

# Tight Bounds for Online Edge Coloring

Ilan Reuven Cohen\*

CWI

Email: ilancohen@gmail.com

Binghui Peng<sup>§</sup>

Columbia University

Email: bp2601@columbia.edu.

David Wajc<sup>†</sup>

CMU

Email: dwajc@cs.cmu.edu.

**Abstract**—Vizing’s celebrated theorem asserts that any graph of maximum degree  $\Delta$  admits an edge coloring using at most  $\Delta + 1$  colors. In contrast, Bar-Noy, Motwani and Naor showed over a quarter century ago that the trivial greedy algorithm, which uses  $2\Delta - 1$  colors, is optimal among online algorithms. Their lower bound has a caveat, however: it only applies to low-degree graphs, with  $\Delta = O(\log n)$ , and they conjectured the existence of online algorithms using  $\Delta(1 + o(1))$  colors for  $\Delta = \omega(\log n)$ . Progress towards resolving this conjecture was only made under stochastic arrivals (Aggarwal et al., FOCS’03 and Bahmani et al., SODA’10).

We resolve the above conjecture for adversarial vertex arrivals in bipartite graphs, for which we present a  $(1 + o(1))\Delta$ -edge-coloring algorithm for  $\Delta = \omega(\log n)$  known a priori. Surprisingly, if  $\Delta$  is not known ahead of time, we show that no  $(\frac{e}{e-1} - \Omega(1))\Delta$ -edge-coloring algorithm exists. We then provide an optimal,  $(\frac{e}{e-1} + o(1))\Delta$ -edge-coloring algorithm for unknown  $\Delta = \omega(\log n)$ . To obtain our results, we study a nonstandard fractional relaxation for edge coloring, for which we present optimal fractional online algorithms and a near-lossless online rounding scheme, yielding our optimal randomized algorithms.

**Keywords**—online algorithms; edge coloring; online coloring; adversarial arrivals

## I. INTRODUCTION

Edge coloring is the problem of assigning a color to each edge of a multigraph so that no two edges with a common endpoint have the same color. This classic problem, even restricted to bipartite graphs, can be used to model scheduling problems arising in sensor networks [27], switch routing [1], radio-hop networks [56] and optical networks [52], among others. Edge coloring can trace its origins back to the 19th-century works of Tait [55] and Petersen [51], who studied this problem in the context of the four color theorem. Shannon [54] later studied edge coloring in the context of color coding wires in electrical units, and proved that any multigraph  $G$  of maximum degree  $\Delta = \Delta(G)$  admits a  $\lfloor \frac{3\Delta}{2} \rfloor$ -edge-coloring; i.e., a coloring using at most

$\lfloor \frac{3\Delta}{2} \rfloor$  colors. (This is tight.) Inspired by this result, Vizing [57] proved that any simple graph can be edge colored using  $\Delta + 1$  colors. Clearly,  $\Delta$  colors are necessary to edge color a graph, and for bipartite graphs multiple near-linear-time  $\Delta$ -edge-coloring algorithms are known [2, 12, 29]. For general graphs, several polytime  $(\Delta + 1)$ -edge-coloring algorithms are known [25, 47, 57], and this too is likely optimal, as determining whether a general graph is  $\Delta$ -edge-colorable is NP-hard [32].

In addition to these optimal polytime algorithms, there exists a simple quasilinear-time  $(2\Delta - 1)$ -edge-coloring greedy algorithm, which colors each edge with the lowest color unused by its adjacent edges. The greedy algorithm is implementable in many restricted models of computation, and improving upon its coloring guarantees, or even matching them quickly in such models, has been the subject of intense research. Examples include PRAM [41], NC and RNC [6, 38, 48], dynamic [7, 14] and distributed algorithms (e.g., [10, 15, 19, 24, 28, 50]).

For online algorithms, little progress was made towards beating the greedy algorithm. The only positive results are under *random-order* edge arrival in “dense” bipartite multigraphs. Specifically, under such stochastic arrivals, Aggarwal et al. [1] showed how to obtain a  $\Delta(1 + o(1))$  edge coloring of  $n$ -vertex multigraphs if  $\Delta = \omega(n^2)$  and Bahmani et al. [4] showed how to obtain a  $1.26\Delta$  edge coloring under the milder assumption that  $\Delta = \omega(\log n)$ . The lack of progress for adversarial arrival order is likely explained by the following theorem of Bar-Noy et al. [5].

**Theorem 1** ([5], informal). *No online edge coloring algorithm can  $2\Delta - 2$  edge-color a graph.*

However, the lower bound of Bar-Noy et al. requires a number of nodes  $n$  exponential in  $\Delta$ : that is, it only holds for some  $\Delta = O(\log n)$ . Therefore, this lower bound can be thought of as an *additive* lower bound of  $\Delta + \Omega(\log n)$ , rather than a multiplicative lower bound of  $\approx 2\Delta$ . Put otherwise, it does not preclude a better-than- $(2\Delta - 1)$  edge-coloring algorithm for  $\Delta = \Omega(\log n)$  large enough. Indeed, Bar-Noy et al. went so far as to conjecture that  $\Delta(1 + o(1))$ -edge-colorings are computable online for large enough  $\Delta$ .

**Conjecture 2** ([5]). *There exists an online algorithm which  $\Delta(1 + o(1))$  edge-colors graphs with  $\Delta = \omega(\log n)$ .*

\*Work done in part while the author was at CMU and the University of Pittsburgh.

<sup>§</sup>Work done in part while the author was an undergraduate in Tsinghua University and visiting CMU as an intern.

<sup>†</sup>Work done in part while the author was visiting EPFL. This work was supported in part by NSF grants CCF-1527110, CCF-1618280, CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808 and a Sloan Research Fellowship.

*Our focus:* In this paper we study edge coloring under the adversarial online vertex-arrival model of Karp et al. [39], where vertices on one side of a bipartite graph arrive over time, with their edges to previously-arrived neighbors. In this model, an online algorithm colors each edge  $e$  upon arrival, immediately and irrevocably. Recall from above that edge coloring in bipartite graphs has multiple applications, including in online settings. Indeed, such an application of edge coloring bipartite graphs to switch routing (with input switches on one side and output switches on the other) was precisely the motivation of Aggarwal et al. [1] to study online edge coloring. The online edge coloring lower bound of [5] holds even for bipartite vertex arrivals. We show that for such arrivals a large enough maximum degree indeed allows to circumvent Bar-Noy et al.’s lower bound, and prove their conjecture. In particular, we present optimal algorithms (up to  $o(1)$  terms) both for the known- $\Delta$  scenario, as well as for the stricter online problem where  $\Delta (= OPT)$  is unknown a priori.

#### A. Our Contributions

We provide the following optimal results for online edge coloring under adversarial vertex arrivals. For conciseness, we state our results in terms of competitiveness, calling an  $\alpha \cdot \Delta$ -edge-coloring algorithm  $\alpha$ -competitive, as the optimal edge coloring requires at least  $\Delta$  colors.

Our first result is an optimal algorithm for known large  $\Delta$ .

**Theorem 3.** *There exists a  $(1 + o(1))$ -competitive randomized edge coloring algorithm for bipartite graphs of known maximum degree  $\Delta = \omega(\log n)$ . A competitive ratio of  $1 + o(1)$  is optimal – no randomized algorithm is  $1 + o(1/\sqrt{\Delta})$  competitive for any  $\Delta$ .*

Like all prior non-trivial online algorithms, the above algorithm assumes a priori knowledge of a critical parameter of the input, namely  $\Delta$ , which is the optimum number of colors needed to color the bipartite graph. However, in many online scenarios, such assumptions are unreasonable. We show that removing this assumption results in a strictly harder problem, though here too greedy is suboptimal. Our main contribution is an optimal online algorithm for unknown large  $\Delta$ .

**Theorem 4.** *There exists an  $(\frac{e}{e-1} + o(1))$ -competitive randomized edge coloring algorithm for bipartite graphs of unknown maximum degree  $\Delta = \omega(\log n)$ . This is optimal (up to  $o(1)$  terms) – no algorithm is better than  $\frac{e}{e-1}$  competitive for unknown  $\Delta$ .*

**Remark 1.** For simplicity we stated our positive results in theorems 3 and 4 for  $\Delta = \omega(\log n)$ . More generally, our algorithms’ competitive ratios are of the form  $\alpha + O(\sqrt{\log n/\Delta})$  for some constant  $c \geq 1$  and  $\alpha = 1$  or  $\alpha = \frac{e}{e-1}$ , respectively (see Section IV). Thus, we obtain

better-than-2 competitive ratios already for sufficiently large  $\Delta = O(\log n)$ .

**Remark 2.** We stated all our positive results for simple graphs, though they hold more generally for any multigraph with maximum edge multiplicity  $o(\Delta)$ . (A necessary condition – see Section XII).

Our upper and lower bounds rely on a nonstandard fractional relaxation for edge coloring. In particular, we present matching upper and lower bounds for this relaxation, and present a nearly-lossless online rounding of solutions to this relaxation. Using this relaxation, we show a complementary result: a separation between online edge coloring on general and bipartite graphs, which we prove by showing a higher lower bound for the former (as well as a better-than-greedy fractional algorithm for the latter).

**Theorem 5.** *No fractional online edge coloring algorithm is better than  $1.606 (> \frac{e}{e-1})$  competitive for general graphs of unknown maximum degree  $\Delta$ . On the other hand, there exists a fractional edge coloring algorithm which is  $1.777$  competitive for general graphs of unknown maximum degree  $\Delta$ .*

To conclude, relying on our new relaxation, we present the first online algorithms beating greedy under any adversarial arrivals. In particular, we prove the conjecture of Bar-Noy et al. and provide tight bounds for the well-studied model of Karp et al., both under known and unknown  $\Delta$ .

#### B. Techniques

*Novel Relaxation:* The classic fractional relaxation for edge coloring asks to minimize  $\sum_M x_M$ , subject to  $\sum_{M \ni e} x_e = 1$  for every  $e \in E$  and  $x_M \geq 0$  for every matching  $M$ . That is, this relaxation fractionally uses integral matchings to cover each edge. For the online problem, this standard relaxation is not particularly useful, as the set of matchings in the input graph is unknown a priori (since the edge set is unknown). Our first insight is a novel fractional relaxation which allows for more “myopic” assignments upon vertex arrivals – its variables  $x_{e,c}$  are the extent to which an edge  $e$  is colored  $c$ . Specifically, our relaxation *integrally* uses *fractional* matchings to cover each edge; i.e.,  $\sum_c x_{e,c} = 1$ . The goal is to minimize the number of non-zero fractional matchings used. This simple change to the relaxation proves particularly useful in the online setting, and underlies both our upper and lower bounds. We believe this relaxation may find applications to other incomplete-information models, such as dynamic, distributed, and local computation algorithms.

*Fractional Algorithms:* Our relaxation admits a trivial 1-competitive solution: set  $x_{e,c} = 1/\Delta$ , for each edge and  $c \in [\Delta]$ . If  $\Delta$  is known a priori, this solution can even be computed by an *online* fractional algorithm (which must fix the values  $x_{ec}$  for each edge  $e$  immediately on arrival). By our lower bounds, for unknown  $\Delta$ , other algorithms are

needed. A natural candidate is the greedy “water-filling” algorithm, which continuously increases an arriving edge’s assignment for all colors minimizing the maximum load for either endpoint. However, this approach may yield an extremely unbalanced allocation. In particular, it can make a vertex  $v$  have load of one in half of the colors used so far and another vertex  $u$  have load of one in the other half. Adding an edge  $(u, v)$  would then force the algorithm to open a new color – resulting in the trivial  $2\Delta - 1$  bound. (See Section VII-C.) Guided by our lower bounds for fractional algorithms, we derive simple but crucial changes to the water-filling algorithm. First, we use an *asymmetric* approach, where we pick colors to use based only on the load on the offline vertex. Secondly, in order to bound the load on the online vertex, we cap the value of each edge-color pair. These changes yield bounded loads for the online vertex and a more balanced allocation, resulting in an *optimal* online fractional edge coloring.

*Online Rounding:* Given an  $\alpha$ -competitive fractional online algorithm, our approach would be to use its  $\alpha\Delta$  fractional matchings to obtain  $\alpha\Delta$  *integral* matchings, or colors, which leave the remaining uncolored subgraph having low maximum-degree ( $o(\Delta)$ ). Then using greedy on the remaining uncolored edges requires a further  $o(\Delta)$  colors, or  $(\alpha + o(1))\Delta$  colors overall. The question is how to compute  $\alpha\Delta$  colors based on the online fractional edge coloring. One natural way to do so is to repeatedly round these fractional matchings online, using a near-lossless online rounding scheme for fractional matchings ([11]). Unfortunately, while maximum-degree vertices stand to be matched  $\approx \Delta$  times during such a rounding stage, a constant fraction of these matches would be along previously-matched (colored) edges. To see this, note that each edge has a constant probability of being matched at least twice this way (since each edge has a constant probability of being colored more than once).

We therefore employ a more elaborate approach, repeatedly rounding *subsets* of *multiple* fractional edge colorings’ fractional matchings. Our guiding intuition is the following simple observation, that the load assigned to the edges of a maximum-degree vertex by an average fractional matching among the  $\alpha\Delta$  matchings is precisely  $1/\alpha$ . Consequently, rounding a randomly-chosen fractional matching will result in this vertex being matched with probability  $\approx 1/\alpha$ . If most such matches are along previously-unmatched(uncolored) edges, then this vertex’s degree in the uncolored graph will decrease at the appropriate rate. However, as exhibited by the previous approach, the probability of matching along an uncolored edge decreases when we use round many fractional matchings. Therefore, we only round a *subset* of fractional matchings, small enough to not decrease the probability of  $v$  being matched along an uncolored edge (due to re-matches), yet large enough to apply tail bound and argue that all high-degree vertices have their uncolored

degree decrease at a rate of  $\approx 1/\alpha$  per color used.

How to sample such a number of fractional matchings, *not too few and not too many, but just right*, for unknown  $\Delta$  is not immediate, however, as we do not even know how many fractional matchings we will use (as  $\Delta$  is unknown and keeps increasing). To address this, we rely on the fact that  $\Delta = \omega(\log n)$ , allowing us to argue that sampling each fractional matching (including matchings which are currently trivial) a priori with appropriate probability  $p$  gives the following. We choose  $p = o(1)$  (guaranteeing few recolors) which also satisfies  $\Delta \cdot p = \omega(\log n)$  (in order to have concentration up to  $(1 \pm o(1))$  factors on the number of non-trivial colors used), we color a  $\approx 1 - p$  fraction of the edges of high-degree vertices (i.e.,  $\approx \Delta \cdot p$ ) edges, while using only  $\approx \alpha\Delta \cdot p$  colors, all with high probability. We then compute *another* fractional edge coloring, but this time only on the residual uncolored subgraph. Repeatedly applying this approach (computing another edge colorings, and rounding a sampled subset of its matchings) therefore allows us to decrease the maximum degree of the uncolored graph at the required rate w.h.p. So, after  $\alpha\Delta$  colors are used this way, we safely run greedy, yielding an  $(\alpha + o(1))$ -competitive edge coloring. Plugging in the optimal algorithms for known and unknown  $\Delta$  into this rounding scheme then yields our main positive results: *optimal* randomized online edge coloring algorithms.

*Lower Bounds:* For our lower bounds (including the tight ones for bipartite graphs) we formulate linear programs capturing constraints satisfied by any  $\alpha$ -competitive fractional online algorithms (for our relaxation) when run on a tailor-made family of edge coloring instances. We then present a family of feasible solutions to the dual program, whose value converges to the claimed lower bounds, implying the lower bound on  $\alpha$  by LP duality. (See [3] for more examples of this approach.)

### C. Related Work

*Online Edge Coloring:* Several previous papers studied edge coloring in online settings [1, 4, 5, 18, 20, 21, 45, 46]. Mikkelsen [45, 46] studied the online edge coloring problem, but with advice about the future. Favrholt et al. [18, 20, 21] studied the “dual” problem of maximizing the number of edges colored using a fixed number of colors. Most relevant to our paper is the work of Motwani et al. [1, 4, 5]. Aggarwal et al. [1] presented a  $(1 + o(1))$ -competitive algorithm for multigraphs with known  $\Delta = \omega(n^2)$ . Bahmani et al. [4], inspired by the distributed algorithm of Panconesi and Srinivasan [50], gave a 1.26-competitive algorithm for multigraphs with known  $\Delta = \omega(\log n)$ . Both algorithms require *random order* edge arrivals, and fall short of the guarantees of Conjecture 2 ([5]), either in the competitive ratio or in the requirement of  $\Delta$ . In contrast, we consider vertex arrivals under the stricter *adversarial* arrival order,

for which we match these conjectured bounds for known  $\Delta$ , and also achieve optimal bounds for (harder) unknown  $\Delta$ .

*Online Matching:* As edge coloring is the problem of partitioning a graph’s edges into matchings, it is natural that our work relates to the long line of work on online matching. This problem was introduced in the seminal work of Karp et al. [39], who presented the classic RANKING algorithm, which is  $(1 - \frac{1}{e})$  competitive for bipartite graphs under one-sided arrivals, and proved its optimality. A simpler argument proves this algorithm’s optimality even among fractional algorithms (see [22]). Alternative analyses of this algorithm were given over the years ([8, 13, 17, 30]) and another optimal fractional algorithm with further applications, WATER FILLING, was given in [36]. Better bounds are known under structural assumptions [9, 11, 49], and under stochastic arrivals [23, 37, 43]. See the survey of Mehta [44] for more on this problem and its extensions. Finally, we note the recent interest in online matching in general graphs [26, 33, 34, 58]. Our complementary results of Theorem 5 for online edge coloring in general graphs are another step towards a better understanding of matching-theory-related problems in online models in general graphs.

## II. THE FRACTIONAL RELAXATION

In this section, we define the online fractional edge coloring relaxation we study and discuss several of its properties.

*The Classic Fractional Relaxation:* The classic relaxation for edge coloring has a nonnegative variable  $x_M$  for each matching  $M$  in  $G = (V, E)$ , corresponding to the (fractional) extent to which this matching is used in the solution. The objective is to minimize  $\sum_M x_M$  subject to  $\sum_{M \ni e} x_M = 1$  for each edge  $e \in E$ . This relaxation clearly lower bounds the chromatic index; i.e., the minimum number of matchings needed to cover  $G$ . A long-standing conjecture of Goldberg and Seymour is that this relaxation is at most one lower than the chromatic index [31, 53]. (See [42, Chapter 7.4] for more discussion of this relaxation.) Unfortunately, this relaxation seems somewhat unwieldy in an online setting, as we outline below.

*The Relaxation We Study:* The standard fractional edge coloring relaxation is difficult to use in online settings, where we do not know the edges which will arrive in the future, let alone which matchings  $G$  will contain. This motivates us to study a more “myopic” relaxation, which allows us to make our (fractional) assignments immediately upon an edge’s arrival (due to one of its endpoints’ arrival). Specifically, rather than relax the integrality of the extent to which we use integral matchings, we relax the integrality of the matchings used. That is, while the classic relaxation fractionally uses integral matchings to color edges, our relaxation *integrally* uses *fractional* matchings to color edges. As we will see, a useful property of this relaxation is that it allows us to rely on machinery for rounding fractional matchings online.

The edge coloring relaxation we consider is thus the following. We say a graph  $G(V, E)$  is *fractionally  $k$ -edge-colorable* if there is a feasible solution to the linear program

$$\begin{aligned} \sum_{c \in [k]} x_{e,c} &= 1 & \forall e \in E \\ \sum_{e \ni v} x_{e,c} &\leq 1 & \forall v \in V, c \in [k] \\ x_{e,c} &\geq 0 & \forall e \in E, c \in [k] \end{aligned}$$

For any graph  $G$ , the minimal number of fractional colors  $k$  is equal to  $G$ ’s maximum degree,  $\Delta$ . We note that in bipartite graphs this relaxation and the classic relaxation are equivalent in an offline sense, in that any solution to one can be transformed to a solution of equal value to the other (for general graphs, there can be a gap of one between the two, as exemplified by the triangle graph). In an online sense it is not clear how to go from one relaxation to the other, and so we will rely only on our new relaxation.

*An LP Formulation.:* For notational simplicity, rather than discuss fractional algorithms using some  $k = \alpha \cdot \Delta$  colors, we will instead use  $k = \Delta$  colors and relax the second constraint to

$$\sum_{e \ni v} x_{e,c} \leq \alpha \quad \forall v \in V, c \in [\Delta]$$

When dealing with fractional solutions, it is easy to “stretch” such a solution to obtain a feasible edge coloring (i.e., satisfying  $\sum_{e \ni v} x_{e,c} \leq 1$ ) while using  $\lceil \alpha \cdot \Delta \rceil \leq \alpha \cdot \Delta + 1$  colors, and this can be done online. Therefore, our goal will be to minimize  $\alpha$  — the competitive ratio.

*Online Algorithms for the LP Relaxation:* An online fractional edge coloring algorithm must assign  $x_{e,c}$  values for all edges  $e$  upon arrival, immediately and irrevocably. For example, if  $\Delta$  is known a priori, assigning each edge-color pair a value of  $\frac{1}{\Delta}$  trivially yields a 1-competitive online fractional algorithm. If  $\Delta$  is *unknown*, the situation is not so simple, as our lower bounds of Section V demonstrate. In the following section we present our online fractional algorithms for unknown  $\Delta$ , including an optimal algorithm for bipartite graphs.

## III. THE FRACTIONAL ONLINE ALGORITHM

Our LP relaxation asks to minimize the maximum *load* of any vertex  $u$  in color  $c$ ,  $L_u(c) \triangleq \sum_{e \ni u} x_{e,c}$ . The greedy water-filling algorithm, upon arrival of edge  $e$ , increases all  $x_{e,c}$  for all colors  $c$  minimizing the maximum load of either endpoint of  $e$ . This natural algorithm is no better than the integral greedy algorithm, however (see Section VII-A). In our algorithm, upon arrival of a vertex  $v$ , we run a variant of the water-filling algorithm on each edge  $(u, v)$  in an arbitrary order. One difference in our algorithm compared to the greedy one is that its greedy choice is *asymmetric*, and is only determined by the current loads of the previously-arrived endpoint,  $u$ . The second difference is that we set a

bound constraint of  $\beta/\Delta$  for each color per edge, where  $\Delta$  is the **current** maximal degree, and  $\beta$  is a parameter of the algorithm which will be determined later. The bound constraints result in bounded load trivially for the online vertex, and by careful analysis, also for the offline vertex. In addition, the bound constraints result in a more balanced allocation, which uses more colors for each edge, but fewer colors overall. A formal description of our algorithm is given in Algorithm 1. Our algorithm is described as a continuous process, but can be discretized easily.

---

**Algorithm 1** Bounded Water Filling

---

**Input:** Online graph  $G(V, E)$  with unknown maximum degree  $\Delta(G)$  under vertex arrivals, parameter  $\beta \in (1, 2)$ .

**Output:** Fractional edge coloring  $\{x_{e,c} \mid e \in E, c \in [\Delta(G)]\}$ .

```

1: (Implicitly)  $x_{e,c} \leftarrow 0$  for all  $e \in E, c \in \mathbb{N}$ .
2: for each arrival of a vertex  $v$  do
3:    $\Delta \leftarrow \max\{\text{current } d(u) \mid u \in V\}$ .
   /*  $\Delta = \text{current max. degree}$  */
4:   for each  $e = (u, v) \in E$  do
5:     while  $\sum_{c \in [\Delta]} x_{e,c} < 1$  do
6:       let  $U := \{c \in [\Delta] \mid x_{e,c} < \beta/\Delta\}$ .
       /* “unsaturated” colors for  $e$  */
7:       let  $C := \{c \in U \mid L_u(c) = \min_{c \in U} L_u(c)\}$ .
       /* “currently active” colors for  $e$  */
8:       for all  $c \in C$  do
9:         increase  $x_{e,c}$  continuously.
       /* update  $L_u(c), L_v(c), U$  and  $C$  */

```

---

*A. Basic properties of the algorithm*

Our water filling algorithm preserves important monotonicity properties on the loads of any previously-arrived vertex  $v$ . In particular, the order obtained by sorting colors by their loads for  $v$  remains invariant following its future neighbors’ arrivals. More formally, for each vertex  $v$ , we define an order permutation  $\sigma_v : Z^+ \rightarrow Z^+$ , where  $\sigma_v(i)$  is the index of  $v$ ’s  $i^{\text{th}}$  most loaded color index after the vertex  $v$  arrives and its edges are fractionally colored (e.g.,  $\sigma_v(1)$  is the most-loaded color index). In addition, we define the load of a color in a vertex with respect to this order; i.e., we denote by  $\ell_u^t(i)$  the load of color  $\sigma_u(i)$  for vertex  $u$  after its  $t^{\text{th}}$  neighbor arrives – which we refer to as *step*  $t$ . In this notation, our monotonicity property will be that  $\ell_u^t(i) \geq \ell_u^t(i+1)$  for each  $u$  and  $i, t \in \mathbb{Z}^+$ .

We denote by  $\delta_u^t$  the global maximum degree after the arrival of the  $t^{\text{th}}$  neighbor of vertex  $u$  and denote by  $A_u$  the degree of  $u$  when it arrives (e.g.,  $A_u = 0$  for offline vertices in bipartite graphs). Next, we prove properties of the load of a specific vertex  $u$  after its arrival (i.e., for steps  $t > A_u$ ), at which point the order  $\sigma_u$  is already set. For ease of notation we omit the subscript  $u$  from variables  $\ell, \delta$  and  $A$  whenever it will be clear from context (i.e., when

considering a single vertex  $u$ ). In addition, as  $\sigma$  will be clear from context, we will use *color*  $k$  as shorthand notation to  $\sigma(k)$ . Moreover, due to space constraints, we defer most proofs to Section VIII.

We first observe that for our bounded water-filling algorithm (as for its unbounded counterpart), the load of  $u$  is monotone decreasing with respect to the  $\sigma_u$  order, and for each step  $t$ , the increase in the load for  $i \leq \delta^t$  is monotone increasing in the  $\sigma_u$  order.

**Observation 6.** For all color indices  $i$ , and any  $t > A$ ,

- $\ell^t(i) \geq \ell^t(i+1)$ .
- $\ell^t(i) - \ell^{t-1}(i) \geq \ell^t(i-1) - \ell^{t-1}(i-1)$ , for all  $i \leq \delta^t$ .

In our analysis, we focus on the *critical* colors at step  $T$  – colors whose load increased at step  $T$  and is higher than the following color load. Formally, color  $k$  is *critical* with respect to vertex  $u$  and its  $T^{\text{th}}$  neighbor if  $\ell^T(k) > \ell^{T-1}(k)$  and  $\ell^T(k) > \ell^T(k+1)$ . Clearly, in order to upper bound the load at step  $T$ , it is sufficient to upper bound the load for critical colors  $k$  for  $T$ . If we let  $V_1^k \triangleq \sum_{i=1}^k \ell^T(i)$  be the total load on colors  $1, 2, \dots, k$  and  $V_2^k \triangleq \sum_{i=k+1}^{\delta^T} \ell^T(i)$  be the total load on colors  $k+1, \dots, \delta^T$ , we will upper bound the load of color  $k$  by

$$\ell^T(k) \leq \frac{V_1^k}{k} \leq \frac{\delta^T - V_2^k}{k}, \quad (1)$$

where the first inequality is due to the monotonicity of the loads, and the second inequality is due to the total load being at most  $\delta^T$ . Therefore, we will upper bound the load by proving a lower bound on the index of any critical color, and a lower bound on the total load after this index.

The next lemma plays a key role in both lower bounds. We show that for any color  $k$  critical at step  $T$  and for all steps  $A < t \leq T$  during which  $k$ ’s load increases, all colors after  $k$  that could be increased (i.e.  $k < i \leq \delta^t$ ) have their load increase by the maximum allowable amount,  $\beta/\delta^t$ .

**Lemma 7.** For a color  $k$  critical at step  $T$ , for all  $A < t \leq T$  such that  $\ell^t(k) > \ell^{t-1}(k)$ , we have

$$\ell^t(i) - \ell^{t-1}(i) = \beta/\delta^t \quad \forall k < i \leq \delta^t.$$

Using the previous lemma, we bound the minimal index of a critical color at step  $T$ .

**Lemma 8.** If  $k$  is a critical color at step  $T$ , then  $k > \delta^T \cdot (1 - 1/\beta)$ .

Next, using Lemma 7 and some useful claims in Section VIII we prove a lower bound on  $V_2^k$ .

**Lemma 9.** If  $k$  is a critical color at step  $T$  and  $k^* \geq$

$\max\{k, \delta^A\}$ , then

$$\begin{aligned} V_2^k &\geq \sum_{j=k+1}^{\delta^T} \left( \ell^T(j) - \ell^{k^*}(j) \right) \\ &\geq \beta \cdot \left( \delta^T - k^* - k \log \frac{\delta^T}{k^*} \right). \end{aligned}$$

*Bounding the maximum load:* Next, we use the previous lemmas in order to bound the maximum load after an assignment of an edge. Specifically, we will bound the load of  $\ell_u$  and  $\ell_v$  after coloring the edge  $(v, u)$ , where  $v$  is the newly-arrived vertex. First, it is easy to bound the load of a vertex  $v$  for each color after its arrival, since we bound each edge-color pair's value  $x_{e,c}$  by  $\beta/\delta_v^{A_v} \leq \beta/A_v$  at arrival of  $v$  (when it has  $A_v$  neighbors).

**Observation 10.**  $\ell_v^{A_v}(i) \leq \beta$  for all  $i \in [\delta^{A_v}]$ .

We next use Lemma 9 and Equation (1) to bound the load of previously-arrived vertex  $u$ .

**Lemma 11.** If  $k > \delta_u^{A_u}$  is a critical color at step  $T$  with respect to  $u$ , then  $\ell_u^T(k) \leq \beta \log \frac{\beta}{\beta-1}$ .

**Lemma 12.** If  $k \leq \delta_u^{A_u}$  is a critical color at step  $T$  with respect to  $u$ , then  $\ell_u^T(k) \leq \beta^2 - \beta + \beta \log \frac{1}{\beta-1}$ .

*Upper Bounding Algorithm 1's Competitive Ratio:* We are now ready to bound the competitive ratio of Algorithm 1. First, we show that Algorithm 1 is  $\frac{e}{e-1}$  competitive for one-sided bipartite graphs. That is,  $G(L, R, E)$  is a bipartite graph and the offline vertices  $L$  arrive before the algorithm starts (i.e.,  $A_u = 0$  for all  $u \in L$ ).

**Theorem 13.** For bipartite graphs under one-sided arrivals, Algorithm 1 is  $\max\{\beta, \beta \log \frac{\beta}{\beta-1}\}$  competitive. Setting  $\beta = \frac{e}{e-1}$ , we obtain an  $(\frac{e}{e-1})$ -competitive algorithm.

*Proof:* We bound the load after coloring of edge  $(v, u)$ , where  $v \in R$  is the  $T^{\text{th}}$  online neighbor of  $u$ . First, we bound the load for any color  $i$  of  $v$ . By Observation 10, we have  $\ell_v(i) = \ell_v^{A_v}(i) \leq \beta$ . For vertex  $u$ , we have  $A_u = \delta^{A_u} = 0$ . Thus, by Lemma 11 we have that  $\max_i \ell_u^T(i) \leq \beta \log \frac{\beta}{\beta-1}$ . ■

Finally, in Section VIII we bound our algorithm's competitive ratio on general graphs, proving that it is better than greedy.

**Theorem 14.** For any graph, Algorithm 1 is  $\beta^2 - \beta + \beta \log \frac{1}{\beta-1}$  competitive. Setting  $\beta = 1.586$ , we obtain a 1.777-competitive algorithm.

#### IV. ONLINE ROUNDING OF FRACTIONAL EDGE COLORING

In this section we show how to round fractional edge-coloring algorithms' output online. Specifically, we will round fractional edge colorings provided by algorithms

which assign at most some (small) value  $\epsilon$  to each edge-color pair, which we refer to as  $\epsilon$ -bounded algorithms. (As we shall see, the optimal fractional algorithms we will plug into this rounding scheme both satisfy this property.) We now state our main technical result of this section: a nearly-lossless rounding process for bounded algorithms on graphs with high enough lower bound on  $\Delta$ .

**Theorem 15.** For all  $\alpha \in [1, 2]$  and  $\epsilon \leq 1$ , if there exists an  $\epsilon$ -bounded  $\alpha$ -competitive fractional algorithm  $\mathcal{A}$  for bipartite graphs with unknown maximum degree  $\Delta \geq \Delta' \geq 2/\epsilon$ , then there exists a randomized integral algorithm  $\mathcal{A}'$  which is  $(\alpha + O(\frac{1}{\sqrt[12]{(\log n)/\Delta'}}))$ -competitive w.h.p on bipartite graphs of unknown maximum degree  $\Delta \geq \Delta' \geq c \cdot \log n$  for some constant  $c$ .

In the end of the section we show how to use this theorem to obtain a  $(1 + o(1))$ -competitive for known  $\Delta$ . For now, we note that plugging in our optimal fractional algorithm for unknown  $\Delta$  into Theorem 15,<sup>1</sup> we get an optimal randomized algorithm for edge coloring graphs with unknown  $\Delta$ .

**Theorem 16.** There exists an  $(\frac{e}{e-1} + O(\frac{1}{\sqrt[12]{(\log n)/\Delta'}}))$ -competitive algorithm for  $n$ -vertex bipartite graphs  $G$  with unknown maximum degree  $\Delta \geq \Delta' \geq c \cdot \log n$  for some absolute constant  $c$ .

**Remark.** The algorithm of Theorem 16 requires only a lower bound  $\Delta' \leq \Delta$  for some  $\Delta' = \omega(\log n)$  in order to output an  $(\frac{e}{e-1} + o(1)) \cdot \Delta$  coloring, and not the exact value of  $\Delta$ . Alternatively, our algorithm uses  $(\frac{e}{e-1} + o(1)) \cdot \max\{\Delta, \Delta'\}$  colors for any unknown  $\Delta$ , where the multiplicative approximation ratio is clearly only worse than  $(\frac{e}{e-1} + o(1))$  for small  $\Delta < \Delta'$  – in which case the additive approximation term is only  $O(\Delta')$ . This result can therefore be read as an asymptotic approximation scheme, trading off between the additive term and the asymptotic competitive ratio.

To describe our rounding scheme, we need the following online rounding scheme of bounded fractional matchings, which motivates our study of bounded fractional edge colorings.

**Lemma 17** (Per-Edge Guarantees [11]). For all  $\epsilon \in [0, 1]$ , there exists an online dependent rounding algorithm, MARKING, which if presented online with a feasible fractional bipartite matching  $\vec{x}$  with an (a priori) guarantee  $\max_e x_e \leq \epsilon$ , outputs a matching  $\mathcal{M}$  which matches each edge  $e$  with probability

$$x_e \cdot \left( 1 - 11 \sqrt[3]{\epsilon \cdot \log(1/\epsilon)} \right) \leq \Pr[e \in \mathcal{M}] \leq x_e.$$

<sup>1</sup>Strictly speaking, our optimal fractional algorithm, Algorithm 1, is not  $2/\Delta'$  bounded. However, setting our initial lower bound on  $\Delta$  to be  $\Delta'$  in Line 3 yields a  $2/\Delta'$ -bounded solution without worsening the competitive ratio. (This is equivalent to adding a dummy star which does not increase the maximum degree.)

We now outline our rounding scheme, which consists of phases, as follows. For each phase  $i$ , let  $U_i$  be the uncolored graph at start of phase  $i$ . (Initially,  $U_1 = G$ .) We compute an  $\alpha$ -competitive fractional edge coloring in  $U_i$  online. Upon the algorithm's initialization, we sample each of the possible  $\alpha \cdot n$  fractional matchings of this fractional coloring, i.i.d with probability  $p$ . We then round and color the sampled fractional matchings in an online fashion, as follows. Whenever a *sampled* fractional matching becomes non trivial, we assign it a new color. Whenever a new vertex  $v$  arrives, for each phase  $i$  in increasing order, we run the next step of MARKING for each of the sampled fractional matchings of phase  $i$ 's fractional coloring, and color all newly-matched edges with the color assigned to the relevant fractional matching. Finally, we greedily color the remaining uncolored edges of  $v$ . Setting  $p = o(1)$  (guaranteeing few re-colors) and also satisfying  $\Delta \cdot p = \omega(\log n)$  (in order to have concentration up to  $(1 \pm o(1))$  factors on number of colors used), this approach will use roughly  $p \cdot \alpha \cdot \Delta(U_i)$  colors for the  $i^{\text{th}}$  phase, while decreasing the uncolored subgraph's maximum degree by roughly  $p \cdot \Delta(U_i)$ , or a  $(1 - p)$  factor. Thus, using  $(1/p) \log(1/p)$  phases yield an uncolored subgraph of maximum degree  $p \cdot \Delta$  (using  $\alpha \cdot \Delta$  colors), which the greedy algorithm colors using  $2p \cdot \Delta$  new colors. This implies Theorem 15.

#### A. Our Online Rounding Scheme

Our online rounding scheme, given an  $\epsilon$ -bounded fractional edge-coloring algorithm  $\mathcal{A}$  which is  $\alpha$  competitive on graphs of maximum degree at least  $2/\epsilon$ , for  $\epsilon = p^4/(12 \log n)$ , works as follows. Let  $p \triangleq \sqrt[12]{24(\log n)/\Delta'}$ . We use  $P \triangleq \lceil (4/p) \log(1/p) \rceil$  many phases. For phase  $i$ , we sample in advance a subset  $\mathcal{S}_i$  of all possible color indices, each taken into  $\mathcal{S}_i$  with probability  $p$ . Let  $U_i$  be the subgraph of edges not colored before phase  $i$ . When online vertex  $v$  arrives, for each phase  $i \in [P]$ , we update a fractional coloring  $x^{(i)}$  using Algorithm  $\mathcal{A}$ , based on  $v$ 's arrival in  $U_i$ . For all sampled  $j \in \mathcal{S}_i$  for which  $x_j^{(i)}$  (the  $j^{\text{th}}$  fractional matching of  $x^{(i)}$ ) is non trivial, we use a distinct color  $c_{i,j}$  to color edges of a matching  $M_{i,j}$  computed online by running MARKING on  $x_j^{(i)}$ . Finally, all remaining uncolored edges of  $v$  are greedily colored using new colors. This is Algorithm 2, below.

#### B. Analysis

We will study changes in the uncolored graph between subsequent phases and the colors used during the phases. For each  $i$ , let  $\Delta_i \triangleq \Delta(U_i)$  be the maximum degree of the online graph not colored by phase  $1, 2, \dots, i-1$ . In this section we will show that during each phase  $i$ , provided  $\Delta_i$  is sufficiently large, Algorithm 2 uses some  $\alpha \cdot \Delta_i \cdot p(1+O(p))$  new colors w.h.p., and obtain an uncolored subgraph  $U_{i+1}$  of maximum degree  $\Delta_{i+1} = \Delta_i \cdot (1-p \pm O(p^2))$  w.h.p. This will imply a degree decrease at a rate of one per  $\alpha + O(p)$

---

#### Algorithm 2 Randomized Edge Coloring for Unknown $\Delta$

---

**Input:** Online  $n$ -vertex bipartite graph  $G(L, R, E)$  with  $\Delta \geq \Delta' \geq c \cdot \log n$ , for  $c$  a constant TBD.

Parameter  $p \triangleq \sqrt[12]{(24 \log n)/\Delta'} (\leq 1/10)$ .

An  $\epsilon$ -bounded fractional online edge-coloring algorithm  $\mathcal{A}$  which is  $\alpha$  competitive on graphs  $U$  with  $\Delta(U) \geq 2/\epsilon$ , for  $\epsilon \triangleq (p^4/12 \log n)$ .

**Output:** Integral  $(\alpha + O(p)) \cdot \Delta$  edge coloring, w.h.p.

- 1: for all  $i$ , set  $\mathcal{S}_i \subseteq [\alpha \cdot n]$  to be such that each  $j \in [\alpha \cdot n]$  is in  $\mathcal{S}_i$  independently with probability  $p$ .
  - 2: for all  $i$ , denote by  $U_i$  the **online** subgraph of  $G$  not colored during phases  $1, 2, \dots, i-1$ .
  - 3: **for** each arrival of a vertex  $v \in R$  **do**
  - 4:     **for** phase  $i = 1, 2, \dots, \lceil (4/p) \log(1/p) \rceil$  **do**
  - 5:          $x^{(i)} \leftarrow$  output of Algorithm  $\mathcal{A}$  on **current**  $U_i$ .  
*/\* run next step of  $\mathcal{A}$  \*/*
  - 6:         **for**  $j \in \mathcal{S}_i$  with  $x_j^{(i)} \neq \vec{0}$  **do**
  - 7:             **if**  $c_{i,j}$  not set **then**
  - 8:                 set  $c_{i,j}$  to next unassigned color index.
  - 9:              $M_{i,j} \leftarrow$  output of MARKING on **current**  $x_j^{(i)}$ .  
*/\* run next step of MARKING \*/*
  - 10:             **if** some  $e \in M_{i,j}$  previously uncolored **then**
  - 11:                 color  $e$  using color  $c_{i,j}$ .  
*/\* note:  $e \ni v$  \*/*
  - 12:     run greedy on uncolored edges of  $v$ , using colors not assigned during the phases.
- 

colors used. Repeating this for  $\lceil (4/p) \log(1/p) \rceil$  phases, will therefore require  $(\alpha + O(p))\Delta$  colors and yield a subgraph of maximum degree  $p \cdot \Delta$ , which we color greedily with  $O(p)\Delta$  new colors, implying Theorem 15.

To upper bound the number of colors used in phase  $i$ , we note that the number of non-trivial (i.e., not identically zero) fractional matchings we round in each iteration is clearly a  $p$ -fraction of the (at most  $\lceil \alpha \cdot \Delta_i \rceil$ ) non-trivial colors of  $x^{(i)}$ . Therefore, by standard Chernoff bounds (Lemma 45), if  $\Delta_i$  is large enough, the number of colors in the phase is small, w.h.p.

**Lemma 18.** *If  $\Delta_i \geq (6 \log n)/p^3$ , then  $C_i$ , the number of colors used in phase  $i$ , satisfies*

$$\Pr[C_i \geq \alpha \Delta_i \cdot p \cdot (1+p)] \leq \frac{1}{n^2}.$$

Lemma 18 upper bounds the number of colors used in phase  $i$  by  $\alpha \Delta_i \cdot p \cdot (1+p)$ . Our main technical lemma, below, whose full proof is deferred to Section IX, asserts that these colors result in a decrease of roughly  $\Delta_i \cdot p$  in the uncolored subgraph's maximum degree during the phase.

**Lemma 19.** *If  $\Delta_i \geq (24 \log n)/p^4$ , then*

- 1)  $\Pr[\Delta_{i+1} \leq \Delta_i \cdot (1-p-4p^2)] \leq 3/n^3$ .
- 2)  $\Pr[\Delta_{i+1} \geq \Delta_i \cdot (1-p+7p^2)] \leq 6/n^2$ .

*Proof Sketch:* Let  $v$  be a vertex of degree  $d_i(v) \geq \Delta_i/2$  in  $U_i$ . By Lemma 17 and the  $\epsilon$ -boundedness of the fractional algorithm  $\mathcal{A}$  (and some simple calculations), each edge  $e \in U_i$  is matched in  $M_{i,j}$  ( $j \in \mathcal{S}_i$ ) with probability  $x_{e,j}^{(i)} \cdot (1 - O(p)) \leq \Pr[e \in M_{i,j}] \leq x_{e,j}^{(i)}$ . That is, we match  $e$  in  $M_{i,j}$  with probability close to its sampled “load” for this color. By Chernoff bounds, as we sample each color of  $x^{(i)}$  with probability  $p$ , the sampled load on  $v$ ’s edges is  $d_i(v) \cdot p(1 \pm O(p))$  w.h.p. So, by linearity and another Chernoff bound, the number of times  $v$  is matched during the  $i^{\text{th}}$  phase satisfies  $M_v \leq d_i(v) \cdot p(1 + O(p))^2 \leq d_i(v) \cdot p(1 + O(p))$ , and  $M_v \geq d_i(v) \cdot p(1 - O(p))^3 \geq d_i(v) \cdot p(1 - O(p))$ .

However,  $M_v$  also counts repeated matchings of edges of  $v$ , which do not contribute to  $v$ ’s degree decrease in the uncolored subgraph. We therefore want to bound  $R_v$  – the number of times a previously-colored edge of  $v$  is matched during the phase. By Chernoff’s bound and  $\epsilon$ -boundedness of the fractional algorithm, the load on each edge in the sampled colors  $\mathcal{S}_i$ , which in expectation is precisely  $p$ , is  $O(p)$  w.h.p. So, intuitively, we would expect  $R_v = \Theta(p) \cdot M_v$  w.h.p., implying  $R_v = \Theta(d_i(v) \cdot p^2)$  w.h.p. Of course, as re-matches are not independent of matches, we cannot simply multiply these expressions this way. However, relying on the theory of negative association (see Section XIII-A), the intuitive claim that  $R_v = \Theta(d_i(v) \cdot p^2)$  w.h.p. can be formalized. We conclude that the degree decrease of vertex  $v$  in the uncolored graph during the  $i^{\text{th}}$  phase is  $M_v - R_v = d_i(v) \cdot p \cdot (1 - \Theta(p))$  w.h.p. Taking union bound over all vertices  $v$ , the lemma follows. ■

Theorem 15 now follows from Lemma 18 and Lemma 19. We sketch a proof of this theorem and defer its full proof to Section IX.

*Proof of Theorem 15 (Sketch):* Clearly, Algorithm 2 colors all edges of  $G$ , due to Line 12. By definition, all color classes computed are matchings. As we shall show, the number of colors used during the phases is at most  $(\alpha + O(p)) \cdot \Delta$  w.h.p., and the greedy algorithm requires some  $O(p) \cdot \Delta$  colors w.h.p., implying our claimed result. We outline this proof using a stronger claim than Lemma 19.

Suppose instead of Lemma 19 we had that with high probability  $\Delta_{i+1} = \Delta_i \cdot (1 - p)$ . Then, by induction we would have  $\Delta_i = \Delta \cdot (1 - p)^i$  and in particular for all  $i \leq (1/p) \log(1/p)$  we would have  $\Delta_i \geq \Delta \cdot p \geq \Delta' \cdot p$ . Taking  $p \geq \sqrt[5]{(24 \log n)/\Delta'}$  would therefore imply that  $\Delta_i \geq \Delta' \cdot p \geq (24 \log n)/p^4$ , which in turn would allow us to appeal to union bound to prove that  $\Delta_i = \Delta \cdot (1 - p)^i$  for all  $i$ , or in other words  $\Delta_i - \Delta_{i+1} = \Delta_i \cdot p$ , and that the number of colors used in each phase  $i$  is at most  $C_i \leq \alpha \cdot \Delta_i \cdot p \cdot (1 + p)$ . Summing over all phases, this would imply that w.h.p., the

number of colors used during the phases is

$$\begin{aligned} \sum_i C_i &\leq \sum_i (\alpha + p(1 + p)) \cdot (\Delta_i - \Delta_{i+1}) \\ &\leq (\alpha + p(1 + p)) \cdot \Delta_0 \\ &= (\alpha + p(1 + p)) \cdot \Delta. \end{aligned}$$

On the other hand, after  $(1/p) \log(1/p)$  phases we would get a final uncolored subgraph of maximum degree  $\Delta \cdot (1 - p)^{(1/p) \log(1/p)} \approx \Delta \cdot p$  w.h.p., and so the greedy step of Line 12 would use at most  $2\Delta \cdot p$  colors. Overall, Algorithm 2 therefore uses at most  $(\alpha + O(p)) \cdot \Delta$  colors for  $p = O(\sqrt[5]{(\log n)/\Delta'})$  and  $\Delta \geq 24 \log n$ . Our more involved bounds are due to the slightly looser bounds for  $\Delta_{i+1}$  in terms of  $\Delta_i$  in Lemma 19. See full proof in Section IX for details. ■

*Applications to Known  $\Delta$ :* Algorithm 2 finds applications for known  $\Delta$ , too. In particular, by Lemma 19 we find that if in each phase  $i$  we assign value  $1/((1 - p + 7p^2)^i \cdot \Delta)$  for each edge-color pair, then we obtain a feasible coloring w.h.p., requiring  $(1 - p + 7p^2)^i \cdot \Delta$  colors when the maximum degree is at least  $(1 - p - 4p^2)^i \cdot \Delta$ , w.h.p.; i.e., this is a  $(1 + O(p^2))$ -competitive fractional algorithm for uncolored subgraph  $U_i$ . Replacing algorithm  $\mathcal{A}$  in Algorithm 2 with this approach then yields, as in the proof of Theorem 15, an optimal randomized algorithm for known  $\Delta$ .

**Theorem 20.** *There exists a  $(1 + O(\sqrt[12]{(\log n)/\Delta}))$ -competitive algorithm for  $n$ -vertex bipartite graphs  $G$  with known maximum degree  $\Delta \geq c \cdot \log n$  for some absolute constant  $c$ .*

In this section we provided optimal online edge coloring algorithms for known and unknown  $\Delta$ . In Section X we improve the  $o(1)$  term in the  $1 + o(1)$  competitive ratio for known  $\Delta$ . In the following section we present our lower bounds for known and unknown  $\Delta$ , proving the optimality of our fractional and randomized algorithms, up to  $o(1)$  terms.

## V. LOWER BOUNDS

In this section we present our lower bounds for online edge coloring. We start with by noting that for known  $\Delta$ , the competitive ratio of  $(1 + o(1))$  we obtain is optimal (up to the exact  $o(1)$  term).<sup>2</sup>

**Observation 21.** *No randomized online edge coloring algorithm is  $(1 + o(1/\sqrt{\Delta}))$ -competitive.*

*Proof:* By [11], no online matching algorithm outputs a matching of expected size  $(1 - o(1/\sqrt{\Delta})) \cdot n$  in  $\Delta$ -regular  $2n$ -vertex bipartite graphs under one-sided arrivals. Given a  $(1 + \epsilon)$ -competitive edge coloring algorithm, we can randomly pick one of the  $(1 + \epsilon) \cdot \Delta$  color classes upon initialization and output that as our matching. For  $\Delta$ -regular graphs on  $2n$

<sup>2</sup>A similar argument implies that  $1 + o(1)$  competitiveness is impossible on arbitrary multigraphs. See Section XII.

vertices, which have  $\Delta \cdot n$  edges, this results in a matching of expected size  $\frac{\Delta \cdot n}{(1+\epsilon) \cdot \Delta} = (1 - O(\epsilon)) \cdot n$ , from which we conclude  $\epsilon = \Omega(1/\sqrt{\Delta})$ . ■

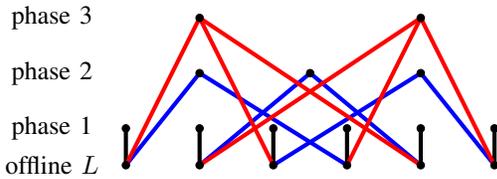
Our main result of the section is a lower bound for unknown  $\Delta$  of  $\frac{e}{e-1}$  on the competitive ratio of any fractional online algorithm for our relaxation (and by extension, for any randomized algorithm). To obtain this result, we derive linear constraints that any  $\alpha$ -competitive fractional online algorithm must satisfy and formulate these constraints as a family of linear programs. Specifically, we will rely on the modified fractional edge coloring formulation, where the competitive ratio  $\alpha \triangleq \max_{v,c} \sum_{e \ni v} x_{e,c}$  is the maximum load of any vertex  $v$  for color  $c$ , and  $x_{e,c} \geq 0$  for all  $c \in [\Delta]$  and  $x_{e,c} = 0$  for all  $c > \Delta$ , for  $\Delta$  the **current** maximum degree. (See Section II.) We then construct feasible dual solutions to these LPs, which by LP duality imply our claimed lower bounds.

#### A. Matching Lower Bound for Bipartite Graphs

Our first lower bound concerns fractionally edge coloring bipartite graphs.

**Theorem 22.** *No fractional online edge coloring algorithm is better than  $\frac{e}{e-1}$  competitive on bipartite graphs under one-sided arrivals.*

*Proof:* Consider the following construction. For any  $m$ , we construct a bipartite graph  $G_m = (L_m, R_m, E_m)$ , where  $L_m$  is the offline side and  $R_m$  is the online side. The offline side,  $L_m$ , contains  $m!$  vertices, denoted by  $v_1, \dots, v_{m!}$ . The online side,  $R_m$ , arrives over  $m$  phases. In phase  $k$  ( $k \in [m]$ ), some  $m!/k$  vertices of degree  $k$  arrive. Each vertex  $u_i$  which arrives in phase  $k$  ( $i \in [m!/k]$ ) neighbors offline vertices  $v_i, v_{m!/k+i}, \dots, v_{m!(k-1)/k+i}$ . We can see that each offline vertex has exactly one more neighbor in phase  $k$  and the maximum degree in phase  $k$  is exactly  $k$ . See Figure 1 for an illustrative example. The algorithm will have to be  $\alpha$  competitive after each phase, as the adversarial sequence can “terminate early”, after essentially presenting disjoint copies of  $G_{m'}$  for some  $m' \leq m$ .



**Figure 1:** The hard instance for bipartite graphs for  $m = 3$ .

We use  $x_{kj} \triangleq \frac{\sum_{e \in \text{phase } k} x_{e,j}}{|\{e \in \text{phase } k\}|}$  to denote the average assignment of color  $j$  to edges of phase  $k$ .

The average load for online vertices of phase  $k$  for color  $j$  is  $k \cdot x_{kj}$ , as each such online vertex has  $k$  edges. Consequently, as their average load is at most  $\alpha$ , we have

the following constraints.

$$k \cdot x_{kj} \leq \alpha \quad \forall 1 \leq j \leq k. \quad (2)$$

Moreover, since each offline vertex has one more edge during phase  $k$ , the average assignment to all edges should cover all edges of phase  $k$ , implying the following constraint.

$$\sum_{j=1}^k x_{kj} \geq 1 \quad \forall k. \quad (3)$$

Finally, as the load of all offline vertices (which have only one edge in phase  $k$ ) for any color  $j$  cannot exceed  $\alpha$  (and so neither can their average), we have the following constraint.

$$\sum_{k=j}^m x_{kj} \leq \alpha \quad \forall j. \quad (4)$$

Combining constraints (2)-(4), yields the following linear program  $\text{LP}_m$ , which lower bounds  $\alpha$ .

$$\begin{aligned} \text{LP}_m \triangleq \min \alpha \\ \sum_{j=1}^k x_{kj} \geq 1 & \quad 1 \leq k \leq m \\ k \cdot x_{kj} \leq \alpha & \quad 1 \leq j \leq k \leq m \\ \sum_{k=j}^m x_{kj} \leq \alpha & \quad 1 \leq j \leq m \\ x_{kj} \geq 0 & \quad 1 \leq j \leq k \leq m. \end{aligned}$$

We construct a series of dual feasible solutions to lower bound  $\alpha$ . First, the dual LP is as follows.

$$\begin{aligned} \max \sum_{k=1}^m y_k \\ \sum_{k=1}^m \sum_{j=1}^k z_{kj} + \sum_{j=1}^m w_j \leq 1 \\ -k \cdot z_{kj} - w_j + y_k \leq 0 & \quad 1 \leq j \leq k \leq m \\ y_k, w_j, z_{kj} \geq 0 & \quad 1 \leq j \leq k \leq m. \end{aligned}$$

Let  $c(m) \triangleq \lfloor m/e \rfloor$ . We know that  $\lim_{m \rightarrow \infty} c(m)/m \rightarrow 1/e$ . Let  $t \triangleq 1/(m+1+c(m) \cdot (H_{c(m)} - H_m))$ , where  $H_k \triangleq \sum_{i=1}^k 1/i$  satisfies  $\lim_{m \rightarrow \infty} H_{c(m)} - H_m \rightarrow \log(c(m)/m) \rightarrow -1$ . We construct a feasible dual solution as follows: We let  $y_1 = \dots = y_m = t$ , and

$$\begin{aligned} w_j &= \begin{cases} t & 1 \leq j \leq c(m) \\ 0 & \text{otherwise} \end{cases} \\ z_{kj} &= \begin{cases} t/k & c(m) + 1 \leq j \leq k \leq m \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

For any  $1 \leq j \leq k \leq m$ , we have that  $k \cdot z_{kj} + w_j = t = y_k$ . For the first dual constraint, we have

$$\begin{aligned} & \sum_{k=1}^m w_k + \sum_{k=1}^m \sum_{j=1}^k z_{kj} \\ &= c(m) \cdot t + \sum_{k=c(m)}^m \left( \frac{k - c(m)}{k} \right) \cdot t \\ &= c(m) \cdot t + (m - c(m) + 1) \cdot t - c(m) \cdot t \cdot (H_m - H_{c(m)}) \\ &= (m + 1 + c(m) \cdot (H_{c(m)} - H_m)) \cdot t = 1. \end{aligned}$$

The above is therefore a feasible dual solution, of value

$$\begin{aligned} \sum_{i=k}^m y_k &= m \cdot t = \frac{m}{m + c(m) \cdot (H_{c(m)} - H_m)} \\ &= \frac{1}{1 + \frac{c(m)}{m} \cdot (H_{c(m)} - H_m)}. \end{aligned}$$

When  $m \rightarrow \infty$ , this tends to  $\frac{1}{1-1/e} = \frac{e}{e-1}$ . Consequently,  $\lim_{m \rightarrow \infty} \text{LP}_m \geq e/(e-1)$ , implying our claimed lower bound for fractional online edge coloring of bipartite graphs.  $\blacksquare$

*Making the Graph Dense:* The above construction yields a sparse graph, as the number of vertices in this graph,  $n = m! + m!(1 + \frac{1}{2} + \dots + \frac{1}{m}) \approx m! \log m$ , is exponential in its maximum degree,  $m$ . However, the following change yields a dense graph where the same lower bound still holds. Fix any integer  $t > 0$ , in the hard instance, we replace each vertex with  $t$  identical copies, and correspondingly, connecting all copies of pairs  $(u, v)$  which are adjacent in the sparse graph. The obtained graph is still bipartite and the maximum degree and the number of vertices both increase by a factor of  $t$ , to  $t \cdot m$  and  $t \cdot m! \log m$ , respectively. Since we can take  $t$  to be arbitrarily large, the graph has maximum degree as high as  $\Omega(n)$ . In order to show that the lower bound still holds, we only need to slightly change the meaning of  $x_{kj}$  to be the average assignment of colors  $(j-1)t + 1, (j-1)t + 2, \dots, jt$  during phase  $k$ . Constraints (2)-(4) still hold with this new meaning in the denser graph. Thus, we conclude that Theorem 22 holds for graphs of arbitrarily high degree.

### B. Lower Bound for General Graphs

Next, we present a lower bound for general graphs. The lower bound is based on the construction for bipartite graphs, but with more alterations. More specifically, recall that in the construction for bipartite graphs, when the online vertices of phase  $k$  arrive, we always connect them to  $k$  offline vertices. However, in general graphs, we have more freedom. In phase  $k$ , there can be two possible futures: in one we continue the sequence for bipartite graphs; in the other we connect all vertices which arrive during phases  $k, k+1, \dots$  to the vertices which arrived in phase  $k-1$ . This example yields a lower bound of 1.606, showing a separation between

bipartite and general graphs.<sup>3</sup> We state this lower bound here and defer its proof to Section XI.

**Theorem 23.** *No fractional online edge coloring algorithm is better than 1.606 competitive in general graphs.*

## VI. CONCLUSION AND OPEN QUESTIONS

In this paper we present optimal online edge coloring algorithms in bipartite graphs under one-sided vertex arrivals, both when the maximum degree is known and when it is not. This work suggests a few follow up questions, most prominent of which is to obtain optimal online edge coloring algorithms under vertex arrivals, or even under edge arrivals. Bar-Noy et al. [5] suggested a candidate algorithm for edge arrivals with known  $\Delta$ , though this algorithm seems challenging to analyze. Is their candidate algorithm  $(1 + o(1))$  competitive? For unknown  $\Delta$ , the problem seems much more challenging, even if one restricts oneself to fractional algorithms.

For vertex arrivals in general graphs we provided a better-than-greedy fractional algorithm. But can this algorithm be rounded without much loss? We note that our online rounding approach of Algorithm 2 works under vertex arrivals in general graphs too, though it requires an online dependent rounding scheme for fractional matching in general graphs with guarantees similar to those of Lemma 17. Such a tool would likely have applications to other online problems beyond edge coloring.

*Acknowledgements:* The authors would like to thank Marek Elias, Seffi Naor and Ola Svensson for comments on an earlier draft of this paper, which helped improve its presentation. The authors would also like to thank Lex Schrijver for asking about extension of our results to multigraphs.

## APPENDIX

### VII. BAD EXAMPLES FOR NATURAL ALGORITHMS

In this section we present bad examples for a variety of natural edge coloring algorithms for known and unknown  $\Delta$ .

#### A. Repeated Maximal Online Matching

Here we give a bad example which shows that a family of natural online algorithms, i.e., algorithms that iteratively find a maximal matching, are no better than 2-competitive, even on dense bipartite graphs under one-sided arrivals. Notice that this family of algorithms includes the natural extension of the RANKING algorithm. I.e., iteratively find the maximal matching via the optimal,  $(1 - 1/e)$ -competitive, online matching algorithm, RANKING – an approach which at first glance one might guess yields an  $\approx (e/(e-1))$ -competitive edge coloring.

<sup>3</sup>In Section VIII-A, we show this example is a tight instance for Algorithm 1, which is 1.777 competitive on it.

The bad example is as follows. The graph is made up of  $\Delta$  stars with  $\Delta - 1$  leaves each, with the stars' centers, which are offline vertices, connected to a common vertex  $v$ , which is the last offline vertex to arrive. It is easy to see that any algorithm that repeatedly uses a maximal online matching for each color  $c = 1, 2, \dots$  would color each star's edges with colors  $1, 2, \dots, \Delta - 1$  following the star's  $\Delta - 1$  leaves' arrivals. The  $\Delta$  edges of  $v$  therefore require a further  $\Delta$  colors. Consequently, such an algorithm would use  $2\Delta - 1$  colors and is thus 2 competitive. Adding  $n - \Delta^2 - 1$  isolated dummy nodes with no edges, we get an example with  $n$  nodes and maximum degree  $\Delta = O(\sqrt{n})$ . This example therefore rules out this natural family of online edge coloring algorithms for all  $\Delta = O(\sqrt{n})$ .

### B. Repeatedly Running Marking

Our online rounding scheme for fractional edge colorings of Section IV applies MARKING to multiple fractional edge colorings. For known  $\Delta$ , this is done by running MARKING on some fractional matchings assigning values  $\frac{1}{\Delta'}$ , which we refer to as MARKING $_{\Delta'}$ , for increasingly smaller value of  $\Delta'$ . Here we show an example underlying the need for rounding multiple edge colorings. In particular, we show that simply iteratively coloring the matching output by MARKING $_{\Delta}$  on the uncolored subgraph – i.e., rounding one trivial edge coloring – results in suboptimally many colors.

To be precise, for  $c = 1, 2, \dots, \Delta$ , the algorithm considered computes  $M_c$ , the  $c$ -th color class, by running MARKING $_{\Delta}$  in the subgraph not colored by the first  $c - 1$  colors,  $U_c \triangleq G \setminus \bigcup_{c'=1}^{c-1} M_{c'}$  (and then reverts to some other algorithm on the uncolored graph, say greedy). For simplicity (though this is too good to be true), let us assume that MARKING $_{\Delta}$  matches each edge  $e$  with probability *precisely*  $1/\Delta$  in  $M_c$  if  $e \in U_c$ . Consider a star graph of degree  $\Delta$ , with the star's center arriving last. Denote by  $p_{e,c} \triangleq \Pr[e \in M_c]$  the probability that  $e$  is colored  $c$ . Then, if we run MARKING $_{\Delta}$  for  $\Delta$  phases on the uncolored graph, the probabilities  $p_{e,c}$  satisfy the recurrence relation  $p_{e,c} = \frac{1}{\Delta} \cdot (1 - \sum_{c' < c} p_{e,c'})$ . But this recurrence captures non-empty bins in a balls and bins process with  $\Delta$  balls (colors) thrown into  $\Delta$  bins (edges). Specifically,  $p_{e,c}$  is the probability of  $c$  being the first ball occupying bin  $e$ . The expected number of unoccupied bins in the above process,  $\Delta - \sum_e \sum_c p_{e,c}$ , is  $(\frac{1}{e} + o(1))\Delta$ , so this process results in at least  $\frac{\Delta}{e} - o(\Delta)$  uncolored edges in the star after  $\Delta$  colors used. Consequently, this approach would use at least  $(1 + \frac{1}{e} - o(1))\Delta$  colors, even on a star of maximum degree  $\Delta$ .

The above bad example rules out running MARKING $_{\Delta}$  for the first  $\Delta$  colors and then resorting to some other algorithm (say, greedy). More generally, extending this idea to any prefix of colors computed using MARKING $_{\Delta}$  and then reverting to greedy can be similarly shown to be suboptimal. For example, standard coupon collector arguments show that

the extreme approach of repeatedly running MARKING $_{\Delta}$  in  $U_c$  for  $c = 1, 2, \dots$  until all edges are colored (i.e., without running greedy) requires at least  $\Delta \log \Delta$  colors(!), even on a star of maximum degree  $\Delta$  whose center arrives last.

### C. Bad Examples for (Unbounded) Water Filling

In this section, we will give bad examples to rule out a natural candidate algorithm for fractional edge coloring; i.e., WATER FILLING. It is easy to see that we can make the level of a color arbitrary large if we only do WATER FILLING on one side. On the other hand, the following algorithm is a natural extension of this algorithm which only conducts WATER FILLING on the *maximum* of the two endpoints' loads for any edge  $(u, v)$  which arrives. More formally, the algorithm is as follows.

---

#### Algorithm 3 WATER FILLING

---

**Input:** Online graph  $G(V, E)$  with unknown  $\Delta' = \Delta(G)$  under vertex arrivals.  
**Output:** Fractional edge coloring  $\{x_{e,c} \mid e \in E, c \in [\Delta']\}$ .

- 1: **for** each arrival of a vertex  $v$  **do**
- 2:      $\Delta \leftarrow \max\{\text{current } d(u) \mid u \in V\}$ .  
       */\*  $\Delta := \text{current max. degree}$  \*/*
- 3:     **for** each  $e = (u, v) \in E$  **do**
- 4:         **while**  $\sum_{c \in [\Delta]} x_{e,c} < 1$  **do**
- 5:             let  $C := \arg \min_{c \in [\Delta]} \max\{L_u(c), L_v(c)\}$ .  
       */\* set of "currently active" colors for  $e$  \*/*
- 6:             **for** all  $c \in C$  **do**
- 7:                 increase  $x_{e,c}$  continuously.  
       */\* update  $L_u(c), L_v(c)$  and  $C$  \*/*

---

Our first observation here is that WATER FILLING is 2 competitive, even under edge arrivals.

**Claim 24.** *The WATER FILLING algorithm is at most 2 competitive under adversarial edge arrivals.*

*Proof:* We only need to prove  $\ell \leq 2$  following updates due to the arrival of an edge  $(u, v)$ . This inequality holds because

$$\begin{aligned} 2\Delta &\geq \sum_{i=1}^{\Delta} \ell_u(i) + \sum_{i=1}^{\Delta} \ell_v(i) = \sum_{i=1}^{\Delta} (\ell_u(i) + \ell_v(i)) \\ &\geq \sum_{i=1}^{\Delta} \max\{\ell_u(i), \ell_v(i)\} \\ &\geq \Delta \cdot \ell, \end{aligned}$$

where the last inequality holds since we run WATER FILLING on the maximum of  $u$  and  $v$ 's loads.  $\blacksquare$

By the above, WATER FILLING is always at most 2 competitive. We will show that there exists a hard instance on which WATER FILLING is 2 competitive, even under one-sided vertex arrivals in bipartite graphs. We start first with a bad example for vertex arrivals in general graphs,

which illustrates the weakness of WATER FILLING, and also motivates the hard instance given in Section V.

*A Bad Example for General Graphs.:* Consider a tree of height  $n + 1$ . The root locates in the first level. Each vertex in level  $k$  has  $n - k + 1$  children. In the online process, vertices arrive from level  $n + 1$  down to 1. See Figure 2 for an example.

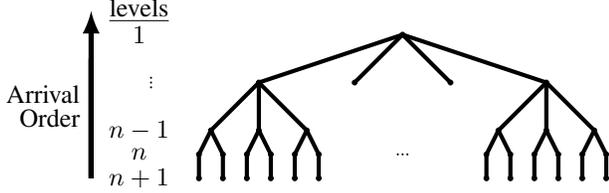


Figure 2: The hard instance for (unbounded) water filling.

The following claim can be proven by induction and we omit the proof here.

**Claim 25.** *The loads for all vertices in level  $k$  when they first arrive are*

- If  $(n - k + 1)$  is odd, the load is  $\left(\frac{2(n-k+1)}{(n-k+2)}, \dots, \frac{2(n-k+1)}{(n-k+2)}, 0, \dots, 0\right)$ .
- If  $(n - k + 1)$  is even, the load is  $\left(\frac{2n-2k+1}{(n-k+1)}, \dots, \frac{2n-2k+1}{(n-k+1)}, \frac{1}{(n-k+1)}, \dots, \frac{1}{(n-k+1)}\right)$ .

When  $n$  is large and we set  $k = 1$ , then the competitive ratio goes to 2.

*A Bad Example for Bipartite Graphs.:* Notice that the bad example above is a bipartite graph, but vertices can arrive from either side. Here we construct a slightly more complicated bipartite example under one-sided arrivals on which WATER FILLING is 2 competitive. In the following claim, we use  $(a, b)$  to denote  $\underbrace{(a, \dots, a)}_{\Delta/2}, \underbrace{b, \dots, b)}_{\Delta/2}$ .

**Claim 26.** *For large enough  $\Delta$ , there exists a sequence of online vertices such that  $(2 - 4/\Delta, 4/\Delta)$  is achievable for offline vertices.*

*Proof:* We only sketch the sequence here.

$$(1, 0) \xrightarrow{(a)} (2/\Delta, 1 + 2/\Delta) \xrightarrow{(b)} (1 + 2/\Delta, 0) \\ \rightarrow \dots \rightarrow (2 - 4/\Delta, 0).$$

The initial state  $(1, 0)$  is achieved by connecting an offline vertex  $v$  to  $\Delta/2$  online vertices one by one. (a) is reached by having an online vertex  $u$  neighbor  $\Delta/2 + 1$  offline vertices  $(1, 0)$  and one offline vertex  $(1 + 1/\Delta, 1/\Delta)$ , which can be produced in the former step. While (b) is achievable by connecting  $(2/\Delta, 1 + 2/\Delta)$  (online) and  $(1, 0)$  (offline). Finally, we repeat the above process in (a)(b) to get  $(2 - 4/\Delta, 0)$ . When  $\Delta$  is large enough, we conclude that the water filling algorithm is exactly 2 competitive. ■

## VIII. OMITTED PROOFS OF SECTION III

Here we provide the missing proofs of lemmas whose proof was deferred from Section III, restated here for ease of reference.

**Lemma 7.** *For a color  $k$  critical at step  $T$ , for all  $A < t \leq T$  such that  $\ell^t(k) > \ell^{t-1}(k)$ , we have*

$$\ell^t(i) - \ell^{t-1}(i) = \beta/\delta^t \quad \forall k < i \leq \delta^t.$$

*Proof:* Suppose there exists  $A < t \leq T$  such that  $k + 1 \leq \delta^t$  and  $\ell^t(k + 1) - \ell^{t-1}(k + 1) < \beta/\delta^t$  and  $\ell^t(k) - \ell^{t-1}(k) > 0$ , then we can immediately derive that  $\ell^t(k) = \ell^t(k + 1)$ , since  $k$  and  $k + 1$  are active at the end of the iteration. But by Observation 6 we know that  $\ell^T(k) = \ell^T(k + 1)$  – a contradiction. Finally,  $\ell^t(i) - \ell^{t-1}(i) \geq \ell^t(k + 1) - \ell^{t-1}(k + 1)$  for all  $k < i \leq \delta^t$  by Observation 6. ■

**Lemma 8.** *If  $k$  is a critical color at step  $T$ , then  $k > \delta^T \cdot (1 - 1/\beta)$ .*

*Proof:* By Lemma 7,  $\ell^T(k) > \ell^T(k + 1)$  and  $\ell^T(k) > \ell^{T-1}(k)$  imply  $\ell^T(i) - \ell^{T-1}(i) = \beta/\delta^T$ , for  $k + 1 \leq i \leq \delta^T$ . Hence, if  $k \leq \delta^T \cdot (1 - \frac{1}{\beta})$ , we would obtain

$$\sum_{i=1}^k (\ell^T(i) - \ell^{T-1}(i)) = 1 - \sum_{i=k+1}^{\delta^T} (\ell^T(i) - \ell^{T-1}(i)) \\ = 1 - (\delta^T - k)\beta/\delta^T \\ < 1 - (\beta/\delta^T) \cdot (\delta^T/\beta) \\ = 0,$$

which would imply  $\ell^T(k) = \ell^{T-1}(k)$  – contradicting the fact that  $k$  is critical. ■

In order to lower bound  $V_k^2$ , we first prove the following two useful claims.

**Claim 27.** *If  $k$  is a critical color at step  $T$ , then for any  $j > k$  and for any  $\mathcal{S} \geq A$ ,*

$$\ell^T(j) - \ell^{\mathcal{S}}(j) = \sum_{\substack{\mathcal{S} < t \leq T \\ \delta^t \geq j}} \frac{\beta}{\delta^t}.$$

*Proof:* We prove that for any  $t \geq A$  and  $\delta^t \geq k$ , then  $\ell^t(k) > \ell^{t-1}(k)$ . Assume not, then we have

$$1 = \sum_{i=1}^{\delta^t} (\ell^t(i) - \ell^{t-1}(i)) \\ = \sum_{i=k+1}^{\delta^t} (\ell^t(i) - \ell^t(i-1)) \\ \leq (\delta^t - k) \cdot \beta/\delta^t \\ \leq (\delta^T - k) \cdot \beta/\delta^T \\ < 1.$$

Where that last inequality is due to  $k > (1 - 1/\beta)\delta^T$ , by Lemma 8. Therefore, by Lemma 7, we have  $\ell^t(j) - \ell^{t-1}(j) = \beta/\delta^t$  for  $j \leq \delta^t$ . Consequently,

$$\begin{aligned}\ell^T(j) - \ell^S(j) &= \sum_{t=S+1}^T (\ell^t(j) - \ell^{t-1}(j)) \\ &= \sum_{t=S+1}^T \mathbb{I}\{\delta^t \geq j\} (\ell^t(j) - \ell^{t-1}(j)) \\ &= \sum_{\substack{S < t \leq T \\ \delta^t \geq j}} \frac{\beta}{\delta^t}.\end{aligned}$$

Next, we bound the *total* load on the colors after a critical color  $k$ .

**Claim 28.** *If  $k$  is a critical color at step  $T$ , then for any  $S \geq A$*

$$\sum_{j=k+1}^{\delta^T} (\ell^T(j) - \ell^S(j)) \geq \sum_{j=S+1}^{\delta^T} \beta \cdot \frac{\delta^j - k}{\delta^j}.$$

*Proof:* By Claim 27, we have

$$\begin{aligned}\sum_{i=k+1}^{\delta^T} (\ell^T(i) - \ell^S(i)) &\geq \sum_{i=k+1}^{\delta^T} \sum_{\substack{S+1 \leq j \leq \delta^T \\ \delta^j \geq i}} \frac{\beta}{\delta^j} \\ &= \sum_{j=S+1}^{\delta^T} \sum_{\delta^j \geq i \geq k} \frac{\beta}{\delta^j} \\ &= \sum_{j=k^*+1}^{\delta^T} \beta \cdot \frac{\delta^j - k}{\delta^j}.\end{aligned}$$

We are now ready to prove the main lower bound volume lemma.

**Lemma 9.** *If  $k$  is a critical color at step  $T$  and  $k^* \geq \max\{k, \delta^A\}$ , then*

$$\begin{aligned}V_2^k &\geq \sum_{j=k+1}^{\delta^T} (\ell^T(j) - \ell^{k^*}(j)) \\ &\geq \beta \cdot \left( \delta^T - k^* - k \log \frac{\delta^T}{k^*} \right).\end{aligned}$$

*Proof:* Substituting  $\mathcal{S}$  with  $k^*$  in Claim 28 (note that,

$k^* \geq \delta^A \geq A$ ), we have

$$\begin{aligned}\sum_{j=k+1}^{\delta^T} (\ell^T(j) - \ell^{k^*}(j)) &= \sum_{j=k^*+1}^{\delta^T} \beta \cdot \frac{\delta^j - k}{\delta^j} \\ &\geq \sum_{j=k^*+1}^{\delta^T} \beta \cdot \frac{j - k}{j} \\ &\geq \beta \cdot (\delta^T - k^*) - \beta \cdot k \log \frac{\delta^T}{k^*} \\ &= \beta \cdot \left( \delta^T - k^* - k \log \frac{\delta^T}{k^*} \right),\end{aligned}$$

where the first inequality is since  $\delta^j \geq j$ .  $\blacksquare$

**Lemma 11.** *If  $k > \delta_u^{A_u}$  is a critical color at step  $T$  with respect to  $u$ , then  $\ell_u^T(k) \leq \beta \log \frac{\beta}{\beta-1}$ .*

*Proof:* As  $k$  is critical at step  $T$ , by Lemma 9, taking  $k^* = k > \delta^A$ , we have

$$\begin{aligned}V_2^k &= \sum_{i=k+1}^{\delta^T} \ell^T(i) \\ &\geq \sum_{i=k+1}^{\delta^T} (\ell^T(i) - \ell^k(i)) \\ &\geq \beta \cdot \left( \delta^T - k - k \log \frac{\delta^T}{k} \right).\end{aligned}$$

In addition, by Lemma 8, we have  $k \geq \delta^T \cdot \left(1 - \frac{1}{\beta}\right)$ . Thus, we find that indeed, by Equation (1)

$$\begin{aligned}\ell^T(k) &\leq \frac{\delta^T - V_2^k}{k} \\ &\leq \frac{\delta^T - \beta \cdot \left( \delta^T - k - k \log \frac{\delta^T}{k} \right)}{k} \\ &= (1 - \beta) \frac{\delta^T}{k} + \beta + \beta \log \frac{\delta^T}{k} \\ &\leq \beta \log \frac{\beta}{\beta-1}.\end{aligned}$$

**Lemma 12.** *If  $k \leq \delta_u^{A_u}$  is a critical color at step  $T$  with respect to  $u$ , then  $\ell_u^T(k) \leq \beta^2 - \beta + \beta \log \frac{1}{\beta-1}$ .*

*Proof:* For ease of notation, in this lemma we will let  $\Delta = \delta^T$ . We will consider two cases and show the bound holds for both cases.

*Case 1:  $\delta^A/\beta \leq k \leq \delta^A$ :* By Lemma 9 with  $k^* = \delta^A \geq k$ , we have

$$V_2^k \geq \beta \cdot \left( \Delta - \delta^A - k \log \frac{\delta^T}{\delta^A} \right).$$

As a consequence, by Equation (1), we have

$$\begin{aligned}
\ell^T(k) &\leq \frac{\Delta - V_2^k}{k} \\
&\leq \left( \Delta - \beta(\Delta - \delta^A) + \beta k \log \frac{\Delta}{\delta^A} \right) / k \\
&= (1 - \beta) \frac{\Delta}{k} + \beta \frac{\delta^A}{k} + \beta \log \frac{\Delta}{\delta^A} \\
&= \frac{\delta^A}{k} \left( (1 - \beta) \frac{\Delta}{\delta^A} + \beta \right) + \beta \log \frac{\Delta}{\delta^A} \\
&\leq \beta \left( (1 - \beta) \frac{\Delta}{\delta^A} + \beta \right) + \beta \log \frac{\Delta}{\delta^A} \\
&\leq \beta^2 - \beta + \beta \log \frac{1}{\beta - 1},
\end{aligned}$$

where the third inequality above holds because  $\frac{\delta^A}{k} \leq \beta$  and  $\frac{\Delta}{\delta^A} \leq \frac{\Delta}{k} \leq \beta/(\beta - 1)$ , by Lemma 8 and the last inequality holds because  $\beta \left( (1 - \beta) \frac{\Delta}{\delta^A} + \beta \right) + \beta \log \frac{\Delta}{\delta^A}$  is maximized when  $\frac{\Delta}{\delta^A} = 1/(\beta - 1)$  (as can be verified by differentiating with respect to  $x = \frac{\Delta}{\delta^A}$ ).

*Case 2:  $k \leq \delta^A/\beta$ :* Note that after the arrival of vertex  $u$ , the color load is at most  $\beta$ , by Observation 10. We may safely assume that  $A \geq \beta k$ , since we can always increase  $A$  to  $\beta k$  without increasing volume in  $V_2^k$  (which we aim to lower bound), by Observation 10.

$$\begin{aligned}
V_2^k &= \sum_{i=k+1}^{\Delta} \ell^{\Delta}(i) \\
&= \sum_{i=k+1}^{\Delta} \ell^A(i) + \sum_{i=k+1}^{\Delta} (\ell^{\delta^A}(i) - \ell^A(i)) \\
&\quad + \sum_{i=k+1}^{\Delta} (\ell^{\Delta}(i) - \ell^{\delta^A}(i)) \\
&\geq (A - \beta k) + \sum_{j=A+1}^{\delta^A} \beta \cdot \frac{\delta^j - k}{\delta^j} \\
&\quad + \beta \cdot \left( \Delta - \delta^A - k \log \frac{\Delta}{\delta^A} \right) \\
&\geq (A - \beta k) + (\delta^A - A) \cdot \beta \cdot \frac{\delta^A - k}{\delta^A} \\
&\quad + \beta \cdot \left( \Delta - \delta^A - k \log \frac{\Delta}{\delta^A} \right) \tag{5} \\
&\geq (\delta^A - \beta k) \cdot \beta \cdot \frac{\delta^A - k}{\delta^A} + \beta \cdot (\Delta - \delta^A) - \beta k \log \frac{\Delta}{\delta^A}.
\end{aligned}$$

The first inequality holds by Observation 10, Claim 27 and Lemma 9 with  $k^* = \delta^A \geq \beta k \geq k$ . The second inequality holds since for  $j > A$ ,  $\delta^j \geq \delta^A$ . For the last inequality, substituting  $A$  with  $\beta k$ , a lower bound of  $A$ , will only decrease Equation (5), since the coefficient of  $A$  is non-negative; i.e.  $1 - \beta + \beta \frac{k}{\delta^A} \geq 1 - \beta + \beta \frac{k}{\Delta} \geq 1 - \beta + \beta \cdot (1 - \frac{1}{\beta}) = 0$ , where the last step follows by Lemma 8. Consequently, by

Equation (1), we have that  $\ell^T(k)$  is at most

$$\begin{aligned}
&\leq \frac{\Delta - V_2^k}{k} \\
&\leq \frac{\Delta - \left( (\delta^A - \beta k) \cdot \beta \cdot \frac{\delta^A - k}{\delta^A} + \beta \cdot (\Delta - \delta^A) - \beta k \log \frac{\Delta}{\delta^A} \right)}{k} \\
&\leq (1 - \beta) \frac{\Delta}{k} + \beta^2 + \beta - \beta^2 \frac{k}{\delta^A} + \beta \log \frac{\Delta}{\delta^A} \\
&= (1 - \beta) \frac{\Delta}{k} + \beta^2 + \beta - \beta^2 \frac{k}{\delta^A} + \beta \left( \log \frac{\Delta}{k} + \log \frac{k}{\delta^A} \right) \\
&= \beta^2 + \beta + \left( \beta \log \frac{k}{\delta^A} - \beta^2 \frac{k}{\delta^A} \right) + \left( \beta \log \frac{\Delta}{k} + (1 - \beta) \frac{\Delta}{k} \right) \\
&\leq \beta^2 + \beta + \left( \beta \log \frac{1}{\beta} - \beta \right) + \left( \beta \log \frac{\beta}{\beta - 1} - \beta \right) \\
&= \beta^2 - \beta + \beta \log \frac{1}{\beta - 1}.
\end{aligned}$$

Finally, we will need the following simple inequalities for our analysis. ■

**Fact 29.** For  $\beta \in (1, 2)$  we have  $\beta \leq \beta^2 - \beta + \beta \log \frac{1}{\beta - 1}$ , as well as  $\beta \log \frac{\beta}{\beta - 1} \leq \beta^2 - \beta + \beta \log \frac{1}{\beta - 1}$ .

*Proof:*

For both inequalities, we rely on  $x - 1 \geq \log(x)$  for all  $x \geq 1$  to obtain the claimed inequalities. For the first, we have

$$\begin{aligned}
&\beta^2 - \beta + \beta \log \frac{1}{\beta - 1} - \beta \\
&= \beta^2 - \beta + \beta \log \frac{1}{\beta - 1} - \beta \\
&= \beta \left( (\beta - 1) - 1 - \log(\beta - 1) \right) \\
&\geq 0.
\end{aligned}$$

For the second inequality, we have

$$\begin{aligned}
&\beta^2 - \beta + \beta \log \frac{1}{\beta - 1} - \beta \log \frac{\beta}{\beta - 1} \\
&= \beta(\beta - 1 - \log \beta) \\
&\geq 0.
\end{aligned}$$

#### A. Tight Example for Bounded Water Filling

Here we give a tight instance for Algorithm 1 in general graphs, showing our analysis in Section III-A is tight. We use the same construction shown in Theorem 23. Moreover, we assume that the state is “old” until phase  $k = (\beta - 1)n$ . We only consider the case when  $b$  is sufficiently large. For sufficiently large  $t$ , where  $t < k$ , we can see the color status for vertex  $v$  is roughly

$$\ell_v^t(i) = \begin{cases} \beta \log \frac{\beta}{\beta - 1} & i \leq (1 - \frac{1}{\beta})t \\ \sum_{x=i}^t \frac{\beta}{x} & (1 - \frac{1}{\beta})t < i \leq k. \end{cases}$$

Meanwhile, the color status for vertex  $u_t$  is roughly

$$\ell_{u_t}^t(i) = \begin{cases} 0 & i \leq (1 - \frac{1}{\beta})t \\ \beta & (1 - \frac{1}{\beta})t < i \leq k. \end{cases}$$

After the arrival of  $u_k$ , the coin is up, and the final color status for  $u_k$  (in round  $n$ ) is

$$\ell_{u_k}^n(i) = \begin{cases} \beta \log \frac{1}{\beta-1} & i \leq \frac{(\beta-1)^2}{\beta} n \\ \beta^2 - \beta + \beta \log \frac{1}{\beta-1} n & \frac{(\beta-1)^2}{\beta} < i < (\beta-1)n \\ \sum_{x=i}^n \frac{\beta}{x} & (\beta-1)n \leq i \leq n. \end{cases}$$

Consequently, our analysis for Algorithm 1 in Section III-A is tight and 1.777 is the best achievable competitive ratio for this algorithm.

#### IX. OMITTED PROOFS OF SECTION IV

Here we provide the missing proofs of lemmas and theorem deferred from Section IV, restated here for ease of reference.

We start by bounding the number of colors used during each phase.

**Lemma 18.** *If  $\Delta_i \geq (6 \log n)/p^3$ , then  $C_i$ , the number of colors used in phase  $i$ , satisfies*

$$\Pr[C_i \geq \alpha \Delta_i \cdot p \cdot (1+p)] \leq \frac{1}{n^2}.$$

*Proof:* As  $\Delta_i \geq (6 \log n)/p^3$ , we have  $\mathbb{E}[C_i] = \mathbb{E}[|\mathcal{S}_i|] \leq \alpha \Delta_i \cdot p \leq \alpha \cdot 6(\log n)/p^2$ . Plugging  $\epsilon = p$  into the upper multiplicative tail bound of Lemma 45, we get

$$\begin{aligned} \Pr[C \geq \alpha \Delta_i \cdot p(1+p)] &\leq \exp\left(-\frac{\alpha \Delta_i \cdot p(1+p)}{3}\right) \\ &\leq \exp\left(-\frac{((6 \log n)/p^3) \cdot p^3}{3}\right) \\ &= 1/n^2. \end{aligned}$$

The main technical lemma of this section, bounding the maximum degree of the uncolored graph  $U_{i+1}$  in terms of its  $i^{\text{th}}$  phase counterpart,  $U_i$ , is as follows.

**Lemma 19.** *If  $\Delta_i \geq (24 \log n)/p^4$ , then*

- 1)  $\Pr[\Delta_{i+1} \leq \Delta_i \cdot (1 - p - 4p^2)] \leq 3/n^3$ .
- 2)  $\Pr[\Delta_{i+1} \geq \Delta_i \cdot (1 - p + 7p^2)] \leq 6/n^2$ .

Before proving this lemma (in turn deferred to Section IX-A), we show how it implies our main theorem, restated below.

**Theorem 15.** *For all  $\alpha \in [1, 2]$  and  $\epsilon \leq 1$ , if there exists an  $\epsilon$ -bounded  $\alpha$ -competitive fractional algorithm  $\mathcal{A}$  for bipartite graphs with unknown maximum degree  $\Delta \geq \Delta' \geq 2/\epsilon$ , then there exists a randomized integral algorithm  $\mathcal{A}'$  which is  $(\alpha + O(\sqrt{(\log n)/\Delta'}))$ -competitive w.h.p on bipartite graphs of unknown maximum degree  $\Delta \geq \Delta' \geq c \cdot \log n$  for some constant  $c$ .*

*Proof:* For our proof, we will require the following fact.

**Fact 30.** *All  $p \in [0, 1/10]$  satisfy  $(1-p-4p^2) \geq \exp(-2 \cdot p)$  and  $(1-p+7p^2) \leq \exp(-p/4)$ .*

For  $p = \sqrt[12]{(24 \log n)/\Delta'} \leq 1/10$  to hold, we need  $\Delta' \geq 24 \cdot 10^{12} \cdot \log n$ . That is,  $c = 24 \cdot 10^{12}$ .

By Lemma 19 and Fact 30,  $\Pr[\Delta_{i+1} \leq \Delta_i \cdot \exp(-2 \cdot p)] \leq \Pr[\Delta_{i+1} \leq \Delta_i \cdot (1 - p - 4p^2)] \leq 3/n^2$ , provided  $\Delta_i \geq (24 \log n)/p^4$ . By our choice of  $p = \sqrt[12]{(24 \log n)/\Delta'}$ , this implies that for all  $i < \lceil (4/p) \log(1/p) \rceil$ ,

$$\Delta \cdot \exp(-2 \cdot p)^i \geq \Delta \cdot p^8 \geq \Delta' \cdot p^8 = (24 \log n)/p^4.$$

Consequently, if we let  $A_i \triangleq [\bigwedge_{j < i} \Delta_j \geq (24 \log n)/p^4]$  be an indicator for the event that  $\Delta_i$  is large enough to appeal to Lemma 18 and Lemma 19 for phase  $i$ , then taking union bound (Lemma 47) over all  $j < i$ , we have

$$\begin{aligned} \Pr[\overline{A}_i] &= \Pr[\Delta_i \leq (24 \log n)/p^4] \\ &\leq \Pr\left[\bigvee_{j < i} (\Delta_j \leq \Delta \cdot \exp(-2 \cdot p)^j)\right] \\ &\leq n \cdot 3/n^3 = 3/n^2. \end{aligned}$$

Now, by Lemma 19 and  $p \leq 1/10$ , we have

$$\begin{aligned} \Pr[\Delta_i - \Delta_{i+1} \leq p(1-7p) \cdot \Delta_i \mid A_i] \\ = \Pr[\Delta_{i+1} \geq \Delta_i \cdot (1-p+7p^2) \mid A_i] \leq 6/n^2. \end{aligned}$$

On the other hand, by Lemma 18, if we denote by  $C_i$  the number of colors used during the  $i^{\text{th}}$  phase, then the probability of any of the  $C_i$  being large is at most

$$\Pr[C_i \geq \alpha \Delta_i \cdot p(1+p) \mid A_i] \leq 1/n^2.$$

Now, by  $\alpha \in [1, 2]$  and  $p \leq 1/10$ , we find that  $\alpha + 54p \leq \alpha(1 + 27p) \leq \alpha \cdot \frac{1+p}{1-7p}$ . Therefore, if we let  $B_i = \mathbf{1}[C_i \geq (\alpha + 54p) \cdot (\Delta_i - \Delta_{i+1})]$  be the bad event that we use a significantly higher number of colors in phase  $i$  than the amount by which we decrease the maximum degree in the uncolored graph in that phase. Then, we have

$$\begin{aligned} \Pr[B_i] &\leq \Pr[C_i \geq (\alpha + 54p) \cdot (\Delta_i - \Delta_{i+1}) \mid A_i] + \Pr[\overline{A}_i] \\ &\leq \Pr[C_i \geq \alpha \cdot \Delta_i \cdot p(1+p) \mid A_i] \\ &\quad + \Pr[\Delta_i - \Delta_{i+1} \leq p(1-7p) \cdot \Delta_i \mid A_i] + \Pr[\overline{A}_i] \\ &\leq 1/n^2 + 6/n^2 + 3/n^2 = 10/n^2. \end{aligned}$$

Therefore, by union bound, we have that with probability at least  $1 - 10/n$ , the number of colors used during the phases is at most

$$\sum_i (\alpha + 54p) \cdot (\Delta_i - \Delta_{i+1}) \leq (\alpha + 54p) \cdot \Delta.$$

Finally, we upper bound the number of colors used by the greedy step of Line 12, by upper bounding the uncolored subgraph's maximum degree before Line 12. We note that

by Lemma 19 and Fact 30, we have  $\Pr[\Delta_{i+1} \geq \Delta_i \cdot \exp(-p/4) \mid A] \leq \Pr[\Delta_{i+1} \geq \Delta_i \cdot (1-p+7p^2) \mid A] \leq 6/n^2$ . Therefore, we find that the final uncolored subgraph  $U$  has maximum degree  $\Delta(U) \leq \Delta \cdot p$ , as the probability  $\Pr[\Delta(U) \geq \Delta \cdot p]$  is at most

$$\begin{aligned} &\leq \Pr[\Delta_{\lceil(4/p) \log(1/p)\rceil} \geq \Delta \cdot \exp(-p/4 \cdot \lceil(4/p) \log(1/p)\rceil)] \\ &\leq \Pr \left[ \bigvee_i (\Delta_{i+1} \geq \Delta_i \cdot \exp(-p/4)) \right] \\ &\leq \Pr \left[ \bigvee_i (\Delta_{i+1} \geq \Delta_i \cdot \exp(-p/4)) \mid A \right] + \Pr[\bar{A}] \\ &\leq n \cdot 6/n^2 + 3/n \\ &= 9/n. \end{aligned}$$

Consequently, the greedy step of Line 12 uses a further  $2\Delta \cdot p$  colors, and so Algorithm 2 is an  $(\alpha+56p)$ -competitive online edge coloring algorithm.  $\blacksquare$

#### A. Progress in degree decrease

In this section we will show that each phase  $i$  of Algorithm 2 with  $\Delta_i \geq 24(\log n)/p^3$  decreases the maximum degree of the uncolored graph by a  $1/(1-p \pm O(p^2))$  factor. That is, we will prove Lemma 19. As outlined in Section IV, our general approach will be to bound the number of times each near-maximum-degree vertex  $v$  in  $U_i$  is matched during the phase and the number of times it is matched without having an edge colored.

For the remainder of this section, we will need the following random variables. First, for any vertex  $v$  and index  $i$ , we let  $d_i(v)$  denote  $v$ 's degree in the uncolored subgraphs  $U_i$ . Moreover, for each edge  $e$  we let  $L_{e,j}^{(i)} = x_j^{(i)}$  if  $j \in \mathcal{S}_i$  and zero otherwise, and similarly  $L_{v,j}^{(i)} \triangleq \sum_{e \ni v} L_{e,j}^{(i)}$ . We refer to the above as the *load* of edge  $e$  and vertex  $v$  in color  $j$  of phase  $i$ . Finally, we denote by  $\ell_e^{(i)} \triangleq \sum_j L_{e,j}^{(i)}$  and  $\ell_v^{(i)} \triangleq \sum_j L_{v,j}^{(i)}$  the load of the edge  $e$  and vertex  $v$  in the sampled colors of phase  $i$ . Clearly, as each color index  $j$  is in  $\mathcal{S}_i$  with probability  $p$ , and as each edge is fractionally matched exactly once, we have that  $\mathbb{E}[\ell_e^{(i)}] = p$  and therefore  $\mathbb{E}[\ell_v^{(i)}] = d_i(v) \cdot p$ . The following lemma asserts that these variables are concentrated around their mean. In all notation, we omit  $i$ , which will be clear from context.

**Lemma 31.** *If  $\Delta_i \geq (24 \log n)/p^3$ , then*

- 1) *for each edge  $e$  we have  $\Pr[\ell_e \geq p(1+p)] \leq 1/n^4$ , and*
- 2) *for each vertex  $v$  of degree  $d_i(v) \geq \Delta_i/2$  in  $U_i$  we have  $\Pr[|\ell_v - d_i(v) \cdot p| \geq d_i(v) \cdot p^2] \leq 2/n^3$ .*

*Proof:* As noted above,  $\mathbb{E}[\ell_e] = p$ . Moreover, by the  $(p^3/12 \log n)$ -boundedness of  $f$  we have that  $\ell_e = \sum_j L_{e,j}$  is the sum of bounded independent variables  $L_{e,j} \in [0, p^3/12 \log n]$ . So, by Chernoff bounds (Lemma 45) with

$\epsilon = p$ , we obtain

$$\begin{aligned} \Pr[\ell_e \geq p(1+p)] &= \Pr[\ell_e \geq \mathbb{E}[\ell_e] \cdot (1+p)] \\ &\leq \exp\left(-\frac{p \cdot p^2}{3p^3/(12 \log n)}\right) \\ &= \exp(-4 \log n) = 1/n^4. \end{aligned}$$

Similarly, as noted above,  $\mathbb{E}[\ell_v] = p \cdot d_i(v)$ . Moreover, as  $x^{(i)}$  is a feasible fractional matching, we have  $|L_{v,j}| \leq 1$  for all  $j$ . So, by Chernoff bounds (Lemma 45), with  $\epsilon = p$ , we obtain

$$\begin{aligned} \Pr[|\ell_v - \mathbb{E}[\ell_v]| \geq p^2 \cdot d_i(v)] &= \Pr\left[\left|\sum_j L_{v,j} - \mathbb{E}[\sum_j L_{v,j}]\right| \geq p \cdot \sum_j \mathbb{E}[L_{v,j}]\right] \\ &\leq 2 \exp\left(-\frac{d_i(v) \cdot p \cdot p^2}{3}\right) \\ &\leq 2 \exp\left(-\frac{\Delta_i \cdot p \cdot p^2}{6}\right) \\ &\leq 2 \exp(-3 \log n) \\ &\leq 2/n^3. \end{aligned}$$

We will now want to bound the number of times a vertex is matched during a phase. We will rely on Lemma 31 together with the following lemma.  $\blacksquare$

**Lemma 32.** *Let  $\vec{x}$  be a fractional matching with  $\max_e x_e \leq p^4/(12 \log n)$ . Then for each edge  $e$ , MARKING run with input  $\vec{x}$  outputs a matching  $\mathcal{M}$  which matches each edge  $e$  with probability*

$$x_e \cdot (1 - 3p) \leq \Pr[e \in \mathcal{M}] \leq x_e$$

*Proof:* The upper bound on  $\Pr[e \in \mathcal{M}]$  is true for all  $\vec{x}$ . For the lower bound, we have that by Lemma 17, as  $p \in [0, 1/10]$  and as we may safely assume  $n \geq 2$  (otherwise the problem is trivial), we have that the probability of  $e$  belonging to  $\mathcal{M}$  is at least

$$\begin{aligned} \Pr[e \in \mathcal{M}] &\geq x_e \cdot (1 - 11p^3 \sqrt{p \cdot \log(12 \log n/p^3)/12 \log n}) \\ &\geq x_e \cdot (1 - 11p^3 \sqrt{3p \log(1/p)/12 \log n + p}) \\ &\geq x_e \cdot (1 - 11p^3 \sqrt{3(1/e)/12 \log n + p}) && p \in [0, 1] \\ &\geq x_e \cdot (1 - 11p^3 \sqrt{3/(e \cdot 12 \log 2) + p}) && n \geq 2 \\ &\geq x_e \cdot (1 - 11p^3 \sqrt{3/(e \cdot 12 \log 2) + 1/10}) && p \leq 1/10 \\ &\geq x_e \cdot (1 - 3p). \end{aligned}$$

Relying on Lemma 31.2 and Lemma 32, we obtain the following bounds on  $M_v$ , the number of times  $v$  is matched during the  $i^{\text{th}}$  phase.  $\blacksquare$

**Lemma 33.** *If  $\Delta_i \geq (24 \log n)/p^4$ , for each vertex  $v$  with degree at least  $d_i(v) \geq \Delta_i/2$ , then  $M_v$ , the number of times  $v$  is matched during the  $i^{\text{th}}$  phase, satisfies*

- 1)  $\Pr[M_v \geq d_i(v) \cdot p(1 + 4p)] \leq 3/n^4$ .
- 2)  $\Pr[M_v \leq d_i(v) \cdot p(1 - 5p)] \leq 3/n^3$ .

*Proof:* Let  $M_v^j$  be an indicator variable for the event that  $v$  is matched in  $M_{i,j}$ . For any instantiation of the variables  $L_{e,j}$ , Lemma 32 implies that each edge  $e$  is matched in  $M_{i,j}$  with probability  $L_{e,j} \cdot (1 - 3p) \leq \Pr[e \in M_{i,j}] \leq L_{e,j}$ , and so by linearity we have  $L_{v,j} \cdot (1 - 3p) \leq \Pr[M_v^j] \leq L_{v,j}$ . In particular, if we let  $A \triangleq [d_i(v) \cdot p(1 - p) \leq \ell_v \leq d_i(v) \cdot p(1 + p)]$ , then, by linearity we have both  $\mathbb{E}[M_v | A] \leq d_i(v) \cdot p(1 + p)$  as well as  $\mathbb{E}[M_v | A] \geq d_i(v) \cdot p(1 - p)(1 - 3p) \geq d_i(v) \cdot p(1 - 4p)$ . Now, clearly,  $M_v = \sum_{j \in \mathcal{S}_i} M_v^j$  is the sum of binary random variables. Moreover, for any subset  $\mathcal{S}_i$  sampled, these  $\{M_v^j | j \in \mathcal{S}_i\}$  are independent, as all matchings  $M_{i,j}$  for  $j \in \mathcal{S}_i$  are computed using independent copies of MARKING. By Chernoff's upper tail bound (Lemma 45) with  $\epsilon = 2p$ , we thus obtain

$$\begin{aligned}
& \Pr[M_v \geq d_i(v) \cdot p(1 + 4p) | A] \\
& \leq \Pr[M_v \geq d_i(v) \cdot p(1 + p)(1 + 2p) | A] \\
& \leq \Pr[M_v \geq \mathbb{E}[M_v | A] \cdot (1 + 2p) | A] \\
& \leq \exp\left(-\frac{\mathbb{E}[M_v | A] \cdot 4p^2}{3}\right) \\
& \leq \exp\left(-\frac{d_i(v) \cdot p(1 - 4p) \cdot 4p^2}{3}\right) \\
& \leq \exp\left(-\frac{(48 \log n)p^3(1 - 4p)}{3p^4}\right) \\
& \leq \exp(-4 \log n) \qquad p \leq 1/5 \\
& \leq 1/n^4.
\end{aligned}$$

Therefore, we obtain the first claim, as

$$\begin{aligned}
& \Pr[M_v \geq d_i(v) \cdot p(1 + 4p)] \\
& \leq \Pr[M_v \geq d_i(v) \cdot p(1 + 4p) | A] + \Pr[\bar{A}] \\
& \leq 3/n^3.
\end{aligned}$$

Similarly, by Chernoff's lower tail bound (Lemma 45)

with  $\epsilon = p$ , we obtain

$$\begin{aligned}
& \Pr[M_v \leq d_i(v) \cdot p(1 - 5p) | A] \\
& \leq \Pr[M_v \leq d_i(v) \cdot p(1 - p)(1 - 3p)(1 - p) | A] \\
& \leq \Pr[M_v \leq \mathbb{E}[M_v | A] \cdot (1 - p) | A] \\
& \leq \exp\left(-\frac{\mathbb{E}[M_v | A] \cdot p^2}{2}\right) \\
& \leq \exp\left(-\frac{d_i(v) \cdot p(1 - p)(1 - 3p) \cdot p^2}{2}\right) \\
& \leq \exp\left(-\frac{12(\log n)p^3(1 - p)(1 - 3p)}{2p^4}\right) \\
& \leq \exp(-3 \log n) \\
& \leq 1/n^3,
\end{aligned}$$

where the second to last inequality holds for all  $p \leq 1/10$ . From the above we obtain the second claim, as

$$\begin{aligned}
& \Pr[M_v \leq d_i(v) \cdot p(1 - 5p)] \\
& \leq \Pr[M_v \leq d_i(v) \cdot p(1 - 5p) | A] + \Pr[\bar{A}] \leq 3/n^3.
\end{aligned}$$

The above lemma asserts that the number of times a vertex  $v$  of high degree in  $U_i$  is matched during the  $i^{\text{th}}$  phase is  $\Theta(d_i(v) \cdot p)$ . The following lemma relies on the theory of Negative Association (NA, see Section XIII-A) to show that all but  $O(d_i(v) \cdot p^2)$  matches of  $v$  during this phase result in an edge of  $v$  being colored. ■

**Lemma 34.** *If  $\Delta_i \geq (24 \log n)/p^3$ , for each vertex  $v$  with degree at least  $d_i(v) \geq \Delta_i/2$ , the number of times  $v$  is matched along a previously colored edge,  $R_v$ , satisfies  $\Pr[R_v \geq 2d_i(v) \cdot p^2] \leq 2/n^2$ .*

*Proof:* Fix the realizations of  $L_{e,j}$  for all  $e, j$ . For any edge  $e \ni v$ , let  $M_{e,j} \triangleq \mathbb{1}[e \in M_{i,j}]$  be an indicator for edge  $e$  being matched in iteration  $j$  of phase  $i$ . By the 0-1 rule, since at most one edge  $e \ni v$  is in any matching, for each  $j$  the binary variables  $\{M_{e,j} | e \ni v\}$  are NA. On the other hand, for  $j \neq j'$  the joint distributions  $\{M_{e,j} | e \ni v\}$  and  $\{M_{e,j'} | e \ni v\}$  are independent. Thus, by closure of NA distributions under independent union (Property 1), the  $\{M_{e,j} | j \in \mathcal{S}_i, e \ni v\}$  are NA. By closure of NA distributions under monotone increasing functions of disjoint variables (Property 2), if we let  $R_e \triangleq \sum_j M_{e,j} \cdot \min\{1, \sum_{j' < j} M_{e,j'}\}$  denote the number of times  $e$  is matched and not colored, then these  $\{R_e | e \ni v\}$  are NA. In this terminology, we have that  $R_v = \sum_{e \ni v} R_e$  is the sum of NA variables. Moreover, as the  $M_{e,j}$  are NA and as  $\mathbb{E}[M_{e,j}] \leq L_{e,j}$  by Lemma 17, we have by the definition

of NA variables (see (39)) that

$$\begin{aligned} \mathbb{E} \left[ \sum_j M_{e,j} \cdot \sum_{j' < j} M_{e,j'} \right] &\leq \sum_j \mathbb{E} [M_{e,j}] \cdot \mathbb{E} \left[ \sum_{j' < j} M_{e,j'} \right] \\ &\leq \sum_j L_{e,j} \cdot \sum_{j' < j} L_{e,j'} \\ &\leq \ell_e \cdot \ell_e. \end{aligned}$$

Let  $A = \mathbb{1}[\forall e \ni v : \ell_e \leq p(1+p)]$  be an indicator for the high probability event that every edge  $e \ni v$  has load at most  $2p$  in the sampled matchings.

$$\begin{aligned} \mathbb{E}[R_e | A] &\leq \mathbb{E} \left[ \sum_j M_{e,j} \cdot \sum_{j' < j} M_{e,j'} \mid A \right] \\ &\leq \mathbb{E}[\ell_e | A] \cdot \mathbb{E}[\ell_e | A] \\ &\leq p^2(1+p)^2. \end{aligned}$$

Therefore, by linearity of expectation,  $\mathbb{E}[R_v] = \sum_{e \ni v} \mathbb{E}[R_e] \leq d_i(v) \cdot p^2(1+p)^2$ . Now, as  $d_i(v) \geq \Delta_i/2 \geq 12(\log n)/p^3$  and as  $R_v = \sum_e R_e$  is the sum of binary NA variables, we can upper bound  $R_v$  using the upper multiplicative Chernoff bound of Lemma 42 with  $\epsilon = \sqrt{p}$  to obtain

$$\begin{aligned} \Pr[R_v \geq d_i(v) \cdot p^2(1+p)^2(1+\sqrt{p}) | A] \\ &\leq \exp \left( -\frac{d_i(v) \cdot p^2(1+p)^2 \cdot p}{3} \right) \\ &\leq \exp \left( -\frac{12 \log n}{3} \right) \leq \frac{1}{n^2}. \end{aligned}$$

Observing that for  $p \leq 1/10$  we have  $2 \leq (1+p)^2(1+\sqrt{p})$ , we find that

$$\begin{aligned} \Pr[R_v \geq 2d_i(v) \cdot p^2 | A] \\ &\leq \Pr[R_v \geq d_i(v) \cdot p^2(1+p)^2(1+\sqrt{p}) | A] \\ &\leq 1/n^2. \end{aligned}$$

Now, by Lemma 31.1 we have for every  $e \ni v$  that  $\Pr[\ell_e \geq p(1+p)] \leq 1/n^3$  and so by union bound we have  $\Pr[\bar{A}] \leq n \cdot 1/n^3 = 1/n^2$ . We therefore conclude that indeed

$$\begin{aligned} \Pr[R_v \geq 2 \cdot d_i(v) \cdot p^2] \\ &\leq \Pr[R_v \geq 2 \cdot d_i(v) \cdot p^2 | A] + \Pr[\bar{A}] \\ &\leq 2/n^2. \end{aligned}$$

Lemma 19, restated below for ease of reference, follows from lemmas 33 and 34 and union bound of relevant subsets of vertices. ■

**Lemma 19.** *If  $\Delta_i \geq (24 \log n)/p^4$ , then*

- 1)  $\Pr[\Delta_{i+1} \leq \Delta_i \cdot (1-p-4p^2)] \leq 3/n^3$ .
- 2)  $\Pr[\Delta_{i+1} \geq \Delta_i \cdot (1-p+7p^2)] \leq 6/n^2$ .

*Proof:* For each vertex  $v$ , the decrease in  $v$ 's degree in the uncolored subgraph during the  $i^{\text{th}}$  phase, denoted by  $D_v \triangleq d_i(v) - d_{i+1}(v)$ , is precisely the number of times  $v$  is matched and its matched edge is colored. That is, in the terminology of Lemma 33 and Lemma 34,  $D_v = M_v - R_v$ . So, by Lemma 33, every maximum degree vertex  $v$  in  $U_i$  (i.e.  $d_i(v) = \Delta_i \geq \Delta_i/2$ ) satisfies

$$\begin{aligned} \Pr[d_{i+1}(v) \leq \Delta_i \cdot (1-p-3p^2)] \\ &= \Pr[d_{i+1}(v) \leq d_i(v) \cdot (1-p-3p^2)] \\ &= \Pr[d_i(v) - d_{i+1}(v) \geq d_i(v) \cdot p(1+3p)] \\ &= \Pr[D_v \geq d_i(v) \cdot p(1+3p)] \\ &\leq \Pr[M_v \geq d_i(v) \cdot p(1+3p)] \\ &\leq 3/n^4. \end{aligned}$$

The first claim then follows by union bound over all maximum degree vertices  $v$  in  $U_i$ .

$$\begin{aligned} \Pr[\Delta_{i+1} \leq \Delta_i \cdot (1-p-3p^2)] \\ &\leq \sum_{v: d_i(v)=\Delta_i} \Pr[d_{i+1}(v) \leq \Delta_i \cdot (1-p-3p^2)] \\ &\leq 3/n^3. \end{aligned}$$

Now, we let  $\lambda \triangleq p(1-7p)$  and note that  $(1-\lambda) \cdot \Delta_i \geq \Delta_i/2$ , since  $p \leq 1/2$ . All vertices  $v$  of degree  $d_i(v) \leq (1-\lambda) \cdot \Delta_i$  in  $U_i$  clearly have  $d_{i+1}(v) \leq d_i(v) \leq (1-\lambda) \cdot \Delta_i$ . On the other hand, for every  $v$  with  $d_i(v) \geq (1-\lambda) \cdot \Delta_i \geq \Delta_i/2$ , we have by lemmas 33 and 34 that

$$\begin{aligned} \Pr[d_{i+1}(v) \geq (1-\lambda) \cdot \Delta_i] \\ &\leq \Pr[d_{i+1}(v) \geq (1-\lambda) \cdot d_i(v)] \\ &= \Pr[d_i(v) - d_{i+1}(v) \leq d_i(v) \cdot \lambda] \\ &= \Pr[D_v \leq d_i(v) \cdot \lambda] \\ &= \Pr[D_v \leq d_i(v) \cdot p(1-7p)] \\ &\leq \Pr[M_v \leq d_i(v) \cdot p(1-5p)] + \Pr[R_v \geq d_i(v) \cdot p \cdot 2p] \\ &\leq 6/n^3. \end{aligned}$$

The second claim then follows by union bound over all vertices  $v$  of degree  $d_i(v) \geq (1-\lambda) \cdot \Delta_i$  in  $U_i$ , recalling that  $\lambda = p(1-7p)$ , since ■

$$\begin{aligned} \Pr[\Delta_{i+1} \geq (1-\lambda) \cdot \Delta_i] \\ &\leq \sum_{v: d_i(v) \geq (1-\lambda) \cdot \Delta_i} \Pr[d_{i+1}(v) \geq (1-\lambda) \cdot \Delta_i] \\ &\leq 6/n^2. \end{aligned}$$

## X. IMPROVED $o(1)$ TERMS FOR KNOWN $\Delta$

In this section we present an improved algorithm for known  $\Delta$ . Recall that for the known  $\Delta$  regime, using our online rounding scheme of Theorem 15 we obtained

a  $1 + O(\sqrt[12]{(\log n)/\Delta})$ -competitive algorithm (see Theorem 20). In this section we will show how to decrease this competitive ratio to  $1 + O(\sqrt[4]{(\log n)/\Delta})$ . That is, we improve the  $o(1)$  term when  $\Delta = \omega(\log n)$ .

We now turn to describing our approach, starting with an offline description. Iterating over  $c \in [\Delta]$ , we compute and color a matching  $M_c$  in the uncolored subgraph  $G \setminus \bigcup_{c'=1}^{c-1} M_{c'}$ . We then color the remaining uncolored subgraph with new colors using the greedy algorithm. This approach can be implemented online, by iteratively running online matching algorithms on the relevant uncolored subgraphs to compute and color matchings. More concretely, when a vertex  $v$  arrives, we iterate over  $c \in [\Delta]$  and update  $M_c$  in the current uncolored graph  $G \setminus \bigcup_{c'=1}^{c-1} M_{c'}$ , as follows. We run the next step of the online matching algorithm used to compute  $M_c$  in the current uncolored graph after  $v$ 's arrival in this subgraph. We then color  $v$ 's newly-matched edge (if any) using color  $c$ . Finally, we run steps of the greedy algorithm on the remaining uncolored edges of  $v$ .

For our analysis, we will analyze the above algorithm according to its offline description. Since the greedy algorithm requires a number of colors linear in its input graph's maximum degree, our objective will be to reduce the uncolored subgraph's maximum degree to  $o(\Delta)$  w.h.p. after computing and coloring the first  $\Delta$  matchings. In particular, this will require us to match each maximum-degree vertex in  $G$  with probability roughly one for each of these  $\Delta$  matchings. One way of matching vertices  $v$  of degree  $\Delta$  in the uncolored subgraph with probability roughly one is to guarantee each edge  $e \ni v$  a probability of roughly  $\frac{1}{\Delta}$  of being matched. An online matching algorithm which does just this is obtained from Lemma 17 applied to the trivial fractional matching which assigns a value of  $\frac{1}{\Delta}$  to each edge. We will refer by  $\text{MARKING}_d$  to the application of  $\text{MARKING}$  to the trivially-feasible fractional matching assigning  $x_e = \epsilon = \frac{1}{d}$  for each edge in a graph of (known) maximum degree at most  $d$ .

**Corollary 35.** *Algorithm  $\text{MARKING}_d$  is an online matching algorithm which in graphs of maximum degree at most  $d$  outputs a matching  $\mathcal{M}$  which matches each edge  $e$  with probability*

$$\frac{1}{d} \cdot \left(1 - 11\sqrt[3]{(\log d)/d}\right) \leq \Pr[e \in \mathcal{M}] \leq \frac{1}{d}.$$

The first natural approach given Corollary 35 is to iteratively run  $\text{MARKING}_\Delta$ . However, as shown in Section VII-B, this approach is suboptimal. Instead, we will increase the probability of high-degree vertices in the uncolored subgraph to have an edge colored, by running  $\text{MARKING}_d$  with a tighter upper bound  $d$  than  $\Delta$  for the uncolored graph's maximum degree for each phase. Unfortunately, upon arrival of some vertex  $v$ , we do not know the uncolored graph's maximum degree for all phases, as this depends on *future* arrivals and random choices of our algorithm. To obtain

a tight (up to  $o(\Delta)$ ) bound  $d$  on the uncolored graph's maximum degree for each phase, we divide the  $\Delta$  coloring iterations into *phases* of  $\ell = \sqrt{\Delta \log n}$  iterations each, during which we use the same upper bound. As  $\ell = o(\Delta)$  and  $\ell = \omega(\log n)$ , this gives us sharply concentrated upper bounds  $d_{i+1}$  on the resulting uncolored graph's maximum degree at the end of each phase  $i$ , which in turn serves as a tight upper bound for the next phase. This results in the desired rate of decrease in the uncolored graph's maximum degree, namely  $1 - o(1)$  per iteration. Greedy thus runs on a subgraph of maximum degree  $o(\Delta)$ . Our  $1 + o(1)$  competitive ratio follows.

#### A. The Improved Algorithm

We now present our online edge coloring algorithm, starting with an offline description. Our algorithm consists of  $\Delta$  iterations, equally divided into  $\sqrt{\Delta \log n}$  phases. During each iteration of phase  $i$ , we color a matching output by  $\text{MARKING}_{d_i}$  run on  $U_i$  – the uncolored subgraph prior to phase  $i$ , for  $d_i \triangleq \Delta - i \cdot (\ell - 8\sqrt{\ell \log n})$ . After all phases, we run greedy with new colors, starting with  $\Delta + 1$ . In the online implementation, after each online vertex  $v$ 's arrival, for phase  $i = 1, 2, \dots$ , we run the next step of  $\ell = \sqrt{\Delta \log n}$  independent runs of  $\text{MARKING}_{d_i}$  in  $U_i$ , color newly-matched edges and update  $U_{i'}$  for  $i' > i$  accordingly. We then greedily color  $v$ 's remaining uncolored edges with new colors. The algorithm's pseudocode is given in Algorithm 4.

---

#### Algorithm 4 Improved Algorithm for Known $\Delta$

---

**Input:** Online bipartite graph  $G(L, R, E)$  with maximum degree  $\Delta = \omega(\log n)$ .

**Output:** Integral  $(1 + o(1))\Delta$  edge coloring, w.h.p.

- 1: let  $\ell \triangleq \lfloor \sqrt{\Delta \log n} \rfloor$ .  $\triangleright$  phase length
  - 2: let  $d_i \triangleq \Delta - i \cdot (\ell - 8\sqrt{\ell \log n})$  for  $i \in [0, \Delta/\ell]$ .  $\triangleright$  degree upper bound for each phase
  - 3: for all  $i$ , denote by  $U_i$  the **online** subgraph of  $G$  not colored by colors  $[i \cdot \ell]$ .
  - 4: **for** each arrival of a vertex  $v \in R$  **do**
  - 5:     **for** phase  $i = 0, 1, \dots, \lfloor \Delta/\ell \rfloor - 1$  **do**
  - 6:         **for** colors  $c \in [i \cdot \ell + 1, (i + 1) \cdot \ell]$  **do**
  - 7:              $M_c \leftarrow$  output of copy  $c$  of  $\text{MARKING}_{d_i}$  on **current**  $U_i$ .  
*/\* run next step of  $\text{MARKING}_{d_i}$  \*/*
  - 8:             **if** some  $e \in M_c$  is previously uncolored **then**
  - 9:                 color  $e$  using color  $c$ .  
*/\* note:  $e \ni v$  \*/*
  - 10:     run greedy on all uncolored edges of  $v$ , using new colors starting from  $\Delta + 1$ .
- 

#### B. Analysis

The crux of our analysis is that for each phase  $i$ , we have  $d_i \geq \Delta(U_i)$  w.h.p. Consequently, the final uncolored subgraph after the  $\Delta$  iterations (and colors) has maximum

degree at most  $d_{\lfloor \Delta/\ell \rfloor} = o(\Delta)$ , so greedily coloring this subgraph requires a further  $o(\Delta)$  colors. The following lemma asserts that if  $d_i \geq \Delta(U_i)$ , then  $d_{i+1} \geq \Delta(U_{i+1})$ , w.h.p.

**Lemma 36.** *For all  $i \in [0, \Delta/\ell - 1]$ , if  $\Delta(U_i) \leq d_i$ , then  $\Pr[\Delta(U_{i+1}) > d_{i+1}] \leq 1/n^3$ .*

*Proof:* If  $\Delta(U_i) \leq d_i - \ell$ , the claim is trivial, as then  $d_{i+1} \geq d_i - \ell \geq \Delta(U_i) \geq \Delta(U_{i+1})$ . We therefore focus on the case  $d_i - \ell \leq \Delta(U_i) \leq d_i$ . For this latter case, we will rely on the fact that for all  $i \leq \Delta/\ell$ , we have  $d_i = \Delta - i \cdot (\ell - 8\sqrt{\ell \log n}) \geq (\Delta/\ell) \cdot 8\sqrt{\ell \log n} > 3\Delta^{3/4} \log^{1/4} n$ .

Vertices of degree less than  $\Delta(U_i) - \ell \leq d_i - \ell < d_{i+1}$  clearly have degree at most  $d_{i+1}$  in  $U_{i+1}$ , as we only decrease their degree in the uncolored subgraph over time. We therefore turn our attention to vertices  $v$  of degree at least  $\Delta(U_i) - \ell$  in  $U_i$ . Such a vertex  $v$  cannot have more than  $\ell$  edges colored during phase  $i$ , regardless of our algorithm's random choices, as at most one of  $v$ 's edges is colored per iteration. So, before each color  $c$  used in the phase,  $v$  has at least  $\Delta(U_i) - 2\ell \geq d_i - 3\ell$  uncolored edges, each of which is matched by  $\text{MARKING}_{d_i}$  with probability at least  $\frac{1}{d_i} (1 - 11 \sqrt[3]{(\log d_i)/d_i})$ . Therefore, if we let  $X_c \triangleq \mathbb{1}[\bigvee_{e \ni v} e \text{ colored } c]$  be an indicator for the event that  $v$  has an edge colored  $c$ , then, regardless of the realization  $\vec{x}$  of variables  $X_{c-1}, X_{c-2}, \dots, X_{(i-1) \cdot \ell + 1}$  corresponding to previous iterations of the  $i^{\text{th}}$  phase, vertex  $v$  will have an edge colored  $c$  with probability at least

$$\begin{aligned} & \Pr[X_c = 1 \mid (X_{c-1}, X_{c-2}, \dots, X_{(i-1) \cdot \ell + 1}) = \vec{x}] \\ & \geq (d_i - 3\ell) \cdot \frac{1}{d_i} \left(1 - 11 \sqrt[3]{(\log d)/d}\right) \\ & \geq 1 - \frac{3\ell}{d_i} - 11 \sqrt[3]{(\log d_i)/d_i} \\ & \geq 1 - \left(\frac{3}{8} + \frac{11}{2}\right) \sqrt[4]{(\log n)/\Delta} \\ & \geq 1 - 6 \sqrt[4]{(\log n)/\Delta}, \end{aligned}$$

where the penultimate inequality follows from  $\ell \leq \sqrt{\Delta \log n}$  and  $n \geq d_i \geq 8\Delta^{3/4} \log^{1/4} n$ .

Therefore, the expected decrease of  $v$ 's degree in the uncolored graph during the  $\ell$  iterations of the  $i^{\text{th}}$  phase is at least  $\mathbb{E}[\sum_{c=i \cdot \ell + 1}^{(i+1) \cdot \ell} X_c] \geq \ell \cdot \left(1 - 6 \sqrt[4]{(\log n)/\Delta}\right) \geq \ell - 6\sqrt{\ell \log n}$ . But the probability of  $v$  having an edge colored  $c$  is at least  $1 - 6 \sqrt[4]{(\log n)/\Delta}$  independently of previous colors during the phase. Consequently, we can appeal to standard coupling arguments (Lemma 44) together with Hoeffding's inequality (Lemma 46) to show that the

sum of these  $\ell$  binary variables satisfies

$$\begin{aligned} & \Pr \left[ \sum_{c=i \cdot \ell + 1}^{(i+1) \cdot \ell} X_c \leq \ell - 6\sqrt{\ell \log n} - \sqrt{2\ell \log n} \right] \\ & \leq \exp \left( -\frac{2(\sqrt{2\ell \log n})^2}{\ell} \right) \\ & = 1/n^4. \end{aligned}$$

Put otherwise,  $v$ 's degree in the uncolored subgraph decreases during phase  $i$  by less than  $\ell - 6\sqrt{\ell \log n} - \sqrt{2\ell \log n} > \ell - 8\sqrt{\ell \log n}$  with probability at most  $1/n^4$ . Thus, as  $v$  has degree at most  $\Delta(U_i)$  in  $U_i$  by definition, we find that vertex  $v$ 's degree in  $U_{i+1}$ , denoted by  $D_v$ , satisfies

$$\begin{aligned} \Pr[D_v \geq d_{i+1}] & = \Pr[D_v \geq d_i - \ell + 8\sqrt{\ell \log n}] \\ & \leq \Pr[D_v \geq \Delta(U_i) - \ell + 8\sqrt{\ell \log n}] \\ & \leq 1/n^4. \end{aligned}$$

Taking union bound over all vertices, the lemma follows.  $\blacksquare$

The above lemma implies this section's main result, given by the following theorem.

**Theorem 37.** *Algorithm 4 is  $(1 + O(\sqrt[4]{(\log n)/\Delta}))$ -competitive w.h.p. in  $n$ -vertex bipartite graphs with known maximum degree  $\Delta = \Omega(\log n)$ .*

*Proof:* Algorithm 4 computes a feasible edge coloring. It colors each edge, by Line 10, and each color class – computed during iterations or by greedy – constitutes a matching (here we rely on the colors used by greedy and the phases being disjoint). It remains to bound the number of colors this algorithm uses. Each phase requires at most  $\ell$  colors, so the phases require at most  $\Delta$  colors. The number of colors the greedy step requires is at most twice the maximum degree of the remaining uncolored subgraph after the phases, which we now bound.

Let  $A_i \triangleq \mathbb{1}[\Delta(U_i) \leq d_i]$  be an indicator for the event that  $d_i$  upper bounds  $\Delta(U_i)$ . By Lemma 36 we have that  $\Pr[\overline{A_i} \mid A_{i-1}, A_{i-2}, \dots] = \Pr[\overline{A_i} \mid A_{i-1}] \leq 1/n^3$ . Also, trivially  $\Pr[\overline{A_0}] = 0$ . Taking union bound over all  $i$ , we find that the probability of any  $A_i$  not being one is at most

$$\begin{aligned} \Pr \left[ \bigvee_{i=0}^{\Delta/\ell} \overline{A_i} \right] & \leq \sum_{i=1}^{\Delta/\ell} \Pr[\overline{A_i}, A_{i-1}, A_{i-2}, \dots] \\ & \leq \sum_{i=1}^{\Delta/\ell} \Pr[\overline{A_i} \mid A_{i-1}, A_{i-2}, \dots] \\ & \leq (\Delta/\ell)/n^3 \\ & \leq 1/n^2. \end{aligned}$$

Consequently, all applications of  $\text{MARKING}_{d_{i+1}}$  during phase  $i+1$  match each edge of  $U_{i+1}$  with probability at least  $\frac{1}{d_{i+1}} (1 - 11 \sqrt[3]{(\log d_{i+1})/d_{i+1}})$ , as required by our analysis for phase  $i+1$ . Moreover,  $d_i \geq \Delta(U_i)$  for all  $i \in [0, \Delta/\ell]$

w.h.p. implies that the uncolored subgraph following the  $\Delta/\ell$  phases has maximum degree at most

$$\begin{aligned} d_{\lfloor \Delta/\ell \rfloor} &= \Delta - \lfloor \Delta/\ell \rfloor \cdot (\ell - 8\sqrt{\ell \log n}) \\ &\leq \ell + (\Delta/\ell) \cdot 8\sqrt{\ell \log n} \\ &\leq \Delta^{1/2} \log^{1/2} n + 8\Delta^{3/4} \log^{1/4} n \\ &\leq 9\Delta^{3/4} \log^{1/4} n. \end{aligned}$$

The greedy algorithm therefore colors the remaining uncolored graph using at most a further  $18\Delta^{3/4} \log^{1/4} n - 1$  colors. That is, it uses  $\Delta \cdot O(\sqrt[4]{(\log n)/\Delta})$  colors in the range  $\Delta + 1, \Delta + 2, \dots$ . The theorem follows, including the stated bound for  $\Delta = \omega(\log n)$ .  $\blacksquare$

**Remark.** Algorithm 4 is  $(1 + O(\sqrt[4]{(\log n)/\Delta}))$  competitive w.h.p. for all  $\Delta = \Omega(\log n)$  large enough, so for  $\Delta = \Omega(\log n)$  large enough it yields a constant competitive ratio strictly smaller than 2.

## XI. OMITTED PROOFS OF SECTION V

Here we present our proof for the lower bound for online edge coloring in general graphs.

**Theorem 23.** *No fractional online edge coloring algorithm is better than 1.606 competitive in general graphs.*

*Proof:* The adversarial instance has  $m + 1$  possible futures. Neither the number of phases  $m$  nor the choice of future are known to the online algorithm. There is a state associated with the input, and there are two possible states, “old” and “new”. Initially, the graph contains  $m!$  vertices and the state is “old”. There are  $m' \leq m$  phases in total. We use  $V_k$  to denote the set of online vertices which arrive in phase  $k$  ( $k \in [m']$ ) and  $V_0$  to denote the initial  $m!$  vertices which arrive in phase 0. Moreover, we use  $v_i^k$  to denote the  $i^{\text{th}}$  vertex arrived in phase  $k$ . In phase  $k$ , newly-arrived vertices have degree  $k$ . If the state is “old”,  $m!/k$  vertices arrive and the  $i^{\text{th}}$  vertex,  $v_i^k$ , is adjacent to  $v_i^0, v_{m!/k+i}^0, \dots, v_{(k-1)m!/k+i}^0$ . On the other hand, if the state is “new” and it changed from “old” to “new” at the end of phase  $t$  ( $k > t$ ), then  $m!/kt$  vertices arrive and the  $i^{\text{th}}$  vertex,  $v_i^k$ , will neighbor  $v_i^t, v_{m!/kt+i}^t, \dots, v_{m!(k-1)/kt+i}^t$ . At the end of phase  $k$ , the adversary decide whether to switch state to “new”. Notice that the state can only transition from “old” to “new”.

Again, we let  $x_{k,j}$  denote the average assignment of color  $j$  to edges of phase  $k$ , but this time only if the state is “old” during this phase. The following constraints still hold for the same reason as Constraints (2) and (3) for the bipartite hard instance of Theorem 22.

$$\sum_{j=1}^k x_{k,j} \geq 1 \quad \forall k. \quad (6)$$

$$\sum_{k=j}^m x_{k,j} \leq \alpha \quad \forall j. \quad (7)$$

Furthermore, We use  $y_{k,j}^t$  to denote the average assignment of color  $j$  to edges between  $V_k$  and  $V_t$  when the state transitions from “old” to “new” in phase  $t$ . (I.e., this is the average assignment of color  $j$  to edges of phase  $k > t$ , for  $t$  the phase at which the transition occurred.)

Again, as each edge between a  $V_t$  vertex and its neighbor in  $V_k$  ( $k > t$ ) must be fractionally colored, we have

$$\sum_{j=1}^k y_{k,j}^t \geq 1 \quad \forall t < k \leq m. \quad (8)$$

Moreover, the maximum load of every vertex for every color is at most  $\alpha$ , and so we have

$$t \cdot x_{t,j} + \sum_{k=t+1}^m y_{k,j}^t \leq \alpha \quad \forall 1 \leq j \leq t \leq m \quad (9)$$

$$\sum_{k=j}^m y_{k,j}^t \leq \alpha \quad \forall 1 \leq t < j \leq m \quad (10)$$

$$k \cdot y_{k,j}^t \leq \alpha \quad \forall 1 \leq t < k \leq m. \quad (11)$$

To summarize, constraints (6)-(11) for any  $m$  are all satisfied by any  $\alpha$ -competitive online fractional edge coloring algorithm on this distribution of inputs. Therefore, the optimal value of an LP with objective of minimizing  $\alpha$  subject to these constraints is a lower bound on the optimal competitive ratio  $\alpha$  of any such online algorithm on general graphs. Using commercial solvers, we solve this LP for  $m = 50$  and find that its optimal value, which lower bounds any algorithm’s competitive ratio on general graphs, is 1.606. Again, using the same trick as Section V-A, we find that this lower bound also holds for dense graphs.  $\blacksquare$

## XII. EXTENSION TO MULTIGRAPHS

In this section we outline the extension of our positive results to multigraphs (the negative results carry over trivially).

*Fractional Algorithms.:* Our fractional results carry over unchanged to multigraphs. To see this, note that our algorithms’ analyses do not require the graph to be simple, as our analysis implies a bound on the maximum load after each edge has its value increased, and the relevant bounds do not require there to be no parallel edges.

*Randomized Algorithms.:* For multigraphs, we “merge” parallel edges into a single edge. When running MARKING on some fractional matching, we have a merged edge’s fractional assignment be the sum of its constituent edges’ fractional assignment. If each edge has multiplicity a sufficiently small  $o(\Delta)$  term, this would assign each edge a value of  $o(1)$ . By the properties of MARKING (Lemma 17), this implies that when we round a fractional matching  $\vec{x}$  to compute a matching  $\mathcal{M}$ , each edge  $e$  is matched in  $\mathcal{M}$  with probability  $x_e \cdot (1 - o(1)) \leq \Pr[e \in \mathcal{M}] \leq x_e$ . Our arguments carry through, though with possibly worse  $o(1)$  terms.

We note that the above stipulation that each edge have bounded multiplicity is necessary in order to obtain  $(1 + o(1))$  competitiveness for known  $\Delta$ .<sup>4</sup>

**Observation 38.** *No algorithm is  $(1 + o(1))$  competitive on multigraphs of arbitrary multiplicity.*

*Proof:* By [11], no online matching algorithm outputs a matching of expected size  $c \cdot n$  in 2-regular  $2n$ -vertex bipartite graphs under one-sided arrivals, for some constant  $c < 1$ . Given an input online 2-regular graph, we simulate the online arrival of a multigraph with  $k$  copies of each edge of the input (simple) graph, to obtain a  $2k$ -regular multigraph with each edge having multiplicity  $k$ . Given a  $(1 + \epsilon)$ -competitive edge coloring algorithm for multigraphs of maximum degree  $\Delta = 2k$ , we can randomly pick one of the  $(1 + \epsilon) \cdot 2k$  color classes upon initialization and output that matching. For 2-regular graphs on  $2n$  vertices, which have  $2 \cdot n$  edges, this results in a matching in the multigraph (which has  $2kn$  edges) of expected size  $\frac{2kn}{(1+\epsilon) \cdot 2k} = n/(1+\epsilon)$ , from which we conclude  $\epsilon = \Omega(1)$ . ■

### XIII. USEFUL PROBABILISTIC INEQUALITIES

For completeness, we cite here some useful probabilistic inequalities and notions of negative dependence, starting with the latter.

#### A. Negative Association and Other Negative Dependence Properties.

In our analysis we rely on several notions of negative dependence between random variables. In particular, one notion we will rely on is the notion of *negative association*, introduced by Khurshid and Lai Saxena [40] and Joag-Dev and Proschan [35].

**Definition 39** (Negative Association [35, 40]). *A joint distribution  $X_1, X_2, \dots, X_n$  is said to be negatively associated (NA) if for any two functions  $f, g$  both monotone increasing or both monotone decreasing, with  $f(\vec{X})$  and  $g(\vec{X})$  depending on disjoint subsets of the  $X_i$ ,  $f(\vec{X})$  and  $g(\vec{X})$  are negatively correlated; i.e.,*

$$\mathbb{E}[f(\vec{X}) \cdot g(\vec{X})] \leq \mathbb{E}[f(\vec{X})] \cdot \mathbb{E}[g(\vec{X})].$$

Clearly, independent random variables are NA. Another class of NA distributions is captured by the *zero-one rule*. This rule asserts that if  $X_1, X_2, \dots, X_n$  are zero-one random variables whose sum is always at most one,  $\sum_i X_i \leq 1$ , then  $X_1, X_2, \dots, X_n$  are NA (see [16]). Additional, more complex, NA distributions can be “built” from simpler NA distributions using the following closure properties.

(P1)

<sup>4</sup>Aggarwal et al. [1] showed that one cannot even achieve  $5/4 - \epsilon$  competitiveness in multigraphs with unbounded edge multiplicities, though under the possibly harder adversarial *edge arrival* model.

- 1) **Independent Union.** If  $X_1, X_2, \dots, X_n$  are NA,  $Y_1, Y_2, \dots, Y_m$  are NA, and  $\{X_i\}_i$  are independent of  $\{Y_j\}_j$ , then  $X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m$  are NA.
- 2) **Concordant monotone functions.** Let  $f_1, f_2, \dots, f_k : \mathbb{R}^n \rightarrow \mathbb{R}$  be functions, all monotone increasing or all monotone decreasing, with the  $f_i(\vec{X})$  depending on disjoint subsets of the  $\{X_i\}_i$ . Then, if  $X_1, X_2, \dots, X_n$  are NA, so are  $f_1(\vec{X}), f_2(\vec{X}), \dots, f_k(\vec{X})$ .

Negative association implies several useful properties, including the applicability of Chernoff-Hoeffding type bounds [16] (we elaborate on this below). In addition, NA clearly implies pairwise negative correlation. More generally, NA implies the stronger notion of *negative orthant dependence*.

**Definition 40.** *A joint distribution  $X_1, X_2, \dots, X_n$  is said to be Negative Upper Orthant Dependent (NUOD), if for all  $\vec{x} \in \mathbb{R}^n$  it holds that*

$$\Pr\left[\bigwedge_{i \in [n]} X_i \geq x_i\right] \leq \prod_{i \in [n]} \Pr[X_i \geq x_i],$$

and Negative Lower Orthant Dependent (NLOD) if for all  $\vec{x} \in \mathbb{R}^n$  it holds that

$$\Pr\left[\bigwedge_{i \in [n]} X_i \leq x_i\right] \leq \prod_{i \in [n]} \Pr[X_i \leq x_i].$$

A joint distribution is said to be Negative Orthant Dependent (NOD) if it is both NUOD and NLOD.

**Lemma 41** (NA variables are NOD ([16, 35])). *If  $X_1, \dots, X_n$  are NA, then they are NOD.*

In our analysis we will prove some scaled Bernoulli random variables are NUOD. To motivate our interest in this form of negative dependence, we note that for binary NUOD variables  $X_1, X_2, \dots, X_n$ , we have that for each set  $I \subseteq [n]$ ,  $\Pr[\bigwedge_{i \in I} X_i = 1] \leq \prod_{i \in I} \Pr[X_i = 1]$ . As shown by Panconesi and Srinivasan [50, proof of Theorem 3.2, with  $\lambda = 1$ ], this property implies that the moment generating function of the sum of the  $X_i$  is upper bounded by the moment generating function of the sum of *independent* copies of the  $X_i$  variables. A simple extension of their argument shows the same holds if the  $X_i$  are NUOD *scaled* Bernoulli variables. As in [50], following the standard proofs of Chernoff-Hoeffding type bounds, this upper bound on the moment generating function implies the applicability of the following upper tail bounds to the sum of NUOD scaled Bernoulli variables “as though these variables were independent”.

**Lemma 42** (Chernoff Bound for NUOD Bernoulli Variables, [50]). *Let  $X = \sum_i X_i$  be the sum of binary NUOD random variables  $X_1, X_2, \dots, X_n$ . Then, for any  $\epsilon \in [0, 1]$  and  $R \geq \mathbb{E}[X]$*

$$\Pr[X > (1 + \delta) \cdot R] \leq \exp\left(\frac{-\epsilon \cdot R}{3}\right).$$

**Lemma 43** (Bernstein’s Inequality for NUOD Scaled Bernoulli Variables). *Let  $X$  be the sum of NUOD random variables  $X_1, X_2, \dots, X_n$  with  $X_i \in \{0, M_i\}$  and  $M_i \leq M$  for each  $i \in [n]$ . Then, if  $\sigma^2 = \sum_{i=1}^n \text{Var}(X_i)$ , we have for all  $a > 0$ ,*

$$\Pr[X > \mathbb{E}[X] + a] \leq \exp\left(\frac{-a^2}{2(\sigma^2 + aM/3)}\right).$$

In addition we will use the following simple coupling argument, stated here for completeness.

**Lemma 44.** *Let  $X_1, X_2, \dots, X_m$  be random variables and  $Y_1, Y_2, \dots, Y_m$  be binary random variables such that  $Y_i = f_i(X_1, X_2, \dots, X_i)$  for all  $i$  such that for all  $\vec{x} \in \mathbb{R}^m$ ,*

$$\Pr\left[Y_i = 1 \mid \bigwedge_{\ell \in [i]} X_\ell = x_\ell\right] \leq p_i.$$

*Then, if  $Z_i = \text{Bernoulli}(p_i)$  are independent random variables, we have*

$$\Pr\left[\sum_i Y_i \geq k\right] \leq \Pr\left[\sum_i Z_i \geq k\right].$$

### B. Standard Concentration Inequalities

In addition, we will need the following standard tail bounds.

**Lemma 45** (Chernoff Bound). *Let  $X_1, X_2, \dots, X_m$  be independent random variables. If  $X_i \in [0, b]$  always, then, for all  $\epsilon \in (0, 1)$  and  $R \geq \mathbb{E}[X]$ , we have*

$$\Pr[X \geq (1 + \epsilon) \cdot R] \leq \exp\left(-\frac{R \cdot \epsilon^2}{3b}\right),$$

$$\Pr[X \leq (1 - \epsilon) \cdot \mathbb{E}[X]] \leq \exp\left(-\frac{\mathbb{E}[X] \cdot \epsilon^2}{2b}\right).$$

**Lemma 46** (Hoeffding’s Inequality). *Let  $X_1, X_2, \dots, X_m$  be independent random variables. If  $X_i \in [a_i, b_i]$  always, then, for all  $\delta$ ,*

$$\Pr[X \geq \mathbb{E}[X] + \delta] \leq \exp\left(-\frac{2\delta^2}{\sum_i (b_i - a_i)^2}\right),$$

$$\Pr[X \leq \mathbb{E}[X] - \delta] \leq \exp\left(-\frac{2\delta^2}{\sum_i (b_i - a_i)^2}\right).$$

### C. Conditional Union Bound

Finally, we will also need the following simple extension of union bound, proven below for completeness.

**Lemma 47.** *Let  $A_1, A_2, \dots, A_n$  be random indicator variables such that  $\Pr[\overline{A_i} \mid \bigwedge_{j < i} A_j] \leq p$ . Then*

$$\Pr\left[\bigvee_i \overline{A_i}\right] \leq n \cdot p.$$

*Proof:* Let  $B_i = \overline{A_i} \wedge \bigwedge_{j < i} A_j$  be the event that  $i$  is the first index  $j$  for which  $\overline{A_j}$  holds. Then

$$\Pr\left[\bigvee_i \overline{A_i}\right] = \Pr\left[\bigvee_i B_i\right] \leq \sum_i \Pr[B_i],$$

by standard union bound. But then, we find that the above is at most

$$\begin{aligned} \sum_i \Pr\left[\overline{A_i} \wedge \bigwedge_{j < i} A_j\right] &= \sum_i \Pr\left[\overline{A_i} \mid \bigwedge_{j < i} A_j\right] \cdot \Pr\left[\bigwedge_{j < i} A_j\right] \\ &\leq \sum_i \Pr[\overline{A_i} \mid A_{i-1}] \\ &\leq n \cdot p. \end{aligned}$$

■

### REFERENCES

- [1] AGGARWAL, G., MOTWANI, R., SHAH, D., AND ZHU, A. 2003. Switch scheduling via randomized edge coloring. In *Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS)*. 502–512.
- [2] ALON, N. 2003. A simple algorithm for edge-coloring bipartite multigraphs. *Information Processing Letters (IPL)* 85, 6, 301–302.
- [3] AZAR, Y., COHEN, I. R., AND ROYTMAN, A. 2017. Online lower bounds via duality. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1038–1050.
- [4] BAHMANI, B., MEHTA, A., AND MOTWANI, R. 2012. Online graph edge-coloring in the random-order arrival model. *Theory of Computing (TOC)* 8, 1, 567–595.
- [5] BAR-NOY, A., MOTWANI, R., AND NAOR, J. S. 1992. The greedy algorithm is optimal for on-line edge coloring. *Information Processing Letters (IPL)* 44, 5, 251–253.
- [6] BERGER, B. AND ROMPEL, J. 1991. Simulating  $(\log^c n)$ -wise independence in NC. *Journal of the ACM (JACM)* 38, 4, 1026–1046.
- [7] BHATTACHARYA, S., CHAKRABARTY, D., HENZINGER, M., AND NANONGKAI, D. 2018. Dynamic algorithms for graph coloring. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1–20.
- [8] BIRNBAUM, B. AND MATHIEU, C. 2008. On-line bipartite matching made simple. *ACM SIGACT News* 39, 1, 80–87.
- [9] BUCHBINDER, N., JAIN, K., AND NAOR, J. S. 2007. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Symposium on Algorithms (ESA)*. 253–264.
- [10] CHANG, Y.-J., LI, W., AND PETTIE, S. 2018. An optimal distributed  $(\Delta + 1)$ -coloring algorithm? In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*. 445–456.

- [11] COHEN, I. R. AND WAJC, D. 2018. Randomized online matching in regular graphs. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 960–979.
- [12] COLE, R., OST, K., AND SCHIRRA, S. 2001. Edge-coloring bipartite multigraphs in  $O(E \log D)$  time. *Combinatorica* 21, 1, 5–12.
- [13] DEVANUR, N. R., JAIN, K., AND KLEINBERG, R. D. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 101–107.
- [14] DUAN, R., HE, H., AND ZHANG, T. 2019. Dynamic edge coloring with improved approximation. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. To Appear.
- [15] DUBHASHI, D., GRABLE, D. A., AND PANCONESI, A. 1998. Near-optimal, distributed edge colouring via the nibble method. *Theoretical Computer Science (TCS)* 203, 2, 225–252.
- [16] DUBHASHI, D. AND RANJAN, D. 1996. Balls and bins: A study in negative dependence. *BRICS Report Series* 3, 25.
- [17] EDEN, A., FELDMAN, M., FIAT, A., AND SEGAL, K. 2018. An economic-based analysis of ranking for online bipartite matching. *arXiv preprint arXiv:1804.06637*.
- [18] EHMSSEN, M. R., FAVRHOLDT, L. M., KOHRT, J. S., AND MIHAI, R. 2010. Comparing first-fit and next-fit for online edge coloring. *Theoretical Computer Science (TCS)* 411, 16-18, 1734–1741.
- [19] ELKIN, M., PETTIE, S., AND SU, H.-H. 2015.  $(2\Delta-1)$ -edge-coloring is much easier than maximal matching in the distributed setting. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 355–370.
- [20] FAVRHOLDT, L. M. AND MIKKELSEN, J. W. 2014. Online dual edge coloring of paths and trees. In *Proceedings of the 12th Workshop on Approximation and Online Algorithms (WAOA)*. 181–192.
- [21] FAVRHOLDT, M. AND NIELSEN, N. 2003. On-line edge-coloring with a fixed number of colors. *Algorithmica* 35, 2, 176–191.
- [22] FEIGE, U. 2018. Tighter bounds for online bipartite matching. *arXiv preprint arXiv:1812.11774*.
- [23] FELDMAN, J., MEHTA, A., MIRROKNI, V., AND MUTHUKRISHNAN, S. 2009. Online stochastic matching: Beating  $1-1/e$ . In *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS)*. 117–126.
- [24] FISCHER, M., GHAFFARI, M., AND KUHN, F. 2017. Deterministic distributed edge-coloring via hypergraph maximal matching. In *Proceedings of the 58th Symposium on Foundations of Computer Science (FOCS)*. 180–191.
- [25] GABOW, H. N., NISHIZEKI, T., KARIV, O., LEVEN, D., AND OSMAU, T. 1985. Algorithms for edge-coloring graphs. Tech. rep., Tohoku University.
- [26] GAMLATH, B., KAPRALOV, M., MAGGIORI, A., SVENSSON, O., AND WAJC, D. 2019. Online matching with general arrivals. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*. To Appear.
- [27] GANDHAM, S., DAWANDE, M., AND PRAKASH, R. 2008. Link scheduling in wireless sensor networks: Distributed edge-coloring revisited. *Journal of Parallel and Distributed Computing* 68, 8, 1122–1134.
- [28] GHAFFARI, M., KUHN, F., MAUS, Y., AND UITTO, J. 2018. Deterministic distributed edge-coloring with fewer colors. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*. 418–430.
- [29] GOEL, A., KAPRALOV, M., AND KHANNA, S. 2013. Perfect matchings in  $O(n \log n)$  time in regular bipartite graphs. *SIAM Journal on Computing (SICOMP)* 42, 3, 1392–1404.
- [30] GOEL, G. AND MEHTA, A. 2008. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 982–991.
- [31] GOLDBERG, M. K. 1973. On multigraphs of almost maximal chromatic class. *Diskret. Analiz* 23, 3, 7.
- [32] HOLYER, I. 1981. The NP-completeness of edge-coloring. *SIAM Journal on Computing (SICOMP)* 10, 4, 718–720.
- [33] HUANG, Z., KANG, N., TANG, Z. G., WU, X., ZHANG, Y., AND ZHU, X. 2018. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*. 17–29.
- [34] HUANG, Z., PENG, B., TANG, Z. G., TAO, R., WU, X., AND ZHANG, Y. 2019. Tight competitive ratios of classic matching algorithms in the fully online model. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. To Appear.
- [35] JOAG-DEV, K. AND PROSCHAN, F. 1983. Negative association of random variables with applications. *The Annals of Statistics*, 286–295.
- [36] KALYANASUNDARAM, B. AND PRUHS, K. R. 2000. An optimal deterministic algorithm for online  $b$ -matching. *Theoretical Computer Science (TCS)* 233, 1, 319–325.
- [37] KARANDE, C., MEHTA, A., AND TRIPATHI, P. 2011. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*. 587–596.
- [38] KARLOFF, H. J. AND SHMOYS, D. B. 1987. Efficient parallel algorithms for edge coloring problems. *J. Algorithms* 8, 1, 39–52.
- [39] KARP, R. M., VAZIRANI, U. V., AND VAZIRANI, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM*

- Symposium on Theory of Computing (STOC)*. 352–358.
- [40] KHURSHEED, A. AND LAI SAXENA, K. 1981. Positive dependence in multivariate distributions. *Communications in Statistics - Theory and Methods* 10, 12, 1183–1196.
- [41] LEV, G. F., PIPPENGER, N., AND VALIANT, L. G. 1981. A fast parallel algorithm for routing in permutation networks. *IEEE transactions on Computers* 2, 93–100.
- [42] LOVÁSZ, L. AND PLUMMER, M. D. 2009. *Matching theory*. Vol. 367. American Mathematical Society.
- [43] MAHDIAN, M. AND YAN, Q. 2011. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*. 597–606.
- [44] MEHTA, A. 2013. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science* 8, 4, 265–368.
- [45] MIKKELSEN, J. W. 2015. Optimal online edge coloring of planar graphs with advice. In *International Conference on Algorithms and Complexity*. Springer, 352–364.
- [46] MIKKELSEN, J. W. 2016. Randomization can be as helpful as a glimpse of the future in online computation. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP)*. 39.
- [47] MISRA, J. AND GRIES, D. 1992. A constructive proof of vizing’s theorem. In *Information Processing Letters (IPL)*.
- [48] MOTWANI, R., NAOR, J. S., AND NAOR, M. 1994. The probabilistic method yields deterministic parallel algorithms. *Journal of Computer and System Sciences* 49, 3, 478–516.
- [49] NAOR, J. S. AND WAJC, D. 2015. Near-optimum online ad allocation for targeted advertising. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*. 131–148.
- [50] PANCONESI, A. AND SRINIVASAN, A. 1997. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing (SICOMP)* 26, 2, 350–368.
- [51] PETERSEN, J. 1898. Sur le théoreme de tait. *L’intermédiaire des Mathématiciens* 5, 225–227.
- [52] RASALA, A. AND WILFONG, G. 2005. Strictly nonblocking wdm cross-connects. *SIAM Journal on Computing (SICOMP)* 35, 2, 449–485.
- [53] SEYMOUR, P. D. 1979. On multi-colourings of cubic graphs, and conjectures of fulkerson and tutte. *Proceedings of the London Mathematical Society* 3, 3, 423–460.
- [54] SHANNON, C. E. 1949. A theorem on coloring the lines of a network. *Journal of Mathematics and Physics* 28, 1-4, 148–152.
- [55] TAIT, P. 1880. Remarks on the colourings of maps. *Proc. R. Soc. Edinburgh* 10, 729.
- [56] TASSIULAS, L. AND EPHREIMIDES, A. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE transactions on automatic control* 37, 12, 1936–1948.
- [57] VIZING, V. G. 1964. On an estimate of the chromatic class of a p-graph. *Diskret analiz* 3, 25–30.
- [58] WANG, Y. AND WONG, S. C.-W. 2015. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP)*. 1070–1081.