# An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions

Thomas Kesselheim[1,⋆], Klaus Radke[2,⋆⋆],
Andreas Tönnis[2,⋆⋆⋆], and Berthold Vöcking[2]

[1] Department of Computer Science, Cornell University, Ithaca, NY, USA
kesselheim@cs.cornell.edu

[2] Department of Computer Science, RWTH Aachen University, Germany
{radke,toennis,voecking}@cs.rwth-aachen.de

**Abstract.** We study online variants of weighted bipartite matching on graphs and hypergraphs. In our model for online matching, the vertices on the right-hand side of a bipartite graph are given in advance and the vertices on the left-hand side arrive online in random order. Whenever a vertex arrives, its adjacent edges with the corresponding weights are revealed and the online algorithm has to decide which of these edges should be included in the matching. The studied matching problems have applications, e.g., in online ad auctions and combinatorial auctions where the right-hand side vertices correspond to items and the left-hand side vertices to bidders.

Our main contribution is an optimal algorithm for the weighted matching problem on bipartite graphs. The algorithm is a natural generalization of the classical algorithm for the secretary problem achieving a competitive ratio of $\mathrm{e} \approx 2.72$ which matches the well-known upper and lower bound for the secretary problem. This shows that the classic algorithmic approach for the secretary problem can be extended from the simple selection of a best possible singleton to a rich combinatorial optimization problem.

On hypergraphs with $(d+1)$-uniform hyperedges, corresponding to combinatorial auctions with bundles of size $d$, we achieve competitive ratio $O(d)$ in comparison to the previously known ratios $O\left(d^2\right)$ and $O(d \log m)$, where $m$ is the number of items. Additionally, we study variations of the hypergraph matching problem representing combinatorial auctions for items with bounded multiplicities or for bidders with submodular valuation functions. In particular for the case of submodular valuation functions we improve the competitive ratio from $O(\log m)$ to e.

⋆ Supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD).

⋆⋆ Supported by the Studienstiftung des deutschen Volkes.

⋆⋆⋆ Supported by the DFG GRK/1298 “AlgoSyn”.

H.L. Bodlaender and G.F. Italiano (Eds.): ESA 2013, LNCS 8125, pp. 589–600, 2013.

## 1 Introduction

Consider the following natural generalization of the classical secretary problem: Suppose an administrator wants to hire people for a set of open positions (rather than only one secretary for a single position). The applicants are interviewed one at a time and in every interview the interviewer learns weights representing the degree of qualification of the current candidate for each of the possible positions. Now, immediately after an interview, the administrator has to either assign the applicant to one of the open positions or the candidate will leave the room and take a job at another company. The administrator is interested in maximizing the sum of the assigned weights.

The described problem corresponds to the weighted online bipartite matching problem. In general, the jobs correspond to the offline vertices on the right-hand side which are known in advance. The vertices on the left-hand side arrive online one by one, each with its incident edges and their respective weights. The decision whether and how to assign the current vertex has to be made online. Unfortunately, for general weights and when the vertices arrive in adversarial order, every algorithm can perform arbitrarily bad. To achieve any reasonable competitive ratio, it is necessary to make additional assumptions on the model. In this work, we assume that the vertices arrive in random order, analogously to the original *secretary problem*.

The weighted online matching problem in the secretary model was introduced by Korula and Pál [19]. It is a generalization of the matroid secretary problem on transversal matroids which was introduced by Babaioff et al. [3] and later improved by Dimitrov and Plaxton [9]. This, respectively, is a generalization of the classical secretary problem. Here, we present the first optimal algorithm for weighted online matching which also matches the lower bound for the secretary problem.

The online matching problem is closely related to combinatorial auctions. Let, e.g., the right-hand side of the graph represent the items and the left-hand side correspond to the bidders. Then the weighted online bipartite matching corresponds to an online combinatorial auction where every bidder can buy at most one item. Now, we extend the graph towards $(d+1)$-uniform hyperedges so that every edge contains exactly one bidder and $d$ items. Thus, every hyperedge represents a bid on a bundle of items in a combinatorial auction. This setting was first analyzed by Korula and Pál [19] on whose results we improve. Additionally, we allow for multiplicities on the items, the right-hand side vertices of the hypergraph, which has applications in ad auctions. Furthermore, we consider submodular[1] weight functions which is a reasonable assumption for these economically motivated problems. Like Korula and Pál, we analyze our algorithms in terms of competitive ratio, i.e. the ratio of the value of an optimal offline solution to the expected weight achieved by the online algorithm.

[1] A set function $f\colon 2^{\Omega} \to \mathbb{R}$ is submodular if for every $X, Y \subseteq \Omega$ we have that $f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y)$.

**Our Contribution:** We provide algorithms for several variants of weighted bipartite matching in the secretary model, i.e. with random arrival order. All our algorithms are generalizations of the classical approach to the secretary problem. First, they gather information on the instance via sampling. Then, in every later step, they solve the known part of the instance optimally and treat the left-hand side vertex that has just arrived according to that locally optimal solution. The most important feature of our analysis is to interpret the random arrival order as a sequence of stochastically independent experiments.

For online bipartite matching we obtain an e-competitive algorithm. This improves on the 8-competitive algorithm by Korula and Pál [19] and matches the lower bound on the classical secretary problem, see e.g. Buchbinder et al. [5]. While Korula and Pál follow a similar approach, their analysis requires them to use a greedy approximation algorithm for the online decision making instead of locally optimal solutions.

When we apply the algorithmic approach to online bipartite hypermatching, we use randomized rounding on a fractional LP solution. Therefore, the obtained competitive ratios are with respect to the fractional offline optimum. When the online bidders are interested in sets of size at most $d$ and every item is available at least $b$ times, we obtain an expected competitive ratio of $O\left(d^{1/b}\right)$. Thus, for classical combinatorial auctions this translates to a $O(d)$-approximation in contrast to the previously known $O\left(d^2\right)$ by Korula and Pál. For multiplicities $b \geq \log(d)$, a common assumption in ad auctions, the competitive ratio becomes a constant $O(1)$. For general valuations on sets of unbounded size, our randomized algorithm is $O(m^{1/(b+1)})$-competitive, where $m$ is the number of items. Furthermore, if valuation functions are submodular, the competitive ratio of our algorithm is again e even for multiplicity one and thus optimal.

All these results are based on a random order of arrivals. Using this assumption, we beat the lower bound of $\Omega\left(b \cdot d^{1/b}\right)$ for any deterministic set packing algorithm in the online adversary model by Azar and Regev [2]. We show that for $b = 1$, every randomized online algorithm in the secretary model, even with unlimited computational power, is $\Omega\left({}^{\ln(d)}/{\ln\ln(d)}\right)$-competitive.

**Related Work:** When analyzing online bipartite matching, it is necessary to make additional assumptions on the model as no algorithm can handle adversarial arrival with general edge weights; see Aggarwal et al. [1] for a proof. A common choice is to assume a random order of the vertices on the left-hand side. Another option is to admit arbitrary order but to make restrictions on the edge weights. Some recent, loosely related papers adopt slight changes to the model and assume budgeted allocations with stochastic arrivals, see e.g. [7,12,22].

The random order model has its origins in the famous secretary problem, where $n$ candidates for a job arrive online in random order and the goal is to pick the best one with maximal probability. This is identical to edge-weighted bipartite matching with only one vertex on the right-hand side. Although the problem was folklore, it was not published until 1960 by Gardner and it was solved many times. See Ferguson [11] for historical details. The optimal algorithm for the secretary problem is e-competitive in expectation.

A generalization of the classical secretary problem is the matroid secretary problem, introduced by Babaioff et al. [3]. Here, the elements of a matroid arrive online in random order and the objective is to select an independent set of maximum weight. For general matroids they gave an $O\big(\log(\rho)\big)$-competitive algorithm, where $\rho$ is the rank of the matroid. This result was later improved to $O\big(\sqrt{\log(\rho)}\big)$ by Chakraborty and Lachish [6]. Various results are known for special kinds of matroids, see [3,13,14,18,19]. Note that transversal matroids are a special case of bipartite matching, where all edges incident to the same left-hand side vertex have identical weight. Babaioff et al. [3] presented a $4d$-competitive algorithm for the case of transversal matroids with bounded left degree $d$. This was improved by Dimitrov and Plaxton [9] who gave a 16-competitive algorithm for transversal matroids. The first result on general bipartite matching in the secretary model is by Korula and Pál [19] who presented an 8-competitive online algorithm.

The research on online bipartite matching with adversarial arrival was initiated by Karp et al. [17] who analyzed the unweighted case. They presented a randomized algorithm obtaining an expected competitive ratio of $^{\mathrm{e}}/_{(\mathrm{e}-1)}$ and a matching lower bound. The proof was later simplified by Goel and Mehta [12] and Birnbaum and Mathieu [4]. A primal-dual analysis was given by Devanur et al. [8]. Karande et al. [16] and independently Mahdian and Yan [21] showed that the lower bound of $^{\mathrm{e}}/_{(\mathrm{e}-1)}$ does not hold when the left-hand side vertices arrive in random, instead of arbitrary, order. Aggarwal et al. [1] were the first to analyze online matching with adversarial order in a general weighted setting. They obtained an expected competitive ratio of $^{\mathrm{e}}/_{(\mathrm{e}-1)}$ as long as all edges incident to the same vertex on the right-hand side have identical weight. Kalyanasundaram and Pruhs [15] presented a deterministic 3-competitive algorithm when the edge weights represent a metric space.

Bipartite hypermatching in the secretary model was first analyzed by Korula and Pál [19]. They obtained an expected competitive ratio of $O\big(d^2\big)$ when the hyperedges have bounded size $d+1$. Krysta and Vöcking [20] investigated online combinatorial auctions with random arrival of the bidders and developed randomized mechanisms that are incentive compatible. For valuations on sets of bounded size $d$ and when each of the $m$ items is available $b$ times, they showed an expected competitive ratio of $O\big(d^{1/b}\log(bm)\big)$. In the case of general valuations, they obtained an expected competitive ratio of $O\big(m^{1/(b+1)}\log(bm)\big)$. When the valuation functions are XOS and $b=1$ the achieved competitive ratio is $O\big(\log(m)\big)$. Feldmann et al. [10] provide constant competitive algorithms for different variants of the secretary problem using submodular weight functions. E.g., they consider a submodular secretary problem on partition matroids.

## 2 Edge-Weighted Bipartite Online Matching

In the *bipartite online matching* problem, we are initially given the set $R$ of an edge-weighted bipartite graph $G=(L\cup R,E)$ and the cardinality $n:=|L|$ of the set $L$. At every step, a new vertex $v\in L$ arrives together with the weights

$w(e) \in \mathbb{R}_{\geq 0}$ of its incident edges. Most importantly, the vertices in $L$ are revealed online and in random order. The algorithm always has to either assign the current vertex to one of its unmatched neighbors in $R$, or decide to leave it unassigned.

Our algorithm is a generalization of the classical approach to the secretary problem. There, a constant fraction of the candidates is ignored. Then, when an online candidate arrives that is better than all previous ones, it is selected. We also start by sampling a constant fraction of the vertices on the left-hand side. Afterwards, whenever a new vertex is presented to the algorithm, we compute an optimum solution on the revealed part of the graph. If, in this local solution, the current vertex on the left-hand side is assigned to an unmatched vertex, we add this edge to our matching.

**Algorithm 1.** Bipartite online matching

```
Input  : vertex set $R$ and cardinality $n = |L|$
Output: matching $M$

Let $L'$ be the first $\lfloor n/\mathrm{e} \rfloor$ vertices of $L$;
$M := \emptyset$;
for each subsequent vertex $\ell \in L - L'$ do                // steps $\lceil n/\mathrm{e} \rceil$ to $n$
    $L' := L' \cup \ell$;
    $M^{(\ell)}$ := optimal matching on $G[L' \cup R]$;     // e.g. by Hungarian method
    Let $e^{(\ell)} := (\ell, r)$ be the edge assigned to $\ell$ in $M^{(\ell)}$;
    if $M \cup e^{(\ell)}$ is a matching then
        add $e^{(\ell)}$ to $M$;
```

For convenience of notation, we will number the vertices in $L$ from 1 to $n$ in the (random) order they are presented to the algorithm. Hence, we will use the variable $\ell$ synonymously as an integer, the name of an iteration and the name of the current vertex.

**Lemma 1.** *Let the random variable $A_v$ denote the contribution of the vertex $v \in L$ to the output, i.e. the weight of the edge $(v, r)$ assigned to $v$ in $M$. And let $OPT$ be the value of a maximum-weight matching in the full graph $G$. For the vertices $\ell \in \{\lceil n/\mathrm{e} \rceil, \dots, n\}$ we have,*

$$\mathbf{E}[A_\ell] \geq \frac{\lfloor n/\mathrm{e} \rfloor}{\ell - 1} \cdot \frac{OPT}{n} .$$

*Proof.* First, we will show that the expected weight of $e^{(\ell)}$, i.e. of the edge assigned to vertex $\ell$ in the matching $M^{(\ell)}$, is a significant fraction of $OPT$. Then, we will analyze the probability of adding this edge to the matching $M$.

The proof relies on the fact that in any step $k$ of the algorithm the choice of the random permutation up to this point can be modeled as a sequence of the following *independent* random experiments: First choose a set of size $k$ from $L$. Then determine the order of these $k$ vertices by iteratively selecting a vertex at random and removing it. We need this interpretation to exploit the randomness in each of these experiments separately.

Now in step $\ell$ we have $|L'| = \ell$ and the algorithm calculates an optimal matching $M^{(\ell)}$ on $G[L' \cup R]$. As explained, the current vertex $\ell$ can be seen as being selected uniformly at random from the set $L'$. Hence, the expected weight of the edge $e^{(\ell)}$ in $M^{(\ell)}$ is $w(M^{(\ell)})/\ell$. Also, since $L'$ can be seen as being uniformly selected from $L$ with size $\ell$ we know $\mathbf{E}\left[w\left(M^{(\ell)}\right)\right] \geq \ell/n \cdot OPT$. Together we have,

$$\mathbf{E}\left[w(e^{(\ell)})\right] \geq \frac{OPT}{n} \; . \tag{1}$$

Note that the above expectation is only over the random choice of the set $L'$ and the choice of the element to be last in their order. The rest of the proof will exploit the randomness in the order of the remaining $\ell - 1$ vertices in $L'$.

The edge $e^{(\ell)} = (\ell, r)$ can only be added to the matching $M$ if $r$ has not already been matched in an earlier step. Consider the vertex $r$. In any of the preceding steps $k \in \{\lceil n/\mathrm{e} \rceil, \ldots, \ell - 1\}$ the vertex $r$ was only matched if it was in $e^{(k)}$, i.e. if in $M^{(k)}$ the vertex $r$ was assigned to the left-hand side vertex $k$. Again, the last vertex in the order can be seen as being chosen uniformly at random from the $k$ participating vertices on the left-hand side. Hence, the probability of $r$ being matched in step $k$ was at most $1/k$. As before, the order of the vertices $1, \ldots, k-1$ is irrelevant for this event. Therefore, also the respective events if some vertex $k' < k$ was matched to $r$ can be regarded as independent. Following this argument inductively from $k = \ell - 1$ down to $\lceil n/\mathrm{e} \rceil$, we get,

$$\mathbf{Pr}\left[r \text{ unmatched in step } \ell\right] = \mathbf{Pr}\left[\bigwedge_{k=\lceil n/\mathrm{e} \rceil}^{\ell-1} r \notin e^{(k)}\right] \geq \prod_{k=\lceil n/\mathrm{e} \rceil}^{\ell-1} \frac{k-1}{k} = \frac{\lceil \frac{n}{\mathrm{e}} \rceil - 1}{\ell - 1} \; .$$

Thus we have $\mathbf{Pr}\left[M \cup e^{(\ell)} \text{ is a matching}\right] \geq \frac{\lfloor n/\mathrm{e} \rfloor}{\ell - 1}$. Together with inequality (1) we obtain the lemma. □

**Theorem 2.** *The online matching algorithm is* e*-competitive in expectation.*

*Proof.* The weight of the matching $M$ is obtained by summing the variables $A_\ell$. Using Lemma 1 we get,

$$\mathbf{E}\left[w(M)\right] = \mathbf{E}\left[\sum_{\ell=1}^{n} A_\ell\right] \geq \sum_{\ell=\lceil n/\mathrm{e} \rceil}^{n} \frac{\lfloor n/\mathrm{e} \rfloor}{\ell - 1} \cdot \frac{OPT}{n} = \frac{\lfloor n/\mathrm{e} \rfloor}{n} \cdot \sum_{\ell=\lfloor n/\mathrm{e} \rfloor}^{n-1} \frac{1}{\ell} \cdot OPT \; .$$

We have $\frac{\lfloor n/\mathrm{e} \rfloor}{n} \geq \frac{1}{\mathrm{e}} - \frac{1}{n}$ and $\sum_{\ell=\lfloor n/\mathrm{e} \rfloor}^{n-1} \frac{1}{\ell} \geq \ln\left(\frac{n}{\lfloor n/\mathrm{e} \rfloor}\right) \geq 1$ which gives,

$$\mathbf{E}\left[w(M)\right] \geq \frac{\lfloor n/\mathrm{e} \rfloor}{n} \cdot \sum_{\ell=\lfloor n/\mathrm{e} \rfloor}^{n-1} \frac{1}{\ell} \cdot OPT \geq \left(\frac{1}{\mathrm{e}} - \frac{1}{n}\right) \cdot OPT \; .$$

□

## 3 Packing Sets of Size at Most $d$ with Capacity $b$

A common generalization of bipartite online matching is the *bipartite online b-hypermatching* problem. Here, the underlying structure is an edge-weighted hypergraph $H = (L \cup R, E)$. We assume that the hyperedges in $E$ are of the form $e = (v, S)$, with $v \in L$, $S \subseteq R$ and $|S| \leq d$. Again, we are initially given the vertex set $R$ together with the size $n$ of the vertex set $L$ and the capacity $b$. The vertices in $L$ are presented to the algorithm online and in a random order. At each step, when a vertex $v \in L$ is revealed, the algorithm also observes all its incident hyperedges $\delta(v) := \{e \in E \mid v \in e\}$ together with their respective weights[2] $w(e) \in \mathbb{R}_{\geq 0}$. As before, the algorithm decides online whether to assign one of the edges in $\delta(v)$, or to leave $v$ unmatched. The objective is a $b$-hypermatching in $H$ of maximum weight, i.e. every vertex $r \in R$ may be contained in up to $b$ edges of the matching but every vertex in $L$ may be matched only once.

In every step $\ell$ we will solve the LP-relaxation of max-weight $b$-hypermatching on the revealed part of the graph computing a fractional solution $x^{(\ell)}$. Note that for a particular subset $L' \subseteq L$ of the left-hand side vertices the restricted hypergraph $H[L' \cup R]$ has exactly the edge set $E' := \{(v, S) \in E \mid v \in L'\}$.

In a matching, a vertex on the left-hand side is assigned to at most one hyperedge. Hence, for every vertex $v \in L'$ a feasible solution to the LP-relaxation satisfies the constraint $\sum_{e=(v,S)\in E'} x_e \leq 1$. Therefore we can interpret the restricted vector $x\big|_{\delta(v)}$ as a probability distribution over all hyperedges incident to $v$. The second LP-constraint is $\sum_{e=(v,S)\in E',\, r\in S} x_e \leq b$ for every vertex $r \in R$.

**Algorithm 2.** Bipartite online $b$-hypermatching

```
Input  : vertex set $R$, cardinality $n = |L|$ and parameter $p < 1$
Output: $b$-hypermatching $M$

Let $L'$ be the first $p \cdot n$ vertices of $L$;
$M := \emptyset$;
for each subsequent vertex $\ell \in L - L'$ do                // steps $pn + 1$ to $n$
    $L' := L' \cup \ell$;
    $x^{(\ell)}$ := optimal fractional solution of LP-relaxation on $H[L' \cup R]$;
    Choose $e^{(\ell)}$ randomly according to the distribution $x^{(\ell)}\big|_{\delta(\ell)}$;
    if $M \cup e^{(\ell)}$ is a $b$-hypermatching then
        add $e^{(\ell)}$ to $M$;
```

The parameter $p < 1$ will be set later. In line with the analysis of bipartite matching, we will number the vertices in $L$ from 1 to $n$ in their online order.

Note that the linear program and the randomized rounding in the above algorithm are only required to maintain polynomial runtime. Furthermore, all following competitive ratios are with respect to the optimal fractional solution.

[2] The weight functions are generally represented implicitly, e.g. by demand oracles, which allows to solve the LP-relaxation in polynomial time, see [23].

**Lemma 3.** *Let the random variable $A_v$ denote the contribution of the vertex $v \in L$ to the output, i.e. the weight of the edge $(v, S)$ assigned to $v$ in $M$. And let $OPT_{LP}$ be the value of a fractional offline optimum, i.e. of the LP-relaxation on the full hypergraph $H$. For the vertices $\ell \in \{pn+1, \dots, n\}$ we have,*

$$\mathbf{E}\left[A_\ell\right] \geq \left(1 - d \cdot \left(\frac{\mathrm{e}(1-p)}{p}\right)^b\right) \frac{OPT_{LP}}{n} \ .$$

*Proof.* In analogy to the proof of Lemma 1, we interpret the random permutation up to any step $k$ as multiple independent experiments: Choose $k$ vertices out of $L$, then pick one of these to be the last in the ordering and remove it. To determine the ordering of the other $k-1$ elements, iteratively select and remove the remaining vertices. Here we have to consider one additional independent random experiment due to the randomized rounding.

In step $\ell$, the algorithm calculates an optimal fractional solution $x^{(\ell)}$ to the LP-relaxation on $H[L' \cup R]$ with value $OPT_{LP^{(\ell)}}$. Since $e^{(\ell)}$ is chosen according to the restricted vector $x^{(\ell)}\big|_{\delta(\ell)}$, we have $\mathbf{E}\left[w\left(e^{(\ell)}\right)\right] = \sum_{e \in \delta(\ell)} w(e) \cdot x_e^{(\ell)}$. Exactly as in the proof of Lemma 1 one can show

$$\mathbf{E}\left[w\left(e^{(\ell)}\right)\right] \geq \frac{OPT_{LP}}{n} \ . \tag{2}$$

The expectation is taken over the choice of the set $L'$, the choice of the vertex to be last in their order and the randomized rounding.

The rounded hyperedge $e^{(\ell)} = (\ell, S)$ is only added to $M$ if every vertex in $S$ is covered by at most $b-1$ other edges in $M$.

We will first bound the probability of covering a vertex $r \in R$ in any preceding step $k \in \{pn+1, \dots, \ell-1\}$. Assume for the moment that, within step $k$, all participating left-hand side vertices did randomized rounding according to their respective restriction of $x^{(k)}$. Let us denote these tentative hyperedges by $h_1$ to $h_k$ and remember that $e^{(k)}$ corresponds to the last one. For $r \in R$ the probability of being covered in step $k$ is at most

$$\begin{aligned}\mathbf{Pr}\left[r \in e^{(k)}\right] &= \mathbf{Pr}\left[\bigvee_{v \in \{1,\dots,k\}} ((v \text{ is last in the order}) \wedge (r \in h_v))\right] \\ &\leq \sum_{v \in \{1,\dots,k\}} \mathbf{Pr}\left[(v \text{ is last in the order}) \wedge (r \in h_v)\right] \ .\end{aligned}$$

The randomized rounding is stochastically independent of the order and we know that the last vertex in the order is chosen uniformly out of $k$ vertices. Hence,

$$\mathbf{Pr}\left[r \in e^{(k)}\right] \leq \frac{1}{k} \cdot \sum_{v \in \{1,\dots,k\}} \mathbf{Pr}\left[r \in h_v\right] \ .$$

The hyperedge $h_v$ was drawn according to the distribution $x^{(k)}\big|_{\delta(v)}$. This gives $\mathbf{Pr}\left[r \in h_v\right] = \sum_{e \in \delta(v),\, r \in e} x_e^{(k)}$. Since $x^{(k)}$ is a feasible LP solution and thus satisfies $\sum_{e \in \delta(v),\, r \in e} x_e^{(k)} \leq b$ for all $r \in R$, we have,

$$\mathbf{Pr}\left[r \in e^{(k)}\right] \leq \frac{1}{k} \cdot \sum_{v \in \{1,\ldots,k\}} \sum_{\substack{e \in \delta(v), \\ r \in e}} x_e^{(k)} \leq \frac{b}{k} , \tag{3}$$

which bounds the probability of $r \in R$ being covered in step $k$.

Finally, we can bound the probability of adding $e^{(\ell)} = (\ell, S)$ to $M$. The attempt fails if any of the vertices in $S$ was already covered $b$ times in previous steps. For any vertex $r \in S$, we have by inequality (3),

$$\mathbf{Pr}\left[r \text{ is covered at least } b \text{ times}\right] \leq \sum_{\substack{C \subseteq \{pn+1,\ldots,\ell-1\}, \\ |C|=b}} \left(\prod_{k \in C} \frac{b}{k}\right) \tag{4}$$

$$\leq \binom{(1-p)n}{b} \cdot \left(\frac{b}{pn}\right)^b \leq \left(\frac{\mathrm{e}(1-p)}{p}\right)^b .$$

Using a union bound over all vertices $r \in S$, and as $|S| \leq d$, we get,

$$\mathbf{Pr}\left[M \cup e^{(\ell)} \text{ is a } b\text{-hypermatching}\right] \geq 1 - d \cdot \left(\frac{\mathrm{e}(1-p)}{p}\right)^b .$$

Together with inequality (2) we obtain the result. □

**Theorem 4.** *Set the parameter $p$ to $\frac{\mathrm{e}(2d)^{1/b}}{1+\mathrm{e}(2d)^{1/b}}$. Then the expected competitive ratio of the $b$-hypermatching algorithm for edges of size at most $d+1$ is $O\big(d^{1/b}\big)$.*

*Proof.* The weight $w(M)$ of the $b$-hypermatching is equal to $\sum_{\ell=1}^{n} A_\ell$. Using Lemma 3 we get,

$$\mathbf{E}\left[w(M)\right] \geq \sum_{\ell=pn+1}^{n} \left(1 - d \cdot \left(\frac{\mathrm{e}(1-p)}{p}\right)^b\right) \frac{OPT_{LP}}{n} .$$

The sum yields a factor of $(1-p) \cdot n$, substituting $p = \frac{\mathrm{e}(2d)^{1/b}}{1+\mathrm{e}(2d)^{1/b}}$, we obtain,

$$\mathbf{E}\left[w(M)\right] \geq \frac{OPT_{LP}}{1+\mathrm{e}(2d)^{1/b}} \cdot \left(1 - d \cdot \left(\frac{\mathrm{e}}{\mathrm{e}(2d)^{1/b}}\right)^b\right) \geq \frac{OPT_{LP}}{2+4\mathrm{e}d^{1/b}} .$$

□

A tighter analysis for the case of $b = 1$ gives a competitive ratio of e$d$.

The above result for hyperedges of bounded size can be generalized to hyperedges of unbounded size using a technique by Krysta and Vöcking [20]. Flip a fair coin to choose one out of two algorithms. In case one, apply the algorithm for hyperedges of bounded size where the instance is restricted to those edges covering at most $d = \lfloor |R|^{b/(b+1)} \rfloor$ vertices on the right-hand side. In the other case, restrict the hyperedges of every vertex on the left-hand side to the single incident hyperedge of maximum weight. Now, apply Algorithm 2 as for sets of size $d = 1$ and with only one vertex on the right-hand side which is available $b$ times. For a proof see full version.

**Corollary 5.** *For online b-hypermatching with general weight functions the described randomized algorithm has an expected competitive ratio of $O(|R|^{1/(b+1)})$.*

## 4 Lower Bound

In Section 3, we presented an $O(d)$-competitive algorithm for online hypermatching with edges of size at most $d+1$. Here, we will complement this result with a lower bound of $\Omega\left(\ln(d)/\ln\ln(d)\right)$. Note that this bound is due to the online nature of the problem and holds even when we admit unbounded computational power. The result is inspired by a lower bound from Babaioff et al. [3].

We will construct a set system with the following more easily imaginable conflict graph, i.e. where we have a vertex for every set and an edge between intersecting sets. The conflict graph is $d$-partite and all partitions are completely connected to each other. Therefore, choosing a single set (resp. vertex in the conflict graph) precludes the selection of any other set whose corresponding vertex in the conflict graph is in a different partition.

**Proposition 6.** *For every prime number $q$, there is a hypergraph $H = (V, E)$ with $|E| = |V| = q^2$ and $|e| = q$ $(\forall e \in E)$, satisfying the following properties:*

1. *the edges $E$ can be partitioned into $q$ disjoint sets $C_i$ each containing $q$ edges,*
2. *the $q$ hyperedges in each set $C_i$ are pairwise disjoint,*
3. *every edge in $C_i$ intersects all $q \cdot (q-1)$ edges that belong to any $C_j$, $j \neq i$.*

For a proof see full version. To turn the graph in Proposition 6 into a lower bound instance for $d = q$, we set $R := V$ and let every vertex in $L$ be incident to exactly one of the hyperedges. So, the edges of the graph arrive online in random order. Every hyperedge is independently assigned the weight 1 with probability $1/d$, and 0 otherwise.

**Theorem 7.** *Any online algorithm obtains a matching of expected weight less than 2. With high probability there is a matching of weight $\Omega\left(\frac{\ln(d)}{\ln\ln(d)}\right)$.*

*Proof.* When an algorithm assigns the first hyperedge $e$, say $e \in C_i$, all the edges that do not belong to $C_i$ are blocked by Property 3. The only edges disjoint to $e$ are those in $C_i$. There are at most $d-1$ such edges, each having an expected weight of $1/d$. So their accumulated expected weight is less than 1.

By Property 2 the $d$ edges of a set $C_i$ form a hypermatching. For any $i$ we have $\mathbf{Pr}\left[\text{at least } \lambda \text{ edges in } C_i \text{ have weight } 1\right] = \binom{d}{\lambda}\left(\frac{1}{d}\right)^{\lambda} \geq \left(\frac{d}{\lambda}\right)^{\lambda}\left(\frac{1}{d}\right)^{\lambda} = \lambda^{-\lambda}$. Choosing $\lambda := \ln(d)/2\ln\ln(d)$, the last term is at least $1/\sqrt{d}$. The probability that every set $C_i$ has less than $\lambda$ heavy edges is at most $(1 - 1/\sqrt{d})^d \leq e^{-d/\sqrt{d}} = e^{-\sqrt{d}}$. Hence, w.h.p. there is a matching of weight $\Omega\left(\ln(d)/\ln\ln(d)\right)$. □

## 5 Submodular Weight Functions

Let us assume that the hypergraph is complete, i.e. $H = (L \cup R, E)$ with $E = L \times 2^R$. Then we can define a weight function $w_v \colon 2^R \to \mathbb{R}_{\geq 0}$ for every vertex $v \in L$ by setting $w_v(S) := w\big((v,S)\big)$, $\forall S \subseteq R$. Now, if all these weight functions are normalized monotone submodular, then we can modify Algorithm 2 to obtain a constant competitive ratio. Note that this setting corresponds to online combinatorial auctions with submodular valuations, where the bidders arrive in random order.

Let us first analyze the case when the vertices in $R$ have multiplicity one. In every online step $\ell \in \{\lceil n/\mathrm{e} \rceil, \dots, n\}$ we solve the LP-relaxation of the revealed part of the instance and randomly round the vector to obtain $e^{(\ell)}$. Hence we have $\mathbf{E}\left[w\left(e^{(\ell)}\right)\right] \geq \frac{OPT_{LP}}{n}$. Remember that in Algorithm 2 we had to completely reject a hyperedge $e^{(\ell)} = (\ell, S)$ if any of the vertices in $S$ was already covered. Here, we can still add the hyperedge $e^{(\ell)\prime} := (\ell, S')$, where $S' \subseteq S$ are those vertices in $S$ that are not yet covered by the matching. For every $r \in R$ the probability of still being unmatched at the beginning of step $\ell$ can be analyzed exactly as in Lemma 1. Thus, we again have $\mathbf{Pr}\left[r \text{ was still unmatched in step } \ell\right] \geq \frac{\lfloor n/\mathrm{e} \rfloor}{\ell-1}$.

A known property of submodular functions is the following, see e.g. [10].

**Proposition 8.** *Given a normalized monotone submodular function $f : 2^R \to \mathbb{R}_{\geq 0}$, a set $S \subseteq R$ and a random set $S' \subseteq S$, where every element of $S$ is contained in $S'$ with probability at least $p$ (not necessarily independently). Then $\mathbf{E}\left[f(S')\right] \geq p \cdot f(S)$.*

Combining the above observations with Proposition 8 we get,

$$\mathbf{E}\left[w\left(e^{(\ell)\prime}\right)\right] \geq \frac{\lfloor n/\mathrm{e} \rfloor}{\ell - 1} \cdot \frac{OPT_{LP}}{n} \, .$$

This inequality is identical to the one in Lemma 1. By the same calculations as in Theorem 2 we obtain our result.

**Theorem 9.** *For online combinatorial auctions with submodular weight functions the algorithm is* e*-competitive.*

Note that for $b$-hypermatching, i.e. when the vertices in $R$ are available with multiplicity $b$, we can obtain the same competitive ratio. Simply replace every vertex in $R$ by $b$ copies, each with multiplicity one, and expand the valuation function in the obvious way. This equivalent instance can then be handled with the above algorithm.

## References

1. Aggarwal, G., Goel, G., Karande, C., Mehta, A.: Online vertex-weighted bipartite matching and single-bid budgeted allocations. In: SODA, pp. 1253–1264 (2011)
2. Azar, Y., Regev, O.: Combinatorial algorithms for the unsplittable flow problem. Algorithmica 44(1), 49–66 (2006)
3. Babaioff, M., Immorlica, N., Kleinberg, R.D.: Matroids, secretary problems, and online mechanisms. In: SODA, pp. 434–443 (2007)
4. Birnbaum, B.E., Mathieu, C.: On-line bipartite matching made simple. SIGACT News 39(1), 80–87 (2008)

5. Buchbinder, N., Jain, K., Singh, M.: Secretary problems via linear programming. In: Eisenbrand, F., Shepherd, F.B. (eds.) IPCO 2010. LNCS, vol. 6080, pp. 163–176. Springer, Heidelberg (2010)
6. Chakraborty, S., Lachish, O.: Improved competitive ratio for the matroid secretary problem. In: SODA, pp. 1702–1712 (2012)
7. Devanur, N.R., Hayes, T.P.: The adwords problem: online keyword matching with budgeted bidders under random permutations. In: ACM Conference on Electronic Commerce, pp. 71–78 (2009)
8. Devanur, N.R., Jain, K., Kleinberg, R.D.: Randomized primal-dual analysis of ranking for online bipartite matching. In: SODA, pp. 101–107 (2013)
9. Dimitrov, N.B., Plaxton, C.G.: Competitive weighted matching in transversal matroids. Algorithmica 62(1-2), 333–348 (2012)
10. Feldman, M., Naor, J., Schwartz, R.: Improved competitive ratios for submodular secretary problems (extended abstract). In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) RANDOM/APPROX 2011. LNCS, vol. 6845, pp. 218–229. Springer, Heidelberg (2011)
11. Ferguson, T.S.: Who solved the secretary problem? Statistical Science, 282–289 (1989)
12. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: SODA, pp. 982–991 (2008)
13. Im, S., Wang, Y.: Secretary problems: Laminar matroid and interval scheduling. In: SODA, pp. 1265–1274 (2011)
14. Jaillet, P., Soto, J.A., Zenklusen, R.: Advances on matroid secretary problems: Free order model and laminar case. CoRR abs/1207.1333 (2012)
15. Kalyanasundaram, B., Pruhs, K.: Online weighted matching. J. Algorithms 14(3), 478–488 (1993)
16. Karande, C., Mehta, A., Tripathi, P.: Online bipartite matching with unknown distributions. In: STOC, pp. 587–596 (2011)
17. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: STOC, pp. 352–358 (1990)
18. Kleinberg, R.D.: A multiple-choice secretary algorithm with applications to online auctions. In: SODA, pp. 630–631 (2005)
19. 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, Part II. LNCS, vol. 5556, pp. 508–520. Springer, Heidelberg (2009)
20. Krysta, P., Vöcking, B.: Online mechanism design (randomized rounding on the fly). In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012, Part II. LNCS, vol. 7392, pp. 636–647. Springer, Heidelberg (2012)
21. Mahdian, M., Yan, Q.: Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In: STOC, pp. 597–606 (2011)
22. Mehta, A., Saberi, A., Vazirani, U.V., Vazirani, V.V.: Adwords and generalized online matching. J. ACM 54(5) (2007)
23. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic game theory. Cambridge University Press (2007)