A Simple Augmentation Method for Matchings with Applications to Streaming Algorithms

Christian Konrad
Department of Computer Science, University of Bristol
Merchant Venturers Building, Woodland Road, BS8 1UB, United Kingdom
christian.konrad@bristol.ac.uk

Abstract

Given a graph G, it is well known that any maximal matching M in G is at least half the size of a maximum matching M^*. In this paper, we show that if G is bipartite, then running the Greedy matching algorithm on a sampled subgraph of G produces enough additional edges that can be used to augment M such that the resulting matching is of size at least $(2-\sqrt{2})|M^*| \approx 0.5857|M^*|$ (ignoring lower order terms) with high probability.

The main applications of our method lie in the area of data streaming algorithms, where an algorithm performs few passes over the edges of an n-vertex graph while maintaining a memory of size $O(n \operatorname{polylog} n)$. Our method immediately yields a very simple two-pass algorithm for Maximum Bipartite Matching (MBM) with approximation factor 0.5857, which only runs the Greedy matching algorithm in each pass. This slightly improves on the much more involved 0.583-approximation algorithm of Esfandiari et al. [ICDMW 2016]. To obtain our main result, we combine our method with a residual sparsity property of the random order Greedy algorithm and give a one-pass random order streaming algorithm for MBM with approximation factor 0.5395. This substantially improves upon the one-pass random order 0.505-approximation algorithm of Konrad et al. [APPROX 2012].

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms, Theory of computation → Graph algorithms analysis

Keywords and phrases Matchings, augmenting paths, streaming algorithms, random order

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.74

1 Introduction

Computing Large Matchings. Given a bipartite graph $G = (A, B, E)$, a matching $M \subseteq E$ in G is a subset of non-adjacent edges. In this paper, we address the Maximum Bipartite Matching (MBM) problem, which consists of finding a matching of maximum size. Many classic algorithms for MBM, such as the Hopcroft-Karp algorithm [20] or Edmonds' algorithm [11], as well as many more recent algorithms, first compute an arbitrary matching and then iteratively improve it by finding augmenting paths until it is of maximum size. A good starting point is a *maximal matching*, i.e., a matching that cannot be enlarged by adding an edge outside the matching to it, which is known to be of size at least $1/2$ times the size of a *maximum matching*, i.e., one of maximum size. A maximal matching is for example produced by the Greedy matching algorithm, which processes the edges of a graph in arbitrary order and adds the current edge to an initially empty matching if the resulting set is still a matching. For an integer $k \geq 1$, a $(2k+1)$-*augmenting path* $P = e_1, e_2, e_3, \dots, e_{2k+1}$ with respect to a matching M is a path of odd length that alternates between edges outside M and edges contained in M such that both e_1 and e_{2k+1} are incident to vertices that are not matched in M. Since P contains $k+1$ edges outside M and k edges of M, removing the matched edges

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 74; pp. 74:1–74:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

in P from M and inserting the unmatched edges in P into M thus increases the size of M by 1. It is known that a non-maximum matching always admits an augmenting path, and, thus, repeatedly finding one and augmenting eventually yields a maximum matching.

To decrease the number of improvement steps required, one common approach is to compute a large *set of disjoint augmenting paths* and augment along each of them simultaneously. This approach is particularly beneficial when designing algorithms in restricted computational models such as the data streaming model (see below) or various distributed computational models, since typically the number of passes (streaming) or rounds (distributed algorithms) grows linearly with the number of augmentation rounds.

Our Results. In this paper, we give a new method that allows us to find a large fraction of disjoint 3-augmenting paths such that, when augmenting along those paths, the resulting matching is of size at least $(2-\sqrt{2})|M^*| - o(|M^*|) \approx 0.5857|M^*| - o(|M^*|)$ with high probability, where M^* is a maximum matching (**Theorem 8**). The strength of our method lies both in its simplicity and effectiveness: It only requires running the Greedy matching algorithm on a random subgraph to produce the necessary edges. Despite its simplicity, it outperforms other more complicated methods and yields improvements over the state-of-the-art one- and two-pass data streaming algorithms for matchings (see below). Our method can also be applied repeatedly and for example yields a 3-pass streaming algorithm that also outperforms the currently best 3-pass streaming algorithm known.

Applications to Data Streaming Algorithms. While our method can be applied in essentially all computational models that allow an implementation of the Greedy matching algorithm, it has been designed with the *data streaming model* in mind. Given an n-vertex graph $G = (V, E)$, a p-pass, s-space data streaming algorithm processing G performs p passes over the edges E of G (the edges may arrive in arbitrary, potentially adversarial order) while maintaining a memory of size s. Since many graph problems require space $\Omega(n \log n)$ (observe that storing a large matching already requires this amount of space) [32], research has focussed on the *semi-streaming model* [16], where a graph streaming algorithm is allowed to use space $O(n \operatorname{polylog} n)$. Concerning the MBM problem, Feigenbaum et al. [16] observed that the Greedy matching algorithm constitutes a one-pass $\frac{1}{2}$-approximation semi-streaming algorithm for MBM. Interestingly, despite intense research efforts, no better one-pass streaming algorithms are known, even if space $O(n^{2-\delta})$ is granted, for any $\delta > 0$, while lower bounds only rule out the existence of semi-streaming algorithms with approximation ratio larger than $1 - 1/e \approx 0.6321$ [22, 18]. Konrad et al. [26] studied minimal extensions to the one-pass semi-streaming model that allow us to improve on Greedy. They showed that approximation ratios strictly larger than $\frac{1}{2}$ can be obtained if either the edges of the input graph arrive in uniform random order, or a second pass is granted. More specifically, they gave a symbolic improvement showing that a $(\frac{1}{2} + 0.005)$-approximation can be obtained if edges arrive in random order, and a $(\frac{1}{2} + 0.02)$-approximation can be achieved if two passes are allowed. Their two-pass result has since been improved by Kale and Tirodkar [21] to $\frac{1}{2} + \frac{1}{16} = \frac{1}{2} + 0.0625$ and independently by Esfandiari et al. to $\frac{1}{2} + 0.083$ [14].

Our method for finding augmenting paths immediately yields a two-pass semi-streaming algorithm with approximation factor 0.5857 (**Theorem 9**), thus slightly improving over the algorithm of Esfandiari et al. [14]. Our algorithm has constant update time (i.e., the running time between two read operations from the stream) and does not need a post-processing step, while the algorithm of Esfandiari et al. requires the computation of a maximum matching in the post-processing step. Our main result is a one-pass random order semi-streaming algorithm with approximation factor 0.5395 (**Theorem 16**), showing that more substantial improvements over $\frac{1}{2}$ than the symbolic improvement given by Konrad et al. [26] are possible

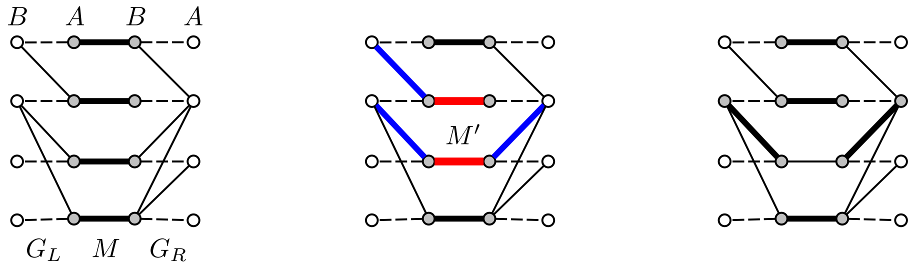

Figure 1 Left: Bipartite graph $G = (A, B, E)$ with maximal matching M. The dotted edges show a perfect matching in G. Matched vertices are grey, free vertices are white. Center: Subset $M' \subseteq M$ is highlighted in red. The blue edges are produced by the runs of GREEDY on G'_L and G'_R. Observe that one 3-augmenting path is found. Right: M after the augmentation.

in the random order scenario. This algorithm is obtained by combining our method for finding augmenting paths with a residual sparsity property of the random order GREEDY matching algorithm (e.g. [25]) that has recently been exploited in various contexts [25, 1, 17].

Techniques. For illustration purposes, consider a bipartite graph $G = (A, B, E)$ that contains a *perfect matching* M^*, i.e., a matching that matches all vertices, and a maximal matching M with $|M| = \frac{1}{2}|M^*|$. It can be seen that $M \oplus M^* := (M \setminus M^*) \cup (M^* \setminus M)$ forms a set of $\frac{1}{2}|M^*|$ disjoint 3-augmenting paths. In other words, there exists a matching of size $\frac{1}{2}|M^*|$ in graph $G_L := G[A(M) \cup \overline{B(M)}]$, where $A(M)$ is the set of matched A-vertices, and $\overline{B(M)} := B \setminus B(M)$, and also one of size $\frac{1}{2}|M^*|$ in $G_R = G[\overline{A(M)} \cup B(M)]$, see Figure 1.

We now sample a random subset of edges $M' \subseteq M$ such that every edge $e \in M$ is included in M' with probability p. Using an argument by Konrad et al. [26], it follows that when running the GREEDY matching algorithm on the subgraph $G'_L := G[A(M') \cup \overline{B(M)}] \subseteq G_L$, then in expectation a $\frac{1}{1+p}$ fraction of the vertices $A(M')$ is matched. Observe that if we chose $p = 1$, then half of the vertices get matched, which is what we expect from the GREEDY matching algorithm. However, if we chose p substantially smaller than 1, then a large fraction of vertices of $A(M')$ is matched. We also apply this argument to subgraph $G'_R := G[B(M') \cup \overline{A(M)}] \subseteq G_R$, which thus allows us to find large matchings in both subgraphs G'_L and G'_R and in turn extract many 3-augmenting paths. Observe that this method directly yields a two-pass semi-streaming algorithm, by computing a maximal matching in the first pass, and augmenting it using the described method in the second pass.

The main shortcoming of this method is that the result by Konrad et al. [26] only holds in expectation, which would imply that our result also only holds in expectation. We therefore strengthen their result and prove that a similar version holds with high probability. Our proof models the execution of the algorithm with a Doob martingale and applies Azuma's inequality to obtain a concentration result. We then use our result and additional combinatorial arguments to bound the number of 3-augmenting paths found.

Our one-pass random order streaming algorithm combines our method for finding 3-augmenting paths with a *residual sparsity* property of the random order GREEDY algorithm. We run GREEDY on the first $\frac{1}{\log n}$ fraction of edges in the stream to produce a matching M. The residual sparsity property states that the residual graph $H = G[V \setminus V(M)]$ contains $O(n \operatorname{polylog} n)$ edges with high probability, which we then collect while processing the remaining edges in the stream. Our main argument is as follows: If $|M|$ is relatively small, then the residual graph H contains a sufficiently large matching. On the other hand, if $|M|$ is relatively large (close to a $\frac{1}{2}$-approximation), then we can use the remainder of the stream to find 3-augmenting paths using the method described above.

Comparison to Esfandiari et al. [14] and Kale and Tirodkar [21]. The two-pass streaming algorithms of Esfandiari et al. and Kale and Tirodkar proceed similarly in that they compute a maximal matching M in the first pass and then find additional edges in the second pass that are used to augment M. Their algorithms are in fact almost identical and only differ in the post-processing stage. With $G_L = G[A(M) \cup \overline{B(M)}]$ and $G_R = G[B(M) \cup \overline{A(M)}]$ being as above, they compute *incomplete semi-matchings* S_L in G_L and S_R in G_R, i.e., subsets of edges such that every vertex in $A(M)$ ($B(M)$) is matched at most once in S_L (resp. S_R) and every vertex $\overline{B(M)}$ (resp. $\overline{A(M)}$) is matched at most k times, for some integer k. Using a Greedy algorithm for computing S_L and S_R, it can be seen that a large fraction of vertices $A(M)$ (resp. $B(M)$) are matched in S_L (resp. S_R). This allows the extraction of multiple 3-augmenting paths. In Kale and Tirodkar, the extraction step is done greedily, which is efficient but leads to a worse approximation factor than in Esfandiari et al. Esfandiari et al. solve an optimization problem in a post-processing phase that allows the extraction of more 3-augmenting paths, which in turn leads to an improved approximation guarantee. Our method is much simpler in this regard, since our additional edges form matchings and it is thus straightforward to extract 3-augmenting paths.

Comparison to Konrad et al. [26]. The one-pass random order algorithm by Konrad et al. proceeds as follows: First, run Greedy on roughly the first third of the edges in the input stream and obtain a matching M. Konrad et al. prove that if Greedy on the entire input stream produces a matching that is close to a $\frac{1}{2}$-approximation, then the matching is built early on, i.e., $|M|$ is relatively large. They then use the remaining part of the stream for finding 3-augmenting paths. To this end, they compute a matching in G_L on roughly the next third of the edges, and then use the last third to compute a matching in G_R to complete the 3-augmenting paths. Their method only yields a marginal improvement over $1/2$ and their result only holds in expectation.

Observe that we also argue that the matching M is large, which we achieve by exploiting the residual sparsity property of Greedy. While Konrad et al. have already processed a third of the edges at this stage, we have only processed a $\frac{1}{\log(n)}$ fraction, and there are thus more remaining edges to our disposal for finding 3-augmenting paths. Further, our method produces more 3-augmenting paths than the method proposed by Konrad et al.

Further Related Work. Matching problems are the most studied graph problems in the data streaming model. Besides the already mentioned works, algorithms have been designed for weighted matchings (e.g. [16, 29, 33, 9, 31]), multiple passes (e.g. [29, 12, 2]), insertion/deletion streams (e.g. [10, 6, 24, 7, 4, 30]), sparse graphs (e.g. [13, 8]), and other variants of the matching problem [27]. Regarding random order streams, Kapralov et al. [23] showed that the size of a maximum matching can be estimated within a poly-log factor using poly-log space, and a $(2/3 - \epsilon)$-approximation can be computed using $\tilde{O}(n^{3/2})$ space [3].

Outline. We proceed as follows. We first give notation and definitions in Section 2. We then present our method for finding a large set of disjoint 3-augmenting paths in Section 3. Implementation details when implementing our method in the adversarial order streaming model are then discussed in Section 4. In Section 5, we give our one-pass random order algorithm. Finally, we conclude in Section 6 with open problems.

2 Preliminaries

Notation. Let $G = (A, B, E)$ be a bipartite graph. We generally use n to denote the number of vertices, i.e., $n = |A| + |B|$, and $m = |E|$ to denote the number of edges. For a subset of vertices $U \subseteq A \cup B$ and a subset of edges $F \subseteq E$, we denote the vertex induced subgraph of G by vertices U by $G[U]$, and the edge induced subgraph of G by edges F by $G[F]$. Let M be a matching in G. We denote by $A(M)$ ($B(M)$) the vertices of A (resp. B) that are matched by M, and we write $V(M) = A(M) \cup B(M)$. Similarly, for an edge $e \in E$, we write $A(e)$ to denote its incident A-vertex, $B(e)$ to denote its B vertex, and $V(e) = \{A(e), B(e)\}$. The complement of a subset $A' \subseteq A$ ($B' \subseteq B$) is denoted by $\overline{A'} = A \setminus A'$ (resp. $\overline{B'} = B \setminus B'$).

The *matching number* of a graph G, i.e., the size of a maximum matching in G, is denoted $\mu(G)$. We write $opt(G)$ to denote an arbitrary but fixed maximum matching in G. For two sets X, Y, we write $X \oplus Y := (X \setminus Y) \cup (Y \setminus X)$ to denote their symmetric difference. For a graph G, $\Delta(G)$ denotes the maximum degree.

Concentration Bounds. In this paper, we will use two concentration bounds. The first one is Azuma's inequality for martingales (we refer the reader to [28] for the an introduction to martingales and Azuma's inequality), and the second is a Chernoff-type bound for weakly dependent random variables.

▶ **Theorem 1** (Azuma's Inequality ([5, 28])). *Suppose that $X_0, X_1, X_2, \dots$ is a martingale and let $|X_i - X_{i-1}| \leq c_i$ for suitable constants c_i. Then:*

$$\mathbb{P}\left[|X_n - X_0| \geq t\right] \leq 2\exp\left(\frac{-t^2}{2\sum_{i=1}^{n} c_i^2}\right) .$$

▶ **Theorem 2** (Chernoff Bound for Weakly Dependent Variables, e.g. [15]). *Let $X_1, X_2, \dots, X_n$ be 0/1 random variables for which there is a $p \in [0,1]$ such that for all $k \in [n]$ the inequality*

$$\mathbb{P}\left[X_k = 1 \,|\, X_1, X_2, \dots, X_{k-1}\right] \leq p$$

holds (i.e., the probability of $X_k = 1$ conditioned on any possible outcome of $X_1, \dots, X_{k-1}$ is at most p). Let further $\mu \geq p \cdot n$. Then, for every $\delta > 0$:

$$\mathbb{P}\left[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu\right] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\mu} .$$

We will say that an event occurs with *high probability in variable x*, if the the probability of the event occurring is at least $1 - x^{-C}$, for some $C \geq 1$. If we do not mention x explicitly, then the high probability statement is in n, the number of vertices of the input graph.

We say that an algorithm is a C-approximation algorithm for MBM if it computes a matching M of size at least $C \cdot \mu(G) - o(\mu(G))$.

3 Finding a Large Set of Disjoint 3-augmenting Paths

We now present an algorithm that, given a maximal matching M in a bipartite graph $G = (A, B, E)$, finds a set of disjoint 3-augmenting paths $\mathcal{P}$ by running the Greedy matching algorithm on a random subgraph of G. The set $\mathcal{P}$ is such that, when augmenting M along the paths $\mathcal{P}$, a matching of size at least $(2-\sqrt{2})\mu(G) - o(\mu(G)) \approx 0.5857\mu(G) - o(\mu(G))$ is obtained.

Our algorithm is illustrated in Algorithm 1. For the sake of a clear presentation, the algorithm employs two invocations of Greedy on two disjoint subgraphs. This is equivalent to invoking Greedy only once on their union. Our algorithm is parametrized by a sampling probability p. To obtain the claimed bound stated above, we will later optimize p.

To obtain a better understanding of our algorithm, we first discuss structural properties that help us locate 3-augmenting paths in G with respect to the matching M.

Input: Bipartite graph $G = (A, B, E)$, maximal matching M, parameter $0 < p < 1$

1. Sample each edge $e \in M$ with probability p; let M' be the resulting sample
2. $M_L \leftarrow \textsc{Greedy}(G[A(M') \cup \overline{B(M)}])$; $M_R \leftarrow \textsc{Greedy}(G[\overline{A(M)} \cup B(M')])$
3. $\mathcal{P} \leftarrow \{\text{paths } b'a, ab, ba' \mid b'a \in M_L, ab \in M, ba' \in M_R\}$
4. **return** $\mathcal{P}$

Algorithm 1. Finding a large set of 3-augmenting paths

Let M^* be a maximum matching in G and let ϵ be such that $|M| = (\frac{1}{2} + \epsilon)|M^*|$. Observe first that $M \oplus M^*$ contains a collection of $(\frac{1}{2} - \epsilon)|M^*|$ disjoint augmenting paths. Further, observe that the endpoints of each augmenting path are a free vertex in A (i.e., a vertex in $\overline{A(M)}$) and a free vertex in B. Hence, the subgraphs $G_L := G[A(M) \cup \overline{B(M)}]$ and $G_R := G[\overline{A(M)} \cup B(M)]$ each contain a matching of size $(\frac{1}{2} - \epsilon)|M^*|$. We summarize this in Observation 3:

▶ **Observation 3.** *Let ϵ be such that $|M| = (\frac{1}{2} + \epsilon)\mu(G)$. Then:*

$$\min\{\mu(G_L), \mu(G_R)\} \geq (\frac{1}{2} - \epsilon)\mu(G) \ .$$

Suppose now that ϵ is small. Further, suppose that we could compute maximum matchings M_L^* in G_L and M_R^* in G_R. Then for almost every edge $e \in M$ there are edges $e_l \in M_L$ and $e_r \in M_R$ such that $e_l e e_r$ forms a 3-augmenting path. We will call e_l a left wing for edge e and e_r a right wing for edge e.

Our augmentation method should of course not be based on computing maximum matchings themselves. We therefore proceed differently. First, observe that if we computed maximal matchings, i.e., $\frac{1}{2}$-approximations, in G_L and G_R, then we may not find any 3-augmenting path at all, since it may happen that we find left wings for half of the edges of M, and right wings for the other half. Our strategy therefore is as follows: We first sample a subset of edges $M' \subseteq M$, where each edge of M is included in M' with probability p, and we attempt to augment only the edges in M' by computing $\textsc{Greedy}$ matchings in the subgraphs $G'_L := G[A(M') \cup \overline{B(M)}]$ and $G'_R := G[\overline{A(M)} \cup B(M')]$. Konrad et al. [26] proved that, in expectation, the resulting matchings are essentially $\frac{1}{1+p} \geq \frac{1}{2}$-approximations, albeit for a slightly different notion of approximation, which is nevertheless suitable for our purposes:

▶ **Theorem 4** (Konrad et al. [26]). *Let $G = (U, V, E)$ be a bipartite graph, and let $U' \subseteq U$ be such that every vertex $u \in U$ is included in U' with probability p ($p \in [0, 1]$). Then, for any arbitrary but fixed order in which* $\textsc{Greedy}$ *processes the edges, the following holds:*

$$\mathbb{E}_{U'}|\textsc{Greedy}(G[U' \cup V])| \geq \frac{p}{1+p}\mu(G) \ .$$

Hence, if ϵ is close to 0, and p is substantially smaller than 1, then it follows from the previous theorem that a large fraction of the vertices $A(M')$ will be matched by $\textsc{Greedy}$ in G'_L, and a large fraction of the vertices of $B(M')$ will be matched by $\textsc{Greedy}$ in G'_R. This in turn implies that a substantial amount of edges of M' both have left and a right wings and are thus included in 3-augmenting paths.

Before we make this intuition formal, we point out one shortcoming of applying the previous theorem by Konrad et al. directly. They prove that the resulting matching is large only *in expectation*, which in turn would imply that our result only holds in expectation.

We therefore first strengthen their result and prove that a similar version holds with high probability. To this end, we first prove a technical lemma that is employed in the proof of our strengthened theorem.

▶ **Lemma 5.** *Let $G = (U, V, E)$ be a bipartite graph and let $u \in U, v \in V$ be arbitrary vertices. Let $U' \subseteq U$ be such that every vertex $u \in U$ is included in U' with probability p. Then, for any arbitrary but fixed order in which* Greedy *processes the edges, the following holds:*

$$0 \le \mathbb{E}_{U'}|\textsc{Greedy}(G[U' \cup V])| - \mathbb{E}_{U'}|\textsc{Greedy}(G[(U' \cup V) \setminus \{u, v\}])| \le 2 \ .$$

Proof. First, observe that

$$\mathbb{E}_{U'}|\textsc{Greedy}(G[U' \cup V])| - \mathbb{E}_{U'}|\textsc{Greedy}(G[(U' \cup V) \setminus \{u, v\}])| =$$
$$\mathbb{E}_{U'}\left(|\textsc{Greedy}(G[U' \cup V])| - |\textsc{Greedy}(G[(U' \cup V) \setminus \{u, v\}])|\right) \ .$$

We will prove next that $0 \le \textsc{Greedy}(G[U' \cup V]) - \textsc{Greedy}(G[U' \cup V - \{u, v\}]) \le 2$ holds for any $U' \subseteq U$, which then proves the lemma. We will in fact argue the stronger statement that for any graph $G = (V, E)$ and any vertex $v \in V$, the inequality $0 \le \textsc{Greedy}(G) - \textsc{Greedy}(G \setminus \{v\}) \le 1$ holds. The result then follows by applying this statement twice.

Consider thus an arbitrary graph $G = (V, E)$ and a vertex $v \in V$. First observe that if $\textsc{Greedy}(G)$ leaves v unmatched, then $\textsc{Greedy}(G) = \textsc{Greedy}(G \setminus \{v\})$. If $\textsc{Greedy}(G)$ matches v, then it is not hard to see that $\textsc{Greedy}(G) \oplus \textsc{Greedy}(G \setminus \{v\})$ consists of one alternating path whose one endpoint is v. This further implies that $\textsc{Greedy}(G \setminus \{v\}) \le \textsc{Greedy}(G) \le \textsc{Greedy}(G \setminus \{v\}) + 1$, which completes the proof. ◀

We now give our strengthened version of Theorem 4.

▶ **Theorem 6.** *Let $G = (U, V, E)$ be a bipartite graph, and let $U' \subseteq U$ be such that every vertex $u \in U$ is included in U' with probability p ($p \in [0, 1]$). Then, for any arbitrary but fixed order in which* Greedy *processes the edges, the following holds with probability at least $1 - (\mu(G))^{-12}$:*

$$|\textsc{Greedy}(G[U' \cup V])| \ge \frac{p}{1+p}\mu(G) - o(\mu(G)) \ .$$

Proof. Let $X := |\textsc{Greedy}(G[U' \cup V])|$. By Theorem 4 we have $\mathbb{E}X \ge \frac{p}{1+p}\mu(G)$.

For $1 \le i \le n$, let Z_i be the ith edge selected by Greedy, and let $Z_i = \perp$ if $i > X$. Let Y_i be the Doob martingale induced by the first i choices of the algorithm, i.e.,

$$Y_i := \mathbb{E}_{Z_{i+1}, Z_{i+2}, \dots, Z_n}(X \mid Z_1, \dots, Z_i) \ .$$

Observe that the expectation in the previous expression is in itself a random variable, since the expectation is only taken over $Z_{i+1}, \dots, Z_n$, while $Z_1, \dots, Z_i$ are random variables. It is not hard to check that the sequence $(Y_i)_i$ always forms a martingale, independently of the underlying sequence Z_i. Observe next that $Y_0 = \mathbb{E}X$ and $Y_n = X$. We thus need to show that $|Y_n - Y_0|$ is small with high probability. To this end, we will apply Azuma's inequality, which requires bounding the differences $|Y_{i+1} - Y_i|$, for every i, first.

First, observe that $|Y_{i+1} - Y_i| = 0$ for every $i \ge X$. Next, we claim that $|Y_{i+1} - Y_i| \le 1$, for every $i < X$. Indeed, observe that Y_i is the expected size of the computed matching conditioned on the first i choices of the algorithm. We can thus rewrite Y_i as:

$$Y_i = i + \mathbb{E}_{U'}|\textsc{Greedy}(H_i)| \ ,$$

where $H_i := G[(U' \cup V) \setminus \cup_{j \le i} V(Z_j)]$ is the residual graph obtained when removing the vertices incident to the first i selected edges. We thus obtain:

$$\begin{aligned} Y_{i+1} - Y_i &= 1 + \mathbb{E}_{U'}|\text{GREEDY}(H_{i+1})| - \mathbb{E}_{U'}|\text{GREEDY}(H_i)| \\ &= 1 + \mathbb{E}_{U'}|\text{GREEDY}(H_i \setminus V(Z_{i+1}))| - \mathbb{E}_{U'}|\text{GREEDY}(H_i)| \in \{-1, 0, 1\} , \end{aligned}$$

where we applied Lemma 5.

Next, since $X \le \mu(G[U' \cup V]) \le \mu(G)$, we have $|Y_{i+1} - Y_i| \le 1$ for every $i \le \mu(G)$, and $|Y_{i+1} - Y_i| = 0$ for every $i > \mu(G)$. Applying Azuma's Inequality (Theorem 1), we obtain:

$$\mathbb{P}\left[|Y_n - Y_0| \ge 5\sqrt{\mu(G)\ln(\mu(G))}\right] \le \mu(G)^{-12} .$$

◀

Equipped with Theorem 6, we now show that our algorithm finds many disjoint 3-augmenting paths, provided that M is close to a $\frac{1}{2}$-approximation.

▶ **Lemma 7.** *Consider Algorithm 1 and suppose that $|M| = (\frac{1}{2} + \epsilon)\mu(G)$. Then, with probability at least $1 - \mu(G)^{-10}$,*

$$|\mathcal{P}| \ge \mu(G)p\left(\frac{1-2\epsilon}{1+p} - \frac{1}{2} - \epsilon\right) - o(\mu(G)) .$$

Proof. First, by an application of a Chernoff bound, we obtain $|M'| = p|M| \pm O(\sqrt{|M|\ln(|M|)})$, with probability at least $1 - |M|^{-C}$, for an arbitrarily large constant C. Next, by Theorem 6 and Observation 3, with probability at least $1 - 2(\mu(G))^{-12}$, we have $|M_L| \ge \frac{p}{1+p}(\frac{1}{2} - \epsilon)\mu(G) - o(\mu(G))$ and $|M_R| \ge \frac{p}{1+p}(\frac{1}{2} - \epsilon)\mu(G) - o(\mu(G))$. Observe that at most $|M'| - |M_L|$ edges of M' do not have a left wing, and at most $|M'| - |M_R|$ edges of M' do not have a right wing. Hence, at least $|M'| - (|M'| - |M_L|) - (|M'| - |M_R|) = |M_L| + |M_R| - |M'|$ edges have both left and right wings and therefore form 3 augmenting paths. We thus obtain:

$$\begin{aligned} |\mathcal{P}| &\ge |M_L| + |M_R| - |M'| \ge 2 \cdot \frac{p}{1+p}(\frac{1}{2} - \epsilon)\mu(G) - o(\mu(G)) - p|M| - O(\sqrt{|M|\ln(|M|)}) \\ &\ge 2 \cdot \frac{p}{1+p}(\frac{1}{2} - \epsilon)\mu(G) - p(\frac{1}{2} + \epsilon)\mu(G) - o(\mu(G)) \\ &= \mu(G)p\left(\frac{1-2\epsilon}{1+p} - \frac{1}{2} - \epsilon\right) - o(\mu(G)) . \end{aligned}$$

By the union bound, the error is bounded by $|M|^{-C} + 2(\mu(G))^{-12} \le (\mu(G))^{-10}$. ◀

We are now ready to prove our main theorem:

▶ **Theorem 8.** *Let M be a maximal matching. Then, setting $p = \sqrt{2} - 1$ in Algorithm 1 guarantees that M augmented by $\mathcal{P}$ gives a matching of size at least $(2 - \sqrt{2})\mu(G) - o(\mu(G)) \approx (\frac{1}{2} + 0.0857)\mu(G) - o(\mu(G))$ with high probability in $\mu(G)$.*

Proof. Observe that the final matching is of size $|M| + |\mathcal{P}|$. Let ϵ be such that $|M| = (\frac{1}{2} + \epsilon)\mu(G)$. By Lemma 7, we have

$$|M| + |\mathcal{P}| \ \ge \ (\frac{1}{2} + \epsilon)\mu(G) + \mu(G)p\left(\frac{1-2\epsilon}{1+p} - \frac{1}{2} - \epsilon\right) - o(\mu(G)) . \qquad (1)$$

It can be seen that for any value of p, the right side of Inequality 1 is minimized for $\epsilon = 0$. On the other hand, for any value of ϵ, the value $p(\epsilon) = \sqrt{\frac{1-2\epsilon}{\frac{1}{2}+\epsilon}} - 1$ maximizes Inequality 1. Using $\epsilon = 0$ and $p(0) = \sqrt{2} - 1$ in Inequality 1 gives $|M| + |\mathcal{P}| \ge (2 - \sqrt{2})\mu(G) - o(\mu(G))$. ◀

Multiple augmentation rounds with decreasing values of p allow further improvements. For example, a second round with $p = \sqrt{\frac{2-\sqrt{2}}{\sqrt{2}-1}} - 1 \approx 0.1892$ guarantees that the resulting matching is of size at least $0.6067\mu(G) - o(\mu(G))$. As we will discuss in the next section, this can give a 3-pass streaming algorithm for MBM with approximation factor 0.6067, which slightly improves the 3-pass 0.605-approximation algorithm by Esfandiari et al. [14].

4 Adversarial Order Streams

Our method for finding augmenting paths given in Section 3 can directly be implemented in the streaming model. In the first pass, we compute a maximal matching M. If the current edge is added to M, then with probability p we add the edge to M' as well. In the second pass, we run GREEDY on the subgraphs G'_L and G'_R and as soon as a 3-augmenting path is completed, we augment M. This can be done with constant update times.

Since we would like our streaming algorithm to succeed with high probability in n, the number of vertices, we need to address the fact that our method as stated in Theorem 8 only succeeds with high probability in $\mu(G)$, the size of a maximum matching in G. If $\mu(G)$ is of size at least, say, $\Omega(n^{\frac{1}{4}})$, our method can also gives a high probability result with respect to n. To deal with the case $\mu(G) = o(n^{\frac{1}{4}})$ we run the 1-pass algorithm of Chitnis et al. [7] in parallel to our algorithm, which computes a subset of edges $E' \subseteq E$ of size $O(n^{\frac{1}{2}})$ that contains a maximum matching provided that $\mu(G) = O(n^{\frac{1}{4}})$. Observe that after the first pass, we know in which of the two cases we are. We then run the Hopcroft-Karp maximum matching algorithm [20] in time $O(\sqrt{n} \cdot \sqrt{n}) = O(n)$ on the set of collected edges. To obtain a streaming algorithm with constant update time, we amortize the previous computation during the processing of the second pass, which is possible under the natural assumption that $m = \Omega(n)$. This gives the following theorem:

▶ **Theorem 9.** *There is a two-pass streaming algorithm for* MBM *with approximation factor* $2 - \sqrt{2} \approx \frac{1}{2} + 0.0857$ *that succeeds with high probability (in n). Using one additional pass, a* 0.6067*-approximation algorithm can be obtained.*

5 1-pass Random Order Streaming Algorithm

In this section, we assume that $\mu(G) = \Omega(n^{\frac{1}{4}})$. To deal with the case $\mu(G) = o(n^{\frac{1}{4}})$ we run the algorithm of Chitnis et al. [7] as outlined in Section 4 in parallel and compute and output a maximum matching after processing the stream. We also assume that the input graph has at least $C_1 \cdot n \log^{C_2} n$ edges, for suitably large constants C_1, C_2. If this is not the case then we could simply store all edges within the semi-streaming space constraint and compute and output a maximum matching.

Our 1-pass random order streaming algorithm combines our method for finding augmenting paths with a *residual sparsity* property of the random order GREEDY matching algorithm:

▶ **Theorem 10** (Residual Sparsity of GREEDY). *Suppose that* GREEDY *processes the edges E of a graph $G = (V, E)$ with $m = |E|$ in uniform random order. Let M_i be the matching produced by* GREEDY *after having processed the ith edge. Then:*

$$\Delta(G[V \setminus V(M_i)]) = O(\frac{m \log n}{i})$$

with probability $1 - n^{-12}$ (over the uniform random ordering of the edges).

This theorem is implied by a similar theorem concerning the random order Greedy algorithm for independent sets as given in [25]. Observe that the Greedy algorithm for matchings on a randomly ordered sequence of the edges of a graph G can be seen as the Greedy algorithm for independent sets on a randomly ordered sequence of the vertices of the line graph $L(G)$.

Our one-pass random order algorithm is parametrized by a probability p, and is illustrated in Algorithm 2. In this listing, we write $\pi = \pi[1], \pi[2], \ldots, \pi[m]$ to be a uniform random ordering of the edges E. For $a < b$ we also write $\pi[a, b]$ to denote edges $\pi[a], \pi[a+1], \ldots, \pi[b]$, and $\pi(a, b]$ to denote edges $\pi[a+1], \pi[a+2], \ldots, \pi[b]$.

Input: Bipartite graph $G = (A, B, E)$ with m edges, parameter $0 < p < 1$

Let $\pi = \pi[1], \pi[2], \ldots \pi[m]$ be the edges of G in uniform random order

1. $M \leftarrow \text{Greedy}(\pi[1, \frac{m}{\log n}])$
2. Let $M' \subseteq M$ be such that every edge of M is included in M' with probability p
3. **while** processing $\pi(\frac{m}{\log n}, m]$ **do in parallel:**
   - **a.** Compute set E_M of edges $ab \in \pi(\frac{m}{\log n}, m]$ with $a, b \notin V(M)$; if $|E_M| \geq C \cdot n \log^2 n$, for some appropriate large constant C, then **abort**
   - **b.** $M_L \leftarrow \text{Greedy}(G_L^r)$, where G_L^r is the subgraph of G induced by all edges $\pi(\frac{m}{\log n}, m]$ between $A(M')$ and $\overline{B(M)}$
   - **c.** $M_R \leftarrow \text{Greedy}(G_R^r)$, where G_R^r is the subgraph of G induced by all edges $\pi(\frac{m}{\log n}, m]$ between $\overline{A(M)}$ and $B(M')$
4. $\mathcal{P} \leftarrow \{\text{paths } b'a, ab, ba' \mid b'a \in M_L, ab \in M, ba' \in M_R\}$
5. **if** $|\mathcal{P}| \geq \mu(G[E_M])$ **then return** M augmented by $\mathcal{P}$
   else return $M \cup opt(G[E_M])$

Algorithm 2. One-pass random order matching algorithm

We run Greedy on the first $\frac{m}{\log n}$ edges to compute a matching M. Theorem 10 implies that the maximum degree in the residual graph $H := G[V \setminus V(M)]$ is $O(\log^2 n)$. This allows us to collect the entire residual graph (i.e., set E_M) within the semi-streaming space bound, since it has $O(n \log^2 n)$ edges with high probability. We abort if $|E_M|$ becomes too large.

In the next stage, we proceed as in our two-pass algorithm: We sample a subset of edges $M' \subseteq M$ and we try to find 3-augmenting paths for M' by computing matchings M_L and M_R in the subgraphs G_L^r and G_R^r. Ideally we would like to search for left and right wings in the subgraphs $G_L := G[A(M) \cup \overline{B(M)}]$ and $G_R := G[\overline{A(M)} \cup B(M)]$. Since however the first $\frac{1}{\log n}$ fraction of edges in the stream has already been processed, we can only search for augmenting paths in G_L^r and G_R^r. Concentration bounds however allow us to prove that not many important edges have arrived among the first $\frac{1}{\log n}$ fraction of edges (Lemma 14).

Our analysis is build on the following important observation. Suppose first that the matching M is small, i.e., $|M| = \alpha|M^*|$, for a small value of α. Then we will argue in the next lemma that a maximum matching in the residual graph is large:

▶ **Lemma 11.** *Let α be such that $|M| = \alpha|M^*|$, and let $H := G[E_M]$ $(= G[V \setminus V(M)])$ be the residual graph. Then:*

$$\mu(H) \geq (1 - 2\alpha)|M^*| \ .$$

Proof. Let M^* be a maximum matching in G. Let $M_1^* \subseteq M^*$ be those edges of M^* that share at least one endpoint with an edge in M, and let $M_2^* = M^* \setminus M_1^*$. Then $|M_1^*| \leq 2|M|$, since each edge of M can only be incident to at most two edges of M^*. Observe further that $M_2^* \subseteq E_M$. Hence: $\mu(H) \geq |M_2^*| = |M^*| - |M_1^*| \geq |M^*| - 2|M| = (1 - 2\alpha)|M^*|$. ◀

By combining M with a maximum matching in H we obtain the following corollary:

▶ **Corollary 12.** *Algorithm 2 finds a matching of size at least* $(1-\alpha)|M^*|$ *with high probability.*

The previous corollary shows that either the matching $M \cup opt(H)$ is large (if α is small), or the matching M itself is already reasonably large (if α is large). This is an important property since we next attempt to augment M, which necessitates that M is already close to a $\frac{1}{2}$-approximation. For this to succeed, we need to show that $\mu(G_L^r)$ and $\mu(G_R^r)$ are large. To this end, let δ be such that $|M| + \mu(G[E_M]) = (\frac{1}{2} + \delta)\mu(G)$. We will first bound $\mu(G_L)$ and $\mu(G_R)$ and then prove a similar bound for $\mu(G_L^r)$ and $\mu(G_R^r)$.

▶ **Lemma 13.** *Suppose that* $|M| + \mu(G[E_M]) = (\frac{1}{2} + \delta)\mu(G)$*. Then:*

$$\min\{\mu(G_L), \mu(G_R)\} \geq (\frac{1}{2} - \delta)\mu(G) \ .$$

Proof. Let M^* be a maximum matching in G and let M_H^* be an arbitrary maximum matching in $H(= G[E_M])$. First, it is not hard to see that $M \cup M_H^*$ is a maximal matching. Next, consider the set of edges $M^* \oplus (M \cup M_H^*)$. Since $|M| + |M_H^*| = (\frac{1}{2} + \delta)\mu(G)$, the set $M^* \oplus (M \cup opt(H))$ contains $(\frac{1}{2} - \delta)\mu(G)$ augmenting paths.

Observe that none of these augmenting paths only contain edges of M^* and M_H^*, since this would imply that M_H^* is not maximum in H. Consider now one such augmenting path P and remove all edges of M_H^* from P. Then P contains at least one augmenting path that only contains edges from M and M^*. Applying this argument to all augmenting paths, this proves that there are matchings in G_L and G_R of sizes $(\frac{1}{2} - \delta)\mu(G)$. ◀

▶ **Lemma 14.** *Suppose that* $|M| + \mu(G[E_M]) = (\frac{1}{2} + \delta)\mu(G)$*. Then, with high probability,*

$$\min\{\mu(G_L^r), \mu(G_R^r)\} \geq (1 - \frac{4}{\log n}) \cdot (\frac{1}{2} - \delta)\mu(G) \ .$$

Proof. We only give the argument for G_L^r, the argument for G_R^r is identical. Let $M_L^* = opt(G_L)$. We will show that most edges of M_L^* are included in $\pi(\frac{m}{\log n}, m]$ with high probability.

By Lemma 13, we have $|M_L^*| \geq (\frac{1}{2} - \delta)\mu(G)$. Let e_i be the i-th edge of M_L^*, let t_i be its position in the stream, and let Y_i be the indicator variable of the event "$t_i \leq \frac{m}{\log n}$". Our aim is to bound the probabilities $\mathbb{P}[Y_i = 1 \,|\, Y_1, \dots, Y_{i-1}]$ and then apply the Chernoff bound stated in Theorem 2.

In the following, all our arguments are conditioned on the event "$|E(G[V \setminus V(M)])| = O(n \log^2 n)$" (without explicitly mentioning it), which we denote by E_1. This implies that the algorithm does not abort in Line 3a. By the residual sparsity property as stated in Theorem 10, E_1 occurs with probability at least $1 - n^{-12}$.

We will argue now that

$$\mathbb{P}\left[\pi[\frac{m}{\log n} + 1] \cup M \text{ is not a matching} \wedge \pi[\frac{m}{\log n} + 1] \notin M_L^* \,|\, Y_1, \dots, Y_{i-1}\right] \geq 1 - \frac{1}{\log^5 n} \ . \quad (2)$$

Since E_1 happens, observe that the second part of the stream consists of $m(1 - \frac{1}{\log n}) - O(n \log^2 n)$ edges that cannot be added to matching M, at most $n/2$ edges of M_L^* (depending on the outcome of variables $Y_1, \dots, Y_{i-1}$), and at most $O(n \log^2 n)$ edges that could extend M. Further, the arrival order of the edges $\pi(\frac{m}{\log n}, m]$ in the second part of the stream is uniform random, since the computed matching M is not affected by their order. Hence,

$$\mathbb{P}\left[\pi[\frac{m}{\log n}+1]\cup M \text{ is not a matching } \wedge \pi[\frac{m}{\log n}+1]\notin M_L^* \,|\, Y_1,\ldots,Y_{i-1}\right]$$

$$\geq \frac{m(1-\frac{1}{\log n})-O(n\log^2 n)}{m(1-\frac{1}{\log n})} \geq 1-\frac{1}{\log^5 n},$$

using the assumption that the graph has at least $C \cdot n \log^{10} n$ edges, for a large enough C.

The key part of our argument is as follows: Let Π be the set of permutations that fulfill the event in Inequality 2. Given Π, we generate a set of permutations Π' with $\Pi' \supseteq \Pi$, which thus implies that the respective event is more likely to happen than the event in Inequality 2. Let $\pi \in \Pi$ be any permutation. Consider edge e_i and let j_i be such that $\pi[j_i] \in M$ is the edge incident to e_i. Since $e_i \in M_L^*$, we know that $t_i > j_i$. Construct now new permutations such that e_i is removed from its position t_i and is inserted at every position $\{t_i+1, t_i+2, \ldots, m\}$ and add the resulting permutations to Π'. Observe that for any permutation π' created this way, the exact same matching M is computed, which uses the fact that $\pi[\frac{m}{\log n}+1]$ cannot be added to M, which is important if e_i is inserted at a position larger than $\frac{m}{\log n}+1$. Observe further that the conditionings Y_j stay the same, which uses the fact that $\pi[\frac{m}{\log n}+1] \notin M_L^*$. Observe that Π' and Π are not identical, since we do not necessarily have that $\pi'[\frac{m}{\log n}+1]\cup M$ is not a matching for $\pi' \in \Pi'$. By construction, at least a $(1-\frac{1}{\log n})$-fraction of the permutations in Π' imply $Y_i = 0$. We thus obtain:

$$\mathbb{P}\left[Y_i = 0 \,|\, Y_1,\ldots,Y_{i-1}\right] \geq$$

$$(1-\frac{1}{\log n})\cdot \mathbb{P}\left[\pi[\frac{m}{\log n}+1]\cup M \text{ is not a matching} \wedge \pi[\frac{m}{\log n}+1]\notin M_L^* \,|\, Y_1,\ldots,Y_{i-1}\right]$$

$$\geq (1-\frac{1}{\log n})(1-\frac{1}{\log^5 n}) \geq 1-\frac{2}{\log n}.$$

We now use the Chernoff bound for dependent variables stated in Theorem 2. Using $k = (\frac{1}{2}-\delta)\mu(G)$, we obtain (using $\mu = 2k/\log n$, and $\delta = 1$ in Theorem 2):

$$\mathbb{P}\left[\sum_{i=1}^{k} Y_i \geq 2\frac{2k}{\log n}\right] \leq \left(\frac{e}{4}\right)^{\frac{2k}{\log n}} \leq n^{-10},$$

using the assumption $\mu(G) = \Omega(n^{\frac{1}{4}})$. The result follows. ◀

In the remaining analysis, with the help of the previous lemma we bound the number of augmenting paths found in Lemma 15. We then conclude with our main theorem, where we show that one of the two computed matchings returned by the algorithm is necessarily large.

▶ **Lemma 15.** *Let $p = \Omega(1)$, suppose that $|M| + \mu(H) = (\frac{1}{2}+\delta)\mu(G)$, and let $|M| = \alpha\mu(G)$. Then, with high probability,*

$$|\mathcal{P}| \geq p\mu(G)\left(\frac{1-2\delta}{1+p}-\alpha\right) - o(\mu(G)).$$

Proof. We follow the structure of the proof of Lemma 7. By an application of a Chernoff bound, we obtain $|M'| = p|M| \pm O(\sqrt{|M|\ln(|M|)})$, with probability at least $1-|M|^{-C}$, for an arbitrarily large constant C. Next, by Theorem 6 and Lemma 14, with high probability in $\mu(G)$ we have

$$\min\{|M_L|, |M_R|\} \geq \frac{p}{1+p}(\frac{1}{2}-\delta)\mu(G) - o(\mu(G)).$$

Since we assumed that $\mu(G) = \Omega(n^{\frac{1}{4}})$, this event also holds with high probability in n. As argued in the proof of Lemma 7, the quantity $|M_L| + |M_R| - |M'|$ bounds the number of 3-augmenting paths found, which then completes the proof:

$$|\mathcal{P}| \geq |M_L| + |M_R| - |M'| \geq \frac{2p}{1+p}(\frac{1}{2} - \delta)\mu(G) - p|M| - o(\mu(G))$$

$$= p\mu(G)\left(\frac{2}{1+p}(\frac{1}{2} - \delta) - \alpha\right) - o(\mu(G)) = p\mu(G)\left(\frac{1-2\delta}{1+p} - \alpha\right) - o(\mu(G)) .$$

◀

▶ **Theorem 16.** *Setting $p = \sqrt{2} - 1$ in Algorithm 2 gives a one-pass random order semi-streaming algorithm for* MBM *with approximation ratio $\frac{1}{2} + \frac{2\sqrt{2}-3}{4\sqrt{2}-10} \geq 0.5390$ that succeeds with high probability.*

Proof. Suppose that $|M| = \alpha\mu(G)$ and $|M| + \mu(H) = (\frac{1}{2} + \delta)\mu(G)$. By Lemma 11, we have $\mu(H) \geq (1 - 2\alpha)\mu(G)$. Hence, $(1 - \alpha)\mu(G) \leq (\frac{1}{2} + \delta)\mu(G)$, which in turn implies $\alpha \geq \frac{1}{2} - \delta$. Plugging this into the bound given in Lemma 15, we obtain (ignoring the $o(\mu(G))$ term):

$$|M| + |\mathcal{P}| \geq \alpha\mu(G) + p\mu(G)\left(\frac{1-2\delta}{1+p} - \alpha\right) = \mu(G)\left(\alpha(1-p) + p\left(\frac{1-2\delta}{1+p}\right)\right)$$

$$\geq \mu(G)\left((\frac{1}{2} - \delta)(1-p) + p\left(\frac{1-2\delta}{1+p}\right)\right) .$$

The quantity $|M| + \max\{|\mathcal{P}|, \mu(H)\}$, i.e., the size of the resulting matching, is minimizes if $|\mathcal{P}| = \mu(H)$. Hence, setting the right side of the previous inequality equal to $(\frac{1}{2} + \delta)\mu(G)$, we obtain $\delta = \frac{p(p-1)}{2p^2-6p-4}$, which is maximized for $p = \sqrt{2} - 1$ (observe that this is the same value as in the proof of Theorem 8). In this case, we obtain $\delta = \frac{2\sqrt{2}-3}{4\sqrt{2}-10} \approx 0.03950$, which completes the proof. ◀

6 Conclusion

In this paper, we gave a new method for finding a set of disjoint 3-augmenting paths that allows the augmentation of a maximal matching such that the resulting matching is of size at least $\sqrt{2} - 2$ times the size of a maximum matching. Our method is simple and only requires running the Greedy matching algorithm on a random subgraph. We applied this method in the data streaming setting and improved over the state-of-the-art one-pass random order algorithm and the state-of-the-art two- and three-pass adversarial order algorithms.

How large a matching can we compute in a single pass in the random order setting? All relevant known lower bounds for matchings [18, 22, 19] are highly sensitive to the arrival order of the edges and do not translate to the random order setting. Can we compute a 2/3-approximation in a single pass in the random order semi-streaming setting? In the adversarial order setting, it is known how to obtain a $2/3 - \delta$ approximation in $O(\frac{1}{\delta})$ passes. How many passes are required to obtain a 2/3-approximation?

References

1 Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2237–2246, 2015. URL: http://jmlr.org/proceedings/papers/v37/ahn15.html.

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

3 Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. *CoRR*, abs/1711.03076, 2017. URL: http://arxiv.org/abs/1711.03076, arXiv:1711.03076.

4 Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1345–1364, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2884435.2884528.

5 Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J. (2)*, 19(3):357–367, 1967. URL: https://doi.org/10.2748/tmj/1178243286, doi:10.2748/tmj/1178243286.

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

7 Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1326–1344, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2884435.2884527.

8 Graham Cormode, Hossein Jowhari, Morteza Monemizadeh, and S. Muthukrishnan. The Sparse Awakens: Streaming Algorithms for Matching Size Estimation in Sparse Graphs. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2017/7849, doi:10.4230/LIPIcs.ESA.2017.29.

9 Michael Crouch and Daniel M. Stubbs. Improved Streaming Algorithms for Weighted Matching, via Unweighted Matching. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 96–104, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2014/4690, doi:10.4230/LIPIcs.APPROX-RANDOM.2014.96.

10 Michael S. Crouch, Andrew McGregor, and Daniel Stubbs. Dynamic graphs in the sliding-window model. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, pages 337–348, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

11 Jack Edmonds. Paths, trees and flowers. *CANADIAN JOURNAL OF MATHEMATICS*, pages 449–467, 1965.

12 Sebastian Eggert, Lasse Kliemann, Peter Munstermann, and Anand Srivastav. Bipartite matching in the semi-streaming model. *Algorithmica*, 63(1):490–508, Jun 2012. doi:10.1007/s00453-011-9556-8.

13 Hossein Esfandiari, Mohammad T. Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete*

Algorithms, SODA '15, pages 1217–1233, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2722129.2722210.

14 Hossein Esfandiari, MohammadTaghi Hajiaghayi, and Morteza Monemizadeh. Finding large matchings in semi-streaming. In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain.*, pages 608–614, 2016. URL: https://doi.org/10.1109/ICDMW.2016.0092, doi:10.1109/ICDMW.2016.0092.

15 Alexander Fanghänel, Thomas Kesselheim, and Berthold Vöcking. Improved algorithms for latency minimization in wireless networks. *Theor. Comput. Sci.*, 412(24):2657–2667, May 2011. URL: http://dx.doi.org/10.1016/j.tcs.2010.05.004, doi:10.1016/j.tcs.2010.05.004.

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

17 Mohsen Ghaffari, Themis Gouleakis, Slobodan Mitrovic, and Ronitt Rubinfeld. Improved massively parallel computation algorithms for mis, matching, and vertex cover. *CoRR*, abs/1802.08237, 2018. URL: http://arxiv.org/abs/1802.08237, arXiv:1802.08237.

18 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485, 2012. URL: http://portal.acm.org/citation.cfm?id=2095157&CFID=63838676&CFTOKEN=79617016.

19 Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. *Algorithmica*, 76(3):654–683, November 2016. URL: http://dx.doi.org/10.1007/s00453-016-0138-7, doi:10.1007/s00453-016-0138-7.

20 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. doi:10.1137/0202019.

21 Sagar Kale and Sumedh Tirodkar. Maximum matching in two, three, and a few more passes over graph streams. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 15:1–15:21, 2017. URL: https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2017.15, doi:10.4230/LIPIcs.APPROX-RANDOM.2017.15.

22 Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1679–1697, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2627817.2627938.

23 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 734–751, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2634074.2634129.

24 Christian Konrad. Maximum matching in turnstile streams. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015*, pages 840–852, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

25 Christian Konrad. MIS in the congested clique model in $O(\log\log\Delta)$ rounds. *CoRR*, abs/1802.07647, 2018. URL: http://arxiv.org/abs/1802.07647, arXiv:1802.07647.

26 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimiza-*

tion. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings, pages 231–242, 2012. URL: https://doi.org/10.1007/978-3-642-32512-0_20, doi:10.1007/978-3-642-32512-0_20.

27 Christian Konrad and Adi Rosén. Approximating semi-matchings in streaming and in two-party communication. *ACM Trans. Algorithms*, 12(3):32:1–32:21, April 2016. URL: http://doi.acm.org/10.1145/2898960, doi:10.1145/2898960.

28 Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics 1989*. Cambridge University Press, Cambridge, 1989.

29 Andrew McGregor. Finding graph matchings in data streams. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 170–181, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

30 Andrew McGregor and Sofya Vorotnikova. A Simple, Space-Efficient, Streaming Algorithm for Matchings in Low Arboricity Graphs. In Raimund Seidel, editor, *1st Symposium on Simplicity in Algorithms (SOSA 2018)*, volume 61 of *OpenAccess Series in Informatics (OASIcs)*, pages 14:1–14:4, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2018/8295, doi:10.4230/OASIcs.SOSA.2018.14.

31 Ami Paz and Gregory Schwartzman. A $2+\epsilon$-approximation for maximum weight matching in the semi-streaming model. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2153–2161, 2017. URL: https://doi.org/10.1137/1.9781611974782.140, doi:10.1137/1.9781611974782.140.

32 Xiaoming Sun and David P. Woodruff. Tight Bounds for Graph Problems in Insertion Streams. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 435–448, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2015/5316, doi:10.4230/LIPIcs.APPROX-RANDOM.2015.435.

33 Mariano Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1):1–20, Feb 2012. URL: https://doi.org/10.1007/s00453-010-9438-5, doi:10.1007/s00453-010-9438-5.