A Note on Online Weighted Bipartite Matchings With Groups

Alex Huang

Abstract

This is an expository note on the online edge-weighted bipartite matching problem with vertex arrivals, also known as the matching secretary problem, introduced by Korula and Pal [15]. We specifically focus on the 1/e-competitive algorithm of Kesselheim et al. [13].We make several modifications to the original analysis. First, we work in a setting where vertices are first grouped by an adversary and groups arrive in uniform random order. This answers an open problem of Korula and Pal [15] regarding constant-competitiveness in the groups setting, but we note that the problem can also be solved Kesselheim et al. [13]'s result on submodular combinatorial auctions. Second, our analysis that removes a key "independence" assumption used in Kesselheim et al. [13] (and related work on online matchings) when lower bounding the availability probability of offline vertices. Third, we also introduce a randomized threshold which allows our algorithm to extend naturally to a continuous time setting, where the number of groups is not known in advance, and the time horizon is fixed.

1 Introduction

In the *online edge-weighted bipartite matching problem with (one-sided) vertex arrivals* (abbrev. online BVM problem), we are given a bipartite graph where we know the right vertices in advance, and the left vertices arrive one-by-one. Whenever a left vertex arrives, we observe all incident edges and their corresponding weights, and must immediately and irrevocably match it to one (if any) of the unmatched right vertices. Each vertex (on both sides) can be matched to at most one other vertex. We know the number of left-vertices in advance, and most importantly, each arrival order for the vertices is equally likely. Our goal is to maximize the weight of the constructed matching. For the online BVM problem, an algorithm is said to be c-competitive, if in expectation, the matching it returns is at least c times the optimal offline matching over the entire graph.

The online BVM problem was first introduced by Korula and Pal [15], who provided a 1/8-competitive algorithm, which was improved on by Kesselheim et al. [13], who provided an optimal $1/e$-competitive algorithm. Korula and Pal also considered another interesting variant of the online BVM problem, where the left vertices are first partitioned into groups by an adversary, with both the grouping and number of groups fixed beforehand[1]. Here, we know the number of groups in advance, and the *groups* arrive one at a time with uniformly random arrival orders. For each arriving group, we observe *all* vertices within the group at once, their incident edges, and corresponding weights, and must immediately and irrevocably decide how to match (some or all of) these left vertices to available unmatched right vertices. We call this the online BVM problem with groups (online BVM-G problem).

In particular, Korula and Pal suggested that an improved understanding of groups—and their contribution to the difficulty of secretary-type problems—is likely to be of interest. They noted that few lower bounds for these problems are known beyond $1/e$ for the original secretary problem, and that obtaining such bounds may require new techniques. Finally, they posed the open problem of finding a constant-competitive algorithm for the online BVM-G problem.

[1]Fixed here means that the adversary cannot form groups in response to the algorithm's choices or the set of vertices seen thus far.

1.1 Our contributions

In this note, we solve the open problem of Korula and Pal [15] by providing a $1/e$-competitive (and thereby optimal) algorithm for the online BVM-G problem. We present two approaches to establish this result.

First, we learned via personal communication with Kesselheim that the online BVM-G problem can be reduced to an online submodular (or XOS) combinatorial auction. We include this reduction with full detail. Second, we provide a direct analysis of a natural modification of Kesselheim et al.'s [13] algorithm for the groups setting. This algorithm skips the first τ groups, where $\tau \sim \text{Binom}(q, 1/e)$ is randomly drawn, then computes a max-weight matching on the subgraph induced by all vertices observed thus far for each arriving group, accepting all edges incident to vertices in the arriving group. In [13], they skip only the first $\lfloor n/e \rfloor$ groups and achieve a competitive ratio of $1/e - 1/n$.[2]. Randomizing the number of elements skipped allows us[3] to remove the undesirable $1/q$ factor, and we prove our algorithm is $1/e$-competitive. We highlight a technical contribution within our direct analysis: an alternative proof of the lemma lower-bounding the probability that a right vertex remains available at a given iteration. Analyses of algorithms based on Kesselheim's approach—where a matching is recomputed on the current subgraph at each step to inform the selection rule—typically rely on this lemma. Prior proofs often rely, either explicitly [13, 9] or implicitly [2], on the independence between the event that a right vertex is matched in a specific iteration and the event that it remained unmatched prior to that iteration. Our proof establishes this lower bound while entirely removing this independence assumption. We believe this alternative proof may be useful in future analyses of Kesselheim et al.'s [13] algorithm under general nonuniform arrival order distributions. We later define a more 'practical' continuous time setting, where the number of groups is unknown, and the time horizon is fixed, and briefly detail how our randomized threshold allows the algorithm to extend to this case.

1.2 Related Work

Over the years, many variations of online matching problems have been explored. For adversarial arrivals and unweighted edges, Karp et al. [11] presented a randomized algorithm attaining a competitive ratio of $1 - 1/e$, with a matching lower bound. When all edges incident to the same vertex are the same weight, this is essentially the vertex-weighted case, for which Aggarwal et al. [1] presented a $1 - 1/e$-competitive algorithm. Subsequently, Devanur et al. [5] provided a far simpler randomized primal-dual analysis for both settings under adversarial arrivals.

For general weighted edges in the adversarial arrival setting, Aggarwal et al. [1] proved that no random algorithm can be constant-competitive, without additional assumptions. One such assumption is the free disposal assumption, which allows edges added to the matching to be discarded at any time. Under this assumption Fahrbach et al. [7] introduced an algorithm achieving a competitive ratio of 0.5086.

Another way of making the general edge-weighted case tractable is by assuming the vertex arrival orders to be uniformly random. For graphs with only one offline vertex and n online vertices, this is then equivalent to the classic secretary problem. This problem has been extensively studied (see Ferguson [8] for historical specifics). The optimal algorithm skips the first $\lfloor n/e \rfloor$ arriving vertices, then matches the first vertex arriving afterwards whose incident edge weight exceeds all previously observed edge weights, achieving the well-known competitive ratio of $1/e$. In light of this, problems involving the uniform random arrival order assumption are often called secretary-type problems; our problem is sometimes dubbed the matching secretary problem.

In addition to the aforementioned results for the matching secretary problem of Korula and Pal [15] and Kesselheim et al. [13], Kaplan et al. [10] presented an order-oblivious proof showing that the greedy-based algorithm of Korula and Pal [15] was in fact 1/5.83-competitive, with a matching lower bound. They also provide a counterexample showing that replacing the greedy subroutine with an optimal matching subroutine for the algorithm in Korula and Pal [15] causes it to perform

[2] If we directly followed their analysis, we would get a competitive ratio of $1/e - 1/q$. However, since an adversary chooses the number of groups q, this is not optimal in our setting.

[3] There are several ways of removing the $1/q$ factor. Randomizing the number of elements skipped, as we do here, is one way, but isn't necessary. Skipping a fixed $\lfloor n/e \rfloor$ elements, as they do in Theorem 2 of [13] can also yield a $1/e$ competitive ratio if one doesn't use the integral lower bound for the sum. We don't include the details here.

arbitrarily poorly. Hoefer and Kodric [9] study modifying the algorithm in Kesselheim et al. [13], where instead of computing the max-weight matching at each step over the vertices observed so far, they compute a greedy-matching. Using the same analysis as Kesselheim et al. [13], they find their algorithm to be $1/2e$-competitive.

While our setting concerns bipartite graphs, Ezra et al. [6] studied the setting with uniform random edge arrival orders and also the setting with uniform random vertex arrival orders over general graphs. For vertex arrivals, they present an algorithm with a tight competitive ratio of 5/12. For edge arrivals, they present a 1/4-competitive algorithm.

In many applications, however, arrival permutations are not uniformly random. Recently, Kesselheim et al. [14] investigated secretary-type problems under non-uniform arrival order distributions. They show that for certain non-uniform distributions with $\Omega(\log \log n)$ entropy, the standard secretary algorithm remains $1/e$-competitive, and they also present a $(1 + o(1))$-competitive algorithm for the k-choice secretary problem. They also show that the greedy-based algorithm of Korula and Pal [15] performs arbitrarily poorly for the online BVM problem under such distributions. A special case of non-uniform arrivals is the group setting, introduced by Korula and Pal [15] for the online BVM problem, where vertices arrive in predefined groups. Korula and Pal [15] showed their greedy-based algorithm fails in this setting also, and left as an open problem whether any constant-competitive algorithm exists for the group setting. Our work directly addresses the latter; we also provide an alternate proof of the algorithm of Kesselheim et al. [13] that may prove useful for establishing guarantees under more general non-uniform distributions, as in [14].

Another closely related setting is that of online submodular combinatorial auctions. When there is only one copy of each item in an auction, this is the online hypermatching setting. When there are multiple copies, this is equivalent to the online b-hypermatching problem. Both of these were studied by Kesselheim et al. [13]–they show that their algorithms for the online hypermatching/b-hypermatching settings attain identical competitive ratios for online submodular combinatorial auctions.

2 Edge-weighted Online Bipartite Matchings with Groups

We formally state the online BVM-G problem below.

We are given an edge-weighted bipartite graph $G = (L \cup R, E)$ with a non-negative weight function on the edges, $w : E \to \mathbb{R}_{\geq 0}$. The set of vertices R is known in advance. The set of vertices L is partitioned into q disjoint groups, $L = \bigcup_{j=1}^{q} B_j$. This partition is adversarially chosen and fixed before the online process begins. The total number of groups, q, is known to the algorithm.

The groups $B_1, \ldots, B_q$ arrive sequentially in a random order, where each of the $q!$ possible permutations of the groups is equally likely. At step $i \in \{1, \ldots, q\}$, a group B arrives. Upon its arrival, the algorithm observes all vertices within B and all edges incident to them, along with their corresponding weights. The algorithm must then immediately and irrevocably decide to match any subset of the vertices in B to currently unmatched vertices in R. The objective is to design an algorithm that maximizes the expected total weight of the final matching, where the expectation is taken over the random arrival order of the groups.

2.1 Reduction to Submodular Combinatorial Auctions

It was brought to our attention by Thomas Kesselheim [12] that this setting can be viewed as a submodular combinatorial auction. We present a proof using this perspective below. Consider the following algorithm, which is adapted from [13] to handle the groups setting.

Algorithm 1 Algorithm for Bipartite Matching Secretary with Groups

1: Set $\tau = \lfloor q/e \rfloor$ and skip the first τ groups.
2: For each subsequent group $B(i)$ (where $i > \tau$):

- Compute the optimal matching, $M_{\text{opt}}^{(i)}$, on the subgraph of G induced by the vertices of all i groups seen thus far.
- For each vertex $v \in B(i)$, if the edge (v, r) is in $M_{\text{opt}}^{(i)}$ and the vertex r is currently unmatched in M, add edge (v, r) to M.

First, we will sketch a reduction of the online BVM-G problem to a submodular combinatorial auction. Then, using a result of [13], we can then conclude that there exists a $1/e$-competitive algorithm for online submodular combinatorial auctions. This immediately implies a $1/e$-competitive algorithm (Algorithm 1) for the online BVM-G problem. Note that for a finite set Ω, we say that a set function $f : 2^{\Omega} \to \mathbb{R}$ is submodular if for every $X, Y \subseteq \Omega$, such that $X \subseteq Y$, and for every $x \in \Omega \setminus Y$, we have that $f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$.

A simplified setting of online combinatorial auctions can be defined as follows: suppose we are running an auction, with n bidders, and m items. For each bidder v, and every subset T of the m items, we may define a valuation function $w_v : 2^m \to \mathbb{R}_{\geq 0}$, so that $w_v(T)$ is the valuation of T for bidder v. Whenever a bidder v_i arrives, we must decide which subset T_i of the m items we would like to allocate to them, without knowing the valuations of bidders arriving after. The bidders arrive in uniform random order, and the valuations may be unbounded. The goal, is then to maximize the total social good $\sum_{i=1}^{n} w_{v_i}(T_i)$. Note that when each item can be assigned to at most one bidder, this corresponds exactly to the online BVM problem.

Now, in the online BVM-G problem, for each of the q groups, we may simply treat each group B_i as an individual bidder. Define a valuation function $w_{B_i}(T)$ so that for every subset of right vertices $T \subseteq R$, $w_{B_i}(T)$ is the weight of the maximum weight matching over the subgraph of G induced by the vertices $B_i \cup T$. Since the groups arrive in uniform random order, it is easy to see that this corresponds to an online combinatorial auction.

Note that Algorithm 1 implicitly defines an allocation: whenever some group $B(k)$ arrives, it computes $M_{\text{opt}}^{(k)}$, and allocates to $B(k)$ the neighboring right-vertices in $M_{\text{opt}}^{(k)}$. The valuation function for that group is the maximum weight matching induced by the vertices of $B(k)$ and the neighboring right-vertices. Moreover, this valuation function is known as the assignment valuation function [17], and also as the OXS valuation [16]. In particular, it is known to be submodular.

We note that our Algorithm 1 is identical to the algorithm from [13] for online submodular combinatorial auctions, but modified only to process groups. From [13], we conclude the following result:

Theorem 1. *For the online BVM-G problem, Algorithm 1 is $1/e$-competitive.*

2.2 An Alternative, Direct Approach

We now present an alternative approach. Consider the following algorithm, which makes two modifications to the 1/e-competitive algorithm used by Kesselheim et al. [13] for the online BVM problem. First, as above, we first modify the algorithm to process vertices group by group. Second, in [13], they skip the first $\tau = \lfloor n/e \rfloor$ elements, and get a competitive ratio of $1/e - 1/n$. Here, we draw $\tau \sim Binom(q, p)$, and skip the first τ groups. We will show that with $p = 1/e$, our algorithm is $1/e$ competitive for the online BVM-G problem. Finally, we note that randomizing τ allows us our analysis to extend easily in the continuous setting, which we formally define later.

We first introduce some notation. In the below, for any $i \in [q]$, let $B(i)$ denote the (random) group arriving at iteration i. For any group B, let $\delta(B)$ be the edges incident to some vertex in B, and for any set of edges $\mathcal{E}$, $w(\mathcal{E}) := \sum_{e \in \mathcal{E}} w(e)$. Let OPT denote the weight of the optimal offline matching over G. For some subset $L' \subseteq L$, let $OPT\,|_{L'}$ denote the weight of the edges in the optimal matching that are incident to some vertex in L'. Lastly, let S_i denote the (random) set of first i groups observed thus far. We analyze the following algorithm.

Algorithm 2 Algorithm for Bipartite Matching Secretary with Groups

1: For $p \in [0,1]$, sample $\tau \sim Binom(q,p)$, and skip the first τ groups.
2: For each subsequent group $B(i)$ (where $i > \tau$):

- Compute the optimal matching, $M_{\text{opt}}^{(i)}$, on the subgraph of G induced by the vertices of all i groups seen thus far.
- For each vertex $v \in B(i)$, if the edge (v,r) is in $M_{\text{opt}}^{(i)}$ and the vertex r is currently unmatched in M, add edge (v,r) to M.

We have the following lemma, which lower bounds the weight of the edges $\delta(B(i))$ that contribute to $M_{opt}^{(i)}$:

Lemma 2. *For any threshold τ, and any $i \in [q]$, $\mathbf{E}\left[w(\delta(B(i)) \cap M_{opt}^{(i)}) \mid \tau\right] \geq \frac{1}{q} \cdot OPT$*

Proof. Note that S_i is equally likely to be any combination of i distinct groups. Fixing $S_i = S$, we have that:

$$\mathbf{E}\left[w(\delta(B(i)) \cap M_{\text{opt}}^{(i)}) \mid S\right] = \frac{1}{i}\sum_{B \in S} w(\delta(B) \cap M_{\text{opt}}^{(i)}) = \frac{1}{i} \cdot w(M_{\text{opt}}^{(i)}(S)) \tag{1}$$

Where $M_{\text{opt}}^{(i)}(S)$ is the max weight matching of iteration i when $S_i = S$. Thus, $w(M_{\text{opt}}^{(i)}(S)) \geq OPT\,|_S$. Since every group is a member of S uniformly with probability i/q, $\mathbf{E}\left[OPT\,|_S\right] = (i/q)OPT$. It follows that:

$$\mathbf{E}_S\left[\mathbf{E}\left[w(\delta(B(i)) \cap M_{\text{opt}}^{(i)}) \mid S\right]\right] = \frac{1}{i} \cdot \mathbf{E}_S\left[w(M_{\text{opt}}^{(i)}(S))\right] \geq \frac{1}{i}\mathbf{E}_S\left[OPT\,|_S\right] = \frac{1}{q} \cdot OPT \tag{2}$$

□

Note that $w(\delta(B(i)) \cap M_{\text{opt}}^{(i)})$ is the weight of the edges that are candidates for selection *before* we check if adding them constitutes a valid matching. For each such edge, $e = (v,r)$, we add it only if r has yet to be matched. We next rigorously lower bound the probability at each iteration t that any right vertex r is available to be matched.

For readability, we introduce a few more definitions. For any group B, let $\pi(B) = i$ denote the event that group B arrives in iteration i. For a right vertex $r \in R$, define $\Phi(r,t)$ to be the event r is matched at any iteration $t' < t$ (and is thus not available at t) and $\phi(r,t)$ to be the event that r is matched in iteration t. Lastly, for some set of first k groups observed S_k let $\sigma(S_k^c)$ be the arrival order of the remaining groups $S_q \setminus S_k$. We have the following lemma:

Lemma 3. *For any threshold τ, any $r \in R$, $t > \tau$, and any choice of first $t-1$ groups S_{t-1}, and any arrival order of the remaining groups $\sigma(S_{t-1}^c)$, $\mathbf{P}\left[\neg\Phi(r,t) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \geq \frac{\tau}{t-1}$, where the probability is taken over arrival orders over S_{t-1}, and all arrival orders are equally likely.*

Proof. We prove the above claim by induction. Fix any right vertex $r \in R$ and any threshold τ. For the base case, when $t = \tau + 1$, the claim holds trivially, since no edges are added before τ. Now suppose that $\mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-2}, \sigma(S_{t-2}^c)\right] \geq \frac{\tau}{t-2}$ holds for all choices of S_{t-2} and $\sigma(S_{t-2}^c)$.

We first prove a fact that we will use twice below. Let S_{t-1} be a fixed set, $\sigma(S_{t-1}^c)$ be a fixed arrival order over the remaining groups, let B be any group in S_{t-1}, and suppose that $\pi(B) = t-1$. The event that S_{t-1} is the first $t-1$ groups we observe and B being the group arriving in iteration $t-1$ is equivalent to S_{t-2} being the first $t-2$ groups that we observe and B being the group arriving in iteration $t-1$. In other words, $S_{t-1} \wedge \pi(B) = t-1 \iff S_{t-2} \wedge \pi(B) = t-1$. Moreover, $\pi(B) = t-1 \wedge \sigma(S_{t-1}^c)$ implies some fixed arrival order $\sigma(S_{t-2}^c)$. Therefore:

$$\mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-1}, \pi(B) = t-1, \sigma(S_{t-1}^c)\right] = \mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-2}, \pi(B) = t-1, \sigma(S_{t-1}^c)\right] \tag{3}$$

$$= \mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-2}, \sigma(S_{t-2}^c)\right] \geq \frac{\tau}{t-2} \tag{4}$$

Where the final inequality is our induction hypothesis. Now by definition:

$$\mathbf{P}\left[\neg\Phi(r,t) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] = \mathbf{P}\left[\neg\phi(r,t-1) \wedge \neg\Phi(r,t-1) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \tag{5}$$

We have the following cases:

- **Case 1: $\mathbf{S_{t-1}}$ is such that r does not belong to $\mathbf{M_{opt}^{(t-1)}}$.** Then r certainly cannot be selected in iteration $t-1$. In other words: $\mathbf{P}\left[\neg\phi(r,t-1) \mid S_{t-1}, \neg\Phi(r,t-1), \sigma(S_{t-1}^c)\right] = 1$. Thus, expanding (5):

$$\begin{aligned}\mathbf{P}\left[\neg\Phi(r,t) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] &= \mathbf{P}\left[\neg\phi(r,t-1) \mid \neg\Phi(r,t-1), S_{t-1}, \sigma(S_{t-1}^c)\right] \\ &\quad \cdot \mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (6)\\ &= \mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (7)\end{aligned}$$

  But 3 and 4 imply that for every $B \in S_{t-1}$, $\mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-1}, \pi(B)=t-1, \sigma(S_{t-1}^c)\right] \geq \frac{\tau}{t-2}$. Therefore, $\mathbf{P}\left[\neg\Phi(r,t) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \geq \frac{\tau}{t-2} > \frac{\tau}{t-1}$

- **Case 2: $\mathbf{S_{t-1}}$ is such that r belongs to $\mathbf{M_{opt}^{(t-1)}}$.** Then exactly one group, say $\mathbf{B}^*$, in the first $t-1$ groups, contains a vertex adjacent to r in $\mathbf{M}_{\text{opt}}^{(\mathbf{t-1})}$. Thus, r is not matched in iteration $t-1$ if and only if any block other than $\mathbf{B}^*$ arrives in iteration $t-1$ or r has been matched at a previous iteration. So, by definition, $\neg\phi(r,t-1) \iff \Phi(r,t-1) \vee \pi(\mathbf{B}^*) \neq t-1$. Thus:

$$\begin{aligned}\mathbf{P}\left[\neg\Phi(r,t) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] &= \mathbf{P}\left[\neg\Phi(r,t-1) \wedge \neg\phi(r,t-1) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (8)\\ &= \mathbf{P}\left[\neg\Phi(r,t-1) \wedge (\Phi(r,t-1) \vee \pi(\mathbf{B}^*) \neq t-1) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (9)\\ &= \mathbf{P}\left[\neg\Phi(r,t-1) \wedge \pi(\mathbf{B}^*) \neq t-1 \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (10)\\ &= \mathbf{P}\left[\neg\Phi(r,t-1) \wedge \bigsqcup_{B \neq \mathbf{B}^*} \pi(B) = t-1 \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (11)\\ &= \sum_{B \neq \mathbf{B}^*} \mathbf{P}\left[\neg\Phi(r,t-1) \wedge \pi(B) = t-1 \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \qquad (12)\end{aligned}$$

  Under uniform random arrival order, for any $B \in S_{t-1}$, $\mathbf{P}\left[\pi(B) = t-1 \mid S_{t-1}, \sigma(S_{t-1}^c)\right] = \mathbf{P}\left[\pi(B) = t-1 \mid S_{t-1}\right] = \frac{1}{t-1}$. Once more, $\mathbf{P}\left[\neg\Phi(r,t-1) \mid S_{t-1}, \pi(B)=t-1, \sigma(S_{t-1}^c)\right] \geq \frac{\tau}{t-2}$. Since there are $t-2$ groups $B \in S_{t-1}$ such that $B \neq \mathbf{B}^*$, we may simplify 12:

$$\mathbf{P}\left[\neg\Phi(r,t) \mid S_{t-1}, \sigma(S_{t-1}^c)\right] \geq (t-2) \cdot \frac{\tau}{t-2} \cdot \frac{1}{t-1} = \frac{\tau}{t-1} \tag{13}$$

□

As an immediate consequence of Lemma 3, we have that for every τ, any $r \in R$, and $t > \tau$:

$$\mathbf{P}\left[\neg\Phi(r,t)\right] \geq \frac{\tau}{t-1}$$

Before we proceed to the main result, we first need the following technical lemmas, which we prove in the appendix:

Lemma 4. *For all n, and any $p \in [0,1]$, let $X(n,p) \sim \text{Binomial}(n,p)$ be a random variable, with cumulative distribution function $F_{X(n,p)}(x) := \Pr[X(n,p) \leq x]$. Then:*

$$\sum_{j=1}^{n} \frac{1}{j} F_{X(n,p)}(j-1) = \sum_{k=1}^{n} \frac{(1-p)^k}{k} \tag{14}$$

Lemma 5. *For all n, and any $p \in [0,1]$, let $X(n,p) \sim \mathrm{Binomial}(n,p)$ be a random variable, and define*

$$H(x) = \begin{cases} 0 & \text{if } x = 0 \\ \dfrac{x}{n} \displaystyle\sum_{i=x+1}^{n} \frac{1}{i-1} & \text{if } x \geq 1 \end{cases}$$

we have that:

$$\mathbf{E}_{X(n,p)}[H(X(n,p))] = p \cdot \sum_{k=1}^{n-1} \frac{(1-p)^k}{k} \tag{15}$$

We are now ready to present the main result:

Theorem 6. *Algorithm 2 with $p = 1/e$ is $\frac{1}{e}$-competitive for the bipartite matching secretary with groups.*

Proof. For $q = 0, 1$, the result is trivial. So suppose that $q \geq 2$. As above, let B be the (random) group arriving at iteration i. For each edge $e = (v, r)$ in $\delta(B) \cap M_{\text{opt}}^{(i)}$, we formally select it if r has not yet been matched (and is thus, still available) by iteration i. Let $\mathcal{A}_i$ denote the expected weight of the edges formally selected in iteration i. Combining Lemma 2, Lemma 3, and the linearity of expectation, for a given τ:

$$\mathbf{E}\left[\mathcal{A}_i \mid \tau\right] = \mathbf{E}\left[\sum_{\substack{e \in \delta(B) \cap M_{\text{opt}}^{(i)} \\ e=(v,r)}} w(e) \mathbf{1}_{\neg \Phi(r,i)} \mid \tau\right] \geq \frac{\tau}{i-1} \mathbf{E}\left[w(\delta(B) \cap M_{\text{opt}}^{(i)}) \mid \tau\right] \geq \frac{\tau}{q \cdot (i-1)} \cdot OPT \tag{16}$$

For $\tau \geq 1$ (we will account for the contribution when $\tau = 0$ later) and summing from $i = \tau + 1$ to q:

$$\mathbf{E}\left[ALG \mid \tau\right] = \sum_{i=\tau+1}^{q} \mathbf{E}\left[\mathcal{A}_i\right] \geq \frac{\tau}{q} \cdot OPT \cdot \sum_{i=\tau+1}^{q} \frac{1}{(i-1)} = OPT \cdot p \cdot \sum_{k=1}^{q-1} \frac{(1-p)^k}{k} \tag{17}$$

Where the final equality follows from Lemma 5.

Therefore, $\mathbf{E}_\tau\left[\mathbf{E}\left[ALG \mid \tau\right]\right] \geq \mathbf{E}\left[ALG \wedge \tau = 0\right] + OPT \cdot p \cdot \sum_{k=1}^{q-1} \frac{(1-p)^k}{k}$. To account for $\mathbf{E}\left[ALG \wedge \tau = 0\right]$, we note that the group arriving in iteration 1 may contribute only if $\tau = 0$. Clearly, all right vertices are available in iteration 1, so the contribution is

$$\mathbf{E}\left[\mathcal{A}_1\right] = \mathbf{E}\left[w(\delta(B) \cap M_{\text{opt}}^{(1)})\right] \geq \frac{1}{q} \cdot OPT$$

Since $\mathbf{P}\left[\tau = 0\right] = (1-p)^q$,

$$\mathbf{E}\left[ALG \wedge \tau = 0\right] \geq \mathbf{E}\left[\mathcal{A}_1 \mid \tau = 0\right] \cdot \mathbf{P}\left[\tau = 0\right] \geq \frac{(1-p)^q}{q} \cdot OPT$$

Summing everything up we have:

$$\mathbf{E}\left[ALG\right] \geq OPT \cdot \left(\frac{(1-p)^q}{q} + p \cdot \sum_{k=1}^{q-1} \frac{(1-p)^k}{k}\right)$$

It suffices to show that there is $p \in [0,1]$ so that $\frac{(1-p)^q}{q} + p \cdot \sum_{k=1}^{q-1} \frac{(1-p)^k}{k} \geq 1/e$ for all q. Now let $Y_q(p) := \frac{(1-p)^q}{q} + p \cdot \sum_{k=1}^{q-1} \frac{(1-p)^k}{k}$ for $q = 2, 3, \ldots$ First note that:

$$Y_q(p) - Y_{q+1}(p) = (1-p)^{q+1}/(q(q+1)) \geq 0, \quad \text{for } p \in [0,1],$$

We recognize the Taylor expansion of $-\ln p$ in the sum term $\sum_{k=1}^{q-1} \frac{(1-p)^k}{k}$. Therefore, $Y_q(p) \searrow Y(p) = -p \ln p$ as $q \to \infty$. Seting $p = 1/e$, $Y(p) = 1/e$, which implies the result. $\square$

2.3 Fixed Time Horizons

The above problem (and many secretary problem settings in general) assumes that we know the number of elements (here, groups) that will be arriving in advance, which we use only to set the number of elements we skip. In real world applications, this is frequently an unrealistic assumption. For instance, a company interviewing candidates to fill positions may not know how many candidates will apply in advance. Instead, we might know that we must hire somebody over some fixed time horizon $[0, t]$. We may then consider a continuous version of the above setting, where we instead don't know the number of groups that arrive in advance. Of the ones that do arrive, we assume that their arrival times $F(z)$ are i.i.d. on the real time interval $[0, t]$. For this setting, we can reuse our above analysis to deduce a $1/e$-competitive algorithm:

Theorem 7. *There exists a $1/e$-competitive algorithm for the continuous online BVM-G (and BVM) problem, where the number of groups is unknown, and arriving groups have i.i.d. arrival times $F(z)$.*

Proof. WLOG, we may assume that $F(z)$ is uniform[4] on $[0, 1]$. Suppose q groups arrive during $[0, t]$. Then, we can modify Algorithm 2 (following the analysis in Bruss [3]) and set $\tau = 1/e$, so that we skip any group that arrives before $t = 1/e$, and run the algorithm as usual for each group arriving after $1/e$. Note that, the number of groups arriving before $1/e$ is a random variable with distribution $Binom(q, 1/e)$. Therefore, by setting $\tau = 1/e$, we are skipping the first $Binom(q, 1/e)$ groups, and for each group that arrives, we compute the max matching, and pair all vertices in the group with their corresponding vertices in the max matching, provided that doing so constitutes a valid matching. This is now exactly identical to Algorithm 2, so the result follows from Theorem 6 if we set $\tau = F^{-1}(1/e)$. □

[4]Since otherwise, we may just introduce a change of time $x = F(z)$ so that in the x time scale, time runs from 0 to 1, and for each group arrival time Z, $X = F(z)$ is uniform on $[0, 1]$. See Theorem 2.1.10 in [4] on the probability integral transform.

Deferred Proofs

We have the following Lemma, which relates the derivative of a binomial distribution function to a binomial density function, which will help us prove Lemma 4:

Lemma 8. *For any n, and some $p \in (0,1)$, let $X(n, p) \sim$ Binomial(n,p), $X(n-1, p) \sim$ Binomial$(n-1,p)$, and define $F_{X(n,p)}(x;p) := \Pr(X(n,p) \leq x \mid p)$.*

$$\frac{d}{dp} F_{X(n,p)}(x;p) = -n\mathbf{P}(X(n-1,\, p) = x)$$

Proof. By definition, $F_X(x;p) = \sum_{k=0}^{x} \binom{n}{k} p^k (1-p)^{n-k}$. Differentiating, we have:

$$\frac{d}{dp} F_X(x;p) = \sum_{k=0}^{x} \binom{n}{k} \frac{d}{dp}\left(p^k (1-p)^{n-k}\right) \tag{18}$$

$$= \sum_{k=0}^{x} \binom{n}{k} \left(kp^{k-1}(1-p)^{n-k} - (n-k)\, p^k (1-p)^{n-k-1}\right) \tag{19}$$

$$= \sum_{k=0}^{x} \left(\binom{n}{k} kp^{k-1}(1-p)^{n-k} - \binom{n}{k}(n-k)\, p^k (1-p)^{n-k-1}\right). \tag{20}$$

To simplify, let $A_k = \binom{n}{k} kp^{k-1}(1-p)^{n-k}$. Using the identity $\binom{n}{k}(n-k) = (k+1)\binom{n}{k+1}$, the second term in the summand can be expressed as:

$$\binom{n}{k}(n-k)\, p^k (1-p)^{n-k-1} = (k+1)\binom{n}{k+1} p^k (1-p)^{n-k-1} = A_{k+1}.$$

Thus, 20 is telescoping:

$$\frac{d}{dp} F_X(x;p) = \sum_{k=0}^{x} (A_k - A_{k+1}) = A_0 - A_{x+1}.$$

Clearly, $A_0 = 0$. The last term is A_{x+1}. Noting that $(x+1)\binom{n}{x+1} = n\binom{n-1}{x}$:

$$\frac{d}{dp} F_X(x;p) = -A_{x+1} = -(x+1)\binom{n}{x+1} p^x (1-p)^{n-(x+1)} = -n\binom{n-1}{x} p^x (1-p)^{n-1-x}$$

Where $\mathbf{P}(X(n-1,\, p) = x) = \binom{n-1}{x} p^x (1-p)^{n-1-x}$.

□

For all of the proofs below, we fix any $n \geq 1$, and any $p \in (0,1)$ we let $X(n,\, p) \sim$ Binomial(n,p), $X(n-1,\, p) \sim$ Binomial$(n-1,p)$, and define $F_{X(n,p)}(x;p) := \Pr(X(n,p) \leq x \mid p)$. We now proceed with the proof of Lemma 4:

Proof of Lemma 4. We first prove that:

$$\sum_{j=1}^{n} \frac{1}{j} F_{X(n,p)}(j-1) = \sum_{k=1}^{n} \frac{(1-p)^k}{k} \tag{21}$$

For $p = 0$, this is trivial. Fix n and let $X(n-1,\, p) \sim$ Binomial$(n-1,p)$. We let $L(p)$, $R(p)$ denote the left and right hand sides of 21. We will differentiate both sides with respect to p. From Lemma 8, $\frac{d}{dp} F_{X(n,p)}(x;p) = -n\mathbf{P}(X(n-1,\, p) = x)$. Therefore:

$$L'(p) = \sum_{j=1}^{n} \frac{1}{j} \frac{\partial F_{X(n,p)}(j-1)}{\partial p} = \sum_{j=1}^{n} \frac{1}{j}\left(-n\binom{n-1}{j-1} p^{j-1}(1-p)^{n-j}\right)$$

Since $\frac{n}{j}\binom{n-1}{j-1} = \binom{n}{j}$, this is equivalent to:

$$L'(p) = -\sum_{j=1}^{n}\binom{n}{j}p^{j-1}(1-p)^{n-j} = -\frac{1}{p}\sum_{j=1}^{n}\binom{n}{j}p^{j}(1-p)^{n-j} \tag{22}$$

The summation represents the probability $\mathbf{P}(X(n,p) \geq 1)$, which is $1 - \mathbf{P}(X(n,p) = 0) = 1 - (1-p)^n$. Thus:

$$L'(p) = -\frac{1-(1-p)^n}{p}$$

For the right hand side:

$$R'(p) = \frac{d}{dp}\sum_{k=1}^{n}\frac{(1-p)^k}{k} = -\sum_{k=1}^{n}(1-p)^{k-1} = -\frac{1-(1-p)^n}{p}$$

Where the final equality results from noting the finite geometric series. Since $L'(p) = R'(p)$, $L(p) = R(p) + C$. Clearly $R(1) = 0$. Now note that for $p = 1$, $X(n,p) = n > j-1$, so $F_{X(n,p)}(j-1) = 0$, which implies $L(1) = 0$, so $C = 0$, and $L(p) = R(p)$. $\square$

We now present the proof of Lemma 5:

Proof of Lemma 5. Recall that

$$H(x) = \begin{cases} 0 & \text{if } x = 0 \\ \dfrac{x}{n}\displaystyle\sum_{i=x+1}^{n}\frac{1}{i-1} & \text{if } x \geq 1 \end{cases}$$

We now show that:

$$\mathbf{E}_{X(n,\,p)}[H(X(n,\,p))] = p \cdot \sum_{k=1}^{n-1}\frac{(1-p)^k}{k} \tag{23}$$

We have that:

$$\begin{aligned}
\mathbf{E}_{X(n,p)}\left(\frac{X(n,p)}{n} \cdot \sum_{i=X(n,p)+1}^{n}\frac{1}{(i-1)}\right) &= \sum_{j=1}^{n-1}\frac{j}{n} \cdot \mathbf{P}(X(n,p) = j)\sum_{i=j+1}^{n}\frac{1}{(i-1)} &(24)\\
&= \sum_{i=2}^{n}\frac{1}{i-1} \cdot \left(\sum_{j=1}^{i-1}\frac{j}{n}\mathbf{Pr}[X(n,p) = j]\right) &(25)\\
&= \sum_{i=1}^{n-1}\frac{1}{i} \cdot \left(\sum_{j=1}^{i}\frac{j}{n}\mathbf{Pr}[X(n,p) = j]\right) &(26)\\
&= \sum_{i=1}^{n-1}\frac{1}{i} \cdot \left(\frac{1}{n}\sum_{j=1}^{i} j \cdot \binom{n}{j}p^j(1-p)^{n-j}\right) &(27)
\end{aligned}$$

Since $j \cdot \binom{n}{j} = n \cdot \binom{n-1}{j-1}$:

$$j \cdot \binom{n}{j}p^j(1-p)^{n-j} = n \cdot \binom{n-1}{j-1}p^j(1-p)^{n-j} = np \cdot \binom{n-1}{j-1}p^{j-1}(1-p)^{(n-1)-(j-1)}.$$

Note that, $P(X(n-1,p) = j-1) = \binom{n-1}{j-1}p^{j-1}(1-p)^{(n-1)-(j-1)}$. We may simplify 27 and swap indices of summation:

$$\sum_{i=1}^{n-1} \frac{1}{i} \cdot \left(\frac{1}{n} \sum_{j=1}^{i} j \cdot \binom{n}{j} p^j \left(1-p\right)^{n-j} \right) = \sum_{i=1}^{n-1} \frac{1}{i} \cdot \sum_{j=1}^{i} p \cdot P\left(X\left(n-1, p\right) = j-1\right) \tag{28}$$

$$= p \sum_{i=1}^{n-1} \frac{1}{i} F_{X(n-1,p)}\left(i-1\right) = p \cdot \sum_{k=1}^{n-1} \frac{\left(1-p\right)^k}{k} \tag{29}$$

Where the final equality follows from Lemma 4.

□

References

[1] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations, 2010. URL `https://arxiv.org/abs/1007.1271`.

[2] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice, 2020. URL `https://arxiv.org/abs/2006.01026`.

[3] F. Thomas Bruss. A Unified Approach to a Class of Best Choice Problems with an Unknown Number of Options. *The Annals of Probability*, 12(3):882 – 889, 1984. doi: 10.1214/aop/1176993237. URL `https://doi.org/10.1214/aop/1176993237`.

[4] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[5] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, page 101–107, USA, 2013. Society for Industrial and Applied Mathematics. ISBN 9781611972511.

[6] Tomer Ezra, Michal Feldman, Nikolai Gravin, and Zhihao Gavin Tang. Tight bounds for secretary matching in general graphs. *Mathematics of Operations Research*, 2024. doi: 10.1287/moor.2022.0206. Published online November 26, 2024.

[7] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. *Journal of the ACM*, 69(6):1–35, November 2022. ISSN 1557-735X. doi: 10.1145/3556971. URL `http://dx.doi.org/10.1145/3556971`.

[8] Thomas S. Ferguson. Who Solved the Secretary Problem? *Statistical Science*, 4(3):282 – 289, 1989. doi: 10.1214/ss/1177012493. URL `https://doi.org/10.1214/ss/1177012493`.

[9] Martin Hoefer and Bojana Kodric. Combinatorial secretary problems with ordinal information, 2017. URL `https://arxiv.org/abs/1702.01290`.

[10] Haim Kaplan, David Naori, and Danny Raz. Online weighted matching with a sample, 2021. URL `https://arxiv.org/abs/2104.05771`.

[11] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Symposium on the Theory of Computing*, 1990. URL `https://api.semanticscholar.org/CorpusID:822904`.

[12] Thomas Kesselheim, 2025. Personal communication.

[13] Thomas Kesselheim, Klaus Radke, Andreas Tonnis, and Berthold Vocking. An optimal on-line algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Algorithms - ESA 2013*, volume 8125 of *Lecture Notes in Computer Science*, pages 589–600. Springer, 2013. doi: 10.1007/978-3-642-40450-4_50.

[14] Thomas Kesselheim, Robert Kleinberg, and Rad Niazadeh. Secretary problems with non-uniform arrival order, 2015. URL `https://arxiv.org/abs/1502.02155`.

[15] Nitish Korula and Martin Pal. Algorithms for secretary problems on graphs and hypergraphs. *CoRR*, abs/0807.1139, 2008. URL `http://arxiv.org/abs/0807.1139`.

[16] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2005.02.006. URL `https://www.sciencedirect.com/science/article/pii/S089982560500028X`. Mini Special Issue: Electronic Market Design.

[17] Lloyd S. Shapley. Complements and substitutes in the opttmal assignment problem. *Naval Research Logistics Quarterly*, 9(1):45–48, 1962. doi: https://doi.org/10.1002/nav.3800090106. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800090106`.