Online Matching with Recourse: Random Edge Arrivals

Aaron Bernstein
Rutgers University, New Brunswick, NJ, USA
bernstei@gmail.com

Aditi Dudeja
Rutgers University, New Brunswick, NJ, USA
aditi.dudeja@rutgers.edu

Abstract

The matching problem in the online setting models the following situation: we are given a set of servers in advance, the clients arrive one at a time, and each client has edges to some of the servers. Each client must be matched to some incident server upon arrival (or left unmatched) and the algorithm is not allowed to reverse its decisions. Due to this no-reversal restriction, we are not able to guarantee an *exact* maximum matching in this model, only an approximate one.

Therefore, it is natural to study a different setting, where the top priority is to match as many clients as possible, and changes to the matching are possible but expensive. Formally, the goal is to always maintain a maximum matching while minimizing the number of changes made to the matching (denoted the *recourse*). This model is called the online model with recourse, and has been studied extensively over the past few years. For the specific problem of matching, the focus has been on vertex-arrival model, where clients arrive one at a time with all their edges. A recent result of Bernstein et al. [1] gives an upper bound of $O\left(n \log^2 n\right)$ recourse for the case of general bipartite graphs. For trees the best known bound is $O(n \log n)$ recourse, due to Bosek et al. [4]. These are nearly tight, as a lower bound of $\Omega(n \log n)$ is known.

In this paper, we consider the more general model where all the vertices are known in advance, but the edges of the graph are revealed one at a time. Even for the simple case where the graph is a path, there is a lower bound of $\Omega(n^2)$. Therefore, we instead consider the natural relaxation where the graph is worst-case, but the edges are revealed in a *random* order. This relaxation is motivated by the fact that in many related models, such as the streaming setting or the standard online setting without recourse, faster algorithms have been obtained for the matching problem when the input comes in a random order. Our results are as follows:

- Our main result is that for the case of general (non-bipartite) graphs, the problem with random edge arrivals is almost as hard as in the adversarial setting: we show a family of graphs for which the expected recourse is $\Omega\left(\frac{n^2}{\log n}\right)$.
- We show that for some special cases of graphs, random arrival is significantly easier. For the case of trees, we get an upper bound of $O\left(n \log^2 n\right)$ on the expected recourse. For the case of paths, this upper bound is $O\left(n \log n\right)$. We also show that the latter bound is tight, i.e. that the expected recourse is at least $\Omega\left(n \log n\right)$.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases matchings, edge-arrival, online model

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2020.11

Funding *Aaron Bernstein*: This work was done while funded by NSF Award 1942010 and the Simon's Group for Algorithms & Geometry.

40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020).
Editors: Nitin Saxena and Sunil Simon; Article No. 11; pp. 11:1–11:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The online matching problem models a scenario in which a set of servers is given in advance, and a set of clients arrive one at a time, with each client incident to some of the servers. In the standard version of this model, the arriving client must be immediately matched to a free server or be left unmatched, and this decision is irrevocable. Due to this constraint, it is not possible to guarantee an *exact* matching, so the goal is to guarantee the best possible approximation. (See the work of Karp et al. [14], which shows that we can't get better than $1 - \frac{1}{e}$ approximation.)

But there are several applications where the top priority is to match *all* the clients (or at least to have a maximum matching), and the irreversibility condition of the standard online model is too restrictive; in applications such as streaming content delivery, web hosting, job scheduling, or remote storage it is preferable to reallocate the clients provided the number of reallocations is small (see [5] for more details). Therefore, over the past decade there have been many papers on the so-called online model with *recourse*, where the goal is to maintain an *exact* solution the problem, while making as few changes to this solution as possible.

In the case of matching in particular, existing results focus on the vertex-arrival model, which is analogous to the similar model in online matching without recourse. In this model, clients arrive one at a time and ask to be matched to a server. The algorithm is allowed to change the matching over time and must always maintain a maximum matching: the goal is then to minimize the total number of changes made to the matching, denoted the *recourse*. Note that the trivial recourse bound is $O(n^2)$ (n changes per client), but one can do significantly better. This model has been studied extensively (see for example, [9, 5, 2, 3, 10, 4, 1]), and the state of the art is an upper bound of $O(n \log^2 n)$ on the total recourse [1]) in bipartite graphs. For the special case of trees, the best known upper bound is $O(n \log n)$ due to [4]. These upper bounds nearly match the lower bound of $\Omega(n \log n)$ for trees due to [9].

In this paper, we consider a more general model where the graph can be non-bipartite and, more importantly, the *edges* in the graph are revealed one at a time; the algorithm must again maintain a maximum matching at all times. Unfortunately, we have very strong lower bounds when the order in which the edges arrive is adversarial; even for the simplest possible case of a path, $\Omega(n^2)$ recourse is necessary. To overcome this lower bound, we consider a natural relaxation of this model where the adversary can still choose the graph, but edges arrive in a *random* order. One of the motivations behind this relaxation is that in several related models, such as the online model without recourse or the streaming model, we have been able to get faster algorithms when the input is assumed to arrive in a random order rather than an adversarial order. (See [13, 16] for online model without recourse, and [15, 12, 8, 7] for the streaming model).

Our results show that for the case of trees and paths, we can do significantly better in the random edge-arrival model: in particular, we show an upper bound of $O(n \log n)$ on the expected recourse in the case of paths (which we show is tight), and a bound of $O\left(n \log^2 n\right)$ in the case of trees. But our main result is that in general graphs, the random arrival setting is provably almost as hard as the adversarial setting. We state our main results formally:

▶ **Theorem 1.** *For any $n > 2^{16}$, there is a (non-bipartite) graph G_n (described in Section 3.1) with n vertices and $\Theta(n \log n)$ edges, such that if edges of the graph arrive in a random order, then the total expected recourse taken by* ***any*** *algorithm that maintains a maximum matching in the graph is $\Omega\left(\frac{n^2}{\log n}\right)$.*

▶ **Theorem 2.** *Let T be a tree on n vertices and let the edges of T arrive one at a time in a random order. Then, the expected total recourse taken by an algorithm that maintains a maximum matching in T is at most $O(n \log^2 n)$.*

▶ **Theorem 3.** *Let P be a path on n vertices, and let the edges of P arrive in a random order. The expected total recourse taken by an algorithm that maintains a maximum matching in P is $O(n \log n)$. Moreover, this bound is tight: the expected recourse taken by* ***any*** *algorithm is $\Omega(n \log n)$.*

▶ Remark 4. For the lower bounds of Theorems 1 and 3, when we say that *any* algorithm has the given lower bound on expected recourse, this bound holds even if the algorithm knows the random permutation in advance. That is, the lower bound holds even if the algorithm is optimal for every possible ordering of the edges.

▶ Remark 5. For the upper bounds in Theorems 2 and 3, the algorithm we use simply changes the matching along an augmenting path whenever such a path becomes available due to the insertion of some edge. If there are multiple augmenting paths the algorithm can take, it chooses between them arbitrarily; the upper bound holds regardless of the choice of path.

We prove our main result, Theorem 1 in Section 3. For proofs of Theorem 2 and 3 we refer the reader to the full version of the paper. We leave as an intriguing open problem whether our lower bound in Theorem 1 also holds for *bipartite* graphs, or whether these graphs allow for expected $o(n^{2-\varepsilon})$ recourse when edges arrive in a random order. See Section 4 for more details.

2 Preliminaries

Let G be an unweighted graph. A matching in G is a set of vertex-disjoint edges. Given any matching M of G, we say that a vertex v is matched if it incident to an edge in M, and free otherwise. Given any two matchings M and M', we use $M \oplus M'$ to denote the symmetric difference. We study the model of online matching with recourse under random edge arrivals. In this model, the adversary fixes any graph $G = (V, E)$ with m edges and n vertices. The vertex set is given in advance, but the edges arrive one at a time; the arrival order $e_1, \ldots, e_m$ is a *random* permutation of E. The goal of the algorithm is to maintain a sequence of matchings $M_1, \ldots M_m$, such that M_i is a maximum matching in the graph $(V, \{e_1, \ldots, e_i\})$. The total recourse of the algorithm is $\sum_{i=1}^{m-1} |M_i \oplus M_{i+1}|$, which is the total number of changes made to the matching throughout the entire sequence of insertions.

Intuitively, an algorithm that minimizes recourse should only change the matching when the maximum matching in the graph increases in size. We formalize this intuition in the remainder of this section.

▶ **Definition 6.** *Define a sequence $M^*_{i_0}, M^*_{i_1}, \cdots, M^*_{i_\eta}$ to be only-augmenting if $M^*_{i_0} = \emptyset$, each $M^*_{i_j}$ is a maximum matching in G_{i_j}, and each symmetric difference $M^*_{i_j} \oplus M^*_{i_{j+1}}$ consists of a single augmenting path; that is, $M^*_{i_j} \oplus M^*_{i_{j+1}}$ consists of an odd-length path P in $\{e_1, \ldots, e_{i_{j+1}}\}$ such that every second edge of P is in M_{i_j}, but the first and last edges of P are not in M_{i_j}. We say that an algorithm is only augmenting if the sequence of* distinct *matchings produced by the algorithm is only-augmenting; in other words, in the sequence of matchings produced by an only-augmenting algorithm, for every $1 \le i \le m-1$, either $M_i = M_{i+1}$, or $M_i \oplus M_{i+1}$ consists of a single augmenting path.*

▶ **Definition 7.** *Let $r(\sigma)$ is the best recourse achievable on permutation σ by an algorithm that knows σ in advance, and let $r^*(\sigma)$ be the best recourse achievable by an only-augmenting algorithm that knows σ in advance. (Knowing σ in advance allows the respective algorithms to pick the best possible matching sequence for permutation σ.)*

▶ **Observation 8.** *Using the above notation, we note that $\mathbb{E}_\sigma[r(\sigma)]$ is a lower bound on the expected recourse of any algorithm, while $\mathbb{E}_\sigma[r^*(\sigma)]$ is a lower bound on the expected recourse of any only-augmenting algorithm. The lower bound applies even if the algorithm knows σ in advance.*

The following Lemma allows us to assume throughout the paper that we are working with an only-augmenting algorithm. The proof of this lemma is relegated to the full version of the paper.

▶ **Lemma 9.** *Given any permutation σ, we have $r(\sigma) = r^*(\sigma)$.*

We now restate our main Theorem with the above lemma in mind.

▶ **Theorem 10.** $\mathbb{E}_\sigma[r^*(\sigma)] = \Omega(n^2/\log(n))$

▶ **Observation 11.** *Observation 8, Lemma 9 and Theorem 10 immediately imply Theorem 1.*

The lower bound proof of Section 3 is devoted entirely to proving Theorem 10

3 Lower Bound on Expected Recourse in General Graphs

This section will be devoted to proving Theorem 10, the main result of our paper. Recall from the preliminaries that we can assume that the algorithm is only-augmenting (See Definition 6) and that it knows the entire permutation σ in advance. In other words, to prove Theorem 1, it is sufficient to prove Theorem 10.

Our proof will proceed as follows. In Section 3.1 we define our candidate graph G_n (we will refer to it as G from now). The main step will be to show that between the times when half the edges of the graph have arrived and a three-quarters of the edges have arrived, the graph induced by non-isolated vertices contains a perfect matching or a near perfect matching throughout (see Definition 15 for a definition of near perfect matching). We will then use this fact to prove Theorem 10.

▶ Remark 12. Before we describe our graph, we describe how we will go about proving the lower bound. Suppose that our algorithm is given graph $G = (V, E)$ as input, where $|E| = m$. In our model this graph is revealed to our algorithm one edge at a time, with the edges arriving in the order prescribed by a random permutation σ . Suppose we look at the graph at time $t < m$, then G_t, the graph at time t has the same distribution as the subgraph of G obtained by randomly sampling t out of m edges. We will show that between the times when $t = 0.5 \cdot m$ and $t = 0.75 \cdot m$, G_t will contain a perfect or a near-perfect matching. To prove this, we will show (in Section 3.2) that the distribution of G_t can be approximated by the following distribution: graph obtained by sampling each edge of G independently with probability $\frac{t}{m}$. Finally, we will prove our aforementioned claims about this new distribution (Section 3.3).

3.1 The Graph

We use n to denote the number of vertices in our graph. In this write-up, $s = 400 \log n$ and $t = \frac{n}{500 \log n}$. Let K_s denote the complete graph on s vertices. Our graph is called G (see Figure 1) and it consists of t copies of K_s that we index as $K_s^{(i)}$ for $1 \le i \le t$. The remaining $\frac{n}{5}$ vertices are partitioned into t sets $\left\{D^{(i)}\right\}_{1\leq i\leq t}$ of size $100\log n$ each. The graph G contains the following edges.

1. For $1 \leq i \leq t-1$, we introduce edges between every vertex of $K_s^{(i)}$ and every vertex of $K_s^{(i+1)}$. Additionally, edges are also introduced between every vertex of $K_s^{(1)}$ and every vertex of $K_s^{(t)}$.
2. For $1 \leq i \leq t$, we fix an arbitrary set $U^{(i)} \subset K_s^{(i)}$ of size $100\log n$. Introduce an arbitrary matching between $U^{(i)}$ and $D^{(i)}$. Call this matching $M^{(i)}$. Let $M = \cup_{i=1}^{t} M^{(i)}$; we add the edges of M to G. We also let $U = \cup_{i=1}^{t} U^{(i)}$ and $D = \cup_{i=1}^{t} D^{(i)}$. For any $u \in D \cup U$, we define $M(u)$ to be the vertex that u is matched to.

We denote the number of edges in G by m. Note that $m = \Theta(n\log n)$.

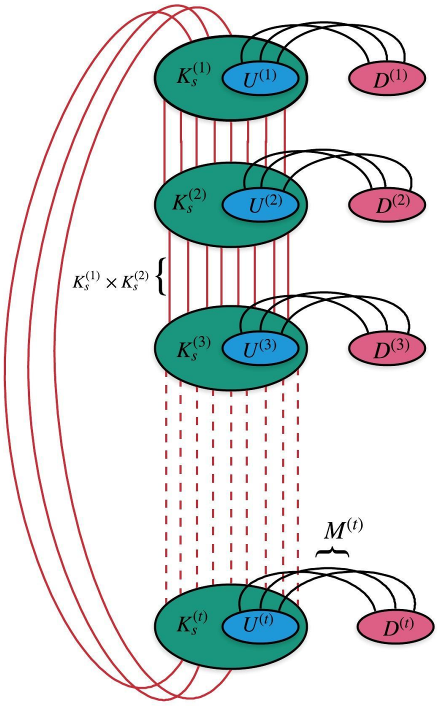

Figure 1 Graph G.

3.2 Relating G_p and $G^{p\cdot m}$

▶ **Definition 13.** *Let $p \in [0,1]$. We define $E_p \subset E(G)$ to be the set of edges obtained by sampling each $e \in E(G)$ with independently probability p.*
Let $V_p = V(G) \setminus \{v \in D$ such that $(v, M(v)) \notin E_p\}$; note that V_p excludes isolated vertices in D. Let G_p be the graph with vertex set V_p and edge set E_p.

▶ **Definition 14.** *Let $E^{p\cdot m} \subset E(G)$ be the set of edges obtained by sampling $p \cdot m$ random edges of $E(G)$. Let $V^{p\cdot m} = V(G) \setminus \{v \in D$ such that $(v, M(v)) \notin E^{p\cdot m}\}$; note that $V^{p\cdot m}$ excludes isolated vertices in D. Let $G^{p\cdot m}$ be the graph with vertex set $V^{p\cdot m}$ and the edge set $E^{p\cdot m}$.*

▶ **Definition 15.** *Let H be a graph with an odd number of vertices. Let $\mathcal{M}$ be any matching of H that leaves exactly one vertex unmatched. Then, $\mathcal{M}$ is called a near perfect matching of H.*

We state the main theorem that we want to prove in this section:

▶ **Theorem 16.** *Let* $p \in \{0.5, \frac{0.5\cdot m+1}{m}, \cdots, \frac{0.75\cdot m-1}{m}, 0.75\}$*, then, the graph* $G^{p\cdot m}$ *contains a perfect matching or a near perfect matching with probability at least* $1 - O\left(\frac{1}{n^3}\right)$*.*

To prove this theorem, we claim that it is sufficient to prove the following theorem:

▶ **Theorem 17.** *Let* $p \in \{0.5, \frac{0.5\cdot m+1}{m}, \cdots, \frac{0.75\cdot m-1}{m}, 0.75\}$*, then, graph* G_p *contains a matching or a near perfect matching with probability at least* $1 - O\left(\frac{1}{n^4}\right)$*.*

To show that Theorem 17 implies Theorem 16, we prove the following lemma:

▶ **Lemma 18.** *Let* $p \in \{0.5, \frac{0.5\cdot m+1}{m}, \cdots, \frac{0.75\cdot m-1}{m}, 0.75\}$*, and let* $G^{p\cdot m}$ *and* G_p *be as described above, and let* $\mathcal{G}$ *be the set of graphs that contain a perfect matching or a near perfect matching, then,*

$$\Pr\left(G^{p\cdot m} \notin \mathcal{G}\right) \leq 10\sqrt{m} \cdot \Pr\left(G_p \notin \mathcal{G}\right).$$

We refer the reader to the full version of this paper for a proof of Lemma 18. For now, we prove Theorem 16 assuming Theorem 17 and Lemma 18:

Proof (Theorem 16). It follows from Lemma 18 that:

$$\begin{aligned}\Pr\left(G^{p\cdot m} \text{ does not contain a matching}\right) &\leq 10\sqrt{m} \cdot \Pr\left(G_p \text{ does not contains a perfect matching}\right) \\ &= 10\sqrt{m} \cdot O\left(\frac{1}{n^4}\right) \qquad \text{(Due to Theorem 17)} \\ &= O\left(\frac{1}{n^3}\right) \qquad \text{(Since } m = \Theta\left(n \log n\right)\text{).}\end{aligned}$$ ◀

The following corollary follows from Theorem 16, via a union bound:

▶ **Corollary 19.** *Let* $I = \{0.5, \frac{0.5\cdot m+1}{m}, \cdots, \frac{0.75\cdot m-1}{m}, 0.75\}$*. Let* $\mathcal{G}$ *be the sequence of graphs* $\{G^{p\cdot m}\}_{p\in I}$*. The probability that every* $G \in \mathcal{G}$ *contains a perfect matching or a near perfect matching is at least* $1 - O\left(\frac{1}{n}\right)$*.*

The bulk of our paper is proving Theorem 17. But first, we provide some intuition for our choice of G by sketching how Corollary 19 implies our main result (Theorem 10).

Proof sketch of Theorem 10. Recall the edges $M \subset E(G)$ which connect the vertices in D, where $|M| = \Theta(n)$ (see 3.1). Consider how the graph $G^{p\cdot m}$ evolves from for $p = \frac{1}{2}$ to $p = \frac{3}{4}$. Let us assume without loss of generality that $G^{\frac{1}{2}\cdot m}$ contains an even number of vertices. Whenever an edge (d, x) from M is inserted into the graph, $d \in D$ is added to $V(G^{p\cdot m})$ (See Definition 13). Since we know from Corollary 19 that $G^{p\cdot m}$ contains a perfect matching whenever $V(G^{p\cdot m})$ is even, we know that after every two edges (d, x) and (d', x') added to M, there is a perfect matching in the resulting graph; thus, the algorithm must take some augmenting path from d to d'. Because G consists of $\Omega\left(\frac{n}{\log(n)}\right)$ consecutive layers, it is easy to see that with probability $\frac{1}{2}$, the shortest path from d to d' has length $\Omega\left(\frac{n}{\log(n)}\right)$. We expect to add $\frac{|M|}{4} = \Omega(n)$ edges to M between $G^{\frac{1}{2}\cdot m}$ and $G^{\frac{3}{4}\cdot m}$, so we have $\Omega(n)$ augmenting paths of expected length $\Omega\left(\frac{n}{\log(n)}\right)$, which implies total augmenting path length $\Omega\left(\frac{n^2}{\log(n)}\right)$. See Section 3.5 for full proof. ◀

3.3 Proving G_p has a Near-Perfect Matching

We now turn to proving Theorem 17. To this end, we introduce some notation:

▶ **Definition 20.** *Given G_p, we define the active subgraph A of G_p as follows: let $V(A) = V(G_p) \setminus \{u \in D \cup U : (u, M(u)) \in G_p\}$. The active subgraph A is the induced subgraph $G_p[V(A)]$.*

▶ **Definition 21.** *We define $A^{(i)}$ to be the following subgraph of G_p: let $V\left(A^{(i)}\right) = V(A) \cap V\left(K_s^{(i)}\right)$ for $1 \le i \le t$. Let $A^{(i)} = G_p\left[V\left(A^{(i)}\right)\right]$ For $1 \le i \le t$, let $|V\left(A^{(i)}\right)| = a_i$. Then,*

1. *If a_i is even, then let $P^{(i)} \cup Q^{(i)}$ be an arbitrary $\frac{a_i}{2}$ by $\frac{a_i}{2}$ bipartition of $V(A^{(i)})$.*
2. *If a_i is odd, then let $v^{(i)}$ be an arbitrary vertex in $V(A^{(i)})$ and let $P^{(i)} \cup Q^{(i)}$ be an arbitrary $\lfloor \frac{a_i}{2} \rfloor$ by $\lfloor \frac{a_i}{2} \rfloor$ bipartition of $V(A^{(i)}) \setminus v^{(i)}$.*

We denote $G(P^{(i)}, Q^{(i)})$ to be the bipartite graph between $P^{(i)}$ and $Q^{(i)}$, with edge set $E\left(P^{(i)}, Q^{(i)}\right) = \left(P^{(i)} \times Q^{(i)}\right) \cap E\left(A^{(i)}\right)$

▷ Claim 22. We observe that $V(A) \cap D = \emptyset$. This follows from the following two facts:

1. Consider any $u \in D$ such that $(u, M(u)) \notin G_p$. Then, $u \notin V(G_p)$. This follows immediately from Definition 13.
2. By Definition 20, we know that any u such that $(u, M(u)) \in G_p$ is not included in $V(A)$.

▷ Claim 23. From Definition 20, we know that $a_i \ge 400 \log n - |U^{(i)}|$. Since $|U^{(i)}| = 100 \log n$ (see Section 3.1 2), it follows that $a_i \ge 300 \log n$.

In order to prove Theorem 17, it is sufficient to prove the following theorem:

▶ **Theorem 24.** *The active subgraph, A contains a perfect matching or a near perfect matching with probability at least $1 - O\left(\frac{1}{n^4}\right)$.*

Proof (Theorem 17). Given a perfect (resp. near-perfect) matching $\mathcal{M}(A)$ of A, we will construct a perfect (resp. near perfect) matching $\mathcal{M}(G_p)$ of G_p. Consider any $u \in V(G_p) \setminus V(A)$. Note that $M(u) \in V(G_p) \setminus V(A)$ and $(u, M(u)) \in G_p$. So we may match u to $M(u)$ in G_p. In particular, $\mathcal{M}(G_p) = \mathcal{M}(A) \cup \{(u, M(u))$ where $u \in V(G_p) \setminus V(A)\}$. Thus, $\mathcal{M}(G_p)$ is a perfect (or a near perfect matching) of G_p if $\mathcal{M}(A)$ is a perfect (or a near perfect matching) of A. ◀

3.4 Near Perfect Matching in Active Subgraph

To prove Theorem 24, we need Chernoff bound, and some existing results on matchings in random bipartite graphs.

▶ **Theorem 25.** *[11] Define $B(n, n, p)$ to be the bipartite graph obtained by deleting edges from $K_{n,n}$ independently with probability $1 - p$. Then,*

$$\Pr\left(B(n,n,p) \text{ does not contain a perfect matching }\right) = O\left(ne^{-np}\right).$$

▶ **Theorem 26** (Chernoff Bounds). *Let $X_0, \cdots, X_k$ be $0 - 1$ random variables that are independent. Let $\mu = \mathbb{E}\left[\sum_{i=1}^k X_i\right]$. Then, for any $0 \le \delta \le 1$,*

$$\Pr\left(\sum_{i=1}^k X_i \le (1-\delta)\mu\right) \le e^{-\frac{\delta^2 \mu}{2}} \text{ and,} \tag{1}$$

$$\Pr\left(\sum_{i=1}^k X_i \ge (1+\delta)\mu\right) \le e^{-\frac{\delta^2 \mu}{3}}. \tag{2}$$

Consider the $A^{(i)}$'s in Definition 21. We mentioned that for some of these $A^{(i)}$'s the corresponding a_i's might be odd. Let $\left\{A^{(i_1)}, \cdots, A^{(i_k)}\right\}$ be this set, with $i_1 < \cdots < i_k$. Let $v^{(i_j)}$ be the vertex left out of the bipartition $P^{(i_j)} \cup Q^{(i_j)}$ of $A^{(i_j)}$ for $1 \leq j \leq k$ (see Definition 21.2). We define the following events:

▶ **Definition 27.** *For $1 \leq i \leq t$, let $\mathcal{A}_i$ be the event that $G(P^{(i)}, Q^{(i)})$ contains a perfect matching (or a near perfect matching). Let $\mathcal{A} = \cap_{i=1}^{t} \mathcal{A}_i$.*

▶ **Definition 28.** *Let $\mathcal{M}'_i$ be a maximum matching of $G\left(P^{(i)}, Q^{(i)}\right)$ for $1 \leq i \leq t$. Let $\mathcal{M}' = \cup_{i=1}^{t} \mathcal{M}'_i$.*

▶ **Definition 29.** *For $1 \leq m \leq \left\lfloor \frac{k}{2} \right\rfloor$, let $\mathcal{B}_m$ be the event that there is an augmenting path between $v^{(i_{2m-1})}$ and $v^{(i_{2m})}$ with respect to $\mathcal{M}'$ in A. Let $\mathcal{B} = \bigcap_{i=1}^{\left\lfloor \frac{k}{2} \right\rfloor} \mathcal{B}_m$.*

In order to prove Theorem 24, we follow these steps:

1. We will prove that each $\mathcal{A}_i$ happens with high probability, and therefore by union bound, $\mathcal{A}$ happens with high probability also.
2. We prove that each $\mathcal{B}_m$, conditioned on $\mathcal{A}$ happens with high probability, and by union bound, $\mathcal{B}$ conditioned on $\mathcal{A}$ also happens with high probability.

In order to prove 2, we will show that for each $1 \leq m \leq \left\lfloor \frac{k}{2} \right\rfloor$ there is an augmenting path between $v^{(i_{2m-1})}$ and $v^{(i_{2m})}$ which only consists of vertices between layers i_{2m-1} and i_{2m}. Therefore, these augmenting paths are vertex-disjoint from each other. These paths can be augmented simultaneously since they don't interfere with each other. So, 1 and 2 combined with this fact imply that the active graph, A contains a perfect matching or a near perfect matching with high probability.

Before we move on to proving 1 and 2, we note that $G(P^{(i)}, Q^{(i)})$ and $V(A^{(i)})$ are both random variables. In particular, $V(A^{(i)}) = \left(V(K_s^{(i)}) \setminus U^{(i)}\right) \cup S$, where S is a random subset of $U^{(i)}$ obtained by excluding every vertex with probability p. However, if we fix the vertex set $V\left(A^{(i)}\right)$, then the edges of $G(P^{(i)}, Q^{(i)})$ have the same distribution as that of a random bipartite graph; we remind the reader that $P^{(i)} \cup Q^{(i)}$ is an arbitrary bipartition of $A^{(i)}$ (see Definition 21). Formally:

▶ **Observation 30.** *For $1 \leq i \leq t$, $G\left(P^{(i)}, Q^{(i)}\right)$ conditioned on $V\left(A^{(i)}\right) = S$, where $|S| = a_i$, has the same distribution as $B\left(\left\lfloor \frac{a_i}{2} \right\rfloor, \left\lfloor \frac{a_i}{2} \right\rfloor, p\right)$.*

Now we prove the following lemma:

▶ **Lemma 31.** *For $1 \leq i \leq t$, $\Pr\left(\neg\mathcal{A}_i\right) = O\left(\frac{1}{n^5}\right)$. Moreover, $\Pr\left(\neg\mathcal{A}\right) = O\left(\frac{1}{n^4}\right)$.*

Proof. We know that:

$$
\begin{aligned}
\Pr\left(\neg\mathcal{A}_i\right) &= \sum_T \Pr\left(\neg\mathcal{A}_i \,\middle|\, V\left(A^{(i)}\right) = T\right) \cdot \Pr\left(V\left(A^{(i)}\right) = T\right) \\
&= \sum_T O\left(|T| \cdot e^{-|T| \cdot p}\right) \cdot \Pr\left(V\left(A^{(i)}\right) = T\right) \text{ (Due to Observation 30 and Lemma 25)} \\
&= \sum_T O\left(\frac{1}{n^5}\right) \cdot \Pr\left(V\left(A^{(i)}\right) = T\right) \text{ (Due to Claim 23 that } a_i \geq 300 \log n \text{ and } p \geq 0.5) \\
&= O\left(\frac{1}{n^5}\right) \text{ (Since we are summing over disjoint events).}
\end{aligned}
$$

By union bound it follows that, $\Pr\left(\neg\mathcal{A}\right) = O\left(\frac{1}{n^4}\right)$. ◀

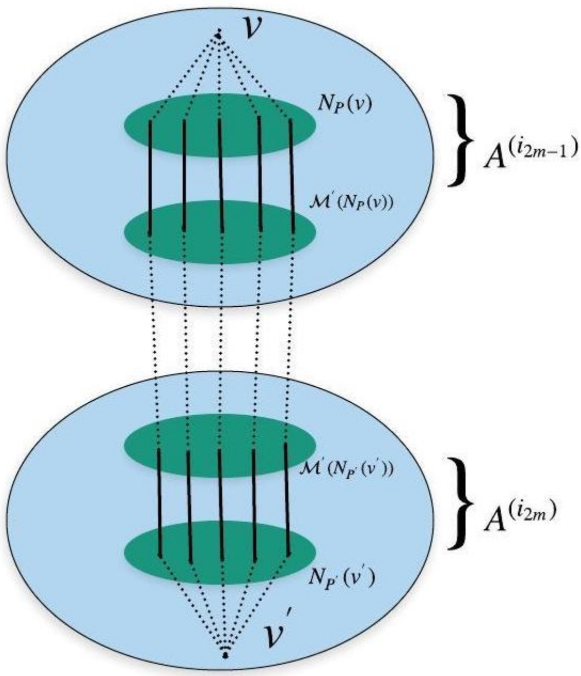

Figure 2 Case 1: When unmatched vertices are in consecutive layers.

▶ **Theorem 32.** *For* $1 \leq m \leq \lfloor \frac{k}{2} \rfloor$, $\Pr(\neg \mathcal{B}_m \mid \mathcal{A}) = O\left(\frac{1}{n^8}\right)$. *Therefore, by union bound it follows that* $\Pr(\neg \mathcal{B} \mid \mathcal{A}) = O\left(\frac{1}{n^7}\right)$.

Proof. To bound $\Pr(\neg \mathcal{B}_m \mid \mathcal{A})$, we consider two cases:

1. **Case 1: $\mathbf{v_{i_{2m-1}}}$ and $\mathbf{v_{i_{2m}}}$ are in consecutive layers.** That is, $i_{2m} = i_{2m-1} + 1$. We will give an overview of what we are about to do. We will use v to denote $v_{i_{2m-1}}$, v' to denote $v_{i_{2m}}$, P and P' to denote $P^{(i_{2m-1})}$ and $P^{(i_{2m})}$, Q and Q' to denote $Q^{(i_{2m-1})}$ and $Q^{(i_{2m})}$ respectively.

   ▶ **Definition 33.** *Let* $N_P(v)$ *(resp.* $N_{P'}(v')$*) denote the set of vertices in* P *(resp.* P'*) adjacent to* v *(resp.* v'*). Let* $deg_P(v)$ *(resp.* $deg_{P'}(v')$*) denote* $|N_P(v)|$ *(resp.* $|N_{P'}(v')|$*).*

   For a set of vertices S, let $\mathcal{M}'(S)$ denote the set of vertices matched to S in $\mathcal{M}'$ (refer to Definition 28 for the definition of $\mathcal{M}'$). We will prove that with high probability $|\mathcal{M}'(N_P(v))|$ and $|\mathcal{M}'(N_{P'}(v'))|$ are large. Conditioned on these sizes being large, we will prove that there is an edge (x, x') in A where $x \in \mathcal{M}'(N_P(v))$ and $x' \in \mathcal{M}'(N_{P'}(v'))$. It follows there is an augmenting path $\mathcal{P} = (v, \mathcal{M}'(x), x, x', \mathcal{M}'(x'), v')$ in A (note that $\mathcal{M}'(x) \in N_P(v)$ and $\mathcal{M}'(x') \in N_{P'}(v')$). (See Figure 2)
   To show this, we first show that $|N_P(v)|$ and $|N_{P'}(v')|$ are large with high probability. We will condition on $\mathcal{A}$, so $|\mathcal{M}'(N_P(v))|$ and $|\mathcal{M}'(N_{P'}(v'))|$ will consequently be large with high probability. It then follows that one of the edges between these two sets is in A with high probability.

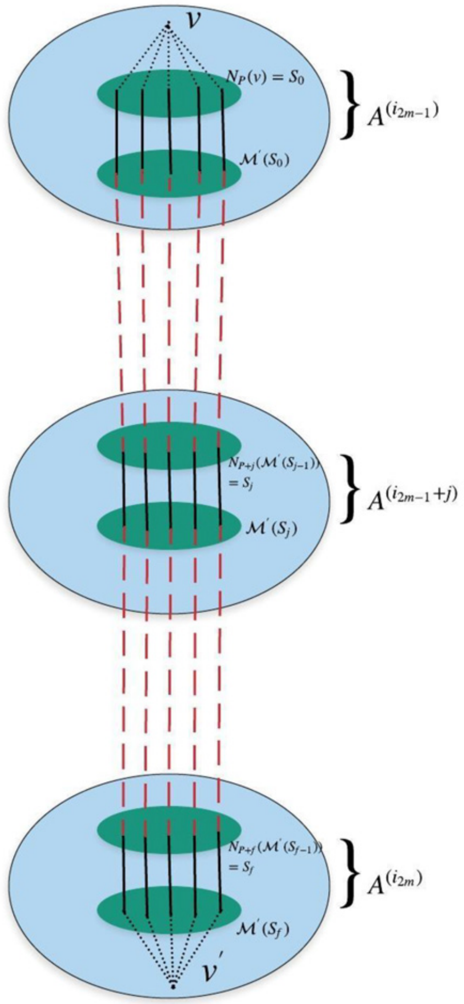

Figure 3 Case 2: When v and v' are not in consecutive layers.

We now turn to the formal proof of case 1. Let X_v and $X_{v'}$ be the random variables denoting $\deg_P(v)$ and $\deg_{P'}(v')$ respectively (see Definition 33). Each edge incident on v and v' in A is sampled independently with probability $p \in [0.5, 0.75]$. This is true even if we condition on the event $\mathcal{A}$. Consequently, $\mathbb{E}[X_v \mid \mathcal{A}] = \mathbb{E}[X_v] \geq 75 \log n$. Since X_v is the sum of $0-1$ independent random variables, we may apply Chernoff bound (see Theorem 26). It follows that:

$$\Pr(X_v \leq 25 \log n \mid \mathcal{A}) = O\left(\frac{1}{n^8}\right).$$

Similarly, we have:

$$\Pr(X_{v'} \leq 25 \log n \mid \mathcal{A}) = O\left(\frac{1}{n^8}\right).$$

Define $\mathcal{Y}$ to be the event that $|\mathcal{M}'(N_P(v))| \geq 25 \log n$ and $|\mathcal{M}'(N_{P'}(v'))| \geq 25 \log n$.

Observe that,

$$\Pr(\neg\mathcal{Y} \mid \mathcal{A}) \leq \Pr(X_v \leq 25\log n \mid \mathcal{A}) + \Pr(X_{v'} \leq 25\log n \mid \mathcal{A}) = O\left(\frac{1}{n^8}\right).$$

Define $\mathcal{Z}$ to be the event that there is an edge between $\mathcal{M}'(N_P(v))$ and $\mathcal{M}'(N_{P'}(v'))$. Observe that,

$$\Pr(\neg\mathcal{Z} \mid \mathcal{A}) \leq \Pr(\neg\mathcal{Y} \mid \mathcal{A}) + \Pr(\neg\mathcal{Z} \mid \mathcal{Y}, \mathcal{A}) = O\left(\frac{1}{n^8}\right) + \frac{1}{n^{O(\log n)}}.$$

The second term follows from the fact that each edge appears independently with probability $p \in [0.5, 0.75]$, and there are $\Omega(\log^2 n)$ edges between $\mathcal{M}'(N_P(v))$ and $\mathcal{M}'(N_{P'}(v'))$ conditioned on $\mathcal{Y}$. It follows that $\Pr(\neg\mathcal{B}_m \mid \mathcal{A}) \leq \Pr(\neg\mathcal{Z} \mid \mathcal{A}) = O\left(\frac{1}{n^8}\right)$. This proves our claim for this case.

2. **Case 2: $\mathbf{i_{2m} > i_{2m-1} + 1}$.** We denote $v_{i_{2m-1}}$ by v, $P^{(i_{2m-1})}$ by P and $v^{(i_{2m})}$ by v'. Let $f = i_{2m} - i_{2m-1}$. For $1 \leq j \leq f$, let $P^{(i_{2m-1}+j)}$ be denoted by $P + j$. We similarly define Q and $Q + j$ (see Figure 3). We also define the following sets:

$$S_0 = N_P(v)$$
$$S_j = N_{P+j}(\mathcal{M}'(S_{j-1})) \text{ for } 1 \leq j \leq f.$$

For $0 \leq j \leq f$, let $\mathcal{X}_j$ be the event that $|\mathcal{M}'(S_j)| \geq 25\log n$. Let $\mathcal{E}$ be the event that there is an edge between v' and $\mathcal{M}'(S_f)$. It is easy to check that the occurrence of $\mathcal{X}_0, \mathcal{X}_1 \cdots, \mathcal{X}_f$ implies that there is an alternating path from v to a large set of vertices (at least $\Omega(\log n)$) in $Q + j$ for all $j \in [f]$. Note that $\mathcal{E}$ implies that there is an edge from $Q + f$ to v'. Combined, $\mathcal{X}_1 \cdots, \mathcal{X}_f, \mathcal{E}$ imply an augmenting path from v to v'. We thus have:

▶ **Observation 34.** *Let $\mathcal{B}_m$ and $\mathcal{X}_1, \cdots, \mathcal{X}_f, \mathcal{E}$ be as defined above (refer to Definition 29 for a definition of $\mathcal{B}_m$), then:*

$$\Pr(\mathcal{B}_m \mid \mathcal{A}) \geq \Pr\left(\cap_{k=0}^{f}\mathcal{X}_k \cap \mathcal{E} \;\middle|\; \mathcal{A}\right).$$

From the above observation, we deduce that in order to upper bound $\Pr(\neg\mathcal{B}_m \mid \mathcal{A})$, it is sufficient to upper bound $\Pr\left(\cup_{k=0}^{f}\neg\mathcal{X}_k \cup \neg\mathcal{E} \;\middle|\; \mathcal{A}\right)$. We know that:

$$\Pr\left(\cup_{k=0}^{f}\neg\mathcal{X}_k \cup \neg\mathcal{E} \;\middle|\; \mathcal{A}\right) \leq \sum_{k=0}^{f}\Pr\left(\neg\mathcal{X}_k \mid \cap_{k=0}^{i-1}\mathcal{X}_k \cap \mathcal{A}\right) + \Pr\left(\neg\mathcal{E} \;\middle|\; \cap_{k=0}^{f}\mathcal{X}_k \cap \mathcal{A}\right).$$

(Follows from the definition of conditional probability)

We computed $\Pr(\neg\mathcal{X}_0 \mid \mathcal{A})$ in case 1. We remind the reader this is just the probability that $|\mathcal{M}'(S_0)| \leq 25\log n$. We now show how to compute $\Pr(\neg\mathcal{X}_j \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_{j-1})$. Consider any $w \in P + j$. We want to compute the probability that w is in the set $N_{P+j}(\mathcal{M}'(S_{j-1})) = S_j$ conditioned on the events $\mathcal{X}_{j-1}$ and $\mathcal{A}$. Since every edge on w is present in the active graph A independently with probability p:

$$\Pr(w \notin S_j \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_{j-1}) \leq (1-p)^{25\log n} \tag{3}$$
$$\leq \left(\frac{1}{2}\right)^{25\log n} \quad \text{(Due to the fact that } p \geq 0.5\text{)}. \tag{4}$$

This implies that:

$$\mathbb{E}\left[|S_j| \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_{j-1}\right] \geq 100 \log n.$$

Since $|S_j|$ is a sum of $0-1$ random variables (it is the sum of $\mathbb{1}_{\{v \in S_j\}}$, that take value 0 with probability $O\left(\frac{1}{n^{25}}\right)$ (due to Equation (4)) and 1 otherwise), we can apply Chernoff bounds (Theorem 26):

$$\Pr\left(|S_j| \leq 25 \log n \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_{j-1}\right) = O\left(\frac{1}{n^9}\right).$$

Since we condition on $\mathcal{A}$ (that is a perfect or, a near perfect matching being present), we know that:

$$|\mathcal{M}'(S_j)| = |S_j|$$

Consequently, we have:

$$\begin{aligned}\Pr\left(|\mathcal{M}'(S_j)| \leq 25 \log n \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_{j-1}\right) &= \Pr\left(|S_j| \leq 25 \log n \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_{j-1}\right) \\ &= O\left(\frac{1}{n^9}\right).\end{aligned}$$

Finally, we want to bound $\Pr\left(\neg\mathcal{E} \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_f\right)$. This can be upper bounded:

$$\begin{aligned}\Pr\left(\neg\mathcal{E} \mid \mathcal{A}, \mathcal{X}_0, \cdots, \mathcal{X}_f\right) &\leq \left(\frac{1}{2}\right)^{25 \log n} \\ &\text{(Edges on } v' \text{ appear independently with probability } p \geq 0.5) \\ &= O\left(\frac{1}{n^{25}}\right).\end{aligned}$$

It is immediate from Observation 34 that:

$$\Pr\left(\neg\mathcal{B}_m \mid \mathcal{A}\right) = O\left(\frac{1}{n^8}\right).$$

From case 1 and case 2, we know that by union bound, $\Pr\left(\neg\mathcal{B} \mid \mathcal{A}\right) = O\left(\frac{1}{n^7}\right)$. ◀

Proof (Theorem 24). From Lemma 31 and Theorem 32 we have that:

$$\Pr\left(A \text{ does not contain a perfect matching }\right) \leq \Pr\left(\neg\mathcal{A}\right) + \Pr\left(\neg\mathcal{B} \mid \mathcal{A}\right) = O\left(\frac{1}{n^4}\right). \quad ◀$$

3.5 Lower Bound On Lengths of Augmenting Paths

We start with some definitions:

▶ **Definition 35.** *For $i \in \{1, \cdots, m\}$, we denote by e_i, the edges arriving at time i. Let $S = \{e_{0.5m}, \cdots, e_{0.75m}\}$.*

This section will be devoted to proving that among the edges in S, $\Omega(n)$ edges will join augmenting paths of expected length $\Omega\left(\frac{n}{\log n}\right)$, and the algorithm is forced to augment along these. Formally,

▶ **Theorem 36.** *With high probability, there exists $S' \subset S$, $|S'| \geq \frac{n}{100}$ such that each $e \in S'$ joins an augmenting path of expected length at least $\Omega\left(\frac{n}{\log n}\right)$.*

We first give a proof of Theorem 10 using Theorem 36:

Proof (Theorem 10). For $i \in [m]$, let $\mathcal{Z}_i$ be the random variable denoting the length of the augmenting path that we augment along when the edge e_i joins. Let $\mathcal{Z} = \sum_{i=1}^{m} \mathcal{Z}_i$, which is the random variable denoting the total length of the augmenting paths taken during the course of the algorithm. We want to compute the quantity $\mathbb{E}[\mathcal{Z}]$. We note that:

$$\begin{aligned}\mathbb{E}[\mathcal{Z}] &= \sum_{i=1}^{m} \mathbb{E}[\mathcal{Z}_i] \geq \sum_{j \in S'} \mathbb{E}[\mathcal{Z}_j] = |S'| \cdot \Omega\left(\frac{n}{\log n}\right) \\ &= \Omega\left(\frac{n^2}{\log n}\right). \\ &\text{(Due to Theorem 36)}\end{aligned}$$

◀

Before we prove Theorem 36, we need certain observations, and the following version of Chernoff for negatively associated random variables:

▶ **Theorem 37.** *[6] Let $X_0, \cdots, X_k$ be $0-1$ random variables that are negatively associated. Let $\mu = \mathbb{E}\left[\sum_{i=1}^{k} X_i\right]$. Then, for any $0 \leq \delta \leq 1$,*

$$\Pr\left(\sum_{i=1}^{k} X_i \leq (1-\delta)\mu\right) \leq e^{-\frac{\delta^2\mu}{2}} \text{ and,} \tag{5}$$

$$\Pr\left(\sum_{i=1}^{k} X_i \geq (1+\delta)\mu\right) \leq e^{-\frac{\delta^2\mu}{3}}. \tag{6}$$

We remind the reader of the edges M in graph G between D and U (refer to Section 3.12). Note that $|M| \geq \frac{n}{5}$. Further, $M = \cup_{i=1}^{t} M^{(i)}$, and $|M^{(i)}| \geq 100 \log n$ for all $i \in [t]$.

We now prove the following claim about S:

▷ Claim 38. Let $\mathcal{R}$ be the event that for all $i \in [t]$, $|M^{(i)} \cap S| \geq 10 \log n$. Then, $\Pr(\mathcal{R}) \geq 1 - O\left(\frac{1}{n^3}\right)$.

Proof. Consider any $M^{(i)}$, and let $e \in M^{(i)}$. Let Z_e be a $0-1$ random variable that takes value 1 if $e \in S$, and 0 otherwise. Let $Z = \sum_{e \in M^{(i)}} Z_e$. This is the random variable that denotes $|M^{(i)} \cap S|$. Further, Z is a sum of negatively associated random variables, and therefore obeys the condition of Theorem 37. We note the following:

$$\Pr(Z_e = 1) = \frac{1}{4} \text{ and, } \mathbb{E}[Z] = 25 \log n.$$

It follows that:

$$\Pr(Z \leq 10 \log n) \leq \exp\left(-(0.6)^2 (0.5) 25 \log n\right) \leq \exp(-4.5 \log n) = O\left(\frac{1}{n^4}\right).$$

Due to union bound, we know that $\Pr(\mathcal{R}) \geq 1 - O\left(\frac{1}{n^3}\right)$. ◁

We also have the following corollary due to Claim 38:

▶ **Corollary 39.** *With probability at least $1 - O\left(\frac{1}{n^3}\right)$, $|M \cap S| \geq \frac{n}{50}$.*

We are ready to define the candidate set S' in Theorem 36, but before that we give a final definition:

▶ **Definition 40.** *Consider edges $e \in M^{(i)}$ and $f \in M^{(j)}$ (see 3.12 for the definition of $M^{(i)}$). Then let $d(e,f) = \min\{t - |i-j|, |i-j|\}$.*

Let $M \cap S = \{e_{i_1}, \cdots, e_{i_q}\}$. Let us assume without loss of generality that before the arrival of e_{i_1}, the set $V(G_{i_1-1})$ is even, so by Theorem 16 the graph G_{i_1-1} has a perfect matching. We define S' to contain every second edge of $M \cap S$: that is, $S' = \left\{e_{i_2}, e_{i_4}, \cdots, e_{i_{2\lfloor \frac{q}{2} \rfloor}}\right\}$. For the rest of the proof we proceed as follows: we will show that with high probability, when $e_{i_{2s}}$ arrives, it will join an augmenting path ending at $e_{i_{2s-1}}$ where $s \in \{1, \cdots, \lfloor \frac{q}{2} \rfloor\}$. Let $e_{i_{2s}} \in M^{(j)}$ and $e_{i_{2s+1}} \in M^{(j')}$. Then, the length of the augmenting path that $e_{i_{2s-1}}$ joins is at least $d(e_{i_{2s-1}}, e_{i_{2s}}) = \min\{t - |j'-j|, |j'-j|\}$. We prove that the expected value of this quantity is at least $\Omega\left(\frac{n}{\log n}\right)$.

We prove the following observation:

▶ **Lemma 41.** *Let e and f be two edges that are chosen uniformly at random from M. Then, $\mathbb{E}[d(e,f)] \geq \frac{n}{2000 \log n}$.*

Proof. The total number of possible choices for $(e,f) = \left(\frac{n}{5}\right) \cdot \left(\frac{n}{5} - 1\right)$. The total number choices for (e,f) such that $d(e,f) = k$, are $\left(\frac{n}{500 \log n}\right) \cdot (100 \log n) \cdot (200 \log n)$. To see this, fix a layer for e, then the number of choices of f for which $d(e,f) = k$ are exactly $200 \log n$. Finally, the total number possible choices of layers for e is $\frac{n}{500 \log n}$. This implies that:

$$\begin{aligned}
\Pr(d(e,f) = k) &= \frac{\left(\frac{n}{500 \log n}\right) \cdot (100 \log n) \cdot (200 \log n)}{\left(\frac{n}{5}\right) \cdot \left(\frac{n}{5} - 1\right)} \\
&\geq \frac{\left(\frac{n}{500 \log n}\right) \cdot (100 \log n) \cdot (200 \log n)}{\left(\frac{n}{5}\right) \cdot \left(\frac{n}{5}\right)} \\
&\geq \frac{1000 \log n}{n}.
\end{aligned}$$

Finally, we have that:

$$\mathbb{E}[d(e,f)] = \sum_{k=0}^{\frac{t}{2}} k \cdot \Pr(d(e,f) = k) \geq \frac{t}{4} = \frac{n}{2000 \log n}. \qquad ◀$$

We state an immediate corollary of Lemma 41:

▶ **Corollary 42.** *For all $s \in \{1, \cdots, \lfloor \frac{q}{2} \rfloor\}$, $\mathbb{E}\left[d\left(e_{i_{2s-1}}, e_{i_{2s}}\right)\right] \geq \frac{n}{2000 \log n}$.*

▶ **Lemma 43.** *If $G^{p \cdot m}$ contains a perfect matching or a near perfect matching for all $p \in \{0.5, \frac{0.5 \cdot m+1}{m}, \cdots, \frac{0.75 \cdot m-1}{m}, 0.75\}$, then for all $s \in \{1, \cdots, \lfloor \frac{q}{2} \rfloor\}$, $e_{i_{2s}}$ joins an augmenting path that ends in $e_{i_{2s-1}}$.*

Proof. We remind the reader that $|V(G^{p \cdot m})|$ is a random variable (check Definition 14) and it's value increases if and only if the edges in M arrive. Recall the assumption that $|V(G_{i_1-1})|$ is even. Upon the arrival of e_{i_1}, we have a near perfect matching in the graph, and this remains the case until e_{i_2} arrives. At this point under our assumption, there must be a perfect matching in the graph. However, the matching that is currently maintained by the algorithm leaves the vertices are the end points of e_{i_1} and e_{i_2} in D unmatched. (Here we use the simplifying assumption from the preliminaries that the algorithm is only-augmenting, so since the arrival of e_{i_1} does not increase the size of the maximum matching, and since the algorithm only changes the matching via augmenting paths, the endpoint of e_{i_1} in D remains free until the arrival of e_{i_2}.) It follows that these endpoints are joined together by an augmenting path. Continuing this way, we can prove the theorem for any $s \in \left\{1, \cdots, \left\lfloor \frac{q}{2} \right\rfloor\right\}$. ◀

Proof (Theorem 36). Let $\mathcal{F}$ be the event that there is an $S' \subset S$, $|S'| \geq \frac{n}{100}$ such that each $e \in S'$ augments along a path of expected length at least $\Omega\left(\frac{n}{\log n}\right)$. Note that the event $\mathcal{F}$ fails to happen if one of these go wrong:

1. $|S'| \leq \frac{n}{100}$. We call this event $\neg\mathcal{U}$. We know from Corollary 39 that $\Pr(\neg\mathcal{U}) = O\left(\frac{1}{n^3}\right)$. This is because S' just takes alternate elements from S.
2. Let $\mathcal{V}$ be the event that for all $p \in \left\{0.5, \frac{0.5\cdot m+1}{m}, \cdots, \frac{0.75\cdot m-1}{m}, 0.75\right\}$, $G^{p\cdot m}$ contain a perfect matching or a near perfect matching. Then, from Lemma 43 we know that $\mathcal{V}$ implies that for all $s \in \left\{1, \cdots, \left\lfloor \frac{q}{2} \right\rfloor\right\}$, $e_{i_{2s-1}}$ joins an augmenting path ending in $e_{i_{2s}}$. From Corollary 42, we know all these paths have expected length at least $\frac{n}{2000 \log n}$. We know from Corollary 19, that $\Pr(\neg\mathcal{V}) = O\left(\frac{1}{n}\right)$.

It follows that the occurrence of $\mathcal{A}$ and $\mathcal{B}$ implies the occurrence of $\mathcal{F}$. Consequently, $\Pr(\mathcal{F}) \geq 1 - \Pr(\neg\mathcal{U}) - \Pr(\neg\mathcal{V}) \geq 1 - O\left(\frac{1}{n}\right)$. ◀

4 Conclusion and Open Problems

We consider the problem of maximum matching with recourse in the random edge-arrival setting. The goal is to compute the expected recourse. As mentioned in the introduction, there are strong lower bounds of $\Omega\left(n^2\right)$ in the adversarial edge-arrival model, even for the case of simple paths. For random edge-arrivals, we can do significantly better for special classes of graphs: we prove an upper bound of $O\left(n \log n\right)$ for the case of paths and $O\left(n \log^2 n\right)$ for the case of trees. This bound is tight up to $\log n$ factors, since we prove that for the case of paths, any algorithm must take expected total recourse of $\Omega\left(n \log n\right)$. But for general graphs, we show that random arrival is basically as hard as adversarial arrival: we give a family of graphs for which the expected recourse is at least $\Omega\left(\frac{n^2}{\log n}\right)$.

An interesting open question is the case of *bipartite* graphs: if edge-arrivals are random, can we prove a similar lower bound of $\Omega\left(\frac{n^2}{\mathrm{polylog}(n)}\right)$ on the expected recourse? Our current lower-bound construction seems hard to extend to the bipartite case, as our proof crucially relies on the fact that after a constant fraction of the edges have arrived, if we focus only on the non-isolated vertices in the lower-bound graph G, then G contains a perfect matching with high probability. This allowed us to force the adversary to take an augmenting path between every new pair of non-isolated vertices. But in the case of bipartite graphs, it seems difficult to guarantee a perfect matching between the non-isolated vertices because the number of non-isolated vertices on the left might not be equal to the number on the right; in fact, they are likely to differ by a $\Theta\left(\sqrt{n}\right)$ factor.

References

1 Aaron Bernstein, Jacob Holm, and Eva Rotenberg. Online bipartite matching with amortized $o(\log^2 n)$ replacements. *Journal of the ACM (JACM)*, 66(5):1–23, 2019.

2 Bartlomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Online bipartite matching in offline time. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 384–393. IEEE, 2014.

3 Bartłomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Shortest augmenting paths for online matchings on trees. In *International Workshop on Approximation and Online Algorithms*, pages 59–71. Springer, 2015.

4 Bartłomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych-Pawlewicz. A tight bound for shortest augmenting paths on trees. In *Latin American Symposium on Theoretical Informatics*, pages 201–216. Springer, 2018.

5 Kamalika Chaudhuri, Constantinos Daskalakis, Robert D Kleinberg, and Henry Lin. Online bipartite perfect matching with augmentations. In *IEEE INFOCOM 2009*, pages 1044–1052. IEEE, 2009.

6 Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

7 Alireza Farhadi, Mohammad Taghi Hajiaghayi, Tung Mai, Anup Rao, and Ryan A. Rossi. Approximate maximum matching in random streams. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1773–1785. SIAM, 2020. doi:10.1137/1.9781611975994.108.

8 Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 491–500, 2019.

9 Edward F Grove, Ming-Yang Kao, P Krishnan, and Jeffrey Scott Vitter. Online perfect matching and mobile computing. In *Workshop on Algorithms and Data Structures*, pages 194–205. Springer, 1995.

10 Anupam Gupta, Amit Kumar, and Cliff Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 468–479. SIAM, 2014.

11 Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random graphs*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 2000. doi:10.1002/9781118032718.

12 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*, pages 734–751. SIAM, 2014.

13 Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 587–596, 2011.

14 Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.

15 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 231–242. Springer, 2012.

16 Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 597–606, 2011.