Almost Tight Bounds for Online Hypergraph Matching

Thorben Tröbst[a,*], Rajan Udwani[b]

[a]*Department of Computer Science*
University of California, Irvine
[b]*Department of Industrial Engineering and Operations Research*
University of California, Berkeley

Abstract

In the online hypergraph matching problem, hyperedges of size k over a common ground set arrive online in adversarial order. The goal is to obtain a maximum matching (disjoint set of hyperedges). A naïve greedy algorithm for this problem achieves a competitive ratio of $\frac{1}{k}$. We show that no (randomized) online algorithm has competitive ratio better than $\frac{2+o(1)}{k}$. If edges are allowed to be assigned fractionally, we give a deterministic online algorithm with competitive ratio $\frac{1-o(1)}{\ln(k)}$ and show that no online algorithm can have competitive ratio strictly better than $\frac{1+o(1)}{\ln(k)}$. Lastly, we give a $\frac{1-o(1)}{\ln(k)}$ competitive algorithm for the fractional *edge-weighted* version of the problem under a free disposal assumption.

Keywords: online algorithms, hypergraph matching, set packing

1. Introduction

In the classic problem of online bipartite matching problem, we have a bipartite graph between resources (offline vertices) and agents (online vertices). Agents arrive sequentially and edges incident on an agent are revealed on arrival. Each agent must be matched to at most one available resource. Matches are immediate and irrevocable with the goal of maximizing the size of the overall matching. We evaluate the performance of an online algorithm using its competitive ratio. The competitive ratio of an algorithm is the worst case ratio between the expected cardinality of matchings produced by said algorithm and the optimal offline solution, which, i.e. the maximum cardinality matching in hindsight.

Online bipartite matching captures the essence of resource allocation under sequential and heterogeneous demand [12]. Fundamental its many variations [18], is the assumption that every agent seeks at most one type of resource. In a variety of settings, ranging from revenue management in airlines [20, 21] to combinatorial auctions [13, 14] and ridesharing [19], agents require a bundle (set) of resources and an allocation that does not include every resource in the bundle has no value. Inspired by this, we study a fundamental generalization of online bipartite matching, namely, online hypergraph matching. A hypergraph is a generalization of a graph where an edge can join any number of vertices. The maximum cardinality of an edge, or hyperedge, is the rank of the hypergraph, i.e. hyperedges of a rank k hypergraph have at most k vertices. A matching in a hypergraph is a set of vertex-disjoint edges. In online hypergraph matching, edges are revealed sequentially in adversarial order. On arrival of an edge, we must make an immediate an irrevocable decision to include the edge in the matching or reject it forever. The objective is to maximize the size of the matching. In the fractional version of the problem, arriving edges can be fractionally included in the matching, subject to the constraint that the total fraction of edges incident on any vertex is at most one.

In case of online bipartite matching, the offline optimum can be found in polynomial time. In sharp contrast, for a rank k hypergraph it is NP hard to find a $\frac{\Omega(\log k)}{k}$ approximation for the offline hypergraph matching problem [9]. As k increases, the separation between these problems widens since the hypergraph matching problem becomes harder to approximate. In this paper, our goal is to find tight upper and lower bounds for online hypergraph matching for large k. We consider both the integral and fractional version of the online hypergraph matching problem.

In the integral case, it can be shown that a simple greedy algorithm that always includes an arriving edge in the matching, if feasible, is $\frac{1}{k}$ competitive. In fact, no deterministic algorithm can do better in the worst case. From the hardness of the offline problem, no polynomial time (online) algorithm can have competitive ratio better than $\frac{\Omega(\log k)}{k}$ (unless P=NP).

In the fractional version, we evaluate the competitive ratio against a fractional relaxation of the offline problem that can be solved in polynomial time using linear programming. Therefore, the computational hardness disappears and a result for the online packing problem [3] gives a $\Omega(\frac{1}{\log k})$ competitive algorithm for online hypergraph matching. This exponential gap between the fractional and integral setting raises the following natural question:

For some $\epsilon > 0$, is there an (exponential time) online algorithm with competitive ratio $\frac{\Omega(1)}{k^{1-\epsilon}}$ for (integral) online hypergraph matching?

[*]Corresponding Author, postal address: Donald Bren Hall, Irvine, CA 92697
Email addresses: `t.troebst@uci.edu` (Thorben Tröbst), `rudwani@berkeley.edu` (Rajan Udwani)

We answer this question in the negative and establish the following result.

Theorem (Informal)**.** *No (randomized and exponential time) online algorithm can achieve a competitive ratio better than $\frac{2+o(1)}{k}$ for online hypergraph matching.*

Unlike the state-of-the-art complexity theoretic upper bound of $\frac{O(\log k)}{k}$, our result is unconditional and arises from the online nature of the problem (as opposed to any computational barriers). For the fractional case, we give new upper and lower bounds, closing the existing constant factor gap for large k.

Theorem (Informal)**.** *There is an efficient algorithm for fractional online hypergraph matching with competitive ratio $\frac{1-o(1)}{\ln k}$. This is the best possible (asymptotic) competitive ratio for any online algorithm.*

2. Model

There are two reasonable variants of online hypergraph matching: edge arrival and vertex arrival. The edge arrival model is the one we have discussed so far: we are given a set of offline vertices and the hyperedges arrive online in adversarial order. However, a more direct generalization of the online bipartite matching problem would be vertex arrival: we are given a set of offline vertices I, online vertices T, and whenever an online vertex t arrives (in adversarial order), all hyperedges incident to t are revealed at once.

Note that edge arrival is strictly harder to deal with than vertex arrival. Moreover, for large k, edge arrival is also no harder than vertex arrival since we can replace any edge arrival instance of rank k with a vertex arrival instance of rank $k+1$ by simply adding one unique vertex for every edge. Since we are interested in the large k case, we will only consider edge arrival throughout the remainder of the paper.

In the fractional version, the optimal offline solution in the edge arrival model is a solution to the following linear program,

$$\textbf{Fractional LP:} \quad \begin{array}{lllr} \max\limits_{\{x_e\}_{e\in E}} & \sum_{e\in E} x_e, & & \\ s.t. & \sum_{e\ni i} x_e & \leq 1, & \forall i \in I, \\ & x_e & \geq 0 & \forall e \in E. \end{array}$$

Decision variable $x_e \in [0,1]$ in the LP captures the fraction of edge e included in the matching. The constraint, $\sum_{e\ni i} x_e \leq 1$, enforces the total fraction of edges incident on a resource $i \in I$ to be at most 1. An online algorithm can include an arbitrary fraction of each arriving edge, subject to the same constraint as the LP.

Finally, note that by adding dummy resources we can assume w.l.o.g., that every edge has exactly k offline vertices. Unless stated otherwise, in the rest of the paper we consider the edge arrival model and assume that instances are k-uniform, i.e., every edge intersects k offline vertices.

2.1. Related Work

Perhaps closest to our setting is the work of Buchbinder and Naor [3] on online packing. They considered an online packing problem that generalizes the fractional online hypergraph matching problem studied here and gave a $O(\log k)$ competitive algorithm. A special case of the online packing problem was considered earlier in [1], in the context of online routing.

Another closely related line of work is on the problem of network revenue management [20, 21]. This is a stochastic arrival setting where seats in flights are offline resources allocated to sequentially arriving customers. A customer with multi-stop itinerary requires a seat on each flight in the itinerary. Recently, Ma et al. [16] gave a $\frac{1}{k+1}$ algorithm for network revenue management.

Another stream of work has focused on hypergraph matching from the perspective of ridesharing. Pavone et al. [19] introduced a hypergraph matching problem with deadlines to capture applications in ridesharing. Their model and results are incomparable to ours. Lowalekar et al. [15] consider a model inspired by ridesharing but with a stochastic arrival sequence. Finally, [13, 14] consider related settings in combinatorial auctions that correspond to online hypergraph matching with stochastic arrivals.

For the offline hypergraph matching problem, Hazan et al. [9] showed that unless P=NP, no polynomial time algorithm can find a matching better than $O(\frac{\log k}{k})$ of the optimum matching. [4, 5] give approximation algorithms for the problem. To the best of our knowledge, the state-of-the-art result is a $\frac{3}{k}$ approximation due to Cygan [5].

Lastly, we want to mention the recent work of Borst et al. [2] which appeared after this paper was first announced (but before it was published). They consider the integral vertex-arrival variant of the online hypergraph matching and give an optimal algorithm in the case of $k = 3$ which turns out to have a competitive ratio of $(e-1)/(e+1) \approx 0.4621$. In addition, they give an algorithm which beats the greedy algorithm for larger k under the assumption that the degree of online vertices is bounded.

3. Integral Matchings

In the following, fix some $k \geq 2$. We will focus on the online hypergraph matching problem in the edge arrival model. We start with a result that is folklore in the literature on online matching [18].

Theorem 1 (Folklore)**.** *There is a $\frac{1}{k}$-competitive algorithm for the online hypergraph matching problem. This is the best possible competitive ratio for deterministic algorithms.*

Proof. Consider the online algorithm that includes an arriving edge e in the matching if it is disjoint with all previously included edges. Let $e_1, \ldots, e_\ell$ be the set of edges included in an offline optimum solution and let $I_o \subseteq I$ denote the set of offline vertices that are covered by the edges chosen in the online algorithm. For each edge e_i, at least one of the resources that it intersects must be included in I_o. Thus, $|I_o| \geq \frac{1}{k}\ell\, k$ and the online algorithm picks at least $\frac{1}{k}\ell$ hyper edges.

To see that this is the best possible competitive ratio, consider two arrival sequences. In the first sequence, we have a single arrival. In the second sequence, we augment the first sequence with k more arrivals such that the optimal offline matching has size k but the first edge intersects every other edge. A deterministic algorithm with non-zero competitive ratio must match the first arrival. □

In our first result, we show that even a randomized and possibly exponential time online algorithm cannot achieve a much better competitive ratio for this problem. To show this, we will use Yao's minimax principle, as stated below.

Lemma 2 (Yao's Principle). *Let α be the best competitive ratio of any randomized algorithm. Let β be the competitive ratio of the best deterministic algorithm against some fixed distribution of instances. Then $\alpha \leq \beta$.*

Before we get to our main result, we will first give a slightly weaker result that serves both as a warm up and as a gadget for the main result.

Theorem 3. *For even k, there does not exist a $\frac{4+\epsilon}{k}$ competitive algorithm for the k-uniform online hypergraph matching problem for any $\epsilon > 0$.*

Proof. Using Yao's principle, we will construct a distribution of instances with even k where OPT $= \frac{k}{2}$ but the best deterministic online algorithm can only achieve an expected matching size of 2.

For any given even value k, the overall (random) instance G_k will consist of $\frac{k}{2}$ "red" edges and $\frac{k}{2}$ "blue" edges constructed in $\frac{k}{2}$ phases. In each phase, there will be one red and one blue edge which look indistinguishable to any online algorithm. The idea is that if the algorithm ever picks a blue edge, it will be locked out of future edges, thus limiting the expected matching size. See Figure 1 for an example of the construction. The construction of the red and blue edge in each phase proceeds as follows:

1. Let A be a set of vertices which intersects every previous blue edge exactly once. Create a new edge e_1 which consists of A and $k - |A|$ many new vertices that have not been in any edges yet. Let e_1 arrive in the instance.
2. Now let A' be a second set of vertices which intersects every previous blue edge *and* e_1 exactly once. Create a new edge e_2 which consists of A' and $k - |A'|$ many new vertices. Let e_2 arrive in the instance.
3. Randomly let one of $\{e_1, e_2\}$ be red and the other blue with equal probability.

Note that the sets A and A' can always be found because each edge contains k vertices and we have $k/2$ phases. The crucial property of this construction is that each blue edge intersects all future edges whereas each red edge is disjoint from all future edges. In particular, the $\frac{k}{2}$ red edges form a maximum size matching, i.e. OPT $= \frac{k}{2}$.

Now consider some deterministic online algorithm $\mathcal{A}$. Let α_i be the probability that $\mathcal{A}$ matches the red edge in phase i and let β_i be the probability that $\mathcal{A}$ matches the blue edge in phase i. Clearly, since the red and blue edges are determined independently and uniformly at random, we must have $\alpha_i = \beta_i$. Moreover, since at most one blue edge can be picked, we know $\alpha_1 + \cdots + \alpha_{k/2} \leq 1$. Thus the expected size of the matching generated by $\mathcal{A}$ is at most

$$\alpha_1 + \cdots + \alpha_{k/2} + \beta_1 + \cdots + \beta_{k/2} \leq 2.$$ □

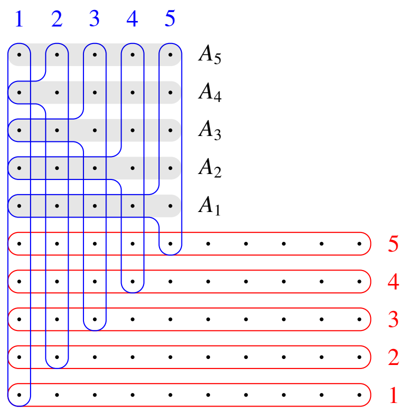

Figure 1: Shown is gadget G_{10} proving that a competitive ratio of $\frac{4}{k} + \epsilon$ is imposisble for $k = 10$. The numbers indicate in which phase each edge was added. The lightly shaded areas represent the vertex sets $A_1, \ldots, A_5$ which are useful for the construction of H_k.

Theorem 4. *If k is a power of two, then here does not exist a $\frac{2+\epsilon}{k}$ competitive algorithm for the online hypergraph matching problem for any $\epsilon > 0$.*

Proof. We will use induction to create a distribution over graphs H_k for powers of two k, with the following properties:

1. There are k red and k blue edges.
2. The edges appear in k phases, each of which consists of one red and one blue edge where the color is chosen uniformly and independently at random.
3. Every blue edge intersects all future edges.
4. Every red edge is disjoint from all future edges.

H_1 is trivial to construct. We will just have a single vertex which is simultaneously in both a red and blue singleton edge. Suppose that we can construct $H_{k/2}$.

Now in order to construct H_k, we first employ the $\frac{k}{2}$ phases of G_k. After this we can construct $\frac{k}{2}$ disjoint sets $A_1, \ldots, A_{k/2}$ of $\frac{k}{2}$ vertices such that each A_i intersects all blue edges and none of the red edges. See again Figure 1. Now, for the remaining $k/2$ phases, we recursively employ the distribution $H_{k/2}$ as follows. In phase $i + \frac{k}{2}$, we extend the the two edges from phase i of $H_{k/2}$ by the set A_i to form sets of size k. This gives us the two edges of rank k for phase $i + \frac{k}{2}$.

Finally, one may check that properties $1 - 4$ are satisfied by induction. Thus, we may conclude the proof similar to the proof of Theorem 3: the optimum solution picks all k red edges whereas any deterministic online algorithm can only get an expected value of 2 since at most one blue edge can be picked and red and blue edges in each phase are indistinguishable for the online algorithm. ☐

4. Fractional Matchings

Inspired by the Waterfilling (or Balancing) algorithm of [11, 17], designed for online bipartite matching and its variants, we propose the following online algorithm for fractional hypergraph matching in the edge arrival model.

Algorithm 1: HYPERGRAPH WATER-FILLING

```
1 For each $e \in E$, set $y_e := 0$.
2 for each edge e which arrives do
3     Increase $y_e$ continuously as long as
      $\sum_{i\in e}(k \ln(k))^{x_i-1} \leq 1$ where $x_i := \sum_{f\in E: i\in f} y_f$.
```

The final value of variable y_e in the algorithm is the fraction of edge e that is included in the matching. At any moment, variable x_i in the algorithm captures the total fraction of edges incident on resource i that have been included in the matching. In other words, the value of x_i is the fraction of i that has already been "matched" by the algorithm. In order to "preserve" resources for future edges, Algorithm 1 stops matching an edge when the value of $\sum_{i\in e}(k \ln(k))^{x_i-1}$ reaches 1. For illustration, consider the following scenarios at the arrival of edge e incident on resources $\{1, 2, \cdots, k\}$.

(i) $x_i = 0$ $\forall i \in [k]$ on arrival of e. Then, the algorithm will match $\frac{\ln(\ln k)}{\ln(k)+\ln(\ln k)}$ fraction of the edge before stopping.

(ii) $x_i = 0.5$ $\forall i \leq \sqrt{k\ln(k)}$ and $x_i = 0$ otherwise. Then, for sufficiently large values of k, the algorithm does not match any fraction of edge e.

We would like to note that Algorithm 1 constructs a fractional matching by augmenting the primal solution y, whereas the online packing algorithm of [3] augments the dual solution. We show that Algorithm 1 achieves the best possible competitive ratio guarantee for large k.

Theorem 5. *Algorithm 1 is $\frac{1-o(1)}{\ln(k)}$-competitive for the fractional online hypergraph matching problem.*

Proof. Given a hypergraph G with resources I and set of hyperedges E, let ALG denote the total revenue of the online algorithm and let OPT denote the value of the optimal fractional offline solution. We use a primal-dual approach inspired by [3, 6] to prove the result. Using weak duality, it suffices to find a feasible solution of non-negative $(r_i)_{i\in I}$ to the following system of linear inequalities,

$$\sum_{i\in I} r_i \leq \text{ALG}, \tag{1}$$

$$\sum_{i\in e} r_i \geq \frac{1-\frac{1}{\ln(k)}}{\ln(k)+\ln(\ln(k))} \quad \forall e \in E. \tag{2}$$

We set the dual variables using the following procedure. In the beginning, all variables r_i are set to 0. When Algorithm 1 is matching edge e in line 3 by some infinitesimal amount dt, we increase r_i by $(k\ln(k))^{x_i-1}\mathrm{d}t$ for all $i \in e$. Note that by the condition in line 3, we know that $\sum_{i\in} r_I \leq$ ALG at the end of the algorithm. It remains to show that (2) is satisfied. Fix an arbitrary edge $e \in E$ and consider the following two cases.

Case 1: Let $e \in E$ be arbitrary and let x_i be the final fill levels of the vertices $i \in e$. If $x_i = 1$ for any i, we know that

$$r_i = \int_0^1 (k\ln(k))^{t-1}\,\mathrm{d}t = \frac{1-1/(k\ln(k))}{\ln(k)+\ln(\ln(k))} \geq \frac{1-1/\ln(k)}{\ln(k)+\ln(\ln(k))},$$

so the this one vertex is already enough for (2).

Case 2: Otherwise, since all $x_i < 1$ at the end of the algorithm, we must have that $P := \sum_{i\in e}(k\ln(k))^{x_i-1} \geq 1$. But in that case, we can compute:

$$\sum_{i\in e} r_i \geq \sum_{i\in e}\int_0^{x_i}(k\ln(k))^{t-1}\,\mathrm{d}t = \frac{P-1/\ln(k)}{\ln(k)+\ln(\ln(k))}$$

which shows the claim and thus the theorem. ☐

Next, we show that this bound is tight, i.e. that no online algorithm can beat the performance of Algorithm 1 for large k.

Theorem 6. *For any $\epsilon > 0$ and k large enough, there does not exist any online algorithm which is $\frac{1+\epsilon}{\ln(k)}$-competitive for the online hypergraph matching problem.*

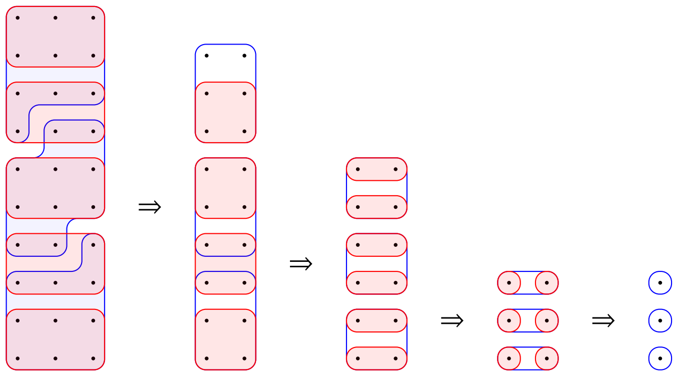

Figure 2: Shown is the upper-bounding construction with $k = 10$, $l = 3$, $\delta = 0.5$. In each step we replace the blue edges with as many red edges of $\frac{1}{1+\delta}$ times the size as possible. Then we pick the l red edges that the algorithm puts the most weight on, make those the new blue edges and repeat until only singleton edges are left.

Proof. Let $\mathcal{A}$ be some algorithm for the fractional online hypergraph matching problem. We can assume w.l.o.g., that $\mathcal{A}$ is deterministic. This is because if $\mathcal{A}$ is randomized, we may create another algorithm $\mathcal{A}'$ which simply fractionally allocates every edge e with the expected value of $\mathcal{A}$. Then $\mathcal{A}'$ is a deterministic algorithm that performs just as well as $\mathcal{A}$.

Fix some large $l \in \mathbb{N}$ and small $\delta > 0$. We will now construct an instance in which some hyperedges may have less than k

vertices. The instance is created according to the following procedure (see Figure 2).

1. Set $m \leftarrow k$ and let l disjoint edges of size k arrive. Let U be the set of all vertices in these l edges.
2. If $m = 0$, stop. Otherwise, set $m \leftarrow \left\lfloor \frac{m}{1+\delta} \right\rfloor$.
3. Partition U into as many disjoint edges of size m as possible and let these arrive.
4. Let $e_1, \ldots, e_l$ be the l of these edges that $\mathcal{A}$ matches the most.
5. Update $U \leftarrow e_1 \cup \cdots \cup e_l$ and go back to step 2.

Now let α be the competitive ratio of $\mathcal{A}$. Our first observation is that steps 2–5 execute $(1 - o(1)) \log_{1+\delta}(k)$ many times as $k \to \infty$. Moreover, in each iteration, we cover l blue edges with

$$\left\lfloor \frac{lm}{\left\lfloor \frac{m}{1+\delta} \right\rfloor} \right\rfloor \geq (1+\delta)l - 1$$

red edges. The optimal solution can thus be increased by at least $\delta l - 1$ by shifting weight from the blue edges to the red edges. Overall, this yields $\text{OPT} \geq (1 - o(1)) \log_{1+\delta}(k)(\delta l - 1)$ and therefore $\text{ALG} \geq \alpha(1 - o(1)) \log_{1+\delta}(k)(\delta l - 1)$.

Let $E^\star \subseteq E$ be the set of edges which are picked at various points in step 4 and let y be the fractional matching constructed by $\mathcal{A}$. Then because these edges are always the l most covered edges we know that

$$y(E^\star) \geq \min_{m \geq 1} \frac{l}{\left\lfloor \frac{lm}{\left\lfloor \frac{m}{1+\delta} \right\rfloor} \right\rfloor} \text{ALG} \geq \frac{1}{1 + \delta - \frac{1}{l}} \text{ALG}.$$

Lastly, we know that that all edges in $E^\star$ overlap in l vertices, namely the l vertices that are contained in the final iteration of the loop. This implies that $y(E^\star) \leq l$. Combining these inequalities, we thus get

$$\begin{aligned}\alpha &\leq \frac{\left(1 + \delta - \frac{1}{l}\right) l}{(1 - o(1)) \log_{1+\delta}(k)(\delta l - 1)} \\ &= \frac{((1+\delta)l - 1)\ln(1+\delta)}{(1 - o(1))(\delta l - 1)} \cdot \frac{1}{\ln(k)}\end{aligned}$$

Finally, observe that for small δ, large l and large k, we get $\alpha < \frac{1+\epsilon}{\ln(k)}$ as claimed. □

5. Edge-Weights

In this section, we consider the free disposal model for online weighted matching [7], and generalize Algorithm 1 to online fractional weighted hypergraph matching with free disposal. In this model, we are allowed to drop previously matched edges with no penalty (though of course we do not count such dropped edges towards our objective function).

Theorem 7. *For any $\epsilon > 0$ and k large enough, Algorithm 2 is $\frac{1-\epsilon}{\ln(k)}$-competitive for online fractional weighted hypergraph matching problem with free disposal.*

Algorithm 2: Hypergraph Weighted Water-Filling

```
1 For each $e \in E$, let $y_e := 0$.
2 For each $i \in V$, let $f_i(t) := \sum_{e: i \in e, w_e \geq t} y_e$ for all $t \geq 0$.
3 for each edge e which arrives do
4     while $\sum_{i \in e} \int_0^{w_e} (k\ln(k))^{f_i(t)-1}\,\mathrm{d}t \leq w_e$ do
5         for $i \in e$ with $x_i = 1$ do
6             Let $e_i^-$ be a minimum weight edge with $i \in e_i^-$ and $y_{e_i^-} > 0$.
7             $y_{e_i^-} \leftarrow y_{e_i^-} - \mathrm{d}s$
8         $y_e \leftarrow y_e + \mathrm{d}s$
```

Proof. The proof will use a similar primal-dual approach as Theorem 5 with non-negative dual variables r_i for all $i \in V$. Once again, when Algorithm 2 is matching an edge e by some amount $\mathrm{d}s$, we increase the dual by (at most) an according amount. Note that the total increase in the matching is $w_e - \sum_{i \in e} w_{e_i^-}$ where $w_{e_i^-} := 0$ if $x_i < 1$.

We increase each r_i by $\int_{w_{e_i^-}}^{w_e} (k\ln(k))^{f_i(t)-1}\,\mathrm{d}t\mathrm{d}s$. By definition of f_i, we know that $f_i(t) = 1$ for all $t < w_{e_i^-}$ and thus

$$\begin{aligned}&\sum_{i \in e} \int_{w_{e_i^-}}^{w_e} (k\ln(k))^{f_i(t)-1}\,\mathrm{d}t \\ &= \sum_{i \in e} \int_0^{w_e} (k\ln(k))^{f_i(t)-1}\,\mathrm{d}t - \sum_{i \in e} \int_0^{w_{e_i^-}} (k\ln(k))^{f_i(t)-1}\,\mathrm{d}t \\ &\leq w_e - \sum_{i \in e} w_{e_i^-}\end{aligned}$$

using the condition in line 4 of the algorithm. This implies that at the end of the algorithm we have $\sum_{i \in V} r_i \leq \text{ALG}$.

Claim: For any $e \in E$, we have

$$\sum_{i \in e} r_i \geq w_e \frac{1 - 1/\ln(k)}{\ln(k) + \ln(\ln(k))}.$$

Proof of Claim: Let f_i be the step function defined in Algorithm 2 at the end of the algorithm. Then the total r_i collected by each $i \in V$ satisfies

$$\begin{aligned}r_i &= \int_0^\infty \int_0^{f_i(t)} (k\ln(k))^{s-1}\,\mathrm{d}s\,\mathrm{d}t \\ &= \int_0^\infty \frac{\sum_{i \in e}(k\ln(k))^{f_i(t)-1} - 1/\ln(k)}{\ln(k) + \ln(\ln(k))}\,\mathrm{d}t\end{aligned}$$

Now let $P(t) := \sum_{i \in e}(k\ln(k))^{f_i(t)-1}$, then this means that

$$\begin{aligned}\sum_{i \in e} r_i &= \int_0^\infty \frac{P(t) - 1/\ln(k)}{\ln(k) + \ln(\ln(k))}\,\mathrm{d}t \\ &\int_0^{w_e} \frac{P(t) - 1/\ln(k)}{\ln(k) + \ln(\ln(k))}\,\mathrm{d}t \\ &= \frac{\int_0^{w_e} P(t)\,\mathrm{d}t - w_e/\ln(k)}{\ln(k) + \ln(\ln(k))}.\end{aligned}$$

Finally, by the condition in line 4 of the algorithm and the fact that $f_i(t)$ only increases during the algorithm for all t, we know that $\int_0^{w_e} P(t)\mathrm{d}t \geq w_e$ which establishes the claim and thus the theorem. □

6. Conclusion

In this paper we have given a tight asymptotic bound for the fractional k-uniform hypergraph matching problem and an almost tight bound for the integral variant. This leaves room for some interesting directions for future research:

A major open problem is to beat the $\frac{1}{k}$ lower bound in the integral setting. In fact, just recently Gamlath et al. [8] showed that for $k = 2$, no algorithm beats $\frac{1}{2}$, even if the underlying graph is bipartite. However, their construction in fact shows this result for the fractional setting and where we know how to beat $\frac{1}{k}$ for large k. It thus remains open whether $\frac{1}{k} + \epsilon$ is achievable for *any* k.

In fact, for small k, one may also explicitly distinguish between edge arrival and vertex arrival models as mentioned in Section 2 or even the fully-online arrival model of Huang et al. [10]. To the best of our knowledge, the only result here is the recent one by Borst et al. [2] who managed to get an optimal algorithm for the $k = 3$ case under vertex arrivals.

Finally, we remark that even in the fractional setting, exactly tight bounds are only known for $k = 2$ and finding a tight non-asymptotic result remains open.

Acknowledgements

We would like to thank Vijay Vazirani for several helpful comments during both the research and writing process.

References

[1] Awerbuch, B., Azar, Y., Plotkin, S.: Throughput-competitive on-line routing. In: Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science. pp. 32–40. IEEE (1993)

[2] Borst, S., Kashaev, D., Koh, Z.K.: Online matching on 3-uniform hypergraphs. arXiv preprint arXiv:2402.13227 (2024)

[3] Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing. Mathematics of Operations Research **34**(2), 270–286 (2009)

[4] Chan, Y.H., Lau, L.C.: On linear and semidefinite programming relaxations for hypergraph matching. Mathematical programming **135**(1), 123–148 (2012)

[5] Cygan, M.: Improved approximation for 3-dimensional matching via bounded pathwidth local search. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. pp. 509–518. IEEE (2013)

[6] Devanur, N.R., Jain, K., Kleinberg, R.D.: Randomized primal-dual analysis of ranking for online bipartite matching. In: Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms. pp. 101–107. SIAM (2013)

[7] Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., Pál, M.: Online ad assignment with free disposal. In: International workshop on internet and network economics. pp. 374–385. Springer (2009)

[8] Gamlath, B., Kapralov, M., Maggiori, A., Svensson, O., Wajc, D.: Online matching with general arrivals. pp. 26–37 (11 2019). https://doi.org/10.1109/FOCS.2019.00011

[9] Hazan, E., Safra, S., Schwartz, O.: On the complexity of approximating k-set packing. computational complexity **15**(1), 20–39 (2006)

[10] Huang, Z., Kang, N., Tang, Z.G., Wu, X., Zhang, Y., Zhu, X.: Fully online matching. J. ACM **67**(3) (may 2020). https://doi.org/10.1145/3390890, `https://doi.org/10.1145/3390890`

[11] Kalyanasundaram, B., Pruhs, K.R.: An optimal deterministic algorithm for online b-matching. Theoretical Computer Science **233**(1-2), 319–325 (2000)

[12] Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. pp. 352–358 (1990)

[13] Kesselheim, T., Radke, K., Tönnis, A., Vöcking, B.: An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In: European symposium on algorithms. pp. 589–600. Springer (2013)

[14] Korula, N., Pál, M.: Algorithms for secretary problems on graphs and hypergraphs. In: International Colloquium on Automata, Languages, and Programming. pp. 508–520. Springer (2009)

[15] Lowalekar, M., Varakantham, P., Jaillet, P.: Competitive ratios for online multi-capacity ridesharing. arXiv preprint arXiv:2009.07925 (2020)

[16] Ma, Y., Rusmevichientong, P., Sumida, M., Topaloglu, H.: An approximation algorithm for network revenue management under nonstationary arrivals. Operations Research **68**(3), 834–855 (2020)

[17] Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. Journal of the ACM (JACM) **54**(5), 22–es (2007)

[18] Mehta, A., et al.: Online matching and ad allocation. Foundations and Trends® in Theoretical Computer Science **8**(4), 265–368 (2013)

[19] Pavone, M., Saberi, A., Schiffer, M., Tsao, M.W.: Online hypergraph matching with delays. Operations Research (2022)

[20] Simpson, R.W.: Using network flow techniques to find shadow prices for market demands and seat inventory control. MIT Press, Cambridge, MA (1989)

[21] Talluri, K., Van Ryzin, G.: An analysis of bid-price controls for network revenue management. Management science **44**(11-part-1), 1577–1593 (1998)