



Online matching on 3-uniform hypergraphs

Sander Borst¹ · Danish Kashaev² · Zhuan Khye Koh³

Received: 20 July 2025 / Accepted: 29 March 2026
© The Author(s) 2026

Abstract

The online matching problem was introduced by Karp, Vazirani and Vazirani (STOC 1990) on bipartite graphs with vertex arrivals. It is well-known that the optimal competitive ratio is $1 - 1/e$ for both integral and fractional versions of the problem. Since then, there has been considerable effort to find optimal competitive ratios for other related settings. In this work, we go beyond the graph case and study the online matching problem on k -uniform hypergraphs. For $k = 3$, we provide an optimal primal-dual fractional algorithm, which achieves a competitive ratio of $(e - 1)/(e + 1) \approx 0.4621$. As our main technical contribution, we present a carefully constructed adversarial instance, which shows that this ratio is in fact optimal. It combines ideas from known hard instances for bipartite graphs under the edge-arrival and vertex-arrival models. For $k \geq 3$, we give a simple integral algorithm which performs better than greedy when the online nodes have bounded degree. As a corollary, it achieves the optimal competitive ratio of $1/2$ on 3-uniform hypergraphs when every online node has degree at most 2. This is because the special case where every online node has degree 1 is equivalent to the edge-arrival model on graphs, for which an upper bound of $1/2$ is known.

Mathematics Subject Classification 68W27 (Online algorithms; streaming algorithms)

An extended abstract of this paper has appeared in [1]. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 805241–QIP). S. Borst and Z. K. Koh — Research was done while the authors were at CWI.

✉ Sander Borst
sborst@mpi-inf.mpg.de

✉ Danish Kashaev
danish.kashaev@cwi.nl

✉ Zhuan Khye Koh
zkkoh@bu.edu

¹ Max Planck Institute for Informatics, Saarbrücken, Germany

² Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

³ Boston University, Boston, USA

1 Introduction

Online matching is a classic problem in the field of online algorithms. It was first introduced in the seminal work of Karp, Vazirani and Vazirani [2], who considered the bipartite version with one-sided vertex arrivals. In this setting, we are given a bipartite graph where vertices on one side are known in advance (offline), and vertices on the other side arrive sequentially (online). When an online vertex arrives, it reveals its incident edges, at which point the algorithm must decide how to match it (or not) irrevocably. The goal is to maximize the cardinality of the resulting matching. Karp et al. [2] gave an elegant randomized algorithm RANKING, which achieves the optimal competitive ratio of $1 - 1/e$.

In certain applications, each offline vertex may be matched more than once. Examples include matching online jobs to servers, or matching online impressions to advertisers. This is the online b -matching model of Kalyanasundaram and Pruhs [3], where $b \geq 1$ is the maximum number of times an offline vertex can be matched. As b and the number of online vertices tend to infinity, it in turn captures the fractional relaxation of the Karp et al. [2] model. This means that the algorithm is allowed to match an online node fractionally to multiple neighbours, as long as the total load on every vertex does not exceed 1. For this problem, it is known that the deterministic algorithm BALANCE (or WATER-FILLING) achieves the optimal competitive ratio of $1 - 1/e$.

1.1 Online hypergraph matching

The online bipartite matching problem can be naturally generalized to hypergraphs as follows. For $k \geq 2$, let $\mathcal{H} = (V, W, H)$ be a k -uniform hypergraph with offline vertices V , online vertices W and hyperedges H . Every hyperedge $h \in H$ contains $k - 1$ elements from V and 1 element from W . Just like before, the online vertices arrive sequentially with their incident hyperedges, and the goal is to select a large matching, i.e., a set of disjoint hyperedges. The greedy algorithm is $1/k$ -competitive. On the other hand, no integral algorithm can be $2/k$ -competitive.¹

For the *fractional* version of the problem, Buchbinder and Naor [5] gave a deterministic algorithm which is $\Omega(1/\log k)$ -competitive. They also constructed an instance showing that any algorithm is $O(1/\log k)$ -competitive. In fact, their results apply to the more general setting of online packing linear program (LP), in which variables arrive sequentially. In the context of hypergraphs, this means that the hyperedges arrive sequentially. Note that for k -uniform hypergraphs, there is a trivial reduction from this *edge-arrival* model to our *vertex-arrival* model on $(k + 1)$ -uniform hypergraphs, by adding degree 1 online nodes.

The aforementioned results show that asymptotically, both integral and fractional versions of the online matching problem on k -uniform hypergraphs are essentially settled (up to constant factors). However, our understanding of the problem for small

¹ In [4], it is shown that no algorithm can be $(2 + f(k))/k$ -competitive for some positive function f with $f(k) = o(1)$. In Appendix D, we give a simple construction showing that no integral algorithm can be $2/k$ -competitive.

values of k (other than $k = 2$) remains poor. Many applications of online hypergraph matching in practice have small values of k . For instance, in ride-sharing and on-demand delivery services [6], $k - 1$ represents the capacity of service vehicles, which is often small. Another example is network revenue management problems [7]. In this setting, given a collection of limited resources, a sequence of product requests arrive over time. When a product request arrives, we have to decide whether to accept it irrevocably. Accepting a product request generates profit, but also consumes a certain amount of each resource. The goal is to devise a policy which maximizes profit. In this context, $k - 1$ represents the maximum number of resources used by a product. As Ma et al. [7] noted, many of these problems have small values of k . In airlines, for example, $k - 1$ corresponds to the maximum number of flight legs included in an itinerary, which usually does not exceed two or three.

1.2 Our contributions

Motivated by the importance of online hypergraph matching for small values of k , we focus on 3-uniform hypergraphs, with the goal of obtaining tighter bounds. Our main result is a tight competitive ratio for the fractional version of this problem.

Theorem 1 *For the online fractional matching problem on 3-uniform hypergraphs, there is a deterministic $(e - 1)/(e + 1)$ -competitive algorithm. Furthermore, every algorithm is at most $(e - 1)/(e + 1)$ -competitive.*

The deterministic algorithm in Theorem 1 belongs to the class of WATER-FILLING algorithms. It uses the function $f(x) := e^x/(e + 1)$ to decide which hyperedges receive load. In particular, for every online vertex w , the incident hyperedges $h = \{u, v, w\} \in \delta(w)$ which minimize $\phi(h) := f(x(\delta(u))) + f(x(\delta(v)))$ receive load until $\phi(h) \geq 1$.

Our main contribution is proving a matching upper bound in Theorem 1. For this, it suffices to consider deterministic algorithms because every randomized algorithm induces a deterministic fractional algorithm with the same expected value. This, in turn, allows us to construct an instance which is adaptive to the actions of the algorithm. The key idea is to combine two hard instances for online matching on *bipartite graphs* [2, 8].

We start with the instance in [8], designed for the edge-arrival model. In this instance, edge arrivals are grouped into phases, such that the size of an online maximum matching increases by one per phase. At the end of every phase, as long as the total fractional value on the revealed edges exceeds a certain threshold, the next phase begins. Otherwise, the instance terminates. For our purpose, we want a more fine-grained control over the actions of the algorithm. So, we apply thresholding at the node level instead, based on fractional degrees, to determine which nodes become incident to the edges arriving in the next phase.

In our construction, we will have multiple copies of this modified edge-arrival instance. The edges in these instances are connected to the online nodes to form hyperedges. The way in which they are connected is inspired by the instance in [2], originally designed for the vertex-arrival model. The idea behind this vertex-arrival

instance is to obfuscate the partners of the online nodes in an offline maximum matching, which is also applicable in our setting.

Our next result concerns the online integral matching problem on k -uniform hypergraphs. We show that one can do better than the greedy algorithm if the online nodes have bounded degree. It is achieved by the simple algorithm RANDOM: for every online vertex w , uniformly select a hyperedge among all the hyperedges incident to w which are disjoint from the current matching.

Theorem 2 *For the online matching problem on k -uniform hypergraphs where online vertices have maximum degree d , the competitive ratio of RANDOM is at least*

$$\min\left(\frac{1}{k-1}, \frac{d}{(d-1)k+1}\right).$$

Note that in Theorem 2, the first term is at most the second term if and only if $d \leq k-1$. Moreover, RANDOM is at least as good as the greedy algorithm, since the latter is $1/k$ -competitive. For 3-uniform hypergraphs, the bound becomes $1/2$ for $d \leq 2$ and $1/(3-2/d)$ otherwise, thus interpolating between $1/3$ and $1/2$. Note that for $d \leq 2$, the bound is optimal, since the online matching problem on graphs under edge arrivals is a special case of this setting (with $k=3$, $d=1$), for which an upper bound of $1/2$ is known even against fractional algorithms on bipartite graphs [8].

Since every randomized algorithm for integral matching induces a deterministic algorithm for fractional matching, the upper bound of $(e-1)/(e+1) \approx 0.4621$ in Theorem 1 also applies to the integral problem on 3-uniform hypergraphs. However, the best known lower bound is $1/3$, given by the greedy algorithm. An interesting question for future research is whether there exists an integral algorithm better than greedy on 3-uniform hypergraphs.

1.3 Related work

Since the online matching problem was introduced in [2], it has garnered significant interest, leading to extensive follow-up work. We refer the reader to the excellent survey by Mehta [9] for navigating this rich literature. The original analysis of RANKING [2] was simplified in a series of papers [10–13]. Many variants of the problem have been studied, such as the online b -matching problem [3], and its extension to the AdWords problem [14–17]. Weighted generalizations have been considered, e.g., vertex weights [18, 19] and edge weights [20]. Weakening the adversary by requiring that online nodes arrive in a random order has also been of interest [21–23]. Another line of research explored more general arrival models such as two-sided vertex arrival [24], general vertex arrival [8], edge arrival [8, 25], and general vertex arrival with departure times [26–28].

In contrast, the literature on the online hypergraph matching problem is relatively sparse. Most work has focused on stochastic models, such as the random-order model. Korula and Pal [29] first studied the edge-weighted version under this model. For k -uniform hypergraphs, they gave an $\Omega(1/k^2)$ -competitive algorithm. This was subsequently improved to $\Omega(1/k)$ by Kesselheim et al. [23]. Ma et al. [7] gave a $1/k$ -

competitive algorithm under ‘nonstationary’ arrivals. Pavone et al. [6] studied online hypergraph matching with delays under the adversarial model. At each time step, a vertex arrives, and it will depart after d time steps. A hyperedge is revealed once all of its vertices have arrived. Note that their model is incomparable to ours because every vertex has the same delay d .

In the prophet IID setting, every online node has a weight function which assigns weights to its incident hyperedges, and these functions are independently sampled from the same distribution. For this problem, [30] gave a $O(\log(k)/k)$ upper bound on the competitive ratio. We refer to [30] for an overview of known results in related settings.

Hypergraph matching on k -uniform hypergraphs is a well-studied problem in the offline setting. It is known to be impossible to approximate within a factor of $\Omega(1/k)$ in polynomial time, unless $\mathbf{NP} \subseteq \mathbf{BPP}$ [31]. Moreover, the factor between the optimal solution and the optimal value of the natural LP relaxation is at least $1/(k - 1 + 1/k)$ [32].

A special case that has also been studied is the restriction to k -partite graphs, where the vertices are partitioned into k sets and every hyperedge contains exactly one vertex from each set. This setting is called *k-dimensional matching*, and the optimal solution is known to be at least $1/(k - 1)$ times the optimal value of the standard LP relaxation [32]. For $k = 3$, the best known polynomial time approximation algorithm gives a $(3/4 - \varepsilon)$ -approximation [33].

1.4 Paper organization

In Section 2, we give the necessary preliminaries and discuss notation. Section 3 presents the optimal primal-dual fractional algorithm for 3-uniform hypergraphs, which shows the first part of Theorem 1. Section 4 complements this with a tight upper bound, proving the second part of Theorem 1. The proof of Theorem 2 is shown in Section 5.

2 Preliminaries

Given a hypergraph $\mathcal{H} = (V, H)$ with vertex set V and hyperedge set H , the maximum matching problem involves finding a maximum cardinality subset of disjoint hyperedges. The canonical primal and dual LP relaxations for this problem are respectively given by:

$$\begin{aligned}
 \max \quad & \sum_{h \in H} x_h \\
 \sum_{h \in \delta(v)} x_h & \leq 1 \quad \forall v \in V \\
 x_h & \geq 0 \quad \forall h \in H
 \end{aligned}
 \qquad
 \begin{aligned}
 \min \quad & \sum_{v \in V} y_v \\
 \sum_{v \in h} y_v & \geq 1 \quad \forall h \in H \\
 y_v & \geq 0 \quad \forall v \in V.
 \end{aligned}$$

We denote by $\text{OPT}_{\text{LP}}(\mathcal{H})$ the offline optimal value of these two LPs. We denote by $\text{OPT}(\mathcal{H})$ the objective value of an offline optimal integral solution to the primal LP, which clearly satisfies $\text{OPT}(\mathcal{H}) \leq \text{OPT}_{\text{LP}}(\mathcal{H})$.

The online matching problem on k -uniform hypergraphs under vertex arrivals is defined as follows. An instance consists of a k -uniform hypergraph $\mathcal{H} = (V, W, H)$, where V is the set of offline nodes and $W = (w_1, w_2, \dots)$ is the sequence of online nodes. The ordering of W corresponds to the arrival order of the online nodes. Every hyperedge $h \in H$ has exactly one node in W and $k - 1$ nodes in V . When an online node $w \in W$ arrives, its incident hyperedges $\delta(w)$ are revealed. A fractional algorithm is allowed to irrevocably increase x_h for every $h \in \delta(w)$, whereas an integral algorithm is allowed to irrevocably pick one of these hyperedges, i.e., setting $x_h = 1$ for some $h \in \delta(w)$.

Given an algorithm \mathcal{A} and an instance \mathcal{H} , we denote by $\mathcal{V}(\mathcal{A}, \mathcal{H}) := \sum_{h \in H} x_h$ the value of the (fractional) matching obtained by \mathcal{A} on \mathcal{H} . An integral algorithm is ρ -competitive if for any instance \mathcal{H} , $\mathcal{V}(\mathcal{A}, \mathcal{H}) \geq \rho \text{OPT}(\mathcal{H})$. Similarly, a fractional algorithm is ρ -competitive if for any instance \mathcal{H} , $\mathcal{V}(\mathcal{A}, \mathcal{H}) \geq \rho \text{OPT}_{\text{LP}}(\mathcal{H})$.

In this paper, we focus on 3-uniform hypergraphs. For a 3-uniform instance $\mathcal{H} = (V, W, H)$, we denote by $\Gamma(\mathcal{H}) = (V, E)$ the graph on the offline nodes with edge set

$$E := \left\{ (u, v) \in V \times V, \quad \exists w \in W \text{ s.t. } \{u, v, w\} \in H \right\}. \quad (1)$$

We remark that $\Gamma(\mathcal{H})$ is not a multigraph. In particular, an edge $(u, v) \in E$ can have several hyperedges in H containing it. A fractional matching x on the hyperedges H naturally induces a fractional matching x' on the edges E , i.e., $x'_e = \sum_{h: e \subseteq h} x_h$ for every $e \in E$. The value obtained by an algorithm \mathcal{A} can thus also be counted as $\mathcal{V}(\mathcal{A}, \mathcal{H}) = \sum_{h \in H} x_h = \sum_{e \in E} x'_e$. For an offline node $u \in V$, we denote its *load* (or *fractional degree*) as $\ell_u = x(\delta(u)) \in [0, 1]$.

3 Optimal fractional algorithm for 3-uniform hypergraphs

In this section, we present a primal-dual algorithm for the online fractional matching problem on 3-uniform hypergraphs under vertex arrivals. This algorithm will turn out to be optimal with a tight competitive ratio of $(e - 1)/(e + 1) \approx 0.4621$. We define the following function $f : [0, 1] \rightarrow [0, 1]$:

$$f(x) := \frac{e^x}{e + 1}. \quad (2)$$

When an online node w arrives, our algorithm chooses to uniformly increase the primal variables of the hyperedges $\{u, v, w\}$ for which $f(x(\delta(u))) + f(x(\delta(v)))$ is minimal. We note that this belongs to the class of *water-filling* algorithms [3]. For this reason, we define the *priority* of a hyperedge $h = \{u, v, w\}$ as:

$$\phi(h) := f(x(\delta(u))) + f(x(\delta(v))). \quad (3)$$

Figure 3 shows the possible values of $x(\delta(u))$ and $x(\delta(v))$ such that $\phi(h) \leq 1$. We now present the algorithm. The dual updates are shaded in gray since they are only needed for the analysis.

Algorithm 3.1 Water-filling fractional algorithm for 3-uniform hypergraphs

Input : 3-uniform hypergraph $\mathcal{H} = (V, W, H)$ with online nodes W .

Output : Fractional matching $x \in [0, 1]^H$

when $w \in W$ **arrives with** $\delta(w) \subseteq H$:

 set $x_h = 0$ for every $h \in \delta(w)$

 increase x_h for every $h = \{u, v, w\} \in \arg \min_{h \in \delta(w)} \{\phi(h)\}$ at rate 1

 increase y_u and y_v at rates $f(x(\delta(u)))$ and $f(x(\delta(v)))$

 increase y_w at rate $1 - f(x(\delta(u))) - f(x(\delta(v)))$

until $x(\delta(w)) = 1$ **or** $\phi(h) \geq 1$ for every $h \in \delta(w)$.

return x

Theorem 3 Algorithm 3.1 is $(e - 1)/(e + 1)$ -competitive for the online fractional matching problem on 3-uniform hypergraphs.

Proof of Theorem 3 We first show that the algorithm produces a feasible primal solution. Note that the fractional value of a hyperedge h is only being increased if $\phi(h) \leq 1$. If $x(\delta(v)) = 1$ for some offline node v then for any hyperedge $h \ni v$, we have:

$$\phi(h) = f(x(\delta(u))) + f(x(\delta(v))) \geq f(1) + f(0) = \frac{e + 1}{e + 1} = 1,$$

where u denotes the second offline node belonging to h . The value of the hyperedge h will thus not be increased anymore, proving the feasibility of the primal solution.

In order to prove the desired competitive ratio, we show that the primal-dual solutions constructed during the execution of the algorithm satisfy:

$$\mathcal{V}(\mathcal{A}) = \sum_{h \in H} x_h = \sum_{v \in V \cup W} y_v \quad \text{and} \tag{4}$$

$$\sum_{v \in h} y_v \geq \rho := \frac{e - 1}{e + 1} \quad \forall h \in H. \tag{5}$$

This is enough to imply the desired competitiveness of our algorithm, since $y/\rho \in \mathbb{R}_+^V$ is then a feasible dual solution, giving:

$$\mathcal{V}(\mathcal{A}) \geq \sum_{v \in V \cup W} y_v \geq \rho \text{OPT}_{\text{LP}}.$$

Note that (4) holds at the start of the algorithm. Let us fix a hyperedge $h = \{u, v, w\} \in H$. When x_h is continuously being increased at rate one, the duals on the incident nodes y_u, y_v and y_w are being increased at rate $f(x(\delta(u))), f(x(\delta(v)))$

and $1 - f(x(\delta(u))) - f(x(\delta(v)))$ respectively. Observe that these rates sum up to one. Hence, $\mathcal{V}(\mathcal{A}) = \sum_{h \in H} x_h$ and $\sum_{v \in V \cup W} y_v$ are increased at the same rate, meaning that (4) holds at all times during the execution of the algorithm.

We now show that (5) holds at the end of the execution of the algorithm. Let us fix an online node $w \in W$. For a given hyperedge $h \in \delta(w)$, note that the algorithm only stops increasing x_h , as soon as either $\phi(h) \geq 1$ or $x(\delta(w)) = 1$ is reached. We distinguish these two cases for the analysis.

Let us first focus on the first case, meaning that $\phi(h) \geq 1$ has been reached for every $h \in \delta(w)$. Consider an arbitrary $h = \{u, v, w\} \in \delta(w)$. For every unit of increase in $x(\delta(u))$, y_u will have been increased by $f(x(\delta(u)))$. If we denote by $\ell_u := x(\delta(u))$ and $\ell_v := x(\delta(v))$ the fractional loads on u and v after the last increase on the hyperedges adjacent to w , then:

$$y_u = \int_0^{\ell_u} f(s) ds = f(\ell_u) - f(0) \quad \text{and} \quad y_v = \int_0^{\ell_v} f(s) ds = f(\ell_v) - f(0), \quad (6)$$

where we have used the fact that f is an antiderivative of itself. Therefore,

$$\begin{aligned} y_u + y_v + y_w &\geq y_u + y_v = f(\ell_u) - f(0) + f(\ell_v) - f(0) \\ &= \phi(h) - 2f(0) \geq 1 - 2f(0) = \frac{e-1}{e+1}. \end{aligned}$$

Suppose now that $x(\delta(w)) = 1$ has been reached. In particular, this means that for each $\{u, v, w\} \in \delta(w)$, the rate at which y_w was increased must have been at least $1 - f(\ell_u) - f(\ell_v)$ at all times, where ℓ_u and ℓ_v denote the fractional loads on u and v after that the algorithm has finished increasing the edges incident to the online node w . Hence, we have:

$$y_w \geq 1 \cdot (1 - f(\ell_u) - f(\ell_v)).$$

By using (6) we see that:

$$y_u + y_v + y_w \geq f(\ell_u) + f(\ell_v) - 2f(0) + (1 - f(\ell_u) - f(\ell_v)) = 1 - 2f(0) = \frac{e-1}{e+1}.$$

This proves (5), and thus completes the proof of the theorem. \square

We note that the above algorithm and its analysis are very similar to those of [5]. Here we optimized the choice of f for the case of 3-uniform hypergraphs. By choosing $f(x) := (k \log(k))^{x-1}$ and setting $\phi(h) := \sum_{u \in h} f(x(\delta(u)))$ for every k -uniform hyperedge h , it can be shown that the natural generalization of Algorithm 3.1 to k -uniform hypergraphs is $\Omega(1/\log(k))$ -competitive [4].

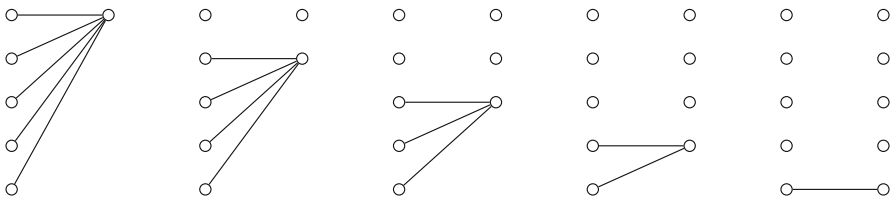


Fig. 1 The hard instance for bipartite graphs under vertex-arrivals. The k th online node connects to the $n - k + 1$ offline nodes with the highest fractional degree at that point in time. In this figure, the offline vertices on the left side of the bipartition are drawn in decreasing order of their fractional degrees. The perfect matching consists of all the horizontal edges

4 Tight upper bound for 3-uniform hypergraphs

We now prove the second part of Theorem 1, i.e., every algorithm is at most $(e - 1)/(e + 1)$ -competitive for the online fractional matching problem on 3-uniform hypergraphs under vertex arrivals.

4.1 Overview of the construction

We construct an adversarial instance that is adaptive to the behaviour of the algorithm. The main idea is to combine the vertex-arrival instance of Karp et al. [2] and the edge-arrival instance of Gamlath et al. [8] for bipartite graphs. We first briefly describe these two instances, and then give a high-level view of our construction.

Bipartite hardness under vertex-arrivals. An illustration of this construction is given in Figure 1. This instance consists of a bipartite graph, where one side of the bipartition has size $n \in \mathbb{N}$ and is known upfront. The other side of the bipartition arrives online, with n online nodes arriving in total. The k th online node has degree $n - k + 1$, with the edges incident to it connecting to the $n - k + 1$ offline nodes with the highest fractional degree at that point in time.

Note that, after all online nodes have arrived, the instance contains a perfect matching of size n . It can however be shown that no fractional algorithm can achieve a competitive ratio better than $1 - 1/e \approx 0.63$ for this instance, due to the uncertainty about the edge adjacent to each online node in the perfect matching, since it is a decision made online by the adversary. This construction can easily be adapted to our model on 3-uniform hypergraphs under vertex arrivals, by replacing each offline node by a pair of two identical offline nodes, and replacing each edge of the graph by a hyperedge of size three.

Bipartite hardness under edge-arrivals. An illustration of this construction is given in Figure 2. Each side of the bipartition has cardinality T and the nodes on both sides are denoted by $\{1, \dots, T\}$. This instance now consists of $T' \leq T$ phases, where T' is unknown a priori and determined online by the adversary given the choices of the algorithm. At each phase $t \in [T']$, the following t new edges, which form a matching,

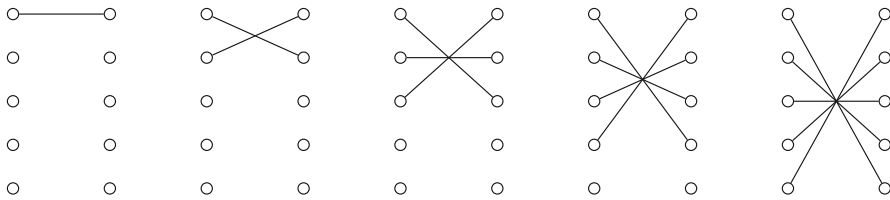


Fig. 2 The hard instance for bipartite graphs under edge-arrivals. The adversary can terminate the instance at any phase if the algorithm does not generate enough value. The optimal matching consists of all the edges which arrived in the last phase

are revealed:

$$\{(i, t - i + 1) : i \in \{1, \dots, t\}\}.$$

Observe that these become the optimal matching at that point in time. If the value generated by the algorithm exceeds αt at the end of phase t for some $\alpha \in [0, 1]$, the instance continues to the next phase. Otherwise, the instance terminates. It is shown in [8] that no fractional algorithm can achieve a competitive ratio better than $1/2$ for this instance. The hardness is due to the uncertainty about the time horizon: the adversary can always make a new set of edges arrive if the algorithm gains enough value in each phase. This construction can easily be adapted to our vertex-arrival model on 3-uniform hypergraphs, by appending an online node with degree 1 to each edge of the bipartite graph.

Our construction. We now give a high-level overview of our construction, which combines both hard instances described above. The offline vertices of the hypergraph are partitioned into m sets C_1, \dots, C_m , which we call *components*. Each component will induce a bipartite graph with bipartition $C_i = U_i \cup V_i$, where U_i and V_i are both ordered vertex sets of size $T \in \mathbb{N}$.

The instance consists of T phases. In each phase $t \in \{1, \dots, T\}$, the adversary first selects a bipartite matching $\mathcal{M}_i^{(t)}$ on each component C_i . Taking the union of these matchings gives a larger matching on the offline nodes: $\mathcal{M}^{(t)} := \bigcup_{i=1}^m \mathcal{M}_i^{(t)}$.

After selecting the matching $\mathcal{M}^{(t)}$ at phase t , the adversary selects the online nodes, with their incident hyperedges, arriving in that phase. The set of online nodes arriving in phase t is denoted by $W^{(t)}$. Each node $w \in W^{(t)}$ connects to a subset of edges $E(w) \subseteq \mathcal{M}^{(t)}$, meaning that the hyperedges incident to w are $\{w\} \cup e : e \in E(w)$.

We briefly explain how the matchings $\mathcal{M}_i^{(t)}$ are constructed and how the edges $E(w)$ are picked:

1. On each component C_i , the matching $\mathcal{M}_i^{(t)}$ is constructed based on the behaviour of the algorithm in phase $t - 1$. It draws inspiration from the edge-arrival instance in [8], together with the function $f(x) = e^x / (e + 1)$ defined in (2). The exact construction is described in Section 4.3 and illustrated in Figure 4.
2. For every online node $w \in W^{(t)}$, the edge set $E(w) \subseteq \mathcal{M}^{(t)}$ is selected based on the behaviour of the algorithm during phase t . This part can be seen as incorporating the vertex-arrival instance in [2]. The exact construction is described in Section 4.5 and illustrated in Figure 7.

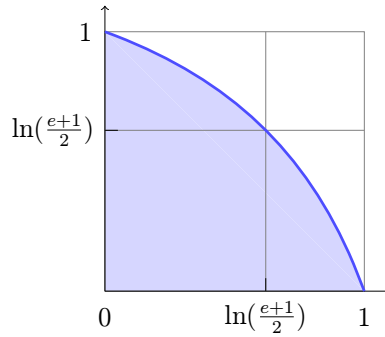


Fig. 3 An illustration of the region $R = \{(a, b) \in [0, 1]^2 : f(a) + f(b) \leq 1\}$. The symmetric point at the boundary of the region has both coordinates $\ln((e + 1)/2) \approx 0.62$. When an online node w arrives, a threshold respecting algorithm ensures that the fractional matching x satisfies $(x(\delta(u)), x(\delta(v))) \in R$ for every hyperedge $h = \{u, v, w\} \in \delta(w)$ with $x_h > 0$ at the end of that iteration

To summarize, the instance is a hypergraph $\mathcal{H} = (V, W, H)$ with offline nodes V , online nodes W and hyperedges H given by

$$V := \bigcup_{i=1}^m C_i = \bigcup_{i=1}^m U_i \cup V_i \quad W := \bigcup_{t=1}^T W^{(t)} \quad H := \bigcup_{t=1}^T \bigcup_{w \in W^{(t)}} \{\{w\} \cup e : e \in E(w)\}.$$

4.2 Assumptions on the algorithm

To simplify the construction and analysis of our instance, we will make two assumptions on the algorithm. First, we need the following definition, which relates the behaviour of an algorithm to the priority function ϕ defined in (3).

Definition 1 Fix $\varepsilon \geq 0$. Let x be the fractional solution given by an algorithm \mathcal{A} after the arrival of an online node w . We say that \mathcal{A} is ε -threshold respecting on w if $\phi(h) = \sum_{v \in h \setminus \{w\}} f(x(\delta(v))) \leq 1 + \varepsilon$ for all incident hyperedges $h \in \delta(w)$ with $x_h > 0$. We also call \mathcal{A} threshold respecting if $\varepsilon = 0$.

Remark 1 We emphasize that the property in Definition 1 only needs to hold for the fractional solution x after w has arrived, and before the arrival of the next online node. In particular, it is possible that $\phi(h) > 1 + \varepsilon$ in later iterations. For a hyperedge $h = \{u, v, w\} \in \delta(w)$, Figure 3 shows the possible values of $x(\delta(u))$ and $x(\delta(v))$ such that $\phi(h) \leq 1$.

The two assumptions that we make are the following. In Appendix B, we show that they can be made without loss of generality.

1. The algorithm is ε -threshold respecting on all online nodes in the first $T - 1$ phases for some arbitrarily small $\varepsilon > 0$.
2. The algorithm is symmetric on each component $C_i = U_i \cup V_i$. In particular, for every $t \in \{1, \dots, T\}$, the t th vertices of U_i and V_i have the same fractional degrees throughout the execution of the algorithm.

4.3 Constructing the matching $\mathcal{M}^{(t)}$

In this section, we construct the matching $\mathcal{M}_i^{(t)}$ for every component $i \in [m]$ and phase $t \in [T]$. We will only describe the matchings for a single component C_i , i.e., $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}, \dots, \mathcal{M}_i^{(T)}$, because the same construction applies to other components. Intuitively, the value obtained by the algorithm in the first $T - 1$ phases is already limited by Assumption 1. So, the goal of this construction is to prevent the algorithm from gaining too much value in the last phase T . In particular, we will show that it can only obtain $O(\sqrt{T}) + \varepsilon O(T^2)$ in the last phase on every component C_i .

The matchings $\mathcal{M}_i^{(1)}, \dots, \mathcal{M}_i^{(T)}$ are adaptive to the behaviour of the algorithm in every phase. It is essentially the instance of [8] with our threshold function incorporated. The vertex set of these matchings is on a bipartite graph, with T nodes on both sides of the bipartition. Let us denote this bipartition as $U_i = \{1, \dots, T\}$ and $V_i = \{1, \dots, T\}$. We index them the same way due to the symmetry assumption of the algorithm (Assumption 2). Each matching $\mathcal{M}_i^{(t)}$ satisfies the invariant that $(u, v) \in \mathcal{M}_i^{(t)}$ if and only if $(v, u) \in \mathcal{M}_i^{(t)}$.

For an offline node $u \in V$, we denote its *load* (or *fractional degree*) at the end of phase t as $\ell_u^{(t)} = x^{(t)}(\delta(u)) \in [0, 1]$, where $x^{(t)}$ is the fractional matching generated by the algorithm at the end of phase t .

- $\mathcal{M}_i^{(1)}$ is a matching of size one that consists of the single edge $(1, 1)$.
- At the end of phase t , we will call a node *active* if it is incident to an edge $e = (u, v) \in \mathcal{M}_i^{(t)}$ satisfying $\phi(e) = f(\ell_u^{(t)}) + f(\ell_v^{(t)}) \geq 1$. All other nodes are said to be *inactive* and will not be used in any of the matchings of later phases. Let $\sigma_t(1) < \sigma_t(2) < \dots < \sigma_t(r_t)$ be the active nodes in U_i at the end of phase t , where r_t denotes the number of such active nodes. By the aforementioned invariant and Assumption 2, the active nodes in V_i are also $\sigma_t(1) < \sigma_t(2) < \dots < \sigma_t(r_t)$.
- The matching at phase $t + 1$ is then of size $r_t + 1$ and is defined as:

$$\mathcal{M}_i^{(t+1)} := \left\{ \left(\sigma_t(k), \sigma_t(r_t + 2 - k) \right), k \in \{1, \dots, r_t + 1\} \right\},$$

where we define $\sigma_t(r_t + 1) := t + 1$ for convenience. In particular, note that $t + 1 \in U_i$ and $t + 1 \in V_i$ are two fresh nodes with zero load, which are always part of the matching $\mathcal{M}_i^{(t+1)}$, but not part of any matching from a previous phase. Clearly, the invariant is maintained. Figure 4 illustrates the construction.

Let us denote $q_t := (r_t + 1)/2$. Observe that the nodes $\sigma_t(k) \in U_i$ and $\sigma_t(k) \in V_i$ for every $k \in \{1, \dots, \lceil q_t \rceil\}$ form a vertex cover of the matching $\mathcal{M}_i^{(t+1)}$, meaning that every edge of the matching in phase $t + 1$ is covered by one of these active nodes at phase t . Intuitively, this construction ensures that as t gets large, these nodes have a high fractional degree. Consequently, the algorithm does not have a lot of room to increase the fractional value on any edge of $\mathcal{M}_i^{(t+1)}$, due to the degree constraints. In order to upper bound the value that the algorithm can get in phase $t + 1$, we will thus lower bound the fractional degree of the active nodes $\sigma_t(i)$ for $i \in \{1, \dots, \lceil q_t \rceil\}$. For

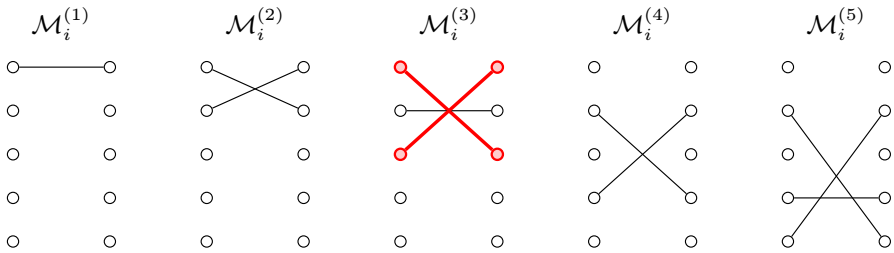


Fig. 4 In this example, the algorithm does not increase the edge (1, 3), and thus by symmetry the edge (3, 1), up to the threshold during phase $t = 3$. Hence, $f(\ell_1^{(3)}) + f(\ell_3^{(3)}) < 1$ and nodes 1 and 3 become inactive from that point on. The maximum matching at the end of phase 5 still has size five and consists of $\mathcal{M}_i^{(5)}$, in addition to the two edges (1, 3) and (3, 1) that are below the threshold

this reason, we define:

$$\ell(t, i) := x^{(t)}\left(\delta(\sigma_t(i))\right) = \sum_{e \in \delta(\sigma_t(i))} x_e^{(t)}.$$

In words, this is the fractional degree of the i^{th} active node at the end of phase t . One can now see $\{\ell(t, i)\}_{t,i}$ as a process with two parameters, which depends on the behaviour of the algorithm. To analyze this process, we will relate it to the CDF of the binomial distribution $B(t, 1/2)$. We will in fact show that

$$\sum_{i=1}^{\lceil q_{T-1} \rceil} 2(1 - \ell(T - 1, i)) = O(\sqrt{T}) + \varepsilon O(T^2). \tag{7}$$

Since the left-hand side is the residual capacity of the vertex cover of $\mathcal{M}_i^{(T)}$, this will yield an upper bound on the value obtained by the algorithm in component C_i during the last phase. For intuition, Figure 5 provides an example of $\ell(t, i)$ if the algorithm exactly reaches the threshold for every edge.

4.4 Bound for the last phase T

In this section, we prove the following theorem by showing (7).

Theorem 4 *During the last phase T, the value gained by the algorithm in each component C_i is at most $O(\sqrt{T}) + \varepsilon O(T^2)$.*

In order to be able to get a lower bound on $\ell(t, i)$, we now relate it to a process which is simpler to analyze, defined as follows on $\mathbb{N} \times \mathbb{Z}/2$:

$$\psi(t, y) = \Pr_{X \sim B(t, \frac{1}{2})} \left[X < \frac{t}{2} + y \right] + \frac{1}{2} \Pr_{X \sim B(t, \frac{1}{2})} \left[X = \frac{t}{2} + y \right],$$

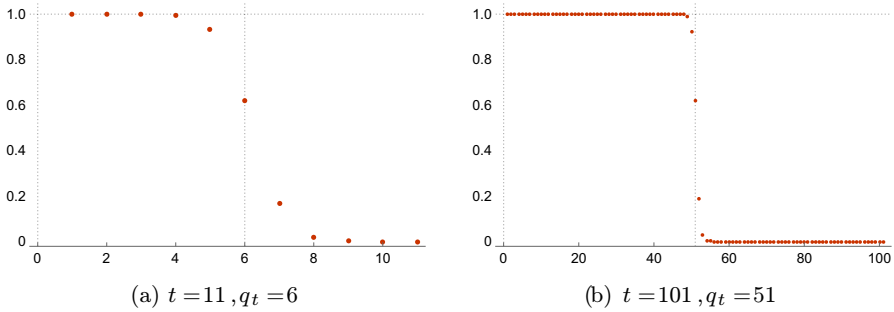


Fig. 5 Plot of the $\ell(t, i)$ process for two different values of t if the algorithm exactly matches the threshold at every phase. Observe that, since t is odd, $(\sigma_t(q_t), \sigma_t(q_t)) \in \mathcal{M}_i^{(t)}$ with the load of node $\sigma_t(q_t)$ staying at $\ln((e + 1)/2) \approx 0.62$

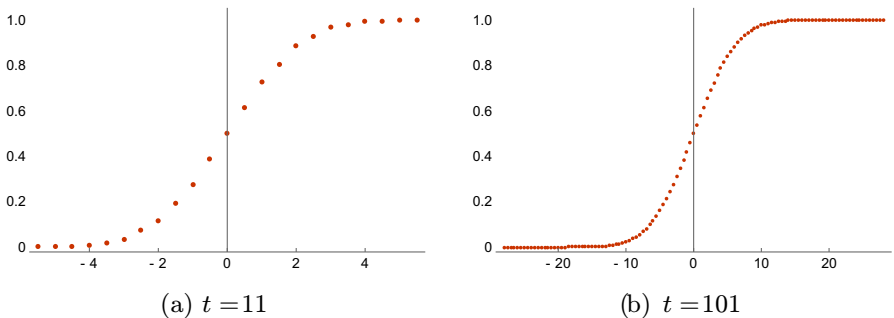


Fig. 6 Plot of $\psi(t, y)$ for two different values of t , the horizontal axis represents $y \in \mathbb{Z}/2$. Here $\epsilon = 0$, as it should be thought of as a very small constant

where $B(t, \frac{1}{2})$ is the binomial distribution with parameters t and $\frac{1}{2}$ (see Figure 6 for an illustration). This process has the following crucial properties, whose proofs are given in Appendix C.

Lemma 4.1 *The function ψ satisfies:*

1. For all t , $\psi(t, y)$ is nondecreasing in y with $\psi(t, 0) = \frac{1}{2}$ and $\psi(t, \frac{t+1}{2}) = 1$.
2. For all t and y , we have: $\psi(t + 1, y) = \frac{1}{2}\psi(t, y - \frac{1}{2}) + \frac{1}{2}\psi(t, y + \frac{1}{2})$.
3. For all t , we have $\sum_{y=0}^{\infty} (1 - \psi(t, \frac{1}{2}y)) \leq 1 + \sqrt{t}$.

We then relate the process $\ell(t, i)$ to a linear transformation of the process $\psi(t, i)$, by defining:

$$\xi(t, i) := a \psi(t, q_t - i) + b - \epsilon t.$$

We choose a and b such that whenever the algorithm exactly hits the threshold for every edge, we have $\xi(t, t/2) + \epsilon t = \ln((e + 1)/2) = \ell(t, t/2)$ for all even t and $\lim_{t \rightarrow \infty} \xi(t, x) + \epsilon t = 1 = \lim_{t \rightarrow \infty} \ell(t, x)$ for all x . Hence, we set $a := 2 - 2 \ln((e + 1)/2) \approx 0.76$ and $b := 2 \ln((e + 1)/2) - 1 \approx 0.24$. This choice will allow us to lower bound $\ell(t, i)$ by $\xi(t, i)$ for all $i \in \{1, \dots, \lceil q_t \rceil\}$ in Lemma 4.3.

We first need one more lemma. As a reminder, r_t is the number of active nodes on each side of the bipartition at the end of phase t and $q_t := (r_t + 1)/2$. For every active node $u = \sigma_t(i)$, where $i \leq r_t$, let us define $d_t(u) = |i - q_t|$, the distance of u to the middle point q_t of the active nodes at time t .

Lemma 4.2 *Let $u = \sigma_t(i)$ for $i \leq r_t$, then the following holds:*

$$d_t(u) \leq d_{t-1}(u) + \frac{1}{2} \text{ if } i < q_t \quad \text{and} \quad d_t(u) \leq d_{t-1}(u) - \frac{1}{2} \text{ if } i > q_t.$$

Proof Let S be the set of indices $i \leq r_{t-1} + 1$ such that $\sigma_{t-1}(i)$ is not active after phase t . Let j be such that $\sigma_{t-1}(j) = u$. Consider the case that $i < q_t$. Let $c = |s \in S : i < s < r_{t-1} + 2 - i|$. We have $d_t(u) = q_t - i = q_{t-1} + \frac{1}{2} - \frac{1}{2}c - i \leq q_{t-1} + \frac{1}{2} - i = d_{t-1}(u) + \frac{1}{2}$. The proof for $i > q_t$ is similar. \square

We are now ready to prove the desired lower bound on the loads.

Lemma 4.3 *For every $t \in \{1, \dots, T - 1\}$ and $i \in \{1, \dots, \lceil q_t \rceil\}$, we have*

$$\ell(t, i) \geq \xi(t, i),$$

where $q_t = (r_t + 1)/2$ and r_t is the number of active nodes at the end of phase t , when $r_t \geq 1$.

Proof We will prove this statement by induction on t .

Base case. For the base case, consider $t = 1$. There are two possibilities, either the edge $(1, 1)$ does not make it to the threshold, i.e. $2f(\ell_1^{(1)}) < 1$, in which case $r_t = 0, q_t = 0$, and the statement is then trivially satisfied. If the edge $(1, 1)$ makes it to the threshold, then $2f(\ell_1^{(1)}) = 2f(\ell(1, 1)) \geq 1$, which is equivalent to $\ell(1, 1) \geq \ln((e + 1)/2)$ by definition of $f(x) = e^x / (e + 1)$. Observe that in this case $r_t = q_t = 1$, leading to

$$\ell(1, 1) \geq \ln((e + 1)/2) = \frac{a}{2} + b = a \psi(1, 0) + b = \xi(1, 1) + \varepsilon t \geq \xi(1, 1),$$

where we have used the fact that $\psi(1, 0) = 1/2$.

Inductive step. Suppose now by induction that the statement holds for $t - 1$, let r_t be the number of active nodes at the end of phase t , let $q_t := (r_t + 1)/2$ and consider an arbitrary $i \in \{1, \dots, \lceil q_t \rceil\}$. We consider the cases where $i = q_t$ and $i < q_t$ separately.

Let us first consider the case where $i = q_t = (r_t + 1)/2$, which can only occur when r_t is odd. Observe that this means the edge $(\sigma_t(i), \sigma_t(i))$ belongs to the matching $\mathcal{M}_i^{(t)}$ and exceeds the threshold, i.e. $\ell(t, i) \geq \ln((e + 1)/2)$. Using the fact that $\psi(t, 0) = 1/2$ for all $t, i = q_t$ and the exact same arguments as above, we get

$$\ell(t, i) \geq \ln((e + 1)/2) = \frac{a}{2} + b = a \psi(t, 0) + b = a \psi(t, q_t - i) + b = \xi(t, i) + \varepsilon t \geq \xi(t, i).$$

Consider now the case where $i < q_t$. Let $e = (u, v) = (\sigma_t(i), \sigma_t(r_t + 1 - i)) \in \mathcal{M}_i^{(t)}$ and observe that e exceeds the threshold, i.e. $f(\ell_u^{(t)}) + f(\ell_v^{(t)}) = f(\ell(t, i)) + f(\ell(t, r_t + 1 - i)) \geq 1$. Let us pick indices j, k such that $u = \sigma_{t-1}(j)$ and $v = \sigma_{t-1}(r_{t-1} + 1 - k)$. We have:

$$\ell(t, i) = \ell(t-1, j) + x_e^{(t)} \quad \text{and} \quad \ell(t, r_t + 1 - i) = \ell(t-1, r_{t-1} + 1 - k) + x_e^{(t)}.$$

If $k = 0$, then v has not appeared in any of the prior matchings, so $\ell_v^{(t-1)} = 0$. In particular, we have $f(\ell_v^{(t-1)}) = f(0) = 1 - f(1) \leq 1 + \varepsilon - f(\xi(t-1, k))$.

Otherwise, we have $(\sigma_{t-1}(k), v) \in \mathcal{M}_i^{(t-1)}$ and by using the fact that the algorithm is ε -threshold respecting, we get $f(\ell(t-1, k)) + f(\ell_v^{(t-1)}) \leq 1 + \varepsilon$. By using the inductive hypothesis $\ell(t-1, k) \geq \xi(t-1, k)$, we get $f(\ell_v^{(t-1)}) \leq 1 + \varepsilon - f(\xi(t-1, k))$. Since edge e exceeds the threshold at the end of phase t , we have $f(\ell_u^{(t-1)} + x_e^{(t)}) + f(\ell_v^{(t-1)} + x_e^{(t)}) \geq 1$, which leads to

$$\begin{aligned} x_e^{(t)} &\geq -\ln\left(f(\ell_u^{(t-1)}) + f(\ell_v^{(t-1)})\right) \\ &\geq -\ln\left(1 + \varepsilon - f(\xi(t-1, k)) + f(\ell_u^{(t-1)})\right) \quad (\text{by induction hypothesis}) \\ &\geq f(\xi(t-1, k)) - f(\ell_u^{(t-1)}) - \varepsilon \quad (\text{since } \ln(1+x) \geq x) \\ &\geq f'(\ell_u^{(t-1)})\left(\xi(t-1, k) - \ell_u^{(t-1)}\right) - \varepsilon \quad (\text{by convexity of } f(x) = \frac{\exp(x)}{e+1}) \\ &= f'(\ell_u^{(t-1)})\left(\xi(t-1, k) - \ell(t-1, j)\right) - \varepsilon. \end{aligned}$$

Finally, since we have $f'(\ell_u^{(t-1)}) = f(\ell_u^{(t-1)}) \geq f\left(\ln((e+1)/2)\right) = 1/2$, so that $x_e^{(t)} \geq \frac{1}{2}\left(\xi(t-1, k) - \ell(t-1, j)\right) - \varepsilon$, and hence:

$$\begin{aligned} \ell(t, i) &= \ell_u^{(t-1)} + x_e^{(t)} \geq \ell_u^{(t-1)} + \frac{1}{2}\left(\xi(t-1, k) - \ell(t-1, j)\right) - \varepsilon \\ &= \frac{1}{2}\ell(t-1, j) + \frac{1}{2}\xi(t-1, k) - \varepsilon \\ &\geq \frac{1}{2}\xi(t-1, j) + \frac{1}{2}\xi(t-1, k) - \varepsilon \quad (\text{by induction hypothesis}) \\ &= \frac{a}{2}\left(\psi\left(t-1, d_{t-1}(u)\right) + \psi\left(t-1, d_{t-1}(v)\right)\right) + b - \varepsilon t \\ &\geq \frac{a}{2}\left(\psi\left(t-1, d_t(u) + \frac{1}{2}\right) + \psi\left(t-1, d_t(v) - \frac{1}{2}\right)\right) + b - \varepsilon t \\ &\quad (\text{by Lemma 4.2}) \\ &= a \psi(t, q_t - i) + b - \varepsilon t \quad (\text{by property (2) in Lemma 4.1}) \\ &= \xi(t, i) \end{aligned}$$

This concludes the proof of the inductive step and thus of the lemma. \square

We are now ready to bound the value obtained by the algorithm in the last phase and thus prove Theorem 4.

Proof of Theorem 4 Consider the end of phase $T - 1$. Observe that the nodes $\sigma_{T-1}(k) \in U_i$ and $\sigma_{T-1}(k) \in V_i$ for $k \in \{1, \dots, \lceil q_{T-1} \rceil\}$ form a vertex cover of the final matching $\mathcal{M}_i^{(T)}$. Because of the degree constraints, this means that the value the algorithm can gain on the last phase T is at most twice the following expression:

$$\begin{aligned} \sum_{k=1}^{\lceil q_{T-1} \rceil} (1 - \ell(T - 1, k)) &\leq \sum_{k=1}^{\lceil q_{T-1} \rceil} (1 - \xi(T - 1, k)) \\ &= a \sum_{k=1}^{\lceil q_{T-1} \rceil} (1 - \psi(T - 1, q_{T-1} - k)) + \varepsilon(T - 1)\lceil q_{T-1} \rceil \\ &\hspace{15em} \text{(definition of } \xi \text{ and } 1 - b = a) \\ &\leq \varepsilon T^2 + a \sum_{y=-1}^{\infty} (1 - \psi(T - 1, \frac{y}{2})) \\ &\hspace{15em} \text{(by change of variables } y := 2(q_{T-1} - k)) \\ &\leq \varepsilon T^2 + a(2 + \sqrt{T - 1}) \hspace{10em} \text{(by property (3) of Lemma 4.1)} \\ &= O(\sqrt{T}) + \varepsilon O(T^2). \end{aligned}$$

□

4.5 Connecting the matching $\mathcal{M}^{(t)}$ to the online nodes

In this section, we connect the matching $\mathcal{M}^{(t)} = \cup_{i=1}^m \mathcal{M}_i^{(t)}$ to the online vertices to form hyperedges, for every phase $t \in [T]$. The way in which they are connected is similar to the vertex-arrival instance of [2] for bipartite graphs. The main idea is to obfuscate the partners of the online nodes in the optimal matching.

The following construction is an adaptation of the vertex-arrival instance for bipartite graphs [2] to tripartite hypergraphs. Given a graph matching \mathcal{M} on the offline nodes, the first online node connects to every edge in \mathcal{M} . After the algorithm sets fractional values on every edge of \mathcal{M} , the second online node connects to $\mathcal{M} \setminus \{e_1\}$, where e_1 is the edge in the matching with the lowest fractional value. More generally, for every $k \in \{1, \dots, |\mathcal{M}|\}$, the k^{th} online node connects to the $|\mathcal{M}| - k + 1$ edges $\mathcal{M} \setminus \{e_1, \dots, e_{k-1}\}$, and e_k is defined as the edge having the lowest fractional value among them at the end of the k^{th} iteration. This instance is illustrated in Figure 7.

We now state the guarantee obtained by this construction, parametrized by the maximum (fractional) degree $\Delta \in [0, 1]$ attained by an offline node. The proof of the following lemma is shown in Appendix A.

Lemma 4.4 *For any graph matching $\mathcal{M} = (V, E)$, there exists an online tripartite hypergraph instance $\mathcal{H} = (V, W, H)$ such that $\Gamma(\mathcal{H}) = \mathcal{M}$ and $\text{OPT}(\mathcal{H}) = |\mathcal{M}|$. Moreover, for any fractional algorithm \mathcal{A} whose returned solution x satisfies $x(\delta(v)) \leq$*

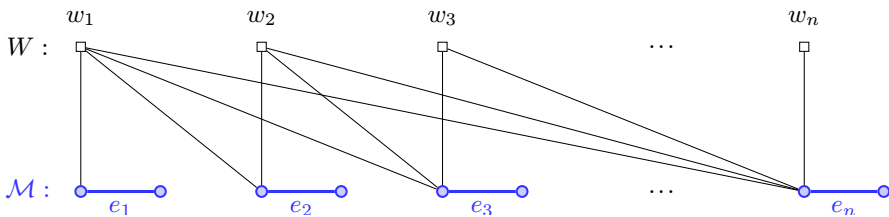


Fig. 7 An illustration of the instance constructed in Lemma 4.4

Δ for all offline nodes $v \in V$, we have

$$\mathcal{V}(\mathcal{A}, \mathcal{H}) \leq (1 - e^{-\Delta})|\mathcal{M}| + 3/2.$$

The way now in which we apply this construction is by partitioning the matching $\mathcal{M}^{(t)}$ into submatchings based on the load (or fractional degree) of the vertices, and applying Lemma 4.4 on each submatching separately. More precisely, let us fix $\eta := |\mathcal{M}^{(t)}|^{-1/3}$ and $N := \lceil 2/\eta \rceil$.

We partition the edges of the matching $\mathcal{M}^{(t)}$ into N^2 submatchings as follows

$$\mathcal{M}^{(t)}(i, j) := \left\{ (u, v) \in \mathcal{M}^{(t)} : \ell_u \in \left[\frac{i-1}{N}, \frac{i}{N} \right], \ell_v \in \left[\frac{j-1}{N}, \frac{j}{N} \right] \right\}$$

for all $i, j \in [N]$. Then, we apply the construction illustrated in Figure 7 to each submatching $\mathcal{M}^{(t)}(i, j)$. This finishes the description of our instance.

4.6 Bound for the first $T - 1$ phases

Recall from Section 4.1 that our instance consists of T phases. For ease of analysis, we will split the total value gained by the algorithm into the value gained in each phase. For an algorithm \mathcal{A} , let $\mathcal{V}^{(t)}(\mathcal{A})$ denote the value obtained by \mathcal{A} in phase t . The next lemma upper bounds $\mathcal{V}^{(t)}(\mathcal{A})$ for a threshold-respecting algorithm \mathcal{A} , in terms of the loads of the offline nodes at the end of phase $t - 1$. Recall that $W^{(t)}$ is the set of online nodes which arrive during phase t .

Lemma 4.5 *If \mathcal{A} is threshold-respecting on $W^{(t)}$, then*

$$\mathcal{V}^{(t)}(\mathcal{A}) \leq \sum_{(u,v) \in \mathcal{M}^{(t)}} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)}) \right) + 15 |\mathcal{M}^{(t)}|^{2/3}.$$

Proof Since we are considering a threshold-respecting algorithm, we can compute an upper bound on the amount that the algorithm can put on an edge $e \in \mathcal{M}^{(t)}(i, j)$ while staying below the threshold:

$$\Delta(i, j) := \max \left\{ x : f\left(\frac{i-1}{N} + x\right) + f\left(\frac{j-1}{N} + x\right) \leq 1 \right\}$$

$$= \ln \left(\frac{e + 1}{\exp\left(\frac{i-1}{N}\right) + \exp\left(\frac{j-1}{N}\right)} \right).$$

A simpler way to write this equation is as follows:

$$\exp(-\Delta(i, j)) = f\left(\frac{i-1}{N}\right) + f\left(\frac{j-1}{N}\right).$$

By Lemma 4.4, we know that there exists sets of online nodes $W^{(t)}(i, j)$ which, together with the matchings $\mathcal{M}^{(t)}(i, j)$, form online hypergraphs $\mathcal{H}^{(t)}(i, j)$ such that

$$\nu(\mathcal{A}, \mathcal{H}^{(t)}(i, j)) \leq \left(1 - \exp(-\Delta(i, j))\right) |\mathcal{M}^{(t)}(i, j)| + \frac{3}{2} \quad \forall i, j \in [N].$$

Now, observe that for an edge $\{u, v\} \in \mathcal{M}^{(t)}(i, j)$ with loads $\ell_u^{(t-1)}$ and $\ell_v^{(t-1)}$, we have

$$f\left(\frac{i-1}{N}\right) \geq f\left(\ell_u^{(t-1)} - \frac{1}{N}\right) \geq f(\ell_u^{(t-1)}) - \frac{1}{N}.$$

The first inequality follows from the fact that f is a non-decreasing function. The second inequality follows from the fact that $f'(x) = f(x) \leq 1$ for every $x \in [0, 1]$. Similarly, we have

$$f\left(\frac{j-1}{N}\right) \geq f(\ell_v^{(t-1)}) - \frac{1}{N}.$$

Hence, the value gained by the algorithm in phase t can be upper bounded as

$$\begin{aligned} \nu^{(t)}(\mathcal{A}) &= \sum_{i, j=1}^N \nu(\mathcal{A}, \mathcal{H}^{(t)}(i, j)) \\ &\leq \sum_{i, j=1}^N \left(1 - f\left(\frac{i-1}{N}\right) - f\left(\frac{j-1}{N}\right)\right) |\mathcal{M}^{(t)}(i, j)| + \frac{3}{2}N^2 \\ &\leq \sum_{i, j=1}^N \sum_{(u, v) \in \mathcal{M}^{(t)}(i, j)} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)}) + \frac{2}{N}\right) + \frac{3}{2}N^2 \\ &= \sum_{(u, v) \in \mathcal{M}^{(t)}} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)})\right) + \frac{2}{N}|\mathcal{M}^{(t)}| + \frac{3}{2}N^2 \\ &\leq \sum_{(u, v) \in \mathcal{M}^{(t)}} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)})\right) + \eta|\mathcal{M}^{(t)}| + 14\eta^{-2}. \end{aligned}$$

For the last inequality, since $N = \lceil 2/\eta \rceil$, we have used the bounds $2/N \leq \eta$ and $N^2 \leq (2/\eta + 1)^2 = (4 + 4\eta + \eta^2)/\eta^2 \leq 9/\eta^2$ because $\eta \in (0, 1]$. In fact, $\eta = |\mathcal{M}^{(t)}|^{-1/3}$

so the bound becomes

$$\mathcal{V}^{(t)}(\mathcal{A}) \leq \sum_{(u,v) \in \mathcal{M}^{(t)}} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)})\right) + 15|\mathcal{M}^{(t)}|^{2/3}.$$

□

From the definition of f and the construction of the matching $\mathcal{M}^{(t)}$, we can convert the previous bound into the following expression. We remark that the threshold-respecting property is only used in the proof of Lemma 4.5.

Lemma 4.6 *If \mathcal{A} is threshold-respecting on $W^{(t)}$, then*

$$\mathcal{V}^{(t)}(\mathcal{A}) \leq \frac{e-1}{e+1}m + 15t^{2/3}m^{2/3}.$$

Proof By splitting the matching $\mathcal{M}^{(t)}$ based on the m components, we can rewrite the bound in Lemma 4.5 as

$$\mathcal{V}^{(t)}(\mathcal{A}) \leq \sum_{i=1}^m \sum_{(u,v) \in \mathcal{M}_i^{(t)}} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)})\right) + 15|\mathcal{M}^{(t)}|^{2/3}. \quad (8)$$

Fix a component $i \in [m]$, and let $\mathcal{E}_i := \left\{e \in \mathcal{M}_i^{(t-1)} \mid f(\ell_u^{(t-1)}) + f(\ell_v^{(t-1)}) \geq 1\right\}$ be the subset of edges in the matching $\mathcal{M}_i^{(t-1)}$ which exceed the threshold at the end of phase $t-1$. By the construction of $\mathcal{M}_i^{(t)}$ in Section 4.3, we know that its node set consists of the nodes incident to \mathcal{E}_i , in addition to two new fresh nodes whose load is 0 at the end of phase $t-1$. This allows us to expand the inner sum in (8) as:

$$\begin{aligned} \sum_{(u,v) \in \mathcal{M}_i^{(t)}} 1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)}) &= 1 - 2f(0) + \sum_{(u,v) \in \mathcal{E}_i} 1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)}) \\ &\leq 1 - 2f(0) = \frac{e-1}{e+1} \end{aligned}$$

where the inequality follows from definition of \mathcal{E}_i . Plugging this into (8) with the bound $|\mathcal{M}^{(t)}| \leq tm$ (which is immediate by the construction in Section 4.3) yields the desired result. □

For an ε -threshold-respecting algorithm, we pick up an extra εtm term.

Corollary 4.1 *If \mathcal{A} is ε -threshold-respecting on $W^{(t)}$ for some $\varepsilon \geq 0$, then*

$$\mathcal{V}^{(t)}(\mathcal{A}) \leq \frac{e-1}{e+1}m + 15(tm)^{2/3} + \varepsilon tm.$$

Proof Fix an edge $e \in \mathcal{M}^{(t)}$. Let h_1, h_2, \dots, h_k be the hyperedges arriving in phase t which contain e , denoted such that h_i arrives before h_j if and only if $i < j$. Let $j \in [k]$ be the smallest index such that $\phi(h_j) > 1$ immediately after \mathcal{A} assigns x_{h_j} to h_j . Let $z_{h_j} \geq 0$ be the largest value such that $\phi(h_j) \geq 1$ if \mathcal{A} were to assign $x_{h_j} - z_{h_j}$ to h_j instead. Define $z_{h_i} := 0$ for all $i < j$, and $z_{h_i} := x_{h_i}$ for all $i > j$. Since \mathcal{A} is ε -threshold-respecting on $W^{(t)}$, we have $\sum_{i=1}^k z_{h_i} \leq \varepsilon$ because f is convex and $f' = f$.

Let z be the vector obtained by repeating this procedure on every edge $e \in \mathcal{M}^{(t)}$. Then, $\mathbb{1}^\top z \leq \varepsilon tm$ as $|\mathcal{M}^{(t)}| \leq tm$. Moreover, observe that the algorithm which assigns $x - z$ in phase t is threshold-respecting on $W^{(t)}$. Thus, we can apply Lemma 4.6 to obtain the desired upper bound on $\mathbb{1}^\top(x - z)$. \square

4.7 Putting everything together

In this section, we complete the proof of Theorem 1. Since we assumed that the algorithm is ε -threshold respecting in the first $T - 1$ phases, we can apply Corollary 4.1 to upper bound the value obtained in the first $T - 1$ phases as

$$\sum_{t=1}^{T-1} \left(\frac{e-1}{e+1} m + \varepsilon tm + O\left((tm)^{2/3}\right) \right) \leq \frac{e-1}{e+1} Tm + \varepsilon T^2 m + O\left(T^{5/3} m^{2/3}\right).$$

By Theorem 4, the value gained by the algorithm on each component C_i during the last phase T is at most $O(\sqrt{T} + \varepsilon T^2)$. Hence, the algorithm gains at most $O(\sqrt{T}m + \varepsilon T^2 m)$ in the last phase.

We now argue that our instance $\mathcal{H} = (V, W, H)$ has a perfect matching.

Lemma 4.7 *Our adversarial instance $\mathcal{H} = (V, W, H)$ satisfies $\text{OPT}(\mathcal{H}) = Tm$.*

Proof We prove that for every $t \in [T]$, there exists a hypergraph matching of size tm at the end of phase t . Let C_i be a component with bipartition $U_i = [T]$ and $V_i = [T]$. It suffices to show that there exists a graph matching $\widetilde{\mathcal{M}}_i^{(t)}$ with vertex set $[t]$ on each side. This is because $\widetilde{\mathcal{M}}^{(t)} := \cup_{i=1}^m \widetilde{\mathcal{M}}_i^{(t)}$ can be extended to a hypergraph matching in \mathcal{H} by our construction (see Lemma 4.4). Let $E_i^{(t)} \subseteq \mathcal{M}_i^{(t)}$ be the edges whose endpoints are not active at the end of phase t . Then, a simple inductive argument on $t \geq 1$ shows that $\cup_{s=1}^{t-1} E_i^{(s)} \cup \mathcal{M}_i^{(t)}$ is a graph matching with vertex set $[t]$ on each side (see Figure 4 for an example). \square

By Lemma 4.7, the competitive ratio of the algorithm is at most

$$\frac{e-1}{e+1} + O(\varepsilon T + T^{2/3} m^{-1/3} + T^{-1/2}).$$

Hence, letting $m \rightarrow \infty$, picking $T = o(\sqrt{m})$ such that $T \rightarrow \infty$ and setting $\varepsilon = o(1/T)$, we conclude that the competitive ratio is upper bounded by $(e - 1)/(e + 1)$, thus finishing the proof of Theorem 1.

5 Integral algorithm for bounded degree hypergraphs

In this section, we show that RANDOM (Algorithm 5.1) performs better than the greedy algorithm when the online nodes have bounded degree.

Algorithm 5.1 RANDOM algorithm for bounded degree hypergraphs

Input : k -uniform hypergraph $\mathcal{H} = (V, W, H)$ with online arrivals of each $w \in W$ with $|\delta(w)| \leq d$.

Output : Matching $\mathcal{M} \subseteq H$

set $\mathcal{M} \leftarrow \emptyset$

when $w \in W$ **arrives with** $\delta(w) \subseteq H$:

pick uniformly at random $h \in \delta(w)$ among the hyperedges that are disjoint from \mathcal{M}

set $y_v = \min\left(\frac{1}{k-1}, \frac{d}{(d-1)k+1}\right)$ for all $v \in h \setminus \{w\}$

set $y_w = \max\left(0, \frac{d-k+1}{(d-1)k+1}\right)$

return \mathcal{M}

We prove Theorem 2, restated below.

Theorem 5 *Algorithm 5.1 is ρ -competitive for k -uniform hypergraphs whose online nodes have degree at most d , where*

$$\rho = \min\left(\frac{1}{k-1}, \frac{d}{(d-1)k+1}\right).$$

Proof Let the algorithm be denoted by \mathcal{A} . We prove the result via a primal-dual analysis, where the random primal solution is given by $x_h := \mathbb{1}_{\{h \in \mathcal{M}\}}$ for every $h \in H$ and the random dual solution is the vector $y \in [0, 1]^{V \cup W}$ constructed during the execution of the algorithm. Observe that the objective values of both solutions are equal at all times during the execution of the algorithm:

$$\mathcal{V}(\mathcal{A}) = |\mathcal{M}| = \sum_{h \in H} x_h = \sum_{v \in V \cup W} y_v. \quad (9)$$

This holds since every time a hyperedge $h \in H$ is matched by the algorithm, increasing the primal value $\mathcal{V}(\mathcal{A})$ by one, the dual objective increases by $\sum_{v \in h} y_v$. Two easy computations that we omit show that the latter is also equal to one in both cases where $d \leq k-1$ and $d \geq k-1$.

We will now show that, in expectation, the dual constraints are satisfied up to a factor of ρ , i.e.

$$\mathbb{E}\left[\sum_{v \in h} y_v\right] \geq \rho \quad \forall h \in H. \quad (10)$$

This will imply the theorem, since the random vector $\mathbb{E}[y]/\rho$ will then be a feasible dual solution, leading to $\mathbb{E}[\mathcal{V}(\mathcal{A})] = \mathbb{E}\left[\sum_{v \in V \cup W} y_v\right] \geq \rho \text{OPT}_{\text{LP}}$ by (9) and (10).

To show this inequality, let $h \in H$ be an arbitrary hyperedge incident to some online node $w \in W$. We now consider the following probabilistic event upon the arrival of w :

$$\mathcal{E} := \left\{ \exists v \in h \setminus \{w\} \text{ which is already matched at the arrival of } w \right\}.$$

We will show (10) by conditioning on \mathcal{E} and on its complementary event $\bar{\mathcal{E}}$, which states that all nodes in $h \setminus \{w\}$ are unmatched when w arrives, and that the hyperedge h is thus available and considered in the random choice of the algorithm in this step. In the first case, if \mathcal{E} happens, then some offline node $u \in h \setminus \{w\}$ has already had its dual value set to $y_u = \min\left(\frac{1}{k-1}, \frac{d}{(d-1)k+1}\right) = \rho$ in a previous step of the algorithm, leading to

$$\mathbb{E} \left[\sum_{v \in h} y_v \mid \mathcal{E} \right] = \sum_{v \in h} \mathbb{E}[y_v \mid \mathcal{E}] \geq \mathbb{E}[y_u \mid \mathcal{E}] = \rho.$$

Otherwise, if $\bar{\mathcal{E}}$ happens, we know that with probability at least $1/d$, the algorithm adds h to the matching. Summing the dual values of the offline nodes contained in h gives

$$\sum_{v \in h \setminus \{w\}} \mathbb{E}[y_v \mid \bar{\mathcal{E}}] \geq \frac{1}{d} \cdot (k-1) \cdot \min\left(\frac{1}{k-1}, \frac{d}{(d-1)k+1}\right).$$

Furthermore, since the algorithm will always match w to a hyperedge in this case, we have:

$$\mathbb{E}[y_w \mid \bar{\mathcal{E}}] = \max\left(0, \frac{d-k+1}{(d-1)k+1}\right).$$

Adding those terms together, we get:

$$\begin{aligned} \sum_{v \in h} \mathbb{E}[y_v \mid \bar{\mathcal{E}}] &\geq \frac{1}{d} \cdot (k-1) \cdot \min\left(\frac{1}{k-1}, \frac{d}{(d-1)k+1}\right) + \max\left(0, \frac{d-k+1}{(d-1)k+1}\right) \\ &= \min\left(\frac{1}{d}, \frac{k-1}{(d-1)k+1}\right) + \max\left(0, \frac{d-(k-1)}{(d-1)k+1}\right) \\ &\geq \min\left(\frac{1}{d}, \frac{k-1}{(d-1)k+1} + \frac{d-(k-1)}{(d-1)k+1}\right) \geq \rho. \end{aligned}$$

This shows that (10) holds, and hence proves that the algorithm is ρ -competitive. \square

6 Concluding remarks

Our main contribution was a tight result for the online fractional matching problem under vertex arrivals for 3-uniform hypergraphs, with matching upper and lower bounds of $(e-1)/(e+1) \approx 0.46$. The biggest open question raised by this work is the integral setting, where the best known algorithm is still the $1/3$ -competitive simple

greedy algorithm. Possible directions to get an improvement would be to develop new rounding techniques of fractional algorithms on hypergraphs, or by trying to find an analogue of the randomized RANKING algorithm for bipartite graphs. This hypergraph setting adds additional difficulties to the online matching problem, for which new techniques seem to be required. Another interesting direction would be to study this problem in different online arrival models, for instance edge-arrivals.

For k -uniform hypergraphs with $k \geq 3$ under vertex arrivals, the integral setting admits upper and lower bounds of $1/k$ and $2/k$, whereas the fractional setting has upper and lower bounds of $\Theta(1/\log k)$. It would still be interesting to derive the optimal closed form solution depending on k for both of these settings, under either vertex or edge arrivals. Note that for large k , both models are very similar, as there is a simple reduction from k -uniform hypergraphs under edge-arrivals to $(k + 1)$ -uniform hypergraphs under vertex-arrivals by appending degree one online nodes.

For both the integral and fractional settings, an additional direction would be to study the variant in which previously added hyperedges can be removed from the matching. This is a natural question, as both our upper bound constructions for the fractional and integral settings rely on the fact that an algorithm cannot revoke previous decisions.

Funding Open Access funding enabled and organized by Projekt DEAL. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement QIP-805241).

Data Availability Not applicable.

Declarations

Competing interests There are no competing interests to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A Proof of Lemma 4.4

Proof Let us fix a fractional algorithm \mathcal{A} and let us fix a matching $\mathcal{M} = (V, E)$ of size n , meaning that $|E| = n$. The adversarial online 3-uniform hypergraph instance \mathcal{H} consists of n online nodes $W = \{w_1, \dots, w_n\}$ arriving and connecting to a subset of edges of the matching \mathcal{M} . For every w_i , we denote by $E(w_i) \subseteq E$ the edges of the matching the online node w_i is connected to, meaning that the 3-hyperedges incident to w_i are $\delta(w_i) = \{w_i \cup e : e \in E(w_i)\}$. Let us denote by $x \in \mathbb{R}^E$ the fractional solution generated online by algorithm \mathcal{A} , and note that this is in fact the induced fractional solution on E .

1. The first online node w_1 connects to every edge of the matching, i.e. $E(w_1) = E$. The algorithm \mathcal{A} now assigns fractional value $x(e)$ to every edge $e \in E$, and we denote by $e_1 \in E$ the edge with the lowest fractional value $x(e_1)$. Observe that $x(e_1) \leq 1/n$.
2. The second online node w_2 connects to $E(w_2) = E \setminus \{e_1\}$. The algorithm \mathcal{A} can thus increase the fractional values $x(e)$ for every $e \in E(w_2)$. We then denote by e_2 the edge in $E(w_2)$ with the lowest fractional value after this iteration, and it is easy to check that $x(e_1) + x(e_2) \leq 2/n + 1/(n - 1)$.
3. More generally, for every $k \in \{1, \dots, n\}$, the online node w_k connects to $n - k + 1$ edges $E(w_k) = E \setminus \{e_1, \dots, e_{k-1}\}$, and e_k is defined as the edge having the lowest fractional value at the end of the iteration of w_k . We thus get a bound of

$$\sum_{k=1}^{\ell} x(e_k) \leq \sum_{k=1}^{\ell} \sum_{i=1}^k \frac{1}{n - i + 1} \quad \forall \ell \in \{1, \dots, n\}. \tag{11}$$

The inner sum in (11) reaches Δ approximately when $k \approx (1 - e^{-\Delta})n$. For higher values of k , it is thus better to use the bound $x(e_k) \leq \Delta$, which holds by assumption. By defining $p := \lfloor e^{-\Delta}n \rfloor$ and $q := n - p$, we can now compute a precise upper bound on the total value generated by the algorithm using (11):

$$\begin{aligned} \mathcal{V}(\mathcal{A}, \mathcal{H}) &= \sum_{k=1}^n x(e_k) \leq \sum_{k=1}^q \sum_{i=1}^k \frac{1}{n - i + 1} + \sum_{k=q+1}^n \Delta = \sum_{i=1}^q \sum_{k=i}^q \frac{1}{n - i + 1} + p\Delta \\ &= \sum_{i=1}^q \frac{q - i + 1}{n - i + 1} + p\Delta = q - (n - q) \sum_{i=1}^q \frac{1}{n - i + 1} + p\Delta \\ &= p\Delta + (n - p) - p \sum_{i=n-q+1}^n \frac{1}{i} = p\Delta + n - p - p(H_n - H_p) \end{aligned} \tag{12}$$

In order to get the desired result for every value of $n \geq 1$, we now need to tightly approximate the difference of the harmonic numbers $H_n - H_p$. In particular, the well known bounds $\ln(n) + 1/n \leq H_n \leq \ln(n + 1)$ for every $n \in \mathbb{N}$ are not enough in this case. We use the equality

$$H_n = \ln(n) + \gamma + \epsilon(n) \quad \text{for some } 0 < \epsilon(n) < \frac{1}{2n} \tag{13}$$

where $\gamma = \lim_{n \rightarrow \infty} (H_n - \ln(n)) \approx 0.58$ is Euler’s constant. Moreover, recall that

$$e^{-\Delta}n - 1 \leq p \leq e^{-\Delta}n. \tag{14}$$

Using (13) and (14) together gives:

$$H_n - H_p = \ln\left(\frac{n}{p}\right) + \epsilon(n) - \epsilon(p) \geq \ln\left(\frac{n}{e^{-\Delta}n}\right) - \frac{1}{2p} = \Delta - \frac{1}{2p}$$

(15)

Finally, plugging (14) and (15) into (12) gets us the desired result for every value of $n \in \mathbb{N}$:

$$\mathcal{V}(A, \mathcal{H}) \leq p\Delta + n - p - p \left(\Delta - \frac{1}{2p} \right) = n - p + \frac{1}{2} \leq (1 - e^{-\Delta})n + \frac{3}{2}.$$

□

B Justification of assumptions in Section 4.2

In this section, we justify the two assumptions made on the algorithm in Section 4.2.

B.1 Assumption 1: Symmetry

We start by justifying the symmetry assumption. For a vertex-arrival hypergraph \mathcal{H} , we denote $V(\mathcal{H})$ as the set of offline nodes, $W(\mathcal{H})$ as the set of online nodes, and $H(\mathcal{H})$ as the set of hyperedges.

Definition 2 For a vertex-arrival hypergraph $\mathcal{H} = (V, W, H)$, an *automorphism* is a permutation σ of the offline nodes V such that for every $S \subseteq V$ and $w \in W$,

$$S \cup \{w\} \in H \iff \{\sigma(v) : v \in S\} \cup \{w\} \in H.$$

For a subset $S \subseteq V$, we write $\sigma(S) := \{\sigma(v) : v \in S\}$. For a hyperedge $h = S \cup \{w\}$ where $S \subseteq V$ and $w \in W$, we also write $\sigma(h) := \sigma(S) \cup \{w\}$.

Definition 3 For a vertex-arrival hypergraph $\mathcal{H} = (V, W, H)$, let Σ be a subset of its automorphisms. We say that a fractional matching x of \mathcal{H} is Σ -*symmetric* if $x_h = x_{\sigma(h)}$ for all $h \in H$ and $\sigma \in \Sigma$.

Since the construction of our vertex-arrival instance depends on the actions of the algorithm, we will overload the notation \mathcal{H} as follows. A (*adaptive*) *vertex-arrival instance* is a function \mathcal{H} which takes as input an algorithm \mathcal{A} and outputs a vertex-arrival hypergraph $\mathcal{H}(\mathcal{A})$. For concreteness, the reader can think of \mathcal{H} as the instance constructed in Section 4. For $i \geq 1$, let \mathcal{H}_i be the subinstance of \mathcal{H} that ends with the arrival of the i th online node w_i . We can assume that $\mathcal{H}_1(\mathcal{A}) = \mathcal{H}_1(\mathcal{A}')$ for any pair of algorithms \mathcal{A} and \mathcal{A}' , as algorithms do not take any actions before the first online node arrives. So, the subinstance \mathcal{H}_1 can be thought of as a hypergraph.

Definition 4 For a vertex-arrival instance \mathcal{H} and an algorithm \mathcal{A} , let Σ be a subset of automorphisms of $\mathcal{H}(\mathcal{A})$. We say that \mathcal{A} is Σ -*symmetric on \mathcal{H}* if it returns a Σ -symmetric fractional matching of $\mathcal{H}(\mathcal{A})$ given \mathcal{H} .

Fix a vertex-arrival instance \mathcal{H} , and let Σ be a subgroup of automorphisms of \mathcal{H}_1 . For an algorithm \mathcal{A} , we define its Σ -*symmetrization* $\text{sym}_\Sigma(\mathcal{A})$ as follows. We first run

\mathcal{A} on \mathcal{H}_1 to obtain a fractional matching x' . Then, $\text{sym}_\Sigma(\mathcal{A})$ sets

$$x_h := \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} x'_{\sigma(h)} \quad \forall h \in H(\mathcal{H}_1).$$

This in turn determines the next set of hyperedges in the hypergraph $\mathcal{H}_2(\text{sym}_\Sigma(\mathcal{A}))$. Note that $\mathcal{H}_2(\text{sym}_\Sigma(\mathcal{A}))$ may not be equal to $\mathcal{H}_2(\mathcal{A})$. As long as Σ remains a subgroup of automorphisms of \mathcal{H}_2 , we repeat the process above. So, for any $i \geq 2$, if x' is the fractional matching obtained by running \mathcal{A} on $\mathcal{H}_i(\text{sym}_\Sigma(\mathcal{A}))$, then $\text{sym}_\Sigma(\mathcal{A})$ sets

$$x_h := \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} x'_{\sigma(h)} \quad \forall h \in H(\mathcal{H}_i(\text{sym}_\Sigma(\mathcal{A}))).$$

Note that $\text{sym}_\Sigma(\mathcal{A})$ is well-defined if and only if Σ is a subgroup of automorphisms of the full hypergraph $\mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$.

Lemma B.1 *If $\text{sym}_\Sigma(\mathcal{A})$ is well-defined, then it is Σ -symmetric on \mathcal{H} .*

Proof Let x be the fractional matching obtained by running $\text{sym}_\Sigma(\mathcal{A})$ on $\mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$, and let x' be the fractional matching obtained by running \mathcal{A} on $\mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$. For any hyperedge h in $\mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$ and any permutation $\tau \in \Sigma$, we have

$$x_h = \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} x'_{\sigma(h)} = \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} x'_{\sigma(\tau(h))} = x_{\tau(h)},$$

where the second equality is due to Σ being a group. □

The following lemma shows that in order to construct a worst-case instance for all algorithms, it suffices to construct a worst-case instance \mathcal{H} for algorithms that are Σ -symmetric on \mathcal{H} . That is, \mathcal{H} can be extended to an instance \mathcal{H}' for all algorithms on which every algorithm has the same performance as its Σ -symmetrization.

Lemma B.2 *For a vertex-arrival instance \mathcal{H} , let Σ be a subgroup of automorphisms of \mathcal{H}_1 . Suppose that Σ remains a subgroup of automorphisms of $\mathcal{H}_i(\mathcal{A})$ for any $i \geq 2$ and any algorithm \mathcal{A} that is Σ -symmetric on \mathcal{H}_{i-1} . Then, there exists a vertex-arrival instance \mathcal{H}' such that for every algorithm \mathcal{A} :*

- $\mathcal{H}'(\mathcal{A}) = \mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$,
- $\mathcal{V}(\mathcal{A}, \mathcal{H}') = \mathcal{V}(\text{sym}_\Sigma(\mathcal{A}), \mathcal{H})$.

Proof Let \mathcal{A} be any algorithm. Observe that $\text{sym}_\Sigma(\mathcal{A})$ is well-defined due to our assumption. Hence, $\text{sym}_\Sigma(\mathcal{A})$ is Σ -symmetric on \mathcal{H} by Lemma B.1. We define $\mathcal{H}'(\mathcal{A}) := \mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$. Let x be the fractional matching obtained by running $\text{sym}_\Sigma(\mathcal{A})$ on $\mathcal{H}(\text{sym}_\Sigma(\mathcal{A}))$, and let x' be the fractional matching obtained by running \mathcal{A} on $\mathcal{H}'(\mathcal{A})$. By definition, we have $x_h := \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} x'_{\sigma(h)}$ for every hyperedge $h \in H(\mathcal{H}(\text{sym}_\Sigma(\mathcal{A})))$. Hence,

$$\sum_h x_h = \sum_h \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} x'_{\sigma(h)} = \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} \sum_h x'_{\sigma(h)} = \sum_h x'_h.$$

□

In Section 4.2, we assumed that the algorithm treats the k th vertex in U_i , say $u_{i,k}$, and the k th vertex in V_i , say $v_{i,k}$, symmetrically. If our constructed hypergraph \mathcal{H} was symmetric with respect to these vertices, i.e. if the permutation σ swapping $u_{i,k}$ and $v_{i,k}$ for all i and k was an automorphism of \mathcal{H} , then Lemma B.2 would show that this assumption can be made without loss of generality. In particular, using the subgroup $\Sigma = (\{\sigma, e\}, \circ)$ where e is the identity permutation, it shows that \mathcal{H} can be extended to all algorithms \mathcal{A} so that \mathcal{A} and $\text{sym}_\Sigma(\mathcal{A})$ have the same performance.

However, one part of the instance that breaks this symmetry is the construction given in the proof of Lemma 4.4 and illustrated in Figure 7. As a reminder, this construction is repeatedly applied to submatchings of $\mathcal{M}^{(t)}$ in Section 4.5. Let us fix one such submatching and denote it by $\mathcal{M} := \mathcal{M}^{(t)}(i, j)$. As described in Section 4.3 and illustrated in Figure 4, if some $u_{i,k} \in \mathcal{M}$, then $v_{i,k} \in \mathcal{M}$ and the submatching is symmetric with respect to this pair, i.e. $\sigma(e) \in \mathcal{M}$ for every edge e in \mathcal{M} . However, due to the Lemma 4.4 construction, $e \cup \{w\}$ might be a hyperedge in \mathcal{H} for some online vertex w , while $\sigma(e \cup \{w\}) = \sigma(e) \cup \{w\}$ might not be a hyperedge in \mathcal{H} .

To fix this, the construction can be slightly tweaked in the following way. An important observation is that the horizontal edges in \mathcal{M} (between $u_{i,k}$ and $v_{i,k}$) are not isomorphic to any other edge in the hypergraph, whereas each of the diagonal edges (non-horizontal edges) are isomorphic to exactly one other edge in \mathcal{M} . For this reason, we can first apply the Lemma 4.4 construction on just the horizontal edges of \mathcal{M} .

We can then apply a slightly modified construction to the diagonal edges, where the pairs of isomorphic edges are treated in the same way. In the original construction, a newly arriving online vertex w would be connected to all edges in \mathcal{M} that were incident to the previous online vertex, except for the one with the smallest fractional value. In the modified construction, we instead consider the online vertices in groups of two. For every two consecutive online vertices, we connect them to all edges in \mathcal{M} that were incident to the previous online vertex, except for the diagonal pair with the smallest total fractional value. This ensures that the symmetry between the diagonal edges is respected.

This modified construction would slightly worsen the upper bound in Lemma 4.4. Let \mathcal{M}_{hor} be the set of horizontal edges in \mathcal{M} and $\mathcal{M}_{\text{diag}}$ be the set of diagonal edges in \mathcal{M} . By applying Lemma 4.4 to the horizontal edges, we get that the value of the matching is at most $(1 - e^{-\Delta})|\mathcal{M}_{\text{hor}}| + 3/2$. The value of the diagonal edges is at most twice the value of the original construction from Lemma 4.4 applied to a transversal of the pairs of diagonal edges, which is at most $(1 - e^{-\Delta}) \cdot \frac{1}{2}|\mathcal{M}_{\text{diag}}| + 3/2$ by Lemma 4.4. So the total value of the matching is at most:

$$(1 - e^{-\Delta})|\mathcal{M}_{\text{hor}}| + 3/2 + 2 \cdot \left((1 - e^{-\Delta}) \cdot \frac{1}{2}|\mathcal{M}_{\text{diag}}| + 3/2 \right) = (1 - e^{-\Delta})|\mathcal{M}| + 9/2.$$

This results in a constant of 9/2 instead of 3/2 in Lemma 4.4, and a constant of 42 instead of 15 in Lemmas 4.5 and 4.6 and Corollary 4.1. This does not affect the

asymptotic upper bound for large m . Hence, it shows that Theorem 1 also holds for non-symmetric algorithms.

B.2 Assumption 2: There is an optimal ε -threshold respecting algorithm

Next, we justify that we restrict to ε -threshold respecting algorithms in our proof. Let \mathcal{H} be the instance constructed in Section 4.1. Let f be the function given by

$$f(x) := \frac{e^x}{e + 1},$$

and recall the definition of ε -threshold respecting with respect to f (Definition 1).

For any algorithm \mathcal{A} and $\varepsilon > 0$, we now show that there exists an algorithm \mathcal{A}' which is ε -threshold respecting on all online nodes before the last phase. Moreover, there exists an instance \mathcal{H}' such that the performance of \mathcal{A} on \mathcal{H}' matches the performance of \mathcal{A}' on \mathcal{H} .

Lemma B.3 *Let \mathcal{H} be the instance constructed in Section 4.1. For any algorithm \mathcal{A} and $\varepsilon > 0$, there exists an algorithm \mathcal{A}' which is ε -threshold respecting on all online nodes before the last phase. Furthermore, there exists an instance \mathcal{H}' such that*

$$\frac{\mathcal{V}(\mathcal{A}, \mathcal{H}'(\mathcal{A}))}{\text{OPT}(\mathcal{H}'(\mathcal{A}))} = \frac{\mathcal{V}(\mathcal{A}', \mathcal{H}(\mathcal{A}'))}{\text{OPT}(\mathcal{H}(\mathcal{A}'))}.$$

Proof From \mathcal{H} , we construct a new instance \mathcal{H}' as follows. Let $N = \lceil 2/\varepsilon \rceil$. For every offline node v in \mathcal{H} , create N offline copies in \mathcal{H}' , denoted v'_1, v'_2, \dots, v'_N . The new algorithm \mathcal{A}' will be defined based on the behaviour of \mathcal{A} on \mathcal{H}' . When the i th online node w_i arrives in $\mathcal{H}(\mathcal{A}')$ for $i \geq 1$, at most N copies of w_i arrives sequentially in $\mathcal{H}'(\mathcal{A})$, denoted $w'_{i,1}, w'_{i,2}, \dots$. When the j th copy $w'_{i,j}$ arrives, for every edge $h = S \cup w_i$ in $\mathcal{H}_i(\mathcal{A}')$, add the edge $h'_j := \{v'_j : v \in S\} \cup w'_{i,j}$ to $\mathcal{H}'(\mathcal{A})$. Now, let $x'_{i,j}$ denote the solution given by \mathcal{A} in \mathcal{H}' after the arrival of $w'_{i,j}$. Consider the following averaged solution

$$x_{i,j}(h) := \frac{1}{N} \sum_k x'_{i,j}(h'_k) \quad \forall h \in H(\mathcal{H}_i(\mathcal{A}')).$$

If $j = N$, or w_i appeared before the last phase and there exists a hyperedge $h \in \delta(w_i)$ such that

$$\sum_{v \in h \setminus \{w_i\}} f(x_{i,j}(\delta(v))) \geq 1,$$

then $w'_{i,j+1}, \dots, w'_{i,N}$ will not arrive in \mathcal{H}' . In this case, \mathcal{A}' sets $x(h) \leftarrow x_{i,j}(h)$ for all $h \in \delta(w_i)$ in \mathcal{H} . Otherwise, we proceed to let the $(j + 1)$ th copy $w'_{i,j+1}$ arrive in \mathcal{H}' . This completes the description of \mathcal{A}' .

Clearly, x is a fractional matching in $\mathcal{H}(\mathcal{A}')$. Moreover,

$$\mathcal{V}(\mathcal{A}', \mathcal{H}(\mathcal{A}')) = \sum_h x(h) = \frac{\mathcal{V}(\mathcal{A}, \mathcal{H}'(\mathcal{A}))}{N}.$$

Next, we claim that $N \cdot \text{OPT}(\mathcal{H}(\mathcal{A}')) = \text{OPT}(\mathcal{H}'(\mathcal{A}))$. Based on the construction of \mathcal{H} , the offline optimal matching in $\mathcal{H}(\mathcal{A}')$ covers all the online nodes in the last phase, and the online nodes on which \mathcal{A}' is strictly threshold respecting. Let W_{OPT} denote the union of these two sets. For each $w_i \in W_{\text{OPT}}$, observe that $w_{i,j}$ is present in $\mathcal{H}'(\mathcal{A})$ for all $j \in [N]$ by our construction of \mathcal{H}' . Hence, the offline optimal matching in $\mathcal{H}'(\mathcal{A})$ covers the following online nodes

$$\{w_{i,j} : w_i \in W_{\text{OPT}}, j \in [N]\}.$$

So, $N \cdot \text{OPT}(\mathcal{H}(\mathcal{A}')) = \text{OPT}(\mathcal{H}'(\mathcal{A}))$ as desired.

It is left to show that \mathcal{A}' is ε -threshold respecting on all online nodes before the last phase. Pick such an online node w_i and let $w_{i,j}$ be its last copy in $\mathcal{H}'(\mathcal{A})$. Note that $x_{i,j}$ is the output of \mathcal{A}' in $\mathcal{H}_i(\mathcal{A}')$. For any $h \in \delta(w_i)$, we have

$$\begin{aligned} \sum_{v \in h \setminus \{w_i\}} f(x_{i,j}(\delta(v))) &\leq \sum_{v \in h \setminus \{w_i\}} f\left(x_{i,j-1}(\delta(v)) + \frac{1}{N}\right) && (x'_{i,j}(\delta(w'_{i,j})) \leq 1) \\ &\leq \sum_{v \in h \setminus \{w_i\}} \left(f(x_{i,j-1}(\delta(v)) + \frac{1}{N})\right) && (f \text{ is 1-Lipschitz}) \\ &< 1 + \frac{2}{N} && (\text{due to } |h| = 3 \text{ and the construction of } \mathcal{H}') \\ &\leq 1 + \varepsilon \end{aligned}$$

□

C Properties of $\psi(t, y)$

Proof of Lemma 4.1 The first two statements follow directly from the definition and the symmetry of $B(t, \frac{1}{2})$ around $\frac{t}{2}$. For the second statement, let $X \sim B(t, \frac{1}{2})$ and $Y \sim B(1, \frac{1}{2})$ be independent. Then, we have:

$$\begin{aligned} \psi(t+1, y) &= \Pr\left[X+Y < \frac{t+1}{2} + y\right] + \frac{1}{2} \Pr\left[X+Y = \frac{t+1}{2} + y\right] \\ &= \Pr[Y=1] \Pr\left[X < \frac{t}{2} + y - \frac{1}{2}\right] + \Pr[Y=0] \Pr\left[X < \frac{t}{2} + y + \frac{1}{2}\right] \\ &\quad + \frac{1}{2} \Pr[Y=0] \Pr\left[X = \frac{t}{2} + y + \frac{1}{2}\right] + \frac{1}{2} \Pr[Y=1] \Pr\left[X = \frac{t}{2} + y - \frac{1}{2}\right] \\ &= \frac{1}{2} \psi\left(t, y - \frac{1}{2}\right) + \frac{1}{2} \psi\left(t, y + \frac{1}{2}\right). \end{aligned}$$

Now, let us prove the last statement. Let $X \sim B(t, \frac{1}{2})$ and observe that $1 - \psi(t, y) \leq \Pr[X \geq \frac{t}{2} + y]$, leading to:

$$\begin{aligned} \sum_{y=0}^{\infty} (1 - \psi(t, \frac{1}{2}y)) &\leq \sum_{y=0}^{\infty} \Pr \left[X - \frac{t}{2} \geq \frac{y}{2} \right] \leq \sum_{y=0}^{\infty} \Pr \left[\left| X - \frac{t}{2} \right| \geq \frac{y}{2} \right] \\ &= \sum_{y=0}^{\infty} (y + 1) \Pr \left[\left| X - \frac{t}{2} \right| = \frac{y}{2} \right] = 1 + 2 \sum_{y=0}^{\infty} \frac{y}{2} \Pr \left[\left| X - \frac{t}{2} \right| = \frac{y}{2} \right] \\ &= 1 + 2 \mathbb{E} \left[\left| X - \frac{t}{2} \right| \right] \leq 1 + 2 \sqrt{\mathbb{E} \left[\left(X - \frac{t}{2} \right)^2 \right]} \quad (\text{by Jensen's inequality}) \\ &= 1 + 2\sqrt{\text{Var}[X]} = 1 + \sqrt{t}. \end{aligned}$$

D Integral upper bound for k -uniform hypergraphs

In this section, we prove a strong upper bound against any randomized integral algorithm, showing that the greedy algorithm is almost optimal, since it achieves a competitive ratio of $1/k$.

Theorem 6 *For the online matching problem on k -uniform hypergraphs, no randomized integral algorithm can be $2/k$ -competitive.*

Proof We prove that any randomized integral algorithm is at most $(2 - 2^{-k+1})/k$ -competitive. To do so, we make use of Yao's principle [34]: it suffices to construct a randomized instance for which any deterministic integral algorithm is at most $(2 - 2^{-k+1})/k$ -competitive in expectation. Let us now describe our randomized construction $\mathcal{H} = (V, W, H)$ for any $k \in \mathbb{N}$.

- The offline nodes are partitioned into $k - 1$ blocks: $V = C_1 \cup \dots \cup C_{k-1}$, where $|C_i| = 2(k - i)$ for each $i \in \{1, \dots, k - 1\}$, meaning that $|V| = k(k - 1)$.
- The instance first consists of $k - 1$ phases with online nodes w_1, \dots, w_{k-1} arriving, all of which are incident to 2 hyperedges. For every $i \in \{1, \dots, k - 1\}$, both hyperedges incident to w_i are disjoint on the offline nodes V and they will both contain $(k - i)$ nodes from C_i , as well as 1 node from each C_j for $j \in \{1, \dots, i - 1\}$.
- We now construct a random subset of hyperedges $H_1 \subseteq H$ in the following way. At the arrival of w_i for every $i \leq k - 1$, pick one of the two incident hyperedges uniformly at random and put it in H_1 . We denote by $V(H_1)$ the offline nodes spanned by H_1 .
- Our construction will now satisfy the following property. For every $i \in \{1, \dots, k - 1\}$, after the arrival of w_i and the random choice described above, we have that

$$|C_j \setminus V(H_1)| = k - i \quad \forall j \in \{1, \dots, i\}. \tag{16}$$

After a certain phase $i - 1 \leq k - 2$, the two hyperedges incident to w_i in the next iteration are then both constructed as follows: take $k - i$ nodes from C_i and complete it by arbitrarily picking one node from $C_j \setminus V(H_1)$ for every $j \leq i - 1$.

- After phase $k - 1$, we have that $|C_j \setminus V(H_1)| = 1$ for every $j \leq k - 1$, by invariant (16). The instance now makes one more online node w_k incident to one hyperedge arrive, whose offline nodes are $C_j \setminus V(H_1)$ for every $j \in \{1, \dots, k - 1\}$. Let us also add this hyperedge to H_1 .

Let us first show that (16) holds by induction. In the first phase, both hyperedges partition C_1 on the offline nodes, since $|C_1| = 2(k - 1)$. One of them is chosen to enter H_1 , meaning that $C_1 \setminus V(H_1) = k - 1$ after phase 1. Let us now fix a phase $i \leq k - 1$ and suppose that (16) holds for all previous phases. By construction, C_i is completely covered by the two hyperedges arriving at phase i , since $|C_i| = 2(k - i)$ and both of these hyperedges contain $k - i$ nodes from C_i . One of these hyperedges enters H_1 at the end of phase i , meaning that $|C_i \setminus V(H_1)| = k - i$ indeed holds. For any other C_j with $j < i$, note that $|C_j \setminus V(H_1)| = k - i + 1$ at the beginning of phase i , by induction hypothesis. Both hyperedges coming at phase i intersect C_j at two different nodes, one of which enters $V(H_1)$ by the random choice, meaning that $|C_j \setminus V(H_1)|$ drops by 1 and equals $k - i$, thus showing (16).

Observe that, by construction, the hyperedges in H_1 are all disjoint from each other. Since we add one hyperedge to H_1 for every online node, we get that $\text{OPT}(\mathcal{H}) = k$.

Let us now upper bound the value that any deterministic algorithm can get on this randomized instance. The key observation is that, if the algorithm picks a hyperedge $h \in \delta(w_i)$ which is not placed in H_1 for some phase $i \in \{1, \dots, k - 1\}$, then it cannot pick any hyperedge arriving in later iterations. This holds, since in that case, $C_i \setminus V(H_1) \subseteq h$, and any hyperedges arriving in later iterations necessarily intersect $C_i \setminus V(H_1)$ by construction.

Let us denote by \mathcal{V}_i the maximum expected value that a deterministic algorithm can get if we were to start the instance from phase i and go up to phase k . Clearly, $\mathcal{V}_k = 1$. For a phase $i \in \{1, \dots, k - 1\}$, the algorithm can either choose not to select anything, or it picks a hyperedge and cannot pick anything in later iterations with probability $1/2$. We thus get the following recurrence relation:

$$\mathcal{V}_i = \max \left\{ \mathcal{V}_{i+1}, \frac{1}{2} + \frac{1}{2}(1 + \mathcal{V}_{i+1}) \right\} = \max \left\{ \mathcal{V}_{i+1}, 1 + \frac{1}{2}\mathcal{V}_{i+1} \right\}.$$

It is easily checked that the solution to this recurrence is a geometric series $\mathcal{V}_{k-i} = \sum_{j=0}^i 2^{-j}$ and thus $\mathcal{V}_1 = \sum_{j=0}^{k-1} 2^{-j} = 2 - 2^{-k+1}$. We have therefore just shown that any algorithm is at most $(2 - 2^{-k+1})/k$ competitive. \square

E Rounding algorithm for online hypergraph b -matching

In this section, we consider the online b -matching problem on k -uniform hypergraphs, in which every (offline and online) node v can be matched to at most b hyperedges. We show that, for $b = \Omega(\log k)$, any fractional algorithm can be converted to a randomized integral algorithm while incurring a small loss in the competitive ratio.

Let \mathcal{A} be a fractional algorithm that is ρ -competitive and let $\mathcal{H} = (V, W, H)$ be an online k -uniform hypergraph instance. We denote by $x \in [0, 1]^H$ the fractional solution constructed by \mathcal{A} on the instance \mathcal{H} . The rounding algorithm is now quite simple and is similar to the methods used in [35–37].

Fix some small $0 < \epsilon < 1$ and initialize two empty sets of hyperedges $S, \mathcal{M} \leftarrow \emptyset$. Upon the arrival of an online vertex $w \in W$ with $\delta(w) \subseteq H$ and $x_h \in [0, 1]$ for every $h \in \delta(w)$, the rounding algorithm is as follows:

- For all $h \in \delta(w)$, independently add h to S with probability $x'_h := (1 - \epsilon)x_h$.
- If h was added to S , add it to \mathcal{M} as long as it does not violate the degree constraints.

The solution outputted is $\mathcal{M} \subseteq H$. Let us denote this rounding algorithm by $R(\mathcal{A}, \epsilon)$.

Lemma E.1 *Let \mathcal{A} be a ρ -competitive fractional algorithm. The randomized integral algorithm $R(\mathcal{A}, \epsilon)$ achieves a competitive ratio of at least $(1 - \epsilon)(1 - k \exp(-\epsilon^2 b/3)) \cdot \rho$.*

Proof Consider an arbitrary node $v \in V \cup W$. To bound the probability that v is matched to more than b hyperedges in S , we use a Chernoff bound [38, Theorem 1.10.1]. Fix a node v and a hyperedge h , and let $X_{v,h} = \sum_{h' \in \delta(v) \setminus \{h\}} \mathbf{1}_{\{h' \in S\}}$. Note that $\mu := \mathbb{E}[X_{v,h}] \leq (1 - \epsilon)b$. If $\mu = 0$, it is clear that $x_{v,h} \leq b$, so assume $\mu > 0$. We now have:

$$\begin{aligned} \Pr[X_{v,h} \geq b] &\leq \exp\left(-\min\left(\left(\frac{b - \mu}{\mu}\right)^2, \frac{b - \mu}{\mu}\right)\mu/3\right) \\ &\leq \exp\left(-\min\left(\frac{(b - \mu)^2}{\mu}, b - \mu\right)/3\right) \\ &\leq \exp\left(-\min(\epsilon^2 b, \epsilon b)/3\right) \leq \exp(-\epsilon^2 b/3), \end{aligned}$$

where the second inequality follows from $b - \mu \geq \epsilon b$ and the last inequality from $b/\mu \geq 1$. We now upper bound the probability that a hyperedge h cannot be included in \mathcal{M} because of the degree constraints:

$$\Pr[h \in S \setminus \mathcal{M} \mid h \in S] \leq \sum_{v \in h} \Pr[X_{v,h} \geq b] \leq k \exp(-\epsilon^2 b/3).$$

Hence, we have:

$$\begin{aligned} \mathbb{E}[|\mathcal{M}|] &= \sum_{h \in H} \Pr[h \in \mathcal{M}] = \sum_{h \in H} \Pr[h \in S] (1 - \Pr[h \in S \setminus \mathcal{M} \mid h \in S]) \\ &\geq \sum_{h \in H} x'_h (1 - k \exp(-\epsilon^2 b/3)) \\ &= (1 - k \exp(-\epsilon^2 b/3)) (1 - \epsilon) \sum_{h \in H} x_h \\ &\geq (1 - k \exp(-\epsilon^2 b/3)) (1 - \epsilon) \rho \text{OPT}_{\text{LP}}. \end{aligned}$$

□

If $b = C \cdot \log(k)$ for some $C > 6$, then by choosing $\epsilon = \sqrt{6/C}$ we get that the competitive ratio is at least $(1 - \sqrt{6/C})(1 - \frac{1}{k})\rho$. By using the $\Omega(1/\log k)$ -competitive fractional algorithm from [5], this gives an $\Omega(1/\log k)$ -competitive integral algorithm for this setting.

References

1. Borst, S., Kashaev, D., Koh, Z.K.: Online Matching on 3-Uniform Hypergraphs. In: Integer Programming and Combinatorial Optimization 2025, pp. 100–113. Springer, Berlin, Heidelberg (2025)
2. 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)
3. Kalyanasundaram, B., Pruhs, K.R.: An optimal deterministic algorithm for online b-matching. *Theoret. Comput. Sci.* **233**(1–2), 319–325 (2000)
4. Tröbst, T., Udwan, R.: Almost tight bounds for online hypergraph matching. *Oper. Res. Lett.* **55**, 107143 (2024)
5. Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing. *Math. Oper. Res.* **34**(2), 270–286 (2009)
6. Pavone, M., Saberi, A., Schiffer, M., Tsao, M.W.: Technical note - online hypergraph matching with delays. *Oper. Res.* **70**(4), 2194–2212 (2022)
7. Ma, Y., Rusmevichientong, P., Sumida, M., Topaloglu, H.: An approximation algorithm for network revenue management under nonstationary arrivals. *Oper. Res.* **68**(3), 834–855 (2020)
8. Gamlath, B., Kapralov, M., Maggiori, A., Svensson, O., Wajc, D.: Online matching with general arrivals. In: 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS), pp. 26–37 (2019). IEEE
9. Mehta, A.: Online matching and ad allocation. *Found. Trends Theor. Comput. Sci.* **8**(4), 265–368 (2013)
10. Birnbaum, B., Mathieu, C.: On-line bipartite matching made simple. *ACM SIGACT News* **39**(1), 80–87 (2008)
11. 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 (2013). SIAM
12. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. *SODA* **8**, 982–991 (2008)
13. Eden, A., Feldman, M., Fiat, A., Segal, K.: An economics-based analysis of ranking for online bipartite matching. In: Symposium on Simplicity in Algorithms (SOSA), pp. 107–110 (2021). SIAM
14. Buchbinder, N., Jain, K., Naor, J.: Online primal-dual algorithms for maximizing ad-auctions revenue. In: European Symposium on Algorithms, pp. 253–264 (2007). Springer
15. Devanur, N.R., Jain, K.: Online matching with concave returns. In: Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing, pp. 137–144 (2012)
16. Huang, Z., Zhang, Q., Zhang, Y.: Adwords in a panorama. In: 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS), pp. 1416–1426 (2020). IEEE
17. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. *Journal of the ACM (JACM)* **54**(5), 22 (2007)
18. Aggarwal, G., Goel, G., Karande, C., Mehta, A.: Online vertex-weighted bipartite matching and single-bid budgeted allocations. In: Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1253–1264 (2011). SIAM
19. Huang, Z., Tang, Z.G., Wu, X., Zhang, Y.: Online vertex-weighted bipartite matching: Beating $1-1/e$ with random arrivals. *ACM Transactions on Algorithms (TALG)* **15**(3), 1–15 (2019)
20. Fahrback, M., Huang, Z., Tao, R., Zadimoghaddam, M.: Edge-weighted online bipartite matching. *J. ACM* **69**(6), 1–35 (2022)
21. Karande, C., Mehta, A., Tripathi, P.: Online bipartite matching with unknown distributions. In: Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing, pp. 587–596 (2011)

22. Mahdian, M., Yan, Q.: Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In: Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing, pp. 597–606 (2011)
23. 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 (2013). Springer
24. Wang, Y., Wong, S.C.: Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In: Automata, Languages, and Programming - 42nd International Colloquium (ICALP), Proceedings, Part I, pp. 1070–1081. Springer, Berlin, Heidelberg (2015)
25. Buchbinder, N., Segev, D., Tkach, Y.: Online algorithms for maximum cardinality matching with edge arrivals. *Algorithmica* **81**(5), 1781–1799 (2019)
26. Huang, Z., Kang, N., Tang, Z.G., Wu, X., Zhang, Y., Zhu, X.: Fully online matching. *J. ACM* **67**(3), 17–11725 (2020)
27. Huang, Z., Tang, Z.G., Wu, X., Zhang, Y.: Fully online matching II: beating ranking and water-filling. In: 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS), pp. 1380–1391 (2020)
28. Ashlagi, I., Burq, M., Dutta, C., Jaillet, P., Saberi, A., Sholley, C.: Edge-weighted online windowed matching. *Math. Oper. Res.* **48**(2), 999–1016 (2023)
29. Korula, N., Pál, M.: Algorithms for secretary problems on graphs and hypergraphs. In: Automata, Languages and Programming, 36th International Colloquium (ICALP), Proceedings, Part II, pp. 508–520. Springer, Berlin, Heidelberg (2009)
30. Marinkovic, J., Soto, J.A., Verdugo, V.: Online combinatorial assignment in independence systems. In: Integer Programming and Combinatorial Optimization - 25th International Conference (IPCO), pp. 294–308. Springer, Berlin, Heidelberg (2024)
31. Lee, E., Svensson, O., Thiery, T.: Asymptotically optimal hardness for k-set packing and k-matroid intersection. In: Proceedings of the 57th Annual ACM Symposium on Theory of Computing, pp. 54–61 (2025)
32. Chan, Y.H., Lau, L.C.: On linear and semidefinite programming relaxations for hypergraph matching. *Math. Program.* **135**(1–2), 123–148 (2012)
33. Cygan, M.: Improved approximation for 3-dimensional matching via bounded pathwidth local search. In: 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 509–518. IEEE Computer Society, Los Alamitos, CA, USA (2013)
34. Yao, A.C.-C.: Probabilistic computations: Toward a unified measure of complexity. In: 18th Annual Symposium on Foundations of Computer Science (sfcs 1977), pp. 222–227 (1977). IEEE Computer Society
35. El Ouali, M., Jäger, G.: The b-matching problem in hypergraphs: Hardness and approximability. In: International Conference on Combinatorial Optimization and Applications, pp. 200–211 (2012). Springer
36. Raghavan, P., Tompson, C.D.: Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* **7**(4), 365–374 (1987)
37. Srivastav, A., Stangier, P.: Weighted fractional and integral k-matching in hypergraphs. *Discret. Appl. Math.* **57**(2–3), 255–269 (1995)
38. Doerr, B.: Probabilistic tools for the analysis of randomized optimization heuristics. In: Doerr, B., Neumann, F. (eds.) *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pp. 1–87 (2019)