Online Matching on 3-Uniform Hypergraphs

Sander Borst[1(✉)], Danish Kashaev[2], and Zhuan Khye Koh[3]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
sborst@mpi-inf.mpg.de
[2] Centrum Wiskunde and Informatica (CWI), Amsterdam, The Netherlands
danish.kashaev@cwi.nl
[3] Boston University, Boston, USA
zkkoh@bu.edu

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.

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 [22], 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

The full version is available at arXiv:2402.13227. 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.

N. Megow and A. Basu (Eds.): IPCO 2025, LNCS 15620, pp. 100–113, 2025.
https://doi.org/10.1007/978-3-031-93112-3_8

cardinality of the resulting matching. Karp et al. [22] 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 [20], 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. [22] 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 $1 - 1/e$.

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 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 [30], $k - 1$ represents the capacity of service vehicles, which is often small. Another example is network revenue management problems [25]. 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

[1] In [31], it is shown that no algorithm can be $(2 + f(k))/k$-competitive for some positive function f with $f(k) = o(1)$. In the full version, we give a simple construction showing that no integral algorithm can be $2/k$-competitive.

et al. [25] 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.

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* [13,22].

We start with the instance in [13], 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 [22], 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 [13].

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.

Related Work. Since the online matching problem was introduced in [22], it has garnered a lot of interest, leading to extensive follow-up work. We refer the reader to the excellent survey by Mehta [28] for navigating this rich literature. The original analysis of RANKING [22] was simplified in a series of papers [3, 10, 11, 14]. Many variants of the problem have been studied, such as the online b-matching problem [20], and its extension to the AdWords problem [4, 9, 19, 29]. Weighted generalizations have been considered, e.g., vertex weights [1, 17] and edge weights [12]. Weakening the adversary by requiring that online nodes arrive in a random order has also been of interest [21, 23, 26]. Another line of research explored more general arrival models such as two-sided vertex arrival [32], general vertex arrival [13], edge arrival [6, 13], and general vertex arrival with departure times [2, 16, 18].

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 [24] 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. [25] gave a $1/k$-competitive algorithm under 'nonstationary' arrivals. Pavone et al. [30] 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, [27] gave a $O(\log(k)/k)$ upper bound on the competitive ratio. We refer to [27] 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 NP-hard to approximate within a factor of $\Omega(\log(k)/k)$ [15]. Moreover, the factor between the optimal solution and the optimal value of the natural LP relaxation is at least $1/(k-1+1/k)$ [7].

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 [7]. For $k = 3$, the best known polynomial time approximation algorithm gives a $(3/4 - \varepsilon)$-approximation [8].

Paper Organization. In Sect. 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 and other missing proofs are in the full version.

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:

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

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

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 $W = (w_1, w_2, \ldots)$ 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 $\mathcal{H}$, $\mathcal{V}(\mathcal{A}, \mathcal{H}) \geq \rho\, \mathsf{OPT}(\mathcal{H})$. Similarly, a fractional algorithm is *ρ-competitive* if for any instance $\mathcal{H}$, $\mathcal{V}(\mathcal{A}, \mathcal{H}) \geq \rho\, \mathsf{OPT}_{\mathsf{LP}}(\mathcal{H})$.

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

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

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 distribution function $f : [0,1] \to [0,1]$:

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

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 [20]. For this reason, we define the *priority* of a hyperedge $h = \{u,v,w\}$ as:

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

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.*

The proof is based on a primal-dual analysis. By construction, the primal and the dual objective values are equal at any point during the execution. In addition, $\sum_{v\in h} y_v \geq (e-1)/(e+1)$ holds for every hyperedge h at the end of the execution, meaning that the vector $(e+1)/(e-1)\, y$ is a feasible dual solution. Details are given in the full version.

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. [22] and the edge-arrival instance of Gamlath et al. [13] for bipartite graphs.

We start by giving a high-level overview of the construction. 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 $|U_i| = |V_i| = T$.

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 [13], together with the function $f(x) = e^x/(e+1)$ defined in (2). The exact construction is described in Sect. 4.3.
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 [22]. The exact construction is described in Sect. 4.4.

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

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

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.

The two assumptions that we make are the following. In the full version, 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 $t \in \{1, \ldots, T\}$, the tth 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., $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}, \ldots, \mathcal{M}_i^{(T)}$, 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 $\mathcal{M}_i^{(1)}, \ldots, \mathcal{M}_i^{(T)}$ are adaptive to the behaviour of the algorithm in every phase. It is essentially the instance of [13] 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 $U_i = \{1, \ldots, T\}$ and $V_i = \{1, \ldots, T\}$. 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 $(u, v) \in \mathcal{M}_i^{(t)}$ 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 $\sigma_t(1) < \sigma_t(2) < \ldots < \sigma_t(r_t)$ 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 $\sigma_t(1) < \sigma_t(2) < \ldots < \sigma_t(r_t)$.

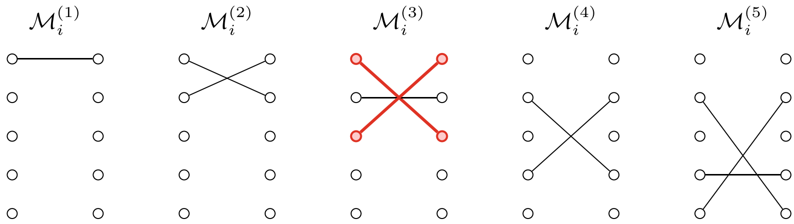

Fig. 1. In this example, the algorithm does not increase the edge $(1, 3)$, and thus by symmetry the edge $(3, 1)$, up to the threshold during phase $t = 3$. Hence, $f(\ell_1^{(3)}) + f(\ell_3^{(3)}) < 1$ and nodes 1 and 3 become inactive from that point on. The maximum matching at the end of phase 5 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.

– The matching at phase $t + 1$ is then of size $r_t + 1$ and is defined as:

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

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 1 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 $k \in \{1, \ldots, \lceil q_t \rceil\}$ 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 $i \in \{1, \ldots, \lceil q_t \rceil\}$. For this reason, we define:

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

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 relate it to the CDF of the binomial distribution $B(t, 1/2)$. We show that

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

Since the left-hand side is the residual capacity of the vertex cover of $\mathcal{M}_i^{(T)}$, this yields an upper bound on the value obtained by the algorithm in component C_i during the last phase.

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

4.4 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 [22] 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 [22] 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 Fig. 2.

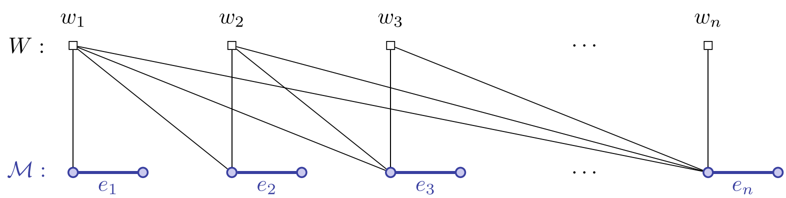

Fig. 2. An illustration of the instance constructed in Lemma 1

We now state the guarantee obtained by this construction, parametrized by the maximum (fractional) degree $\Delta \in [0, 1]$ attained by an offline node.

Lemma 1. *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 $\mathsf{OPT}(\mathcal{H}) = |\mathcal{M}|$. Moreover, for any fractional algorithm $\mathcal{A}$ whose returned solution x satisfies $x(\delta(v)) \leq \Delta$ 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 1 on each submatching separately. More precisely, let us fix $N := \lceil 2|\mathcal{M}^{(t)}|^{1/3}\rceil$. 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 Fig. 2 to each submatching $\mathcal{M}^{(t)}(i,j)$. This finishes the description of our instance.

Recall from Sect. 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 2. *If $\mathcal{A}$ is threshold-respecting on $W^{(t)}$, then*

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

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 2.

Lemma 3. *If $\mathcal{A}$ is threshold-respecting on $W^{(t)}$, then*

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

Proof. By splitting the matching $\mathcal{M}^{(t)}$ based on the m components, we can rewrite the bound in Lemma 2 as

$$\mathcal{V}^{(t)}(\mathcal{A}) \le \sum_{i=1}^{m} \sum_{(u,v)\in\mathcal{M}_i^{(t)}} \left(1 - f(\ell_u^{(t-1)}) - f(\ell_v^{(t-1)})\right) + 15\,|\mathcal{M}^{(t)}|^{2/3}. \tag{4}$$

Fix a component $i \in [m]$, and let $\mathcal{E}_i := \left\{e \in \mathcal{M}_i^{(t-1)} \mid f(\ell_u^{(t-1)}) + f(\ell_v^{(t-1)}) \ge 1\right\}$ 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 Sect. 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 (4) as:

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

where the inequality follows from definition of $\mathcal{E}_i$. Plugging this into (4) with the bound $|\mathcal{M}^{(t)}| \le tm$ (which is immediate by the construction in Sect. 4.3) yields the desired result. □

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

Corollary 1. *If $\mathcal{A}$ is ε-threshold-respecting on $W^{(t)}$ for some $\varepsilon \geq 0$, then*

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

Proof. Fix an edge $e \in \mathcal{M}^{(t)}$. Let $h_1, h_2, \ldots, h_k$ 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, $1^\top z \leq \varepsilon tm$ 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 3 to obtain the desired upper bound on $1^\top(x - z)$. □

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

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 1). 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 Fig. 1 for an example). □

By Lemma 4, 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 $m \to \infty$, picking $T = o(\sqrt{m})$ such that $T \to \infty$ 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.

References

1. 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. SIAM (2011)
2. 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)
3. Birnbaum, B., Mathieu, C.: On-line bipartite matching made simple. ACM SIGACT News **39**(1), 80–87 (2008)
4. Buchbinder, N., Jain, K., Naor, J.S.: Online primal-dual algorithms for maximizing ad-auctions revenue. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 253–264. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75520-3_24
5. Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing. Math. Oper. Res. **34**(2), 270–286 (2009)
6. Buchbinder, N., Segev, D., Tkach, Y.: Online algorithms for maximum cardinality matching with edge arrivals. Algorithmica **81**(5), 1781–1799 (2019)
7. Chan, Y.H., Lau, L.C.: On linear and semidefinite programming relaxations for hypergraph matching. Math. Program. **135**(1–2), 123–148 (2012)
8. 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 (2013)
9. 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)
10. Devanur, N.R., Jain, K., Kleinberg, R.D.: Randomized primal-dual analysis of ranking for online bipartite matching. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 101–107. SIAM (2013)
11. 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. SIAM (2021)
12. Fahrbach, M., Huang, Z., Tao, R., Zadimoghaddam, M.: Edge-weighted online bipartite matching. J. ACM **69**(6), 1–35 (2022)
13. 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. IEEE (2019)
14. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: SODA, vol. 8, pp. 982–991 (2008)
15. Hazan, E., Safra, S., Schwartz, O.: On the complexity of approximating k-set packing. Comput. Complex. **15**, 20–39 (2006)
16. Huang, Z., et al.: Fully online matching. J. ACM **67**(3):17:1–17:25 (2020)
17. Huang, Z., Tang, Z.G., Wu, X., Zhang, Y.: Online vertex-weighted bipartite matching: beating 1–1/e with random arrivals. ACM Trans. Algorithms (TALG) **15**(3), 1–15 (2019)
18. 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. IEEE (2020)
19. 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. IEEE (2020)

20. Kalyanasundaram, B., Pruhs, K.R.: An optimal deterministic algorithm for online b-matching. Theor. Comput. Sci. **233**(1–2), 319–325 (2000)
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. 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)
23. Kesselheim, T., Radke, K., Tönnis, A., Vöcking, B.: An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In: Bodlaender, H.L., Italiano, G.F. (eds.) ESA 2013. LNCS, vol. 8125, pp. 589–600. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40450-4_50
24. Korula, N., Pál, M.: Algorithms for secretary problems on graphs and hypergraphs. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5556, pp. 508–520. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02930-1_42
25. 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)
26. 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)
27. Marinkovic, J., Soto, J.A., Verdugo, V.: Online combinatorial assignment in independence systems. In: Integer Programming and Combinatorial Optimization - 25th International Conference (IPCO). LNCS, vol. 14679, pp. 294–308. Springer, Cham (2024)
28. Mehta, A.: Online matching and ad allocation. Found. Trends Theor. Comput. Sci. **8**(4), 265–368 (2013)
29. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. J. ACM (JACM) **54**(5), 22–es (2007)
30. Pavone, M., Saberi, A., Schiffer, M., Tsao, M.W.: Technical note - online hypergraph matching with delays. Oper. Res. **70**(4), 2194–2212 (2022)
31. Tröbst, T., Udwani, R.: Almost tight bounds for online hypergraph matching. Oper. Res. Lett. **55**, 107143 (2024)
32. Wang, Y., Wong, S.C.: Two-sided online bipartite matching and vertex cover: beating the greedy algorithm. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015. LNCS, vol. 9134, pp. 1070–1081. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47672-7_87