Maximum Matching in Semi-streaming with Few Passes*

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

[1] Reykjavik University, Reykjavik, Iceland
christiank@ru.is
[2] CNRS, LIAFA, Université Paris Diderot, Sorbonne Paris-Cité, 75205 Paris, France
magniez@cnrs.fr
[3] CNRS, Département d'Informatique UMR CNRS 8548, École Normale Supérieure, Paris, France
claire@cs.brown.edu

Abstract. In the *semi-streaming model*, an algorithm receives a stream of edges of a graph in arbitrary order and uses a memory of size $O(n \operatorname{polylog} n)$, where n is the number of vertices of a graph. In this work, we present semi-streaming algorithms that perform one or two passes over the input stream for Maximum Matching with no restrictions on the input graph, and for the important special case of bipartite graphs that we refer to as Maximum Bipartite Matching. The Greedy matching algorithm performs one pass over the input and outputs a $1/2$ approximation. Whether there is a better one-pass algorithm has been an open question since the appearance of the first paper on streaming algorithms for matching problems in 2005 [Feigenbaum et al., SODA 2005]. We make the following progress on this problem:

In the one-pass setting, we show that there is a deterministic semi-streaming algorithm for Maximum Bipartite Matching with expected approximation factor $1/2 + 0.005$, assuming that edges arrive one by one in (uniform) random order. We extend this algorithm to general graphs, and we obtain a $1/2 + 0.003$ approximation for Maximum Matching.

In the two-pass setting, we do not require the random arrival order assumption (the edge stream is in arbitrary order). We present a simple randomized two-pass semi-streaming algorithm for Maximum Bipartite Matching with expected approximation factor $1/2 + 0.019$. Furthermore, we discuss a more involved deterministic two-pass semi-streaming algorithm for Maximum Bipartite Matching with approximation factor $1/2 + 0.019$ and a generalization of this algorithm to general graphs with approximation factor $1/2 + 0.0071$.

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 for instance 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. Another typical situation is a continuous flow of traffic logs sent to a router. Streaming algorithms sequentially scan the input piece by piece while using sublinear memory space. The analysis of Internet traffic [1] 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 like optical disks or hard drives where random access is too costly and hence only sequential access is possible. 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. However, polylogarithmic memory is too restrictive for many graph problems: as shown in [2], deciding basic graph properties such as bipartiteness or connectivity of an n-vertex graph already requires $\Omega(n)$ space. Muthukrishnan [3] suggests to study massive graphs in a *semi-external* model, that is, not the entire graph but the vertex set can be stored in memory. In that model, a graph is given by a stream of edges arriving in

* Most of the work was done while Christian Konrad was a PhD student at LIAFA, Université Paris Diderot. Frédéric Magniez is supported by the French ANR Blanc project ANR-12-BS02-005 (RDAM). Claire Mathieu is supported by NSF Grant CCF-0964037.

arbitrary order. A *semi-streaming* algorithm has memory $O(n \operatorname{polylog} n)$, and the graph vertices are usually known before processing the stream of edges.

Matchings. 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 [4]. Currently, a variety of semi-streaming matching algorithms for particular settings exist (unweighted/weighted, bipartite/general graphs). Most works consider the multipass scenario [5, 6] 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 [5]. Ahn and Guha [5] provide an overview of the current best algorithms.

In this work, we focus on semi-streaming algorithms that perform one or two passes. In the one-pass setting, in the unweighted case, the greedy matching algorithm is still the best known algorithm (We note that there are algorithmic results in the weighted case [4], [7], [8] [9], but when the edges are unweighted those algorithms are of no help.). The greedy matching algorithm 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 $M \cup \{e\}$ 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 is to address the following question:

Is the greedy matching algorithm best possible, or is there a semi-streaming algorithm with an approximation ratio better than $1/2$?

This is an open question at least since the publication of the first paper [4] on computing matchings in the semi-streaming in 2005. On the negative side, Michael Kapralov showed in [10] that there is no semi-streaming algorithm for maximum matching (even for bipartite graphs) with approximation factor asymptotically better than $1 - 1/e$. This, however, still leaves room between $1/2$ and $1 - 1/e$. To get an approximation ratio better than $1/2$, prior multipass semi-streaming algorithms require at least 3 passes, for instance the algorithm of [6] 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 extensively studied in a variety of settings. The most relevant reference [11] 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, 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 first been studied for the problems of sorting and selecting in limited space by Munro and Paterson [12]. Guha and McGregor [13] gave an exponential separation between random order and adversarial order models. Recently, Kapralov, Khanna and Sudan showed in [14] that under the random order arrival assumption, the size of a maximum matching can be approximated within a constant factor using only polylogarithmic space. 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. Maximum bipartite matching 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 Vazirani and Vazirani [15] (KVV algorithm) 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 [16, 17].

The online model for bipartite matching carries over to the streaming model. In the so-called *vertex arrival order* model, the input edge stream is sorted with respect to the vertices of one bipartition [18, 10]. The KVV algorithm can also be seen as a streaming algorithm when the incoming edge sequence is in vertex arrival order. Surprisingly, it turns out that the KVV algorithm is *optimal*, that is, no semi-streaming algorithm for Maximum Bipartite Matching can achieve an approximation factor better than $1 - 1/e$ [10]. Goel, Kapralov and Khanna showed in [18] that there is a deterministic counterpart to the KVV algorithm in the semi-streaming model that achieves the same approximation factor. This separates the online setting from the vertex arrival order setting in streaming since it is well-known that any deterministic online algorithm for Maximum Bipartite Matching cannot achieve an approximation ratio better than $1/2$.

Our results. In this paper, we present semi-streaming algorithms for maximum matching for bipartite graphs and general graphs with approximation factor strictly larger than $1/2$. Our algorithms make one or two passes over the input. Our one-pass semi-streaming algorithm for bipartite graphs is deterministic and achieves an expected approximation ratio $1/2 + 0.005$ for any graph (**Theorem 1**) assuming that the edges arrive one by one in (uniform) random order. Furthermore, we modify the latter algorithm in order to obtain an algorithm for general graphs, and we obtain an approximation ratio of $1/2 + 0.003$ (**Theorem 2**).

Our 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 bipartite graph and for any arrival order (**Theorem 4**). We achieve the same approximation ratio with a more involved deterministic semi-streaming algorithm (**Theorem 5**). Last, we extend the latter algorithm to general graphs and we obtain an approximation ratio of $\frac{1}{2} + 0.0071$ (**Theorem 6**). Figure 1 provides an overview of our algorithms.

Bipartite/General Graphs	Deterministic/Randomized	Approximation Factor
1 **pass, uniform random order:**		
bipartite	deterministic	$\frac{1}{2} + 0.005$ (**Theorem 1**)
general	deterministic	$\frac{1}{2} + 0.003$ (**Theorem 2**)
2 **passes, arbitrary order:**		
bipartite	randomized	$\frac{1}{2} + 0.019$ (**Theorem 4**)
bipartite	deterministic	$\frac{1}{2} + 0.019$ (**Theorem 5**)
general	deterministic	$\frac{1}{2} + 0.0071$ (**Theorem 6**)

Fig. 1. Overview of our semi-streaming algorithms for maximum matching.

Techniques. The one-pass algorithms as well as the randomized two-pass algorithm each apply three times the greedy matching algorithm on different and carefully chosen subgraphs. The deterministic two-pass algorithms are more complicated as they use subroutines that compute particular subsets of edges besides the greedy algorithm. There is a general idea that is common to all our algorithms that we are going to explain for bipartite graphs:

If we had three passes at our disposal (see for instance Algorithm 2 in [4]), 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 with respect to 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 matched in M_0 and A vertices that are free with respect to M_0 whose combination with M_0 and M_1 forms paths of length 3 that can be used to augment the 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 $(\beta m, m]$ for $0 < \alpha < \beta < 1$, and we 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 2), 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 3 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 and is also used in [19]. 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 see 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. The deterministic one-pass algorithm for bipartite graphs and the deterministic two-pass algorithm for bipartite graphs both extend to general 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 [7] overcomes this problem by repeatedly sampling bipartite graphs from the general graph. Such a strategy, however, is not necessary for our one-pass algorithm. Since the input sequence is in uniform random order, we can show that undesired triangles simply do not appear *too often* allowing our techniques to still work. For our deterministic two-pass algorithm, a combinatorial argument is used to bound the number of those *bad* triangles.

Conference Version. This work builds on the article [20] that was presented at the 15th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2012). Besides a more detailed presentation of the results of [20], the extensions of the algorithms for bipartite graphs to general graphs are discussed.

Outline. We start our presentation with notations and definitions in Section 2. In Section 3, we discuss a well-known result that is reused in all following sections. We point out how the Greedy matching algorithm can be used in 3 passes to obtain an approximation ratio strictly larger than 1/2. In Section 4, we discuss the one-pass algorithm for bipartite graphs and its extension to general graphs. Then, in Section 5 we present our randomized two-pass algorithm for bipartite graphs. Finally, we conclude with the discussion of our deterministic two-pass algorithms for bipartite graphs and general graphs in Section 6.

2 Preliminaries

Let $G = (V, E)$ be a graph with vertex set V and edge set E. If G is bipartite with bipartitions A and B then we write $G = (A, B, E)$ and we denote $V = A \cup B$. Let $n = |V|$ and $m = |E|$. For an edge $e \in E$ with end points $u, v \in V$, we denote e by uv. For a subset of edges $S \subseteq E$ and a vertex $v \in V$, we write $\deg_S(v)$ for the degree of v in S, meaning the number of edges in S that have v as one of its endpoints.

We define now matchings, maximum matchings and maximal matchings.

Definition 1 (Matching). *A* matching *in a graph $G = (V, E)$ is a subset of edges $M \subseteq E$ such that $\forall v \in V : \deg_M(v) \leq 1$. A* maximum matching *$M^*$ is a matching such that for any other matching M' : $|M^*| \geq |M'|$. A* maximal matching *M is a matching that is inclusion-wise maximal, a.e. $\forall e \in E \backslash M : M \cup \{e\}$ is not a matching.*

The MAXIMUM BIPARTITE MATCHING problem consists of computing a maximum matching in a bipartite graph and we abbreviate it by MBM.

The MAXIMUM MATCHING problem consists of computing a maximum matching in a general graph and we abbreviate it by MM.

For a subset of edges $F \subseteq E$, we denote by $\text{opt}(F)$ a maximum matching in the graph G restricted to edges F. We may write $\text{opt}(G)$ for $\text{opt}(E)$, and M^* for $\text{opt}(G)$. 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 $S \subseteq V$, we write $\text{opt}(S)$ for $\text{opt}(G|_S)$, that is a maximum matching in the subgraph of G induced by vertices S. In case of bipartite graphs, for $S_A \subseteq A$ and $S_B \subseteq B$ we write $\text{opt}(S_A, S_B)$ for $\text{opt}(G|_{S_A \cup S_B})$. Moreover, for two sets S_1, S_2 we denote by $S_1 \oplus S_2$ the symmetric difference $(S_1 \setminus S_2) \cup (S_2 \setminus S_1)$ of the two sets.

A standard technique to increase the size of matchings is to search for *augmenting paths*. We define augmenting paths as follows.

Definition 2 (Augmenting Path). *Let $p \geq 3$ be an odd integer. Then a length p augmenting path with respect to a matching M in a graph $G = (V, E)$ is a path $P = (v_1, \ldots, v_{p+1})$ such that $v_1, v_{p+1} \notin V(M)$ and for $i \leq 1/2(p-1) : v_{2i}v_{2i+1} \in M$, and $v_{2i-1}v_{2i} \notin M$.*

An augmenting path of length p ($p \geq 3, p$ odd) with respect to a matching M in a graph $G = (V, E)$ is a path that starts and ends at nodes that are not matched in M. We call such nodes *free* nodes. All internal nodes of the path are matched in M, and we call these nodes *matched* nodes. The path alternates between edges outside M and edges of M. Removing from M the edges of the augmenting path that are also in M and inserting into M the edges outside M increases the size of M by 1.

The input graph G is given as a graph stream, i.e. as a sequence of edges arriving one by one in some order. Let $\Pi(G)$ be the set of all edge sequences of G. An input stream for our streaming algorithms is then an edge sequence $\pi \in \Pi(G)$. We write $\pi[i]$ for the i-th edge of π, and $\pi[i, j]$ for the subsequence $\pi[i]\pi[i+1]\ldots\pi[j]$. In this notation, a round bracket 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.

Definition 3 (Semi-streaming Algorithm). *A $p(n)$-pass* semi-streaming algorithm ***S*** *on input graph G with update time $t(n)$ is an algorithm such that, for every input stream $\pi \in \Pi(G)$:*

1. ***S*** *performs at most $p(n)$ passes on stream π,*
2. ***S*** *maintains a random access memory of size $O(n \,\text{polylog}\, n)$,*
3. ***S*** *has running time $O(t(n))$ between two consecutive read operations from the stream.*

Furthermore, preprocessing time (the time before the first read operation) and postprocessing time (the time after the last read operation and the output of the result) is $O(t(n))$. We assume that read operations on any stream require constant time.

We say that an algorithm **A** computes a *c-approximation to the maximum matching problem* 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.

The Greedy matching algorithm is illustrated in Algorithm 1. It is easy to see that this algorithm can be seen as a semi-streaming algorithm with approximation factor 1/2 and update time $O(1)$.

Algorithm 1 The Greedy Matching Algorithm

```
M ← ∅
while edge stream not empty do
    e = v1v2 ← next edge in stream
    if {v1, v2} ∩ V(M) = ∅ then M ← M ∪ {e} end if
end while
return M
```

3 Three-pass Semi-Streaming Algorithm for Bipartite Graphs on Adversarial Order

To improve on the Greedy matching algorithm with three passes, a simple strategy is to, firstly, compute a maximal matching M_G in one pass, and then use the second and the third pass to search for 3-augmenting paths to augment M_G.

Suppose that M_G is close to a 1/2-approximation.Then almost all edges of M_G are 3-*augmentable*. We say that an edge $e \in M_G$ is 3-augmentable if the removal of e from M allows the insertion of two edges $f, g \in M^* \setminus M$ into M. More formally, the following lemma holds.

Lemma 1. *Let $\epsilon \geq 0$. 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. □

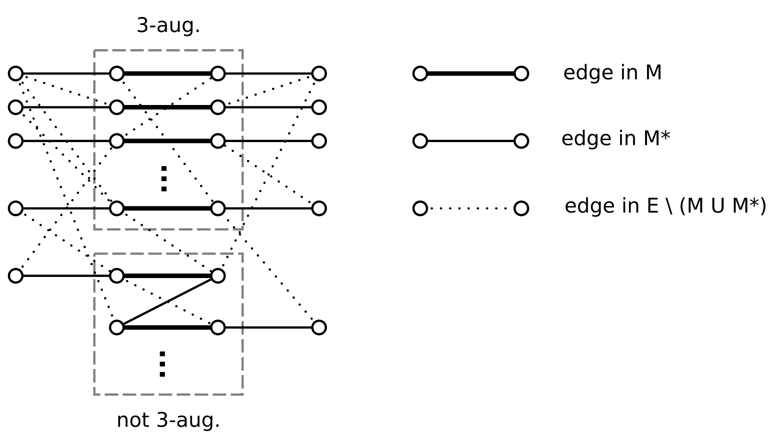

Fig. 2. Illustration of Lemma 1. If $|M| \leq (1/2 + \epsilon)|M^*|$, then at least $(\frac{1}{2} - 3\epsilon)|M^*|$ edges of M are 3-augmentable.

We search for 3-augmenting paths as follows. Firstly, we compute a maximal matching M_L via the Greedy algorithm between the A vertices that are matched in M_G and the free B vertices. Under the assumption that M_G is close to a 1/2 approximation, most of the edges of M_G are 3-augmentable. There exists hence a large matching, and since M_L is a maximal matching, M_L will be at least of size 1/2 times the number of 3-augmentable edges. Edges from M_L will serve as the start of length 3-augmenting paths. Then in the third pass, we compute another maximal matching M_R in order to complete 3-augmenting paths with the edges of M_G and M_L. This algorithm is stated in Algorithm 2, and illustrated in Figure 3. This idea was already used in [4]. The authors present there an $O((\log \frac{1}{\epsilon})/\epsilon)$-pass semi-streaming algorithm that computes a $2/3 - \epsilon$ approximation to the maximum bipartite matching problem. An analysis for Algorithm 2 can be derived from their work.

Algorithm 2 Three-pass Bipartite Matching Algorithm

Require: The input stream π is an edge stream of a bipartite graph $G = (A, B, E)$
1: $M_G, M_L, M_R \leftarrow \varnothing$
2: **1st pass:** $M_G \leftarrow \text{Greedy}(\pi)$
3: $G_L \leftarrow$ complete graph between $A(M_G)$ and $B \setminus B(M_G)$
4: **2nd pass:** $M_L \leftarrow \text{Greedy}(\pi \cap G_L)$
5: $G_R \leftarrow$ complete graph between $\{b \in B(M_G) : A(M_G(b)) \in A(M_L)\}$ and $A \setminus A(M_G)$
6: **3rd pass:** $M_R \leftarrow \text{Greedy}(\pi \cap G_R)$
7: **return** maximum matching in $M_G \cup M_L \cup M_R$

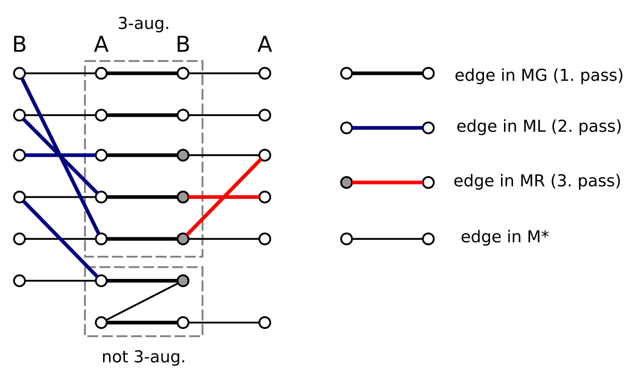

Fig. 3. Illustration of Algorithm 2. The graph contains a perfect matching of size 13. In the first pass, M_G is computed and it has size 7. This is close to a 1/2 approximation and by Lemma 1, M has many (here 5) 3-augmentable edges. There exists hence a matching of size at least 5 between $A(M_G)$ and the free B vertices. Since M_L is maximal, it is of size at least 5/2 (here 4). Then, a maximal matching is computed between the solid vertices, which are the B vertices of edges of M_G that may potentially be completed to 3-augmenting paths, and the free A vertices. In this example, two 3-augmenting paths were found.

4 One-pass Matching Algorithm on Random Order

We discuss now, how the 3-pass algorithm from the previous section can be simulated with a single pass if the input is in random order. First, we present in Subsection 4.1 a lemma about the convergence of the Greedy matching algorithm if the input is in random order. This lemma is the main ingredient for our one-pass algorithms. Then, in Subsection 4.2 we discuss our one-pass algorithm on random order for bipartite graphs, and we extend it to general graphs in Subsection 4.3.

4.1 A Lemma on the Convergence of the Greedy Algorithm

We identify a property about the convergence of the Greedy algorithm that is required for the construction of our one-pass algorithms on random order. We show that if in expectation over all input edges sequences the Greedy algorithm computes a matching that is close to a 1/2 approximation, then Greedy builds this matching early on, or in other words, the Greedy algorithm converges quickly, see Lemma 2.

Lemma 2. *If* $\mathbb{E}_\pi |\text{Greedy}(\pi)| \leq (\frac{1}{2}+\epsilon)|M^*|$ *for some* $0 < \epsilon < 1/2$*, then for any* $0 < \alpha \leq 1$,

$$\mathop{\mathbb{E}}_{\pi} |\text{Greedy}(\pi[1,\alpha m])| \geq |M^*|(\frac{1}{2} - (\frac{1}{\alpha} - 2)\epsilon).$$

Proof. Let $M_0 = \text{Greedy}(\pi[1,\alpha m])$. 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 $\chi[X] = 0$. We show in Lemma 3 that

$$\Pr[a \text{ and } b \notin V(M_0)] \leq (\frac{1}{\alpha} - 1)\Pr[e \in M_0]. \tag{1}$$

Taking expectations and using Inequality 1,

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

We will show in Lemma 4 that for a maximum matching M^* and any maximal matching M_G, we have $|M_G \cap M^*| \leq 2(|M_G| - 1/2|M^*|)$. Using this, and since M_0 is just a subset of the edges of M_G, we obtain by linearity of expectation

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

Combining gives the Lemma. ☐

We now prove Lemma 3 that was used in the proof of Lemma 2.

Lemma 3. *Suppose that* $\mathbb{E}_\pi |\text{Greedy}(\pi)| \leq (\frac{1}{2}+\epsilon)|M^*|$ *for some* $0 < \epsilon < 1/2$*. Let* $M_0 = \text{Greedy}(\pi[1,\alpha m])$ *for some* $0 < \alpha \leq 1/2$*. Then:*

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

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 $b \notin V(M_0 \setminus \{e\})$. More 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 $[1,\alpha m]$. Thus,

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

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. Then we conclude the Lemma by the random order assumption $\Pr[e \in \pi[1,\alpha m]] = \alpha$. ☐

Lemma 4 shows that an optimal matching and a maximal matching that is far from this optimal matching in size do not have many edges in common.

Lemma 4. *Let M be a maximal matching of a graph 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. □

4.2 Bipartite Graphs

Algorithm We simulate the 3-pass algorithm, Algorithm 2, in one pass as follows. We split the input graph stream $\pi \in \Pi(G)$ into three phases $\pi[1, \alpha m]$, $\pi(\alpha m, \beta m]$, and $\pi(\beta m, m]$ (for $0 < \alpha < \beta < 1$), and we build a matching in each phase. M_0 is built during the first phase and corresponds to matching M_G of our 3-pass algorithm. M_1 is built in the second phase and M_2 in the third, and they correspond to M_L and M_R of our 3-pass algorithm, respectively. Assume that Greedy performs badly on the input graph G. Lemma 1 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 4.

Algorithm 3 One-pass Bipartite Matching on Random Order

```
α ← 0.4312, β ← 0.7595
M_G ← Greedy(π)
M_0 ← Greedy(π[1, αm]), matching obtained by Greedy on the first ⌊αm⌋ edges
F_1 ← complete bipartite graph between B(M_0) and ‾A(M_0)‾
M_1 ← Greedy(F_1 ∩ π(αm, βm]), matching obtained by Greedy on edges ⌊αm⌋ + 1 through βm that intersect F_1
A′ ← {a ∈ A | ∃b ∈ B(M_1) : ab ∈ M_0}
F_2 ← complete bipartite graph between A′ and ‾B(M_0)‾
M_2 ← Greedy(F_2 ∩ π(βm, m]), matching obtained by Greedy on edges ⌊βm⌋ + 1 through m that intersect F_2
M ← matching obtained from M_0 augmented by M_1 ∪ 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, the 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.

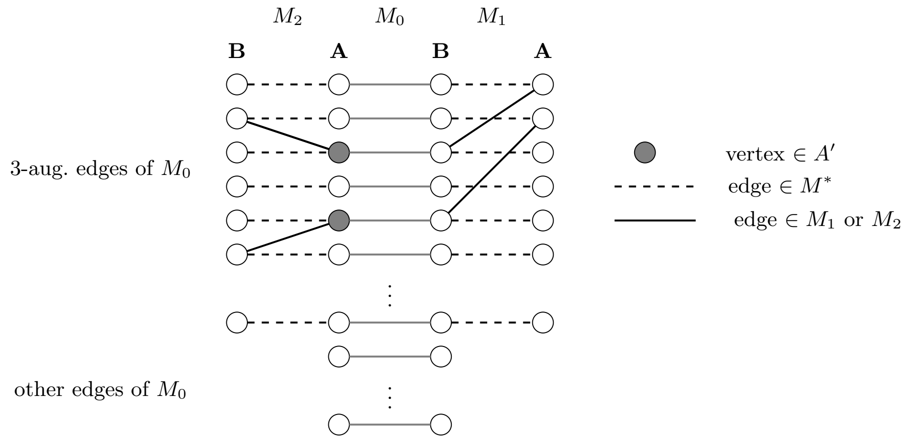

Fig. 4. Illustration of Algorithm 3. Note that every edge of M_2 completes a 3-augmenting path consisting of one edge of M_1 (on the right hand side of the picture) followed by one edge of M_0 (center) followed by one edge of M_2 (on the left hand side of the picture).

Analysis We use the notations of Algorithm 3. Consider α and β as variables with $0 \leq \alpha \leq \frac{1}{2} < \beta < 1$.

Lemma 5. *Assume that $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2} + \epsilon)|M^*|$. Then the expected size of a 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:*

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

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 1, 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 2 applied to M_0 concludes the proof. ☐

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,

$$\mathop{\mathbb{E}}_{\pi} |M_1| \geq \frac{1}{2} \mathop{\mathbb{E}}_{\pi} |\text{opt}(\overline{A(M_0)}, B(M_0)) \cap \pi(\alpha m, \beta m]|.$$

By independence of 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 in 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:

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

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 1 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.

Theorem 1. *Algorithm 3 is a deterministic one-pass semi-streaming algorithm for* MBM *with expected approximation ratio* $\frac{1}{2} + 0.005$ *against (uniform) random order for any graph, and can be implemented with* $O(1)$ *update time.*

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

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

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

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

Combining,

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

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. □

4.3 Extension to General Graphs

In this section, we show how the one-pass algorithm of Section 4.2 can be adapted to general graphs $G = (V, E)$.

Algorithm Algorithm 4 follows the same line as Algorithm 3 for the bipartite case. While in the bipartite case, edges from M_1 extend M_0 on only one bipartition, and those edges do not interfere with edges from M_2, this structure is no longer given in the general setting. Here, M_1 is a Greedy matching between the matched vertices in M_0 and all free vertices. This may already produce some 3-augmenting paths, however, it may also happen that by taking a *bad* edge into M_1, this rules out any possibility of finishing the 3-augmenting paths containing these edges. We call the edge of M_0 *blocked* if it can not be completed to a 3-augmenting path, see Definition 4.

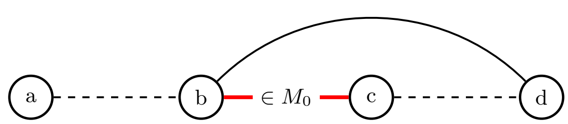

Fig. 5. If edge bd is taken into M_1 and edge $ac \notin E$, this may block the 3-augmenting path ab, bc, cd. In that case we call bc blocked.

We show in Lemma 11 that the probability that an edge of M_0 will become blocked is at most 1/2. This guarantees that we can finalize many 3-augmenting paths by the Greedy matching M_2.

Aug is a set of length 3 paths. $|Aug|$ denotes the number of length 3 paths in *Aug*. For some vertex $a \in V$ (resp. some edge $e \in E$), we write $a \in Aug$ (resp. $e \in Aug$) if a (resp. e) is part of some length 3 path.

Algorithm 4 One-pass Matching on Random Order for General Graphs

$\alpha \leftarrow 0.413, \beta \leftarrow 0.708$
$M_G \leftarrow \text{Greedy}(\pi)$
$M_0 \leftarrow \text{Greedy}(\pi[1, \alpha m])$, matching obtained by Greedy on the first $\lfloor \alpha m \rfloor$ edges
$F_1 \leftarrow$ complete bipartite graph between $V(M_0)$ and $\overline{V(M_0)}$
$M_1 \leftarrow \text{Greedy}(F_1 \cap \pi(\alpha m, \beta m])$, matching obtained by Greedy on edges $\lfloor \alpha m \rfloor + 1$ through βm that intersect F_1
$Aug \leftarrow$ length 3 paths in $M_0 \oplus M_1$
$V_1 \leftarrow \{u \in V \setminus V(Aug) \,|\, \exists v \in V(M_1) : uv \in M_0\}$
$V_2 \leftarrow \overline{V(M_0)} \setminus V(Aug)$
$F_2 \leftarrow$ maximal bipartite graph between V_1 and V_2 such that $\nexists\, m_0 \in M_0 \setminus Aug, m_1 \in M_1 \setminus Aug, f_2 \in F_2$ st. they form a triangle
$M_2 \leftarrow \text{Greedy}(F_2 \cap \pi(\beta m, m])$, matching obtained by 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

Analysis We bound the size of a maximum matching between $V(M_0)$ and $\overline{V(M_0)}$.

Lemma 9. *Assume that* $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2} + \epsilon)|M^*|$. *Then:*

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

Proof. The size of a maximum matching between $\overline{V(M_0)}$ and $V(M_0)$ is at least twice the number of augmenting paths of length 3 in $M_0 \oplus M^*$. By Lemma 1, 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 $(\frac{1}{2} - 3\epsilon)|M^*| - (\mathbb{E}\,|M_G| - \mathbb{E}\,|M_0|)$ edges that are 3-augmentable. Lemma 2 applied to M_0 concludes the proof. □

Lemma 10. *Assume that* $\mathbb{E}_\pi |M_G| \leq (\frac{1}{2} + \epsilon)|M^*|$. *Then:*

$$\mathbb{E}\,|M_1| \geq \frac{1}{2}(\beta - \alpha)(\mathbb{E}\,|\text{opt}(V(M_0), \overline{V(M_0)})| - \frac{1}{1 - \alpha}).$$

Proof. The proof is identical to the proof of Lemma 6. □

Definition 4 (Blocked edge). *Let* $e = uv \in M_0$ *such that* e *is* 3*-augmentable by edges* $o_1 = uu', o_2 = vv' \in M^*$. *We call* e *blocked, if:*

1. *either* $uv' \in E$ *or* $u'v \in E$ *(not both of them), and*
2. *if* $uv' \in E$ *then* $uv' \in M_1$, *otherwise* $u'v \in M_1$.

Lemma 11.

$$\Pr[e \text{ } blocked \text{ } | \, e \in M_0] \leq \frac{1}{2}.$$

Proof. W.l.o.g. let $uv' \in E$ and $u'v \notin E$.

$$\begin{aligned}\Pr[e \text{ blocked } | \, e \in M_0] &= \Pr[e \notin Aug \text{ and } uv' \in M_1 \,|\, e \in M_0] \\ &\leq \Pr[uv' \in M_1 \,|\, e \in M_0 \setminus Aug].\end{aligned}$$

Since $\Pr[uv' \in M_1 \,|\, e \in M_0 \setminus Aug] = \Pr[vv' \in M_1 \,|\, e \in M_0 \setminus Aug]$, and since the events $(uv' \in M_1 \,|\, e \in M_0 \setminus Aug)$ and $(vv' \in M_1 \,|\, e \in M_0 \setminus Aug)$ exclude each other, the result follows. □

Lemma 12.

$$\mathbb{E}\,|\text{opt}(F_2)| \geq \max\{\frac{1}{2}(\mathbb{E}\,|M_1| - 4|Aug| - 4\epsilon|M^*|), 0\}.$$

Proof. The size of a maximum matching in F_2 is at least the number of length 2 paths in $M_0 \oplus M_1$ that can be completed to a 3-augmenting path. Denote by k_2 the number of length two paths in $M_0 \oplus M_1$. Then, $|M_1| = 2|Aug| + k_2$. A length 3 path may block at most 2 other length 2 paths from being completed.

By Lemma 1, the number of edges of $|M_G|$ that are not 3-augmentable is in expectation at most $(\frac{1}{2} + \epsilon)|M^*| - (\frac{1}{2} - 3\epsilon)|M^*| = 4\epsilon|M^*|$. Since M_0 is a subset of M_G, it follows that at most $4\epsilon|M^*|$ edges from M_0 are not 3-augmentable. Hence, the number of M_0 edges for which a length two path was found and which is 3-augmentable is at least $(k_2 - 2|Aug| - 4\epsilon|M^*|)$. In expectation, by Lemma 11, at most half of these edges are blocked. The Lemma follows. □

Lemma 13.

$$\mathbb{E}\,|M_2| \geq \frac{1}{2}\left((1-\beta)\,\mathbb{E}\,|\mathrm{opt}(F_2)| - 1\right).$$

Proof. This proof is identical to the proof of Lemma 8. □

We now present the proof of the main theorem, Theorem 2.

Theorem 2. *Algorithm 4 is a deterministic one-pass semi-streaming algorithm for* Maximum Matching *with approximation ratio* $\frac{1}{2} + 0.00363$ *in expectation over (uniform) random order for any graph, and can be implemented with* $O(1)$ *update time.*

Proof. The expected matching size is

$$\mathbb{E}\,|M| \geq \mathbb{E}\,|M_0| + |Aug| + \frac{1}{2}\,\mathbb{E}\,|M_2|, \tag{2}$$

since, by construction, at least half of the edges of M_2 can be used to complete a 3-augmenting path. Firstly, we bound $|M_2|$ by Lemma 13 and Lemma 12 and we obtain

$$\mathbb{E}\,|M_2| \geq \max\{0, (1-\beta)(\frac{1}{4}\,\mathbb{E}\,|M_1| - |Aug| - \epsilon|M^*|) - O(1)\}. \tag{3}$$

By Lemma 10 and Lemma 9, we bound the size of M_1 and we obtain

$$\mathbb{E}\,|M_1| \geq \frac{1}{2}|M^*|(\beta-\alpha)(1 - 2(\frac{1}{\alpha} + 2)\epsilon) - O(1). \tag{4}$$

Using Inequality 4 in Inequality 3, we obtain

$$\mathbb{E}\,|M_2| \geq \max\{0, (1-\beta)(\frac{1}{8}|M^*|\left((\beta-\alpha)(1-2(\frac{1}{\alpha}+2)\epsilon) - \epsilon\right) - |Aug|) - O(1)\}. \tag{5}$$

Furthermore, in Lemma 2 we show that $\mathbb{E}_\pi\,|M_0| \geq |M^*|(\frac{1}{2} - (\frac{1}{\alpha} - 2)\epsilon)$. We use this and Inequality 5 in Inequality 2 and we obtain an Inequality for $\mathbb{E}\,|M|$ that depends on $\alpha, \beta, |Aug|$ and ϵ. It is easy to see that this Inequality is minimized if $|Aug| = 0$.

The expected value of the output of the Algorithm is at least $\min_\epsilon \max\{(\frac{1}{2} + \epsilon)|M^*|, \mathbb{E}_\pi\,|M|\}$. By a numerical search we optimize parameters α, β. Setting $\alpha = 0.413, \beta = 0.708$, we obtain $\epsilon \approx 0.00363$. which proves the Theorem. □

5 Randomized Two-pass Algorithm on any Order

We present now a randomized two-pass semi-streaming algorithm for Maximum Bipartite Matching with approximation ratio strictly greater than $\frac{1}{2}$. This algorithm simulates the three passes of the 3-pass algorithm of Section 3 in two passes. We require a new property of the Greedy algorithm that may be of independent interest. In Subsection 5.1, we discuss this new property. Then, we present in Subsection 5.2 our two-pass randomized algorithm for bipartite graphs.

Algorithm 5 Matching a Random Subset of Vertices (Bipartite Graphs)

1: Take independent random sample $A' \subseteq A$ st. $\Pr[a \in A'] = p$, for all $a \in A$
2: Let F be the complete bipartite graph between A' and B
3: **return** $M' = \text{Greedy}(F \cap \pi)$

5.1 Matching Many Vertices of a Random Vertex Subset

Consider a bipartite graph $G = (A, B, E)$. For a fixed parameter $0 < p \leq 1$, Algorithm 5 generates an independent random sample of vertices $A' \subseteq A$ such that $\Pr[a \in A'] = p$, for all $a \in A$, and runs then the Greedy algorithm on the subgraph $G|_{A' \times B}$.

We prove in Theorem 3 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.

The proof of Theorem 3 will use Wald's equation for super-martingales, see [21], Wald's Equation, p.300, section 12.3.[4]

Lemma 14 (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. *$\mathbb{E}[\Phi(S_{i+1}) - \Phi(S_i) \mid S_i] \leq \mu$ for all $i < D$.*

Then:

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

Theorem 3. *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 $\pi \in \Pi(G)$:*

$$\mathop{\mathbb{E}}_{A'} |\text{Greedy}(F \cap \pi)| \geq \frac{p}{1+p} |\text{opt}(G)|.$$

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 14. 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 with respect to 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

[4] The theorem cited in the book is actually weaker than the one we need, but our statement follows from the proof of that Theorem.

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:

$$\mathbb{E}(\Delta\phi(i) \mid S_i) \leq 1 + p. \tag{6}$$

Indeed, since $\Delta\phi(i)$ only takes on values 0, 1 or 2, we can write that

$$\mathbb{E}(\Delta\phi(i) \mid S_i) \leq 1 + \Pr[\Delta\phi(i) = 2 \mid S_i].$$

To bound the latter probability, let $e = ab$ denote the edge of $M'_{i+1} \setminus M'_i$ and let t be such that $e = \pi[t]$. In order for e to change ϕ by 2, it must be that b is matched in M^* to a node a' that is also in A''. Furthermore, it is required that a' was unmatched before edge e arrived. Since a' was unmatched up to 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

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

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

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

implying Inequality 6. Applying Wald's Stopping Lemma, we obtain

$$\mathbb{E}\,\phi(D) \leq (1+p)\,\mathbb{E}\,D.$$

Finally, since $\mathbb{E}\,\phi(D) = \mathbb{E}\,|A''| = p \cdot |\text{opt}(G)|$ and $D \leq |\text{Greedy}(F \cap \pi)|$, and the Theorem follows. □

5.2 A Randomized Two-pass Algorithm for Bipartite Graphs

Based on Theorem 3, we design our randomized two-pass algorithm for bipartite graphs $G = (A, B, E)$. Assume that Greedy(π) returns a matching that is close to a $\frac{1}{2}$-approximation. In order to apply Theorem 3, 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 6 and Figure 7 for an illustration. 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. In the prove of Theorem 4, we show that if Greedy(π) is close to a $\frac{1}{2}$-approximation, then we find many 3-augmenting paths.

Algorithm 6 Two-pass Randomized Bipartite Matching Algorithm

```
Let p ← √2 − 1.
Take an independent random sample A' ⊆ A st. Pr[a ∈ A'] = p, for all a ∈ A
Let F_1 be the set of edges with one endpoint in A'.
First pass: M_0 ← Greedy(π) and M' ← Greedy(F_1 ∩ π)
M_1 ← {e ∈ M' | e goes between B(M_0) and overline(A(M_0))}
A_2 ← {a ∈ A(M_0) : ∃b, c : ab ∈ M_0 and bc ∈ M_1}.
Let F_2 ← {da : d ∈ overline(B(M_0)) and a ∈ A(M_0) and ∃b, c : ab ∈ M_0 and bc ∈ M_1}.
Second pass: M_2 ← Greedy(F_2 ∩ π)
Augment M_0 by edges in M_1 and M_2 and store it in M
return the resulting matching M
```

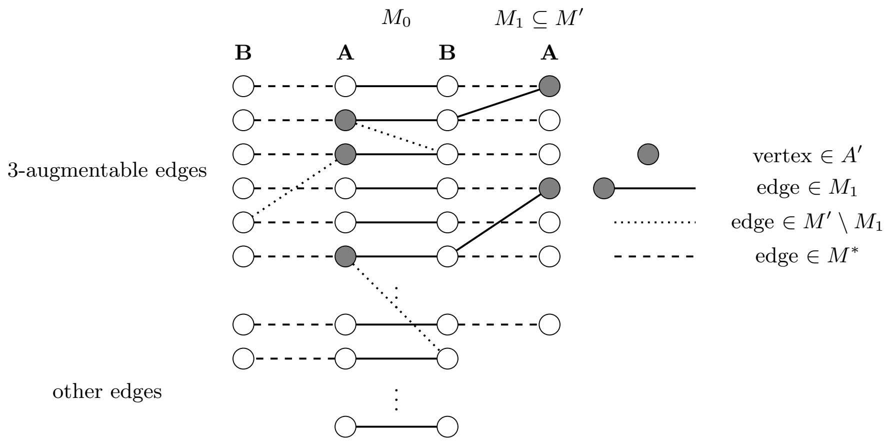

Fig. 6. Illustration of the first pass of Algorithm 6. By Theorem 3, nearly all vertices of A' are matched in M', in particular those that are not matched in M_0.

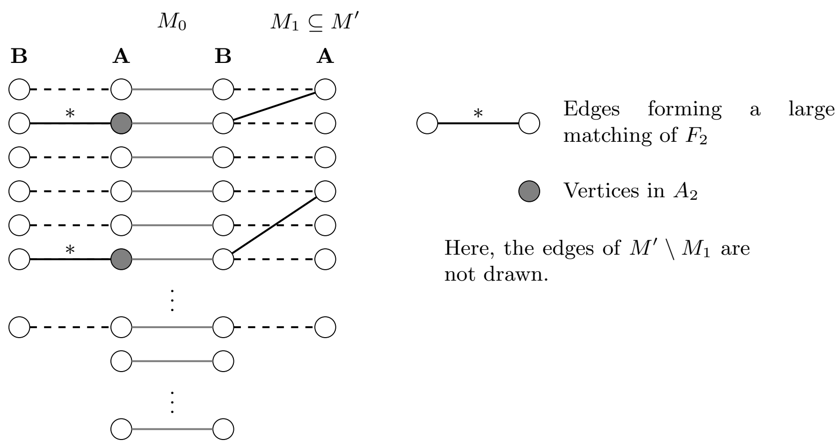

Fig. 7. Analysis of the second pass of Algorithm 6. Here, we see that $M_0 \oplus M_1$ has two paths of length 2, and that both of those paths can be extended into 3-augmenting paths using M^*: this illustrates $|\text{opt}(F_2)| \geq 2$. Matching M_2, being a 1/2 approximation, will find at least one 3-augmenting path.

Theorem 4. *Algorithm 6 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* $O(1)$ *update time.*

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:

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

Lemma 1 gives $|\{ab \in M_0 \mid ab \text{ is not 3-augmentable with } M^*\}| \le 4\epsilon|\mathrm{opt}(G)|$. It remains to bound $|\{ab \in M_0 \mid \exists c : bc \in M_1\}|$ from below. By definition of M' and of $M_1 \subseteq M'$, and by maximality of M_0,

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

Taking expectations, by Theorem 3 and by independence of M_0 from A':

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

Combining:

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

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. □

6 Deterministic Two-pass Algorithm on any Order

We discuss now deterministic two-pass streaming algorithms for Maximum Bipartite Matching and Maximum Matching for input streams in adversarial order. We start our presentation with an algorithm for bipartite graphs in Section 6.1. Then, we show how this idea can be extended to general graphs in Section 6.2.

6.1 Bipartite Graphs

Algorithm The deterministic two-pass algorithm, Algorithm 8, follows the same line as its randomized version, Algorithm 6. In a first pass, we compute a Greedy matching M_0 and some additional edges S that we compute by Algorithm 7. If M_0 is close to 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. In Algorithm 6, 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 $\lambda \ge 2$, an *incomplete λ-bounded semi-matching* S of a bipartite graph $G = (A, B, E)$ is a subset $S \subseteq E$ such that $\deg_S(a) \le 1$ and $\deg_S(b) \le \lambda$, 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 15, we show that Algorithm 7, a straightforward greedy algorithm, computes an incomplete λ-bounded semi-matching that covers at least $\frac{\lambda}{\lambda+1}|M^*|$ vertices of A. Now, assume that the greedy matching

Algorithm 7 Incomplete λ-bounded Semi-matching iSEMI(λ)

```
S ← ∅
while ∃ edge ab in stream
  if deg_S(a) = 0 and deg_S(b) ≤ λ − 1 then S ← S ∪ {ab}
return S
```

algorithm computes a M_0 close to a $\frac{1}{2}$-approximation. Then, for $\lambda \ge 2$ 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 8 where λ is set to 3, see Theorem 5.

We show two figures illustrating the first pass (Figure 8) and the second pass (Figure 9) of Algorithm 8.

Algorithm 8 Two-pass Deterministic Bipartite Matching Algorithm

First pass: $M_0 \leftarrow \text{Greedy}(\pi)$ and $S \leftarrow \text{iSEMI}(3)$
$S_1 \leftarrow \{e \in S \mid e = ab \text{ such that } a \in \overline{A(M_0)} \text{ and } b \in B(M_0)\}$
$A_2 \leftarrow \{a \in A(M_0) \mid \exists bc : ab \in M_0 \text{ and } bc \in S_1\}$
$F \leftarrow \{e \mid e = ab \text{ such that } a \in A_2 \text{ and } b \in \overline{B(M_0)}\}$
Second pass: $M_2 \leftarrow \text{Greedy}(\pi \cap F)$
Augment M_0 by edges in S_1 and M_2 and store it in M
return M

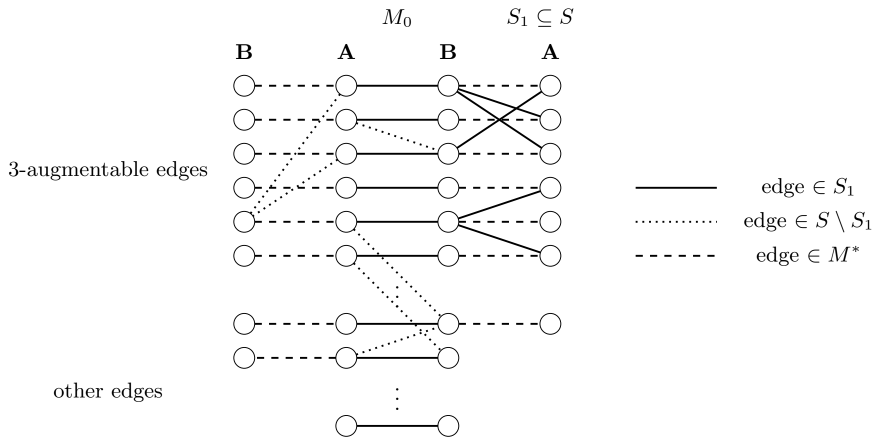

Fig. 8. Illustration of the first pass of Algorithm 8. In this example we set $\lambda = 2$ and we compute an incomplete degree 2 limited semi-matching S. By Lemma 15, we match at least $\frac{2}{3}|M^*|$ A vertices. Since $|M| \approx \frac{1}{2}|M^*|$, some A vertices that are not matched in M_0 are matched in S. The edges incident to those define S_1.

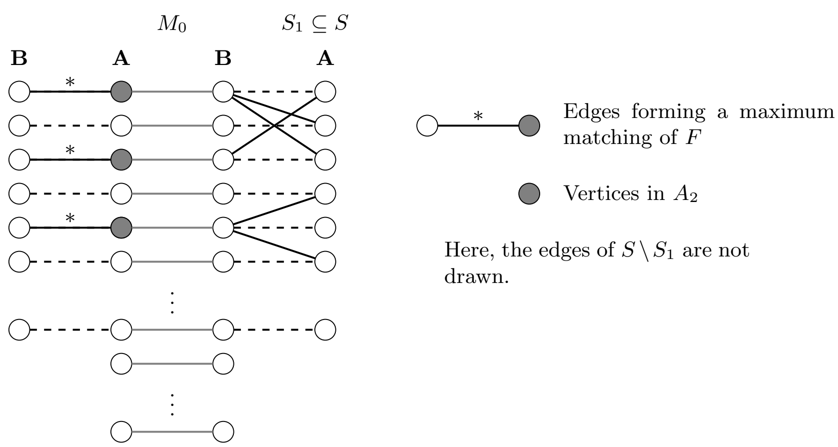

Fig. 9. Analysis of the second pass of Algorithm 8. In this example, we set $\lambda = 2$. Here, we see that $M_0 \oplus S_1$ has five paths of length 2. These paths are not disjoint, but since the maximal degree in S is 2, $M_0 \oplus S_1$ has at least $\frac{1}{2} \cdot 5$ disjoint paths, and hence $|A_2| = 3 \geq \frac{1}{2} \cdot 5$. A maximum matching in F 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.

Analysis We firstly present a lemma, Lemma 15, that analyses Algorithm 7. This lemma is then used in the proof of the main theorem, Theorem 5.

Lemma 15. *Let $S = \text{iSEMI}(\lambda)$ be the output of Algorithm 7 for some $\lambda \geq 2$. Then S is an incomplete λ-bounded semi-matching such that $|A(S)| \geq \frac{\lambda}{\lambda+1}|M^*|$.*

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 $|M^*| - |A(S)| \leq |A(M^*) \setminus A(S)|$. □

Theorem 5. *Algorithm 8 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* $O(1)$ *update time.*

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 S_1 that allows the construction of a 3-augmenting path. Each 3-augmenting path increases the matching M_0 by 1. See also Figure 9. Since $|M_2|$ is a maximal matching of the graph induced by the edges F, we obtain

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

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

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

A_2 are those vertices matched also by M_0 such that there exists an edge in S_1 matching the mate of the A_2 vertex. Since the maximal degree in S_1 is λ, we can bound $|A_2|$ by

$$|A_2| \geq \frac{1}{\lambda}|S_1|.$$

Note that $|S_1| = |A(S) \setminus A(M_0)|$ 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, $|S_1| = |S_{\overline{M_0}}|$ since an edge of S is taken into S_1 if it is in $S_{\overline{M_0}}$. Lemma 15 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

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

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

6.2 Extension to General Graphs

Algorithm The deterministic two-pass algorithm for general graphs follows the same line as the deterministic two-pass algorithm for bipartite graphs. In the first pass, Greedy matching M together with some additional edges F are computed. F forms an *incomplete b-bounded forest.*

Definition 5 (incomplete b-bounded forest). *Given an integer b, an incomplete b-bounded forest F is a cycle free subset of the edges of a graph $G = (V, E)$ with maximal degree b.*

If $F \oplus M$ contains 3-augmenting paths, we augment M by a maximal set of disjoint 3-augmenting paths and store the result in M'. Those edges of F that were not used in the previous augmentation and that form length-2 paths with edges of M' are stored in M_R. In a second pass, length-2 paths of $M' \cup M_R$ are completed to 3-augmenting paths by computing a matching M_L. A maximal set of disjoint 3-augmenting paths of $M' \cup M_L \cup M_R$ is then used to augment M'. Algorithm 9 is a greedy algorithm that constructs a

---

Algorithm 9 b-bounded Forest: FOREST(b)

Require: b

$S \leftarrow \varnothing$
while stream not empty **do**
    $uv \leftarrow$ next edge in stream
    if ($\deg_S(u) = 0$ and $\deg_S(v) \le b - 1$) **or** ($\deg_S(u) \le b - 1$ and $\deg_S(v) = 0$) **then**
        $S \leftarrow S \cup \{uv\}$
    end if
end while
return S

Algorithm 10 Two-pass Deterministic Matching Algorithm for General Graphs

Require: b

1: $Aug \leftarrow \varnothing$
2: $M \leftarrow$ Greedy() and $F \leftarrow$ FOREST(b) {**first pass**}
3: $M' \leftarrow M$ augmented by a maximal set of 3-augmenting paths in $M \oplus F$
4: $M_R \leftarrow$ maximal subset of F such that $\forall uv \in M_R : u \in V(M')$ and $\deg_{M_R}(v) = 1$
5: $V' \leftarrow \{v \in V(M') \,:\, v' = M'(v)$ and $\exists v'u \in M_R : u \notin V(M')\}$
6: **while** stream not empty **do** {**second pass**}
7:     $vw \leftarrow$ next edge in stream
8:     **if** $v \in V'$ and $w \notin V(M')$ and vw completes a 3-augmenting path with edges $uv \in M', tu \in M_R$ **then**
9:         $Aug \leftarrow Aug \cup \{vw, tu\}$, remove all edges from M_R incident to u
10:     **end if**
11: **end while**
12: $M'' \leftarrow M'$ augmented with Aug
13: **return** M''

---

forest F such that the maximal degree of a node in F is b, for some $b \ge 1$. For a large enough b, all but a small fraction of the vertices of the graph are covered by an edge in F.

The situation of the algorithm after the first pass is illustrated in Figure 10. Note that M_R is an incomplete b-bounded semi-matching in the induced bipartite graph with vertex sets $V \setminus V(M')$ and $V(M')$.

Analysis The analysis refers to the variables that are used in the algorithm. Furthermore, let M^* denote a maximum matching in the input graph and let ϵ be such that $|M| = (1/2 + \epsilon)|M^*|$. Let $\alpha = \frac{|M'|}{|M|} - 1$, or in other words, the set of disjoint 3-augmenting paths found in Line 3 is of size $\alpha|M|$.

The analysis of the algorithm requires a lemma concerning the structure of forests.

Lemma 16. *Let T be a forest with at least k nodes of degree at least d. Then:*

$$|T| \ge (d-1)k.$$

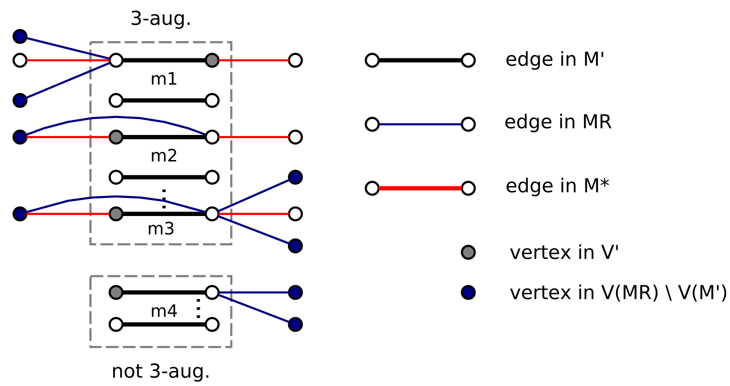

Fig. 10. Situation of Algorithm 10 after the first pass. M' is the resulting matching that is obtained by augmenting M with edges from F. M_R is a maximal subset of the edges of F that were not used for augmenting M such that vertices that are free in F with respect to M' have a degree of one.

Proof. Consider the directed Graph D that is obtained from T by directing the edges from the roots of the trees of T towards the leaves. Let $v_1, \ldots, v_k$ denote the nodes that have degree at least d. Then for all $i \neq j : \Gamma_D(v_i) \cap \Gamma_D(v_j) = \varnothing$. Furthermore, for each $i : |\Gamma_D(v_i)| \geq (d-1)$. The result follows. □

Lemma 17. *Let M^* denote a maximum matching in $G = (V, E)$. Consider the state of F after the first pass. Then:*

$$|F| \geq (b-1)|V(M^*) \setminus V(F)|. \tag{7}$$

Proof. By induction it is easy to see that F is a forest with maximal degree b. We argue that F has at least $|V(M^*) \setminus V(F)|$ nodes of degree b. The result then follows by applying Lemma 16. Let $u \in V(M^*) \setminus V(F)$ and denote by v the mate of u in M^*. Since uv is not taken, the degree of v was already b upon arrival of uv. Hence, for each node $u \in V(M^*) \setminus V(F)$ the partner $M^*(u)$ has degree b in F. □

Lemma 18. *Let $|M| = (\frac{1}{2} + \epsilon)|M^*|$. Consider the state of F after the first pass. Then*

$$|V(F) \setminus V(M)| \geq (1 - 2\epsilon - \frac{2}{b})|M^*|.$$

Proof. By Lemma 17, $|F| \geq (b-1)|V(M^*) \setminus V(F)|$. Then

$$\begin{aligned} |V(F) \setminus V(M)| &\geq |V(F)| - |V(M)| \geq (b-1)|V(M^*) \setminus V(F)| - 2|M| \\ &= (b-1)|V(M^*) \setminus V(F)| - (1+2\epsilon)|M^*|. \end{aligned} \tag{8}$$

Furthermore,we also have $|V(F)| \geq 2|M^*| - |V(M^*) \setminus V(F)|$, and hence

$$\begin{aligned} |V(F) \setminus V(M)| &\geq |V(F)| - |V(M)| \geq 2|M^*| - |V(M^*) \setminus V(F)| - 2|M| \\ &= 2|M^*| - |V(M^*) \setminus V(F)| - (1+2\epsilon)|M^*| \\ &= (1-2\epsilon)|M^*| - |V(M^*) \setminus V(F)|. \end{aligned} \tag{9}$$

Then $|V(M^*) \setminus V(F)| = \frac{2|M^*|}{b}$ minimizes $\max\{8, 9\}$ and we obtain $|V(F) \setminus V(M)| \geq \frac{2|M^*|}{b}$. □

Lemma 19. *Consider the state of the variables of the algorithm before the second pass. Let $M'_a \subseteq M'$ such that $\forall m \in M'_a$ there is an edge $m_R \in M_R$ and an edge $m_L \in E$ such that m_R, m, m_L forms a 3-augmenting path. Then:*

$$|M'_a| \geq \frac{1}{b}\left(|V(M_R) \setminus V(M')| - |M'|\right) - 4(\epsilon + \frac{1}{2}\alpha + \alpha\epsilon)|M^*|.$$

Proof. The set M'_a is precisely the subset of edges uv of M' that fulfill the following two conditions.

1. uv is 3-augmentable, and
2. uv has an edge of M_R incident that is not a *blocking* edge.

We say that an edge $m_R = u'v \in M_R$ is a blocking edge, if uv is the incident edge of M', uu', vv' are the edges incident to uv in $M' \oplus M^*$, and the edge $u'v$ is not in the graph G. See Figure 11 for an illustration. Note that there are at most $|M'|$ blocking edges in the graph.

We consider the vertices that are matched in M_R but are free in M'. Each vertex $v \in V(M_R) \setminus V(M')$ is connected by an edge of M_R to an edges of M'. We remove from $V(M_R) \setminus V(M')$ these vertices that have a blocking edge incident. There are at most $|M'|$ blocking edges. Since the maximal degree in M_R is b, there are at least $1/b(|V(M_R) \setminus V(M')| - |M'|)$ edges in M' that fulfill condition (2). By Lemma 1, there are at most $4(\epsilon + \frac{1}{2}\alpha + \alpha\epsilon)|M^*|$ edges in M' that are not 3-augmentable, and the result follows. □

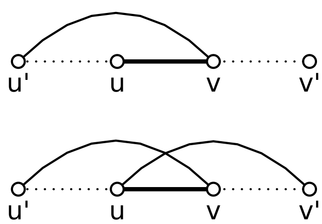

Fig. 11. Illustration of a blocking edge. In the first setting, the edge $u'v$ is a blocking edge, since the edge uv' is not in the graph. The edge $u'v$ blocks edge uu' from augmenting uv. In the second setting, neither $u'v$ nor uv' are blocking edges. $u'v$ blocks the edge vv', however, the edge $u'v$ is an alternative for the node v for being augmented. This alternative is not present in the first figure.

Theorem 6. *Algorithm 10 with $b = 8$ is a deterministic 2-pass semi-streaming algorithm for* Maximum Matching *with approximation ratio $1/2 + 1/140 \approx 1/2 + 0.007142$ for any graph and any arrival order.*

Proof. By construction, the computed matching M'' is of size $|M'| + |Aug|$. Since $|M'| = (1+\alpha)|M|$ and $|M| = (\frac{1}{2} + \epsilon)|M^*|$, we obtain

$$|M''| = (1+\alpha)(\frac{1}{2} + \epsilon)|M^*| + |Aug|. \tag{10}$$

It remains to lower bound $|Aug|$.

In Lemma 19, we show that there is a subset $M'_a \subseteq M'$ such that

$$|M'_a| \geq \frac{1}{b}(|V(M_R) \setminus V(M')| - |M'|) - 4(\epsilon + \frac{1}{2}\alpha + \alpha\epsilon)|M^*|,$$

and for each edge of M'_a there is a 3-augmenting path with an edge from M_R and another edge from the stream. Any 3-augmenting path that is added in Line 9 of Algorithm 10 to Aug may block at most 2 further edges of M'_a from being augmented, see Figure 12. We will find hence at least $\frac{1}{3}|M'_a|$ 3-augmenting paths, and we obtain

$$|Aug| \geq 1/3|M'_a| \geq \frac{1}{3}\left(\frac{1}{b}(|V(M_R) \setminus V(M')| - |M'|) - 4(\epsilon + \frac{1}{2}\alpha + \alpha\epsilon)|M^*|\right). \tag{11}$$

Note that by construction, $|V(M_R) \setminus V(M')| = |V(F) \setminus V(M')|$. We bound now $|V(F) \setminus V(M')|$. By Lemma 18, $|V(F) \setminus V(M)| \geq (1 - 2\epsilon - \frac{2}{b})|M^*|$. Note that M' is the matching that is obtained by augmenting

M with edges from F. Each augmented edge of M has two edges incident from F that are used for the augmentation. Hence,

$$|V(F) \setminus V(M')| \geq (1 - 2\epsilon - \frac{2}{b})|M^*| - 2\alpha|M|. \tag{12}$$

Using Inequality 12 and Inequality 11 in Inequality 10, we obtain

$$|M''| \geq \left(\frac{1}{2} + \frac{1}{6b} - \frac{1}{3}(\alpha\epsilon + \epsilon + \frac{\alpha}{2}) - \frac{1}{b}(\alpha\epsilon + \epsilon + \frac{\alpha}{2} + \frac{2}{3b})\right)|M^*|. \tag{13}$$

Note that we also have

$$|M''| \geq |M'| \geq |M^*|(\frac{1}{2} + \epsilon + \frac{\alpha}{2} + \alpha\epsilon). \tag{14}$$

We determine ϵ_0 as a function of α and b that minimizes the maximum of the right sides of Inequality 13 and Inequality 14. For any α and ϵ_0, M'' is maximized by setting $b = 8$. This leads to an approximation factor $1/2 + 1/140 \approx 1/2 + 0.007142$. □

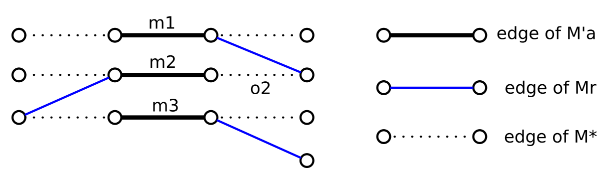

Fig. 12. m_1, m_2, m_3 have each an edge of M_R incident and can be augmented with this edge and an incident edge from M^*. If m_2 is augmented with its incident edge from M_R and o_2, then this may prevent m_1 and m_3 from being augmented.

References

1. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. J. of Computer and System Sciences **58**(1) (1999) 137–147
2. Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., Zhang, J.: Graph distances in the streaming model: the value of space. In: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms. SODA '05, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2005) 745–754
3. Muthukrishnan, S.: Data streams: Algorithms and applications. In: Foundations and Trends in Theoretical Computer Science. Now Publishers Inc (2005)
4. Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., Zhang, J.: On graph problems in a semi-streaming model. Theoretical Computer Science **348**(2-3) (2005) 207–216
5. Ahn, K.J., Guha, S.: Linear programming in the semi-streaming model with application to the maximum matching problem. Inf. Comput. **222** (2013) 59–79
6. Eggert, S., Kliemann, L., Munstermann, P., Srivastav, A.: Bipartite matching in the semi-streaming model. Algorithmica **63**(1-2) (2012) 490–508
7. McGregor, A.: Finding graph matchings in data streams. In: Proceedings of 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems. (2005) 170–181
8. Zelke, M.: Weighted matching in the semi-streaming model. Algorithmica **62**(1-2) (2012) 1–20
9. Epstein, L., Levin, A., Mestre, J., Segev, D.: Improved approximation guarantees for weighted matching in the semi-streaming model. SIAM J. on Discrete Mathematics **25**(3) (2011) 1251–1265
10. Kapralov, M.: Better bounds for matchings in the streaming model. In: In: Proc. of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013 (2013). (2005)

11. Dyer, M., Frieze, A.: Randomized greedy matching. Random Structures & Algorithms **2**(1) (1991) 29–46
12. Munro, J., Paterson, M.: Selection and sorting with limited storage. Theoretical Computer Science **12** (1980) 211–219
13. Guha, S., McGregor, A.: Stream order and order statistics: Quantile estimation in random-order streams. SIAM J. of Computing **38**(1) (2009) 2044–2059
14. Kapralov, M., Khanna, S., Sudan, M.: Approximating matching size from random streams. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '14 (2014) to appear.
15. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. STOC '90, New York, NY, USA, ACM (1990) 352–358
16. Karande, C., Mehta, A., Tripathi, P.: Online bipartite matching with unknown distributions. In: Proceedings of the 43rd annual ACM symposium on Theory of computing. STOC '11, New York, NY, USA, ACM (2011) 587–596
17. Mahdian, M., Yan, Q.: Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In: Proceedings of the 43rd annual ACM symposium on Theory of computing. STOC '11, New York, NY, USA, ACM (2011) 597–606
18. Goel, A., Kapralov, M., Khanna, S.: On the communication and streaming complexity of maximum bipartite matching. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '12, SIAM (2012) 468–485
19. Konrad, C., Rosén, A.: Approximating semi-matchings in streaming and in two-party communication. In: Proceedings of the 40th international conference on Automata, Languages, and Programming - Volume Part I. ICALP'13, Berlin, Heidelberg, Springer-Verlag (2013) 637–649
20. Konrad, C., Magniez, F., Mathieu, C.: Maximum matching in semi-streaming with few passes. In: Proceedings of 15th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems. (2012) 231–242
21. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press (2005)