vertex a if a is live with respect to M'_i and dead with respect to M'_{i+1}.

We use Lemma 9. Here, by "time", we mean the number of edges in M', so between time $i-1$ and time i, during the execution of Greedy, several edges arrive and all are rejected except the last one which is added to M'. We use a potential function $\phi(i)$ which we define as the number of dead vertices wrt. M'_i. We define the stopping time D as the first time when the event $\phi(i) = |A''|$ holds.

We only need to check that the three assumptions of the Stopping Lemma hold. First, initially all nodes of A'' are live, so $\phi(0) = 0$. Second, the potential function ϕ is non-decreasing and uniformly bounded: since adding an edge to M' can kill at most two vertices of A'', we always have $\Delta\phi(i) := \phi(i+1) - \phi(i) \leq 2$. Third, let S_i denote the state of the process at time i, namely the information about the entire sequence of edge arrivals up to that time, hence, in particular, the set of i edges currently in M'. Observe that, here, G and M^* are fixed. Then D is indeed a stopping time, since the event $D \geq i+1$ can be inferred from the knowledge of S_i.

We now claim that:

$$\mathbb{E}(\Delta\phi(i) \mid S_i) \leq 1 + p. \tag{1}$$

Indeed, since $\Delta\phi(i)$ only takes on values 0, 1 or 2, we can write that $\mathbb{E}(\Delta\phi(i)|S_i) \leq 1 + \Pr[\Delta\phi(i) = 2|S_i]$. To bound the latter probability, let $e = ab$ denote the edge of $M'_{i+1} \setminus M'_i$, let t be such that $e = \pi[t]$, and let a' be the mate of b in matching M^*. In order for e to change ϕ by 2, it must be that a' is in A' and that a' was unmatched before edge e arrived. Since a' was unmatched up to

[2] The theorem cited in the book is actually weaker than the one we need, but our statement follows from the proof of that Theorem. Another source is available online at `http://greedyalgs.info/blog/stopping-times/`

arrival t, no edge $a'b'$ had been seen among the first t edges of stream π, such that b' was free at arrival time (of $a'b'$). Thus

$$\Pr[\Delta\phi(i) = 2|S_i] \leq \\ \Pr[a' \in A' \text{ and } \nexists a'b' \in \pi[1,t] \text{ st. } b' \text{ was free when } a'b' \text{ arrived}|S_i].$$

Now, given that no edge $f = a'b'$ arrived before t such that b' was free when $a'b'$ arrived, the outcome of the random coin determining whether $a' \in A'$ was never looked at, and could have been postponed until t. Thus

$$\Pr[a' \in A' \,|\, (\nexists a'b' \in \pi[1,t] \text{ such that } b' \text{ was free when } a'b' \text{ arrived}, S_i)] = \\ \Pr[a' \in A'] = p,$$

implying Inequality 1. Applying Walds' Stopping Lemma, we obtain $\mathbb{E}\,\phi(D) \leq (1+p)\,\mathbb{E}\,D$. Finally, observe that $\mathbb{E}\,\phi(D) = \mathbb{E}\,|A''| = p \cdot |\text{opt}(G)|$ and that $D \leq |\text{Greedy}(F \cap \pi)|$, and the Theorem follows.

□

Application: a randomized two-pass algorithm. Based on Theorem 2, we design our randomized two-pass algorithm. Assume that Greedy(π) returns a matching that is close to a $\frac{1}{2}$ approximation. In order to apply Theorem 2, we pick an independent random sample $A' \subseteq A$ such that $\Pr[a \in A'] = p$ for all a. In a first pass, our algorithm computes a Greedy matching M_0 of G, and a Greedy matching M' between vertices of A' and B. M' then contains some edges that form parts of 3-augmenting paths for M_0: see Figure 2 and Figure 3 for an illustration in Appendix B. Let $M_1 \subset M'$ be the set of those edges. It remains to complete these length 2 paths $M_0 \cup M_1$ in a second pass by a further Greedy matching M_2. Theorem 3 states then that if Greedy(π) is close to a $\frac{1}{2}$ approximation, then we find many 3-augmenting paths.

Algorithm 3 Two-pass bipartite matching

Let $p \leftarrow \sqrt{2} - 1$.
Take an independent random sample $A' \subseteq A$ st. $\Pr[a \in A'] = p$, for all $a \in A$
Let F_1 be the set of edges with one endpoint in A'.
First pass: $M_0 \leftarrow \text{Greedy}(\pi)$ and $M' \leftarrow \text{Greedy}(F_1 \cap \pi)$
$M_1 \leftarrow \{e \in M' \,|\, e \text{ goes between } B(M_0) \text{ and } \overline{A(M_0)}\}$
$A_2 \leftarrow \{a \in A(M_0) : \exists b, c : ab \in M_0 \text{ and } bc \in M_1\}$.
Let $F_2 \leftarrow \{da : d \in \overline{B(M_0)} \text{ and } a \in A(M_0) \text{ and } \exists b, c : ab \in M_0 \text{ and } bc \in M_1\}$.
Second pass: $M_2 \leftarrow \text{Greedy}(F_2 \cap \pi)$
Augment M_0 by edges in M_1 and M_2 and store it in M
return the resulting matching M

Theorem 3. *Algorithm 3 is a randomized two-pass semi-streaming algorithm for* MBM *with expected approximation ratio* $\frac{1}{2} + 0.019$ *in expectation over its internal random coin flips for any graph and any arrival order, and can be implemented with* $O(1)$ *processing time per letter.*

Proof. By construction, each edge in M_2 is part of a 3-augmenting path, hence the output has size: $|M| = |M_0| + |M_2|$.

Define ϵ to be such that $|M_0| = (\frac{1}{2} + \epsilon)|\text{opt}(G)|$. Since M_2 is a maximal matching of F_2, we have $|M_2| \geq \frac{1}{2}|\text{opt}(F_2)|$. Let M^* be a maximum matching of G. Then $|\text{opt}(F_2)|$ is greater than or equal to the number of edges ab of M_0 such that there exists an edge bc of M_1 and an edge da of M^* that altogether form a 3-augmenting path of M_0:

$$\begin{aligned}|\text{opt}(F_2)| &\geq |\{ab \in M_0 \,|\, \exists c : bc \in M_1 \text{ and } \exists d : da \in M^*\}| \\ &\geq |\{ab \in M_0 \,|\, \exists c : bc \in M_1\}| - |\{ab \in M_0 \,|\, ab \text{ not 3-augmentable}\}|.\end{aligned}$$

Lemma 2 gives $|\{ab \in M_0 \,|\, ab \text{ is not 3-augmentable with } M^*\}| \leq 4\epsilon|\text{opt}(G)|$. It remains to bound $|\{ab \in M_0 \,|\, \exists c : bc \in M_1\}|$ from below. By definition of M' and of $M_1 \subseteq M'$, and by maximality of M_0,

$$\begin{aligned}|\{ab \in M_0 \,|\, \exists c : bc \in M_1\}| &= |M'| - |\{ab \in M' \,|\, a \in A(M_0)\}| \\ &\geq |M'| - |A(M_0) \cap A'|.\end{aligned}$$

Taking expectations, by Theorem 2 and by independence of M_0 from A':

$$\mathop{\mathbb{E}}_{A'} |M'| - \mathop{\mathbb{E}}_{A'} |A(M_0) \cap A'| \geq \frac{p}{1+p}|\text{opt}(G)| - p(\frac{1}{2} + \epsilon)|\text{opt}(G)|.$$

Combining:

$$\mathop{\mathbb{E}}_{A'} |M| \geq (\frac{1}{2} + \epsilon)|\text{opt}(G)| + \frac{1}{2}\left(|\text{opt}(G)|p(\frac{1}{1+p} - \frac{1}{2} - \epsilon) - 4\epsilon|\text{opt}(G)|.\right)$$

For ϵ small, the right hand side is maximized for $p = \sqrt{2} - 1$. Then $\epsilon \approx 0.019$ minimizes $\max\{|M|, |M_0|\}$ which proves the theorem.

□

5 Deterministic two-pass algorithm on any order

The deterministic two-pass algorithm, Algorithm 5, follows the same line as its randomized version, Algorithm 3. In a first pass we compute a Greedy matching M_0 and some additional edges S, computed by Algorithm 4. If M_0 is not much more than a $\frac{1}{2}$-approximation then S contains edges that serve as parts of 3-augmenting paths. These are completed via a Greedy matching in the second pass.

The way we compute the edge set S is now different. Before, S was a matching M' between B and a random subset A' of A. Now, S is not a matching but a relaxation of matchings as follows. Given an integer $\lambda \geq 2$, an *incomplete λ-bounded semi-matching* S of a bipartite graph $G = (A, B, E)$ is a subset $S \subseteq E$ such that $\deg_S(a) \leq 1$ and $\deg_S(b) \leq \lambda$, for all $a \in A$ and $b \in B$. This notion is closely related to semi-matchings. A semi-matching matches all A vertices to B vertices without limitations on the degree of a B vertex. However, since we require that the B vertices have constant degree, we loosen the condition that all A vertices need to be matched.

In Lemma 10, we show that Algorithm 4, a straightforward greedy algorithm, computes an incomplete λ-bounded semi-matching that covers at least $\frac{\lambda}{\lambda+1}|M^*|$ vertices of A.

Lemma 10. *Let $S = \text{SEMI}(\lambda)$ be the output of Algorithm 4 for some $\lambda \geq 2$. Then S is an incomplete λ-bounded semi-matching such that $|A(S)| \geq \frac{\lambda}{\lambda+1}|M^*|$.*

Algorithm 4 incomplete degree λ limited semi-matching SEMI(λ)

$S \leftarrow \varnothing$
while $\exists$ edge ab in stream
  if $\deg_S(a) = 0$ and $\deg_S(b) \leq \lambda - 1$ **then** $S \leftarrow S \cup \{ab\}$
return S

Proof. By construction, S is an incomplete degree λ bounded semi-matching. We bound $A(M^*) \setminus A(S)$ from below. Let $a \in A(M^*) \setminus A(S)$ and let b be its mate in M^*. The algorithm did not add the optimal edge ab upon its arrival. This implies that b was already matched to λ other vertices. Hence, $|A(M^*) \setminus A(S)| \leq \frac{1}{\lambda}|A(S)|$. Then the result follows by combining this inequality with $|M^*| - |A(S)| \leq |A(M^*) \setminus A(S)|$. □

Now, assume that the greedy matching algorithm computes a M_0 close to a $\frac{1}{2}$-approximation. Then, for $\lambda \geq 2$ there are many A vertices that are not matched in M_0 but are matched in S. Edges incident to those in S are candidates for the construction of 3-augmenting paths. This argument can be made rigorous, leading to Algorithm 5 where λ is set to 3, in Theorem 4.

Algorithm 5 two-pass deterministic algorithm

First pass: $M_0 \leftarrow$ Greedy(π) and $S \leftarrow$ SEMI(3)
$M_1 \leftarrow \{e \in S \mid e \text{ is between } B(M_0) \text{ and } \overline{A(M_0)}\}$
$A_2 \leftarrow \{a \in A(M_0) \mid \exists bc : ab \in M_0 \text{ and } bc \in M_1\}$
$F \leftarrow \{e \mid e \text{ goes between } A_2 \text{ and } \overline{B(M_0)}\}$
Second pass: $M_2 \leftarrow$ Greedy($\pi \cap F$)
Augment M_0 by edges in M_1 and M_2 and store it in M
return M

Theorem 4. *Algorithm 5 is a deterministic two-pass semi-streaming algorithm for* MBM *with approximation ratio* $\frac{1}{2} + 0.019$ *for any graph and any arrival order and can be implemented with* $O(1)$ *processing time per edge.*

Proof. The computed matching M is of size $|M_0| + |M_2|$ since, by construction, for each edge in M_2 there is at least one distinct edge in M_1 that allows the construction of a 3-augmenting path. Each 3-augmenting path increases the matching M_0 by 1. See also Figure 5 in Appendix C. Since $|M_2|$ is a maximal matching of the graph induced by the edges F, we obtain

$$|M| \geq |M_0| + \frac{1}{2}\text{opt}(F).$$

Let ϵ be such that $|M_0| = (\frac{1}{2} + \epsilon)|M^*|$. By Lemma 2, at most $4\epsilon|M^*|$ edges of M_0 are not 3-augmentable, hence

$$\text{opt}(F) \geq |A_2| - 4\epsilon|M^*|.$$

A_2 are those vertices matched also by M_0 such that there exists an edge in M_1 matching the mate of the A_2 vertex. Since the maximal degree in M_1 is λ, we can bound $|A_2|$ by

$$|A_2| \geq \frac{1}{\lambda}|M_1|.$$

Note that $|M_1| = |A(S) \setminus A(M_0)|$ since the degree of an A vertex matched by S in S is one, and S can be partitioned into $S_{M_0}, S_{\overline{M_0}}$ such that edges in S_{M_0} couple an A vertex also matched in M_0, and edges in $S_{\overline{M_0}}$ couple an A vertex that is not matched in M_0. Now, $|M_1| = |S_{\overline{M_0}}|$ since an edge of S is taken into M_1 if it is in $S_{\overline{M_0}}$.

Lemma 10 allows us to bound the size of the set $A(S) \setminus A(M_0)$ via

$$|A(S) \setminus A(M_0)| \geq |A(S)| - |A(M_0)| \geq (\frac{\lambda}{\lambda+1} - \frac{1}{2} - \epsilon)|M^*|.$$

Using the prior Inequalities, we obtain

$$|M| \geq (\frac{1}{2} - \epsilon + \frac{1}{2\lambda+2} - \frac{1}{4\lambda} - \frac{\epsilon}{2\lambda})|M^*|.$$

Since we have also $|M| \geq |M_0| = (\frac{1}{2} + \epsilon)|M^*|$, we set

$$\begin{aligned} \epsilon_0 &= \min_{\epsilon} \max\{(\frac{1}{2} - \epsilon + \frac{1}{2\lambda+2} - \frac{1}{4\lambda} - \frac{\epsilon}{2\lambda})|M^*|, (\frac{1}{2} + \epsilon)|M^*|\} \\ &= \frac{\lambda - 1}{8\lambda^2 + 10\lambda + 2}, \end{aligned}$$

which is maximized for $\lambda = 3$ leading to an approximation factor of $\frac{1}{2} + \frac{1}{52} \approx \frac{1}{2} + 0.019$.

Concerning the processing time per edge, note that once an edge is added in the second pass, a corresponding 3-augmenting path can be determined in time $O(1)$.

□

References

[1] K. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. In *ICALP*, pages 526–538, 2011.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. of Computer and System Sciences*, 58(1):137–147, 1999.

[3] M. Dyer and Al. Frieze. Randomized greedy matching. *Random Structures & Algorithms*, 2(1):29–46, 1991.

[4] S. Eggert, L. Kliemann, P. Munstermann, and A. Srivastav. Bipartite matching in the semi-streaming model. *Algorithmica*, 63(1-2):490–508, 2012.

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

[6] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Theoretical Computer Science*, 348(2-3):207–216, 2005.

[7] Joan Feigenbaum, Sampath Kannan, Andrew Mcgregor, Siddharth Suri, and Jian Zhang. Graph distances in the streaming model: the value of space. In *SODA*, pages 745–754, 2005.

[8] A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. In *SODA*, 2012. To appear.

[9] S. Guha and A. McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM J. of Computing*, 38(1):2044–2059, 2009.

[10] C. Karande, A. Mehta, and P. Tripathi. Online bipartite matching with unknown distributions. In *STOC*, pages 587–596, 2011.

[11] R. Karp, U. Vazirani, and V. Vazirani. An optimal online bipartite matching algorithm. In *STOC*, pages 352–358, 1990.

[12] M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs. In *STOC*, pages 597–605, 2011.

[13] A. McGregor. Finding graph matchings in data streams. In *APPROX*, pages 170–181, 2005.

[14] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[15] J. Munro and M. Paterson. Selection and sorting with limited storage. *Theoretical Computer Science*, 12:211–219, 1980.

[16] S. Muthukrishnan. Data streams: Algorithms and applications. In *Foundations and Trends in Theoretical Computer Science*. Now Publishers Inc, 2005.

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

A Figure for the one-pass algorithm on random order

Figure 1 illustrates Algorithm 1.

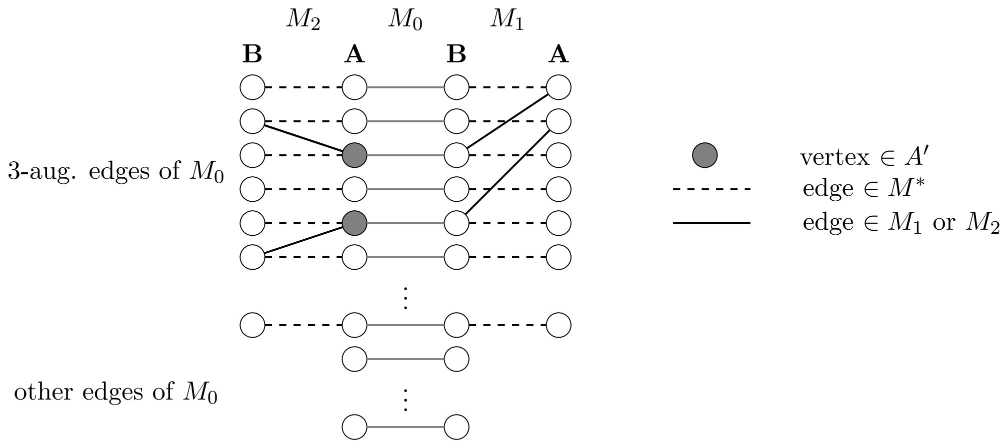

Figure 1: Illustration of Algorithm 1. Note that every edge of M_2 completes a 3-augmenting path consisting of one edge of M_1 (on the right hand side of the picture) followed by one edge of M_0 (center) followed by one edge of M_2 (on the left hand side of the picture).

B Figures for the randomized two-pass algorithm

We provide two figures illustrating the first pass (Figure 2) and the second pass (Figure 3) of Algorithm 3.

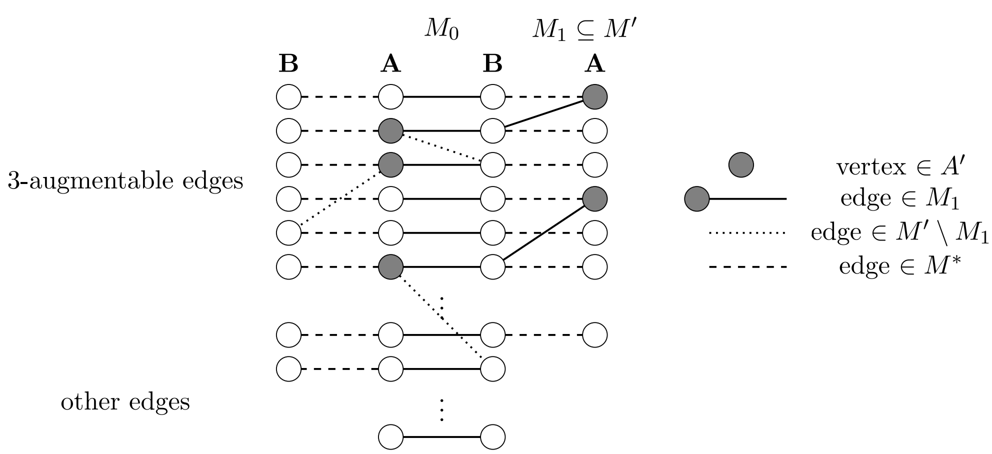

Figure 2: Illustration of the first pass of Algorithm 3. By Theorem 2, nearly all vertices of A' are matched in M', in particular those that are not matched in M_0.

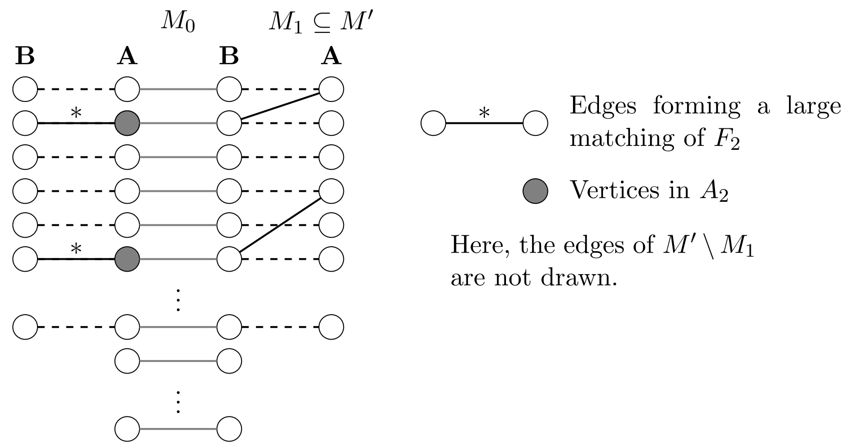

Figure 3: Analysis of the second pass of Algorithm 3. Here, we see that $M_0 \oplus M_1$ has two paths of length 2, and that both of those paths can be extended into 3-augmenting paths using M^*: this illustrates $|\text{opt}(F_2)| \geq 2$. Matching M_2, being a 1/2 approximation, will find at least one 3-augmenting path.

C Figures for the deterministic two-pass algorihtm

We show two figures illustrating the first pass (Figure 4) and the second pass (Figure 5) of Algorithm 5.

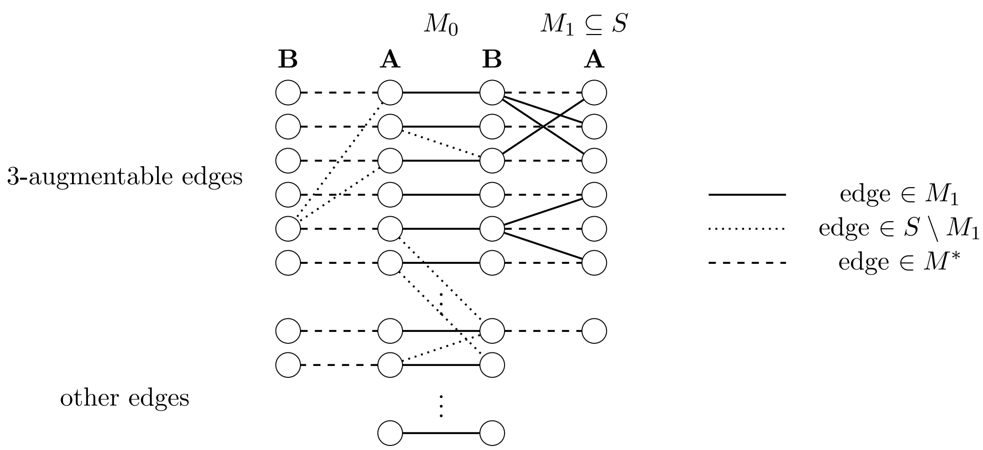

Figure 4: Illustration of the first pass of Algorithm 5. In this example we set $\lambda = 2$ and we compute an incomplete degree 2 limited semi-matching S. By Lemma 10, we match at least $\frac{2}{3}|M^*|$ A vertices. Since $|M| \approx \frac{1}{2}|M^*|$, some A vertices that are not matched in M_0 are matched in S. The edges incident to those define M_1.

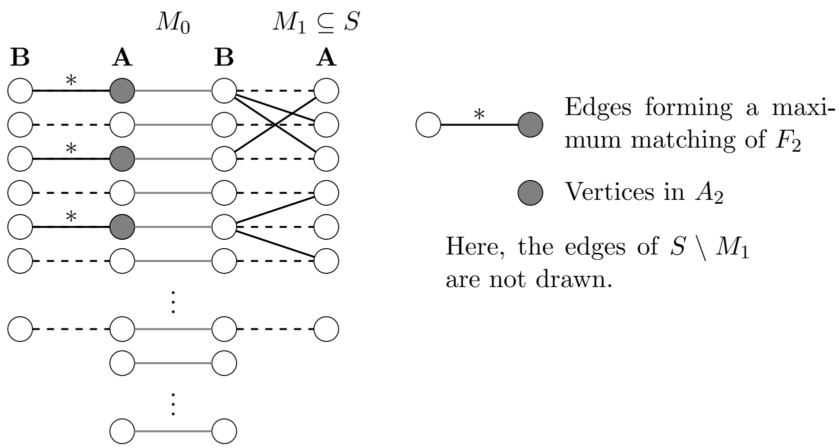

Figure 5: Analysis of the second pass of Algorithm 5. Here, we see that $M_0 \oplus M_1$ has five paths of length 2. These paths are not disjoint, but since the maximal degree in S is 2, $M_0 \oplus M_1$ has at least $\frac{1}{2} \cdot 5$ disjoint paths, and hence $|A_2| = 3 \geq \frac{1}{2} \cdot 5$. A maximum matching in F_2 is of size 3, and in the second pass, Greedy will find at least half of them leading to at least two 3-augmenting paths.