Secretary Markets with Local Information

Ning Chen * Martin Hoefer † Marvin Künnemann ‡ Chengyu Lin§
Peihan Miao ¶

Abstract

The secretary model is a popular framework for the analysis of online admission problems beyond the worst case. In many markets, however, decisions about admission have to be made in a distributed fashion. We cope with this problem and design algorithms for secretary markets with limited information. In our basic model, there are m firms and each has a job to offer. n applicants arrive sequentially in random order. Upon arrival of an applicant, a value for each job is revealed. Each firm decides whether or not to offer its job to the current applicant without knowing the actions or values of other firms. Applicants accept their best offer.

We consider the social welfare of the matching and design a decentralized randomized thresholding-based algorithm with a competitive ratio of $O(\log n)$ that works in a very general sampling model. It can even be used by firms hiring several applicants based on a local matroid. In contrast, even in the basic model we show a lower bound of $\Omega(\log n/(\log\log n))$ for all thresholding-based algorithms. Moreover, we provide a secretary algorithm with a constant competitive ratio when the values of applicants for different firms are stochastically independent. In this case, we show a constant ratio even when we compare to the firm's individual optimal assignment. Moreover, the constant ratio continues to hold in the case when each firm offers several different jobs.

1 Introduction

In the *secretary problem* [15, 36] a firm interviews a set of applicants who arrive in an online fashion. When an applicant arrives, his non-negative value is revealed, and the firm needs to make an immediate and irrevocable decision on whether to make an offer to the applicant, without knowing the values of future potential applicants. The objective is to maximize the (expected) value of the hired applicant. As a fundamental online hiring scenario, this problem is well studied both in social science and computer science. It is well known that the secretary problem, with an adversarial order, does not admit an algorithm with any bounded competitive ratio. However, if applicants arrive in uniform random order, there is an online algorithm that hires the best applicant with optimal probability approaching $1/e$ (see, e.g., [7]). For a more detailed discussion on the secretary problem, we refer to, e.g., [3, 19].

The secretary problem constitutes a popular basis to study online admission scenarios with various applications. In many of these scenarios, however, there are multiple firms that make offers and accept applicants in a distributed fashion with limited information and without central coordination (e.g., online resource allocation problems in large networks, hiring in job markets, online dating, school admission, casting shows like "The Voice", etc.). Surprisingly little is known about how decision makers can successfully coordinate in such scenarios to achieve an allocation that is good – from an individual or a social point of view.

*Nanyang Technological University, Singapore. ningc@ntu.edu.sg
†Goethe University, Frankfurt am Main, Germany. Supported by DFG Cluster of Excellence MMCI at Saarland University. mhoefer@cs.uni-frankfurt.de
‡Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany. marvin@mpi-inf.mpg.de
§Chinese University of Hong Kong, Hong Kong. cylin@cse.cuhk.edu.hk
¶University of California, Berkeley, USA. peihan@cs.berkeley.edu

In this paper, we study a natural generalization of the secretary problem from one firm to multiple firms and from one hire to multiple hires. Applicants arrive sequentially in random order. Firms are decision makers; that is, there is no centralized authority, and every firm can choose different hiring strategies (based on observed information). Upon arrival of a new applicant, each firm can only observe its local information, i.e., it has no knowledge about the values of other (firm, applicant)-pairs and the selected strategies of other firms. Firms decide simultaneously about whether to make an offer to the applicant, and the applicant accepts the offer of the most preferred firm. Our goal is to design and analyze strategies for the firms in such a decentralized environment that allow the firms to obtain good allocations. The algorithms are evaluated globally: We hope the outcomes achieve good social welfare (i.e., the total value obtained by all firms). Thus, we measure the competitive ratio compared to the social welfare of the optimal allocation in hindsight.

We provide a strategy to approximate social welfare within a logarithmic factor, and we show almost matching lower bounds on the competitive ratio for a very natural class of algorithms inspired by the classic secretary algorithm. Since overcoming the obstacles posed by the lower bound seems difficult given the limited feedback of our decentralized setting, our results give some evidence that, in the general case, centralized control seems to be necessary to achieve good social welfare. Moreover, we identify a natural setting that admits algorithms with small constant competitive ratio both globally and individually (with respect to the best set of applicants for each firm). This implies that in some cases, algorithms can obtain a good allocation, despite decentralized control and very limited feedback.

We stress that this work focuses on the challenges introduced by the *distributed* setting of a natural online admission problem in the presence of multiple firms. In particular, we study how to obtain good allocations even under *limited information*. In contrast, related work [9, 24, 25, 27] views firms as *strategic and self-interested* entities, and aims to characterize properties of equilibrium behavior in secretary online markets. Unfortunately, this is a highly intricate task, and such results have been restricted mostly to quite special cases: all firms have the same preference over applicants, all applicants have the same preference over firms, each firm can fully observe the strategies employed by every other firm, and each firm strives to hire only the best applicant (for a review of the literature, see Section 1.3 below). In our setting, however, information about values and strategies played by other firms is revealed only very indirectly. Moreover, each (firm, applicant)-pair has a possibly arbitrary non-negative value and our algorithms generalize even to firms having multiple positions.

As suggested by the proof of our lower bound, particularly the limited feedback provides a major challenge to improve upon our main algorithm: Only if a firm makes an offer that is rejected by the applicant, this firm could, in principle, deduce some information about the existence and the number of other firms in the market (as well as their values for the applicants, or possibly even their strategies used for hiring) – this is the only time when the existence of other firms makes a notable difference in the feedback of a given firm. However, learning such information appears to require a minimum number of job positions for systematic testing of possible competition in the market. A firm would have to increase the number of early offers to obtain information about competition, at the expense of offering the job to possibly many bad applicants and a high chance of terminating the learning/hiring process early with very suboptimal results. Whether such strategies may lead to better allocations remains an interesting question for further work.

1.1 Model

We first outline our *basic model*, a decentralized online scenario for hiring a single applicant per firm with random arrival. There is a complete bipartite graph $G = (U, V, w)$ with sets $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$ of firms and applicants, respectively. We assume that each firm can hire at most one applicant.

There is a *value* or *weight function* $w : U \times V \to \mathbb{R}^+$. The weights describe an implicit preference of each individual to the other side. Each firm $u \in U$ prefers applicants according to the decreasing order of $w(u, \cdot)$ of the edges incident to u; similarly, each applicant $v \in V$ prefers firms according to the decreasing order of $w(\cdot, v)$ of the edges incident to v. Note that in this definition, the preferences are symmetric: applicants and firms use the same weights to determine their preference. A canonical more general preference setting would specify an additional weight function $w' : V \times U \to \mathbb{R}^+$, such that applicant v has a preference $w'(v, u)$ to firm u that is possibly different from u's preference $w(u, v)$ to v. Since already the symmetric weights setting provides a natural scenario to study the challenges raised by the decentralized setup, we do not consider the more general case in the present paper and leave it open for further research. Furthermore, for ease of presentation, we assume that no two edges have the same weight; for our results, this assumption can easily be lifted, e.g., when the algorithm internally applies a tiny random perturbation to each weight.

Applicants in V arrive one by one to the market. Upon the arrival, each applicant v reveals to each firm u the corresponding edge weight $w(u, v)$. Each firm immediately decides whether to make an offer to the applicant or not; after collecting all job offers, the applicant then picks the one that she prefers the most, i.e., the one with the largest weight. Note that each firm u can only see its own weights $w(u, \cdot)$ for the applicants arrived so far. It has no information about future applicants; in addition, all decisions cannot be revoked. A firm can make multiple offers over time until it succeeds to hire an applicant. In this paper, we mostly concentrate on the random permutation model, i.e., weights are fixed by an adversary but applicants arrive in a uniformly random order. We also briefly mention extensions of our results to other standard models, such as the iid model (weights drawn iid from known distributions), prophet-inequality model (different known distributions, adversarial arrival), and more general models based on different mixtures of stochastic and adversarial elements.

Our goal is to design decentralized algorithms that enable each firm to make decisions based only on its own previously seen information, without any centralized authority that coordinates different firms. Due to the online arrival some performance loss is unavoidable, and there are two natural objectives to quantify this loss. The standard benchmark is the *social welfare*, defined to be the total weight of assigned firm-applicant pairs. Formally, let M be a matching in G and define $w(M) = \sum_{(u,v)\in M} w(u, v)$. For an algorithm $\mathcal{A}$, we say that the algorithm has a *competitive ratio* of α if, for all instances, we have $w(M^*)/\mathbb{E}\left[w(M^{\mathcal{A}})\right] \leq \alpha$, where M^* is the maximum weight matching in G, and $M^{\mathcal{A}}$ is the matching returned when every firm runs algorithm $\mathcal{A}$. Here the expectation is taken over the random permutation and, if the algorithm is randomized, over its internal random bits. In addition, we examine the *individual optimum* for each firm (i.e., the value of its best applicant) and the possibilities to obtain a constant competitive ratio for this benchmark. This goal is obviously much more demanding than social welfare. It can be impossible, e.g., if there is a single applicant that is extremely valuable for every firm, while all others are not valuable at all. Consequently, a constant competitive ratio for individual optima can be achieved only in domains with additional structure. In this paper, we obtain them when applicant values result from a stochastic process with a sufficient degree of independence among firms.

1.2 Motivation, Contribution and Techniques

The starting point of our work is the observation that some interesting online hiring scenarios should not be regarded as isolated instances of the secretary problem. Consider, for instance, job markets with "public performance displays": In these markets, we have performers and sponsors. In a performance, each performer publicly demonstrates his or her qualities. Interested sponsors have to decide whether or not to make them offers. Such hiring markets can be time-critical, thus we assume that this decision should be made immediately. It is not hard to see that this setting closely fits our basic model, which thus provides an avenue to study especially the aspect that all sponsors are influenced by the offers of other sponsors as well as of the decision of the performers. In particular, as we shall see below, viewing the decentralized problem as isolated secretary problems may quickly lead to globally unsustainable allocations in the market.

Specifically, in our basic model consider the case where every firm runs the classic secretary algorithm [15, 36]. In this algorithm, each firm u samples the first $r(u)$ applicants, records the best weight seen in the sample, and then makes an offer to every applicant that exceeds this threshold (see Algorithm 1 for a formal description). Here, $r(u)$ is a parameter chosen freely by firm u. In the classic secretary problem, i.e., if only a single firm is present, its optimal choice (based on the number of applicants) is well known to satisfy $r(u) = \Theta(n)$, and $r(u) \approx n/e$ when n becomes large.

It turns out that such a strategy fails miserably in a decentralized market, even if we allow each firm u_i to run the classic algorithm with a possibly different parameter $r(u_i) \in \{1, 2, \ldots, n\}$. The proof is based

Algorithm 1: The classic secretary problem algorithm for firm u.

Parameter: Sample size $r(u) \in \{1, 2, \ldots, n\}$

Reject the first $r(u)$ applicants, denote the set of these applicants by $R(u)$

$T_u \leftarrow \max_{j \in R(u)} w(u, v_j)$

for *applicant* v_t *arriving in round* $t = r(u) + 1, \ldots, n$ **do**

  if $w(u, v_t) > T_u$ **then**

    Make an offer to v_t, stop if v_t accepts

on two instances. If there is at least one firm u_i with parameter $r(u_i) \leq \sqrt{n}$, then the adversary presents an instance that effectively reduces the problem to finding the best applicant for the single firm u_i. Due to the short sampling period, the probability of this event becomes small. Otherwise, if all parameters satisfy $r(u_i) \geq \sqrt{n}$, the adversary presents an instance with many applicants that are "bad" for all firms and only a few ($\Theta(\sqrt{n} \log n)$) applicants that are "good" for all firms. With high probability, there will be at least one "good" applicant in the common sampling period. Then all firms compete over the "good" applicants later on, but at most $O(\sqrt{n} \log n)$ firms can hire one of them. In contrast, to achieve an optimal social welfare each firm should hire an applicant.

Proposition 1. *For every choice of parameters $r : U \to \{1, 2, \ldots, n\}$, when each firm u_i uses the classic secretary algorithm with parameter $r(u_i)$, the resulting assignment has a competitive ratio of $\Omega(\sqrt{n}/\log n)$.*

Proof. Suppose there are n applicants and $m = n$ firms. First, if there is a firm u_i with $r(u_i) \leq \sqrt{n}$, then we consider an instance in which the best applicant for firm u_i has weight 1 for firm u_i. We let all other (firm, applicant)-pairs have distinct weights in the interval $[0, \varepsilon]$, for some sufficiently small $\varepsilon > 0$, in such a way that every applicant has the largest weight for firm u_i. Thus, every applicant prefers to join firm u_i. Observe that by choosing ε sufficiently small, the competitive ratio is dominated by the inverse of the probability that firm u_i hires its best applicant. Let j_k be the k-th best applicant for firm u_i, for $k \geq 1$. Suppose the best applicant in the sample set $R(u_i)$ is j_{k+1}. Then the probability that the algorithm of firm u_i hires j_1 is $1/k$. By iteratively drawing the positions for the best applicants, we see that

$$\begin{aligned}
\mathbf{Pr}\left[j_1 \text{ hired}\right] &= \sum_{k=1}^{n-r(u_i)} \mathbf{Pr}\left[j_1 \text{ hired} \mid j_{k+1} \text{ best in } R(u_i)\right] \cdot \mathbf{Pr}\left[j_{k+1} \text{ best in } R(u_i)\right] \\
&= \sum_{k=1}^{n-r(u_i)} \frac{1}{k} \cdot \left(\frac{r(u_i)}{n} \cdot \prod_{j=0}^{k-1} \frac{n - r(u_i) - j}{n - 1 - j} \right) \quad \leq \quad \frac{r(u_i)}{n} \cdot \sum_{k=1}^{n} \frac{1}{k} \quad = \quad O\left(\frac{\log n}{\sqrt{n}} \right) .
\end{aligned}$$

Otherwise, if $r(u_i) > \sqrt{n}$ for all firms u_i, then we consider an instance in which applicants come in two types: $n_g = \lceil \frac{1}{2} \sqrt{n} \ln n \rceil$ 'good' applicants with weight 2 for all edges incident to them, and the remaining 'bad' applicants with weight 1 for all edges incident to them. (Recall that to avoid ties, we can add a small perturbation $\epsilon_{u,v}$ on all pairs). Regardless of the permutation of the applicants, we have $w(M^*) = m + n_g$. Next, we consider the matching $M^{\mathcal{A}}$ returned by the algorithm and give an upper bound on $\mathbb{E}\left[w(M^{\mathcal{A}})\right]$. Let $r = \lceil \sqrt{n} \rceil$, then in the first r rounds, the probability that no good applicant arrives is

$$p \;=\; \prod_{j=0}^{r} \frac{n - n_g - j}{n - j} \;\leq\; \left(1 - \frac{n_g}{n}\right)^r \;\leq\; e^{-\frac{n_g}{n} r} \;\leq\; e^{-\frac{1}{2} \cdot \frac{r}{\sqrt{n}} \cdot \ln n} \;\leq\; n^{-\frac{1}{2}}.$$

by the choices of n_g and r.

Since each firm u_i samples $r(u_i) \geq r$ applicants, a "good" applicant arriving in the first r rounds is observed by all firms. Additionally, since a good applicant is good for all firms, all thresholds will be set to

2, and no 'bad' applicant can be hired. Hence, $\mathbb{E}\left[w(M^{\mathcal{A}})\right] \le p(m+n_g) + (1-p)\cdot 2n_g \le p(m+n_g) + 2n_g$. We conclude that

$$\frac{\mathbb{E}\left[w(M^{\mathcal{A}})\right]}{w(M^*)} \quad \le \quad p + \frac{2n_g}{m+n_g} \le n^{-\frac{1}{2}} + \frac{2\lceil \frac{1}{2}\sqrt{n}\ln n\rceil}{n+\lceil \frac{1}{2}\sqrt{n}\ln n\rceil} \quad = \quad \Theta\left(\frac{\log n}{\sqrt{n}}\right) \ .$$

□

Thus, an attempt to cope with the decentralized setting by using an optimal algorithm for isolated secretary problems may be unacceptable both globally and individually. In this article, we study alternatives to the classical secretary algorithm which (when followed by all participating firms) overcome the distributed nature of the scenario and manage to provide reasonably competitive solutions.

In particular, in Section 3 we present an $O(\log n)$-competitive solution based on sampling and thresholds. Interestingly, the algorithmic approach we use here was devised by Babaioff, Immorlica and Kleinberg [4] for a different extension of the secretary problem, namely to obtain a logarithmic guarantee for the matroid secretary problem. We give a novel analysis of this approach in our decentralized setting. The main obstacle we face are correlations between decisions of different firms, since all firms are presented the same, yet random, arrival order. Roughly speaking, the key tool to overcome these difficulties is an abstraction that bundles all stochastic decisions (due to the random arrival order) in a way that allows us to treat correlations among firms conveniently using linearity of expectation. The remaining effects of applicant preferences and competition can then be analyzed in a pointwise, worst-case fashion.

This algorithm can be applied very generally beyond the basic model. In fact, we prove the guarantee in a more general scenario in which each firm u_i has a private matroid $\mathcal{S}_i$ and can accept any subset of applicants that forms an independent set in $\mathcal{S}_i$. Furthermore, as shown in Appendix B, our analysis extends to a general sampling model due to Göbel et al. [20] that encompasses the secretary model (random arrival, worst-case weights), prophet-inequality model (worst-case arrival, stochastic weights), as well as a variety of other mixtures of stochastic and worst-case assumptions.

Returning to the basic model, we contrast this upper bound with an almost matching lower bound for thresholding-based algorithms. A thresholding-based algorithm samples a number of applicants, determines a threshold, and then makes offers to every remaining applicant that has a weight above the threshold. Although such algorithms are optimal in the centralized setting, every thresholding-based algorithm must have a competitive ratio of at least $\Omega(\log n/\log\log n)$ in the decentralized setting. This shows that Babaioff et al.'s approach is an almost optimal way to obtain such a threshold in the decentralized setting. Furthermore, it illustrates the obstacles we need to overcome to obtain constant-competitive solutions. In particular, in the proof of the lower bound, we carefully construct a challenge to guess how many firms contribute to the social welfare, which is necessary to avoid overly high concentration of offers on a small number of valuable applicants. Given the extremely limited feedback about the presence of other firms in our model, this seems hardly possible in the general case.

In Section 4, however, we show that this challenge can be overcome if there is stochastic independence between the weights of an applicant to different firms. We study this property in a generalization of our basic model, namely *decentralized secretary matching*: here, each firm u_i has k_i different jobs to offer. Upon arrival, an applicant reveals k_i weights for each firm u_i, one for each position. If each firm uses a variant of the optimal e-competitive algorithm for bipartite matching [28], we prove that independence between weights of different firms yields a constant competitive ratio. Moreover, each firm even manages to recover a constant fraction of the individual optimum matching, and therefore almost plays a best response strategy.

Finally, we conclude in Section 5 with a discussion of open problems.

1.3 Related Work

The secretary model is a classic domain of stopping and online admission problems [15, 36]. The classic algorithm results from analyzing and optimizing a recurrence that strives to optimize the probability of hiring the best applicant. If n grows to infinity, the optimal probability approaches $1/e$. There has been a very large research interest in the secretary problem and its variants. For much of the earlier related work we refer to standard surveys [3,19]. Here we discuss only directly related work on algorithms with competitive ratios for secretary problems with combinatorial structure.

This work prominently addresses secretary models for packing problems with random arrival of elements. This domain became prominent especially for the case of the matroid secretary problem [4], where elements of a weighted matroid arrive in random order and reveal their weight upon arrival. The first algorithm for this problem was $O(\log k)$-competitive, where k is the rank of the matroid. More recently, the ratio was reduced to $O(\log \log k)$ [16, 35]. Constant-factor competitive algorithms have been obtained for numerous special matroid classes, such as graphic [32], co-graphic [38], transversal [11], laminar [26], regular and decomposable [12] and more [23,30]. More recently, these algorithms have also been adjusted and extended to submodular objective functions [6,18,21]. It remains a fascinating open problem whether a constant-factor competitive algorithm exists for all matroids or not.

Another popular domain is bipartite matching in the secretary model, where the firm has multiple positions. This problem was first studied as transversal matroid, in which the firm needs to commit on the allocation of applicants to positions only in the end of the hiring process. More recently, there has also been interest in variants where the firm must commit on a specific a position at the time when hiring an applicant. There have been several constant-factor algorithms for this problem [11,32]. Some of them apply specifically when applicant values for positions have a product-form, which is common in ad-auctions [2]. In Section 4 we use an algorithm that applies to general non-negative weight and achieves an optimal approximation factor of e [28]. Ideas for bipartite matching are also useful for solving secretary versions of packing linear problems [10, 29, 37]. The ratios then depend asymptotically on the number of non-zero entries in each column and the minimum ratio between right-hand-side value and constraint coefficients.

Secretary problems with multiple firms have started to attract attention recently in a game-theoretic direction [9, 24, 25, 27]. These works assume that firms have full information about all arrived applicants, their preferences, the preferences of all firms with respect to arrived applicants, and their hiring strategies. Moreover, the existing works address the case with a uniform preference of firms over applicants and a uniform preference of applicants over firms. In their setting, each firm has a single job, and the goal is to hire the best applicant (and not necessarily the expected value of an applicant). In [25] a continuum of arriving agents is studied, and earliest offering times in a Nash equilibrium are analyzed. A finite variant and Nash equilibria of this problem are studied in [9], whereas algorithms for computing subgame-perfect equilibria are given in [27]. In a slightly different direction, in dueling scenarios with two players the goal is to hire a better applicant than the competitor [24]. In contrast, our work studies markets with significantly different assumptions, since we explore markets with both decentralized control and restricted feedback. Moreover, in our study firms can have several positions to offer, and preferences of firms and applicants can be highly non-uniform.

Other variants of secretary problems with multiple decision makers are, e.g., variants where the goal is to minimize blocking pairs of firms and applicants [5], or secretary problems with k queues [17]. Moreover, secretary variants of combinatorial packing problems with ordinal feedback have been studied recently [22].

A related domain of work addresses the slightly different prophet inequality model [33, 34]. Here each applicant has a known probability distribution for its value. The realization becomes known only upon arrival, and arrival order is adversarial. In the basic model with a single job, there is an optimal stopping rule that yields a 2-approximation. This factor 2 can, in fact, be extended to arbitrary matroids, and a constant-factor guarantee exists even for polymatroids [14] or a intersection of a constant number of matroids [31]. Recently, there has been increased interest in this work, especially in the context of Bayesian mechanism design and posted-price mechanisms [1]. We also refer to [13] for current state-of-the-art approximation guarantees in this domain.

Our analysis of the algorithm for the general case applies in a unifying sampling model recently proposed as a framework for online maximum independent set in graphs [20]. It encompasses many stochastic adversarial models for online optimization – the secretary model, the prophet inequality model, and various other mixtures of stochastic and worst-case adversaries.

A preliminary version of this paper was published as an extended abstract in the proceedings of ICALP 2015 [8].

2 Preliminaries

In this section, we give a more detailed account of our basic model and introduce further concepts to adapt this model later on.

2.1 Basic Model

Our basic model consists of three components: the decentralized setting, random arrival order of worst-case weights and measuring the global performance of algorithms by the social welfare. We remark that while all of our results hold for this basic model, we generalize all upper bounds to more general settings and show our lower bound already for a specialization of this model.

Decentralized Setting. We are given a complete and weighted bipartite graph $G = (U, V, w)$, where $U = \{u_1, \ldots, u_m\}$ denotes the set of firms, $V = \{v_1, \ldots, v_n\}$ denotes the set of applicants and $w : U \times V \to \mathbb{R}^+$ denote the set of *weights* or *preferences*. There is a (possibly random) *arrival order* of the applicants; by a slight abuse of notation, we name the applicants such that $v_1, \ldots, v_n$ always denotes the sequence of arriving applicants.

An algorithm $\mathcal{A}$, when running for firm u_i, receives as input a sequence of weights $w(u_i, v_1), \ldots, w(u_i, v_n)$ and decides upon arrival of v_j whether or not u_i makes an offer to v_j. This decision must be based solely on the following information: (1) the weights of all $w(u_i, v_1), \ldots, w(u_i, v_j)$ as well as (2) for all previous applicants $v_\ell, \ell < j$ for which $\mathcal{A}$ has decided to make an offer, v_ℓ's decision of whether or not v_ℓ accepts the offer.

Let $\boldsymbol{\mathcal{A}} := (\mathcal{A}_i)_{i \in [m]}$ be a collection of algorithms. We denote by $M^{\boldsymbol{\mathcal{A}}}$ the allocation returned by the following process: For each $j = 1, \ldots, n$, the applicant v_j receives the set $S \subset [m]$ of offers, i.e., the set of all $i \in [m]$ such that u_i makes an offer to v_j, where each firm $u_i, i \in [m]$ runs its corresponding algorithm $\mathcal{A}_i$. Among this set, v_j accepts the offer maximizing v_j's preference, i.e., v_j accepts the offer of $\text{argmax}_{i \in S} w(u_i, v_j)$. Then $M^{\boldsymbol{\mathcal{A}}}$ is the set of all pairs (u_i, v_j) such that v_j accepts the offer of u_i. Observe that by this definition, $M^{\boldsymbol{\mathcal{A}}}$ is always a matching in G.

The weight $w(M)$ of any matching M in G is defined as the total weight $\sum_{(u_i, v_j) \in M} w(u_i, v_j)$ of all matched firm-applicant pairs. The social welfare of the collection of algorithms $\boldsymbol{\mathcal{A}}$ is the weight $w(M^{\boldsymbol{\mathcal{A}}})$ of the allocation obtained by the firms running $\boldsymbol{\mathcal{A}}$.

Arrival Order and Preference Generation. We consider different possibilities to specify the preferences between applicants and firms. In our basic model, an adversary specifies worst-case weights $w(u_i, v_j)$. In more specialized settings, we let the preferences be randomly generated, e.g., (1) each applicant v_j draws an independent applicant weight $w(v_j)$ from an applicant weight distribution D_j and is assigned the same preference $w(u_i, v_j) = w(v_j)$ for all firms u_i (independence among applicants), or (2) for each applicant v_j, the weights $\{w(u_i, v_j)\}_{i \in [m]}$ could be drawn independently from potentially different distributions for each firm (independence among firms). These alternatives are studied in Sections 3.2 and Sections 4, respectively.

As a realistic input assumption, our basic model assumes the classical *secretary model*: the applicants arrive in a uniform random order.

Competitive Ratio. For a collection of algorithms $\boldsymbol{\mathcal{A}} := (\mathcal{A}_i)_{i \in [m]}$, we say that $\boldsymbol{\mathcal{A}}$ has a *competitive ratio* of α if for all instances, we have $w(M^*)/\mathbb{E}\left[w(M^{\boldsymbol{\mathcal{A}}})\right] \leq \alpha$, where the expectation is taken over the internal randomness of all algorithms $\mathcal{A}_i$ and the random arrival order. In Sections 3.2 and 4, where we consider instances with randomly generated weights, we extend this notion canonically: in this case, the criterion changes to $\mathbb{E}[w(M^*)]/\mathbb{E}[w(M^{\boldsymbol{\mathcal{A}}})] \leq \alpha$, where the expectations are taken over the randomly generated weights in addition to the algorithms' internal randomness and the random arrival order. We make this more formal in the corresponding sections.

2.2 Further Concepts and Conventions

Let $[n] := \{1, \ldots, n\}$. Generally, there are two types of randomness occurring in our analysis: Internal random bits of an algorithm $\mathcal{A}$, and randomness inherent to the input instance I (which includes the random arrival order of applicants and the possibly random weight generation process). Whenever we intend to stress the distinction, we write $\mathbb{E}_{\mathcal{A}}[\cdot]$ and $\mathbb{E}_I[\cdot]$ to denote the expectation over $\mathcal{A}$'s internal random bits and the random generation process of I, respectively. Similarly, when arguing over a distribution $\mathcal{I}$ over instances I, we write $\mathbb{E}_{\mathcal{I}}[\cdot] = \mathbb{E}_{I \leftarrow \mathcal{I}}[\mathbb{E}_I[\cdot]]$ to take the expectation over first sampling an instance I and then generating weights and an ordering of the applicants according to I.

In Section 3, we use the notion of matroids to generalize our basic model, which enables us to strengthen our results. A matroid $\mathcal{S} = (E, \mathcal{I})$ consists of a non-empty ground set E and a non-empty family of *independent sets* $I \subseteq E$ satisfying the following two properties: (1) for any $B \in \mathcal{I}$ and $A \subseteq B$, we have $A \in \mathcal{I}$ and (2) for any $A, B \in \mathcal{I}$, if $|A| < |B|$, there exists an element $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$.

3 General Preferences

For general weights $w : U \times V \to \mathbb{R}^+$, Proposition 1 shows that the classic secretary algorithm may perform poorly in a decentralized market. A reasonable strategy has to be more careful in adopting a threshold to avoid extensive competition over a few applicants. We overcome this obstacle with a randomized thresholding strategy similar to [4], and we analyze it in a very general distributed matroid scenario. In Appendix B, we show that our bounds apply even within a general sampling model [20] that encompasses the secretary model, the prophet-inequality model, and many other approaches for stochastic online optimization.

For the combinatorial structure of the scenario, we consider the case that each firm u_i holds a possibly different matroid $\mathcal{S}_i$ over the set of applicants. In this setting, firm u_i may accept an applicant as long as the set of accepted applicants forms an independent set in $\mathcal{S}_i$. Special cases include hiring a single applicant or any subset of at most k_i many applicants. As the canonical generalization of the objective in the basic model, each firm now strives to maximize the sum of the weights of hired applicants. In our algorithms, the structure of $\mathcal{S}_i$ does not have to be known in advance. It suffices that firm u_i has an oracle to test if a set of arrived applicants is an independent set in $\mathcal{S}_i$.

As a simple baseline, we can trivially obtain the following guarantee. Suppose there is an α-competitive algorithm $\mathcal{A}$' for a single firm. We assume that every firm u_i executes $\mathcal{A}$' in exactly the same way as if it was the only firm in the market. In particular, if u_i would actually be alone in the market and $\mathcal{A}$' would make an offer to an applicant v_j, then v_j will accept it. Based on this, $\mathcal{A}$' possibly makes subsequent offers to another applicant $v_{j'}$ later on. Since u_i is typically not alone in the market, the offer by $\mathcal{A}$' to v_j could be turned down. This would have an impact on the subsequent offers that $\mathcal{A}$' makes to $v_{j'}$ (and other applicants). However, for the following guarantee we assume this is not the case – for the decision about whether to offer a position to an applicatn, $\mathcal{A}$' pretends that all previous offers were accepted (even though, in reality, some applicants might actually have turned u_i down). This serves to ensure that throughout each firm viewing its applicant values acts exactly as if it would be alone in the market.

Proposition 2. *Let algorithm $\mathcal{A}$' be any α-competitive algorithm for a single firm. Suppose every firm u_i runs a version $\mathcal{A}$ that pretends every applicant getting an offer from u_i also accepts it. Then algorithm $\mathcal{A}$ is $m\alpha$-competitive.*

Proof. For each firm u_i, consider the individual optimum M_i^* in hindsight. Clearly, there is one firm $u_{i'}$ for which this individual optimum has $w(M_{i'}^*) \geq w(M^*)/m$. Using $\mathcal{A}$', $u_{i'}$ makes offers to a set of applicants that constitute an α-approximation to the individual optimum. If an applicant decides against the offer of $u_{i'}$, it accepts a better offer from a different firm, so it secures an even larger weight in the solution $M^{\mathcal{A}}$. Hence, $\mathbb{E}[w(M^{\mathcal{A}})] \geq w(M_{i'}^*)/\alpha \geq w(M^*)/(m\alpha)$. □

For general matroids it implies a competitive ratio of $O(m \log\log k_{\max})$ using the currently best algorithm [16, 35], where $k_{\max}$ is the maximum rank of any of the matroids $\mathcal{S}_i$. In the following section, we describe an algorithm that significantly improves upon this trivial guarantee when m grows large.

Algorithm 2: Thresholding algorithm for u_i with matroids.

1 $k \leftarrow Binom(n, 1/2)$
2 Reject the first k applicants, denote this set by V_i^S
3 $m_i \leftarrow \arg\max_{v_j \in V_i^S} w(u_i, v_j)$
4 $X_i \leftarrow \text{Uniform}(0, 1, \ldots, \lceil \log_2 b \rceil + 1)$, where $b \geq |M^*|$
5 $t_i \leftarrow w(u_i, m_i)/2^{X_i}$, $M_i \leftarrow \emptyset$
6 **for** *all remaining* v_j *over time* **do**
7   **if** $w(u_i, v_j) \geq t_i$ *and* $M_i \cup \{v_j\}$ *is independent set in* $\mathcal{S}_i$ **then**
8   make an offer to v_j
9   **if** v_j *accepts* **then** $M_i \leftarrow M_i \cup \{v_j\}$

3.1 Logarithmic Approximation

Algorithm 2 is executed in parallel by all firms u_i. We first sample a fraction of $k \leftarrow Binom(n, 1/2)$ applicants and determine a random threshold based on the maximum weight seen by firm u_i in its sample. Firm u_i then greedily makes an offer only to those applicants whose values are above the threshold. This idea and extensions have been successful for approximating combinatorial secretary problems with a single firm [4, 16, 32, 35].

In line 4, the algorithm relies on an upper bound $b \geq |M^*|$. A simple example is $b = n$, which is always known and results in an $O(\log n)$-competitive algorithm. In case there is additional knowledge about the cardinality of the optimum solution, the guarantee can be improved. For example, if all firms know the number of firms m and the maximum rank $k_{\max}$ of the matroids, then with $b = mk_{\max}$ the algorithm is $O(\log m + \log k_{max})$-competitive. In particular, if all firms know m in the basic model, the algorithm is $O(\log m)$-competitive.

Theorem 1. *Algorithm 2 is* $32(\lceil \log_2 b \rceil + 2)$*-competitive.*

Proof. We denote by V_i^S the set of applicants in the sample and by V_i^I the other applicants. To ease the analysis, we simulate the algorithm as follows: first, we assign every v_j independently and uniformly to V_i^S or V_i^I, then compute t_i, and finally consider applicants from V_i^I in random order[1]. To see that this is correct, one uses the principle of deferred decisions: Fix any set $S \subseteq V$. When running Algorithm 2, we first use the internal randomness of (a) choosing k and (b) determining the prefix of the first $|S|$ elements of the random arrival order to obtain $\Pr[V_i^S = S] = \Pr[k = |S|] \cdot \Pr[v_1, \ldots, v_{|S|} \text{ is a permutation of } S] = \binom{n}{|S|} 2^{-n} \cdot \binom{n}{|S|}^{-1} = 2^{-n}$. Then, the permutation of the remaining $n - |S|$ elements in the random arrival order is chosen independently and uniformly at random. Thus, V_i^S is chosen uniformly among all subsets of applicants, and V_i^I is a random permutation of the remaining applicants. This is equivalent to the above described generation process. Note that in particular, the event $v_j \in V_i^S$ is independent of $v_{j'} \in V_i^S$ for all other applicants $j' \neq j$, and occurs with probability $\mathbf{Pr}\left[v_j \in V_i^S\right] = \mathbf{Pr}\left[v_j \in V_i^I\right] = 1/2$.

Let $v_i^{\max} = \text{argmax}_{v_j} w(u_i, v_j)$ and $v_i^{\text{2nd}} = \text{argmax}_{v_j \neq v_i^{\max}} w(u_i, v_j)$ be the best and second best applicant for firm u_i, respectively. In addition, we denote by $w_i^{\max} = w(u_i, v_i^{\max})$ and $w_i^{\text{2nd}} = w(u_i, v_i^{\text{2nd}})$ their weights for firm u_i. For most of the analysis, we consider another weight function, the *capped weights* $\tilde{w}(u_i, v_j)$, based on the thresholds t_i set by the algorithm. Intuitively, the capped weights give a sufficiently good lower bound on the actual weights while at the same time, from firm u_i's perspective, equalizing the weights of many applicants. Roughly speaking, by arguing that our algorithm provides u_i with a large-*cardinality* independent set among a subset of applicants of equal capped weights, we can then bound the solution quality in terms of the optimal solution.

[1] Such a simulation is used for the analysis of secretary algorithms in, e.g., [20, 32].

Formally, we define the capped weights as follows

$$\tilde{w}(u_i, v_j) = \begin{cases} w_i^{\max} & \text{if } v_j \in V_i^I, t_i = w_i^{\text{2nd}}, \text{ and } w(u_i, v_j) > w_i^{\text{2nd}}, \\ t_i & \text{else, if } v_j \in V_i^I \text{ and } w(u_i, v_j) \geq t_i, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that the definition of $\tilde{w}$ relies on several random events, namely, $v_j \in V_i^I$ and the choice of the thresholds t_i. For any outcome of these events, however, we have that $\tilde{w}(u_i, v_j) \leq w(u_i, v_j)$ for all pairs (u_i, v_j), since if $t_i = w_i^{\text{2nd}}$ and $w(u_i, v_j) > w_i^{\text{2nd}}$, then $v_j = v_i^{\max}$ (recall that we assume no ties in w).

By the following lemma, in expectation over all the correlated random events, an optimal offline solution with respect to $\tilde{w}$ still gives an approximation to the optimal offline solution with respect to w.

Lemma 1. *Denote by $w(M)$ and $\tilde{w}(M)$ the weight and capped weight of a solution M, respectively. Let $\tilde{M}^*$ and M^* be optimal solutions for $\tilde{w}$ and w, respectively. Then,*

$$\mathbb{E}\left[\tilde{w}(\tilde{M}^*)\right] \quad \geq \quad \frac{1}{16(\lceil \log_2(b) \rceil + 2)} \cdot w(M^*).$$

Proof. Let $(u_i, v_j) \in M^*$ be an arbitrary pair. First, assume that v_j maximizes $w(u_i, v_j)$, i.e., $v_j = v_i^{\max}$. With probability at least $1/4$, we have $v_j \in V_i^I$ and $v_i^{\text{2nd}} \in V_i^S$. For any such outcome, we have with probability $1/(\lceil \log_2(b) \rceil + 2)$ that $t_i = w_i^{\text{2nd}}$ and $\tilde{w}(u_i, v_j) = w_i^{\max}$. This yields $\mathbb{E}[\tilde{w}(u_i, v_j)] \geq w(u_i, v_j)/(4(\lceil \log_2(b) \rceil + 2))$.

Second, for any $v_j \neq v_i^{\max}$ with $w(u_i, v_j) > w_i^{\max}/(2b)$, we know that $v_j \in V_i^I$ and $v_i^{\max} \in V_i^S$ with probability $1/4$. Furthermore, there is some $1 \leq k' \leq \lceil \log_2(b) \rceil + 1$, with $w(u_i, v_j) > w_i^{\max}/2^{k'} \geq w(u_i, v_j)/2$. Under $v_i^{\max} \in V_i^S$, we have with probability $1/(\lceil \log_2(b) \rceil + 2)$ that $X_i = k'$ and $\tilde{w}(u_i, v_j) = t_i \geq w(u_i, v_j)/2$. Thus, with probability $1/(4(\lceil \log_2(b) \rceil + 2))$, we have $v_j \in V_i^I$ and $\tilde{w}(u_i, v_j) \geq w(u_i, v_j)/2$, which yields $\mathbb{E}[\tilde{w}(u_i, v_j)] \geq w(u_i, v_j)/(8(\lceil \log_2(b) \rceil + 2))$.

Finally, we denote by $M^>$ the set of pairs $(u_i, v_j) \in M^*$ for which $w(u_i, v_j) > w_i^{\max}/(2b)$. The expected weight of the best assignment with respect to the capped weights is thus

$$\begin{aligned} \mathbb{E}\left[\tilde{w}(\tilde{M}^*)\right] \quad &\geq \quad \sum_{(u_i, v_j) \in M^*} \mathbb{E}[\tilde{w}(u_i, v_j)] \quad \geq \quad \sum_{(u_i, v_j) \in M^>} \frac{w(u_i, v_j)}{8(\lceil \log_2(b) \rceil + 2)} \\ &= \quad \frac{1}{8(\lceil \log_2(b) \rceil + 2)} \cdot (w(M^*) - w(M^* \setminus M^>)) \\ &\geq \quad \frac{1}{16(\lceil \log_2(b) \rceil + 2)} \cdot w(M^*), \end{aligned}$$

where the last inequality results from $\sum_{(u_i, v_j) \in M^* \setminus M^>} w_i^{\max}/(2b) \leq \max_i w_i^{\max}/2 \leq w(M^*)/2$. □

The previous lemma bounds the weight loss due to using the capped weights. In particular, by losing only a logarithmic factor, we arrive in a setting where *all applicants accepting an offer from u_i contribute the same value to the social welfare*. The next lemma bounds the remaining loss due to random arrival of elements in V_i^I. Crucially, we relate the result of the algorithm *under the actual weights* to the optimal solution *under the capped weights*.

Lemma 2. *Suppose subsets V_i^I and thresholds t_i are fixed arbitrarily and consider the resulting weight function $\tilde{w}$. Let $M^{\mathcal{A}}$ be the feasible solution resulting from Algorithm 2 using the thresholds t_i, for any arbitrary arrival order of applicants in $\bigcup V_i^I$. Then $w(M^{\mathcal{A}}) \geq \tilde{w}(\tilde{M}^*)/2$.*

Proof. We show $\tilde{w}(\tilde{M}^*) \leq 2 \cdot w(M^{\mathcal{A}})$ by the following accounting scheme: We charge the weight of each edge $(u_i, v_j) \in \tilde{M}^*$ under $\tilde{w}$ to the original weight $w(u_{i'}, v_{j'})$ of a pair $(u_{i'}, v_{j'}) \in M^{\mathcal{A}}$, using each pair in $M^{\mathcal{A}}$ at most twice. W.l.o.g., we may assume $\tilde{w}(u_i, v_j) > 0$ for all $(u_i, v_j) \in \tilde{M}^*$, which implies that $w(u_i, v_j) \geq t_i$. Consider any firm u_i. We classify all applicants v_j receiving an offer from u_i by defining the condition

$$v_j \text{ gets a better offer from another firm } u_{i'} \neq u_i \text{ with } w(u_{i'}, v_j) > w(u_i, v_j). \qquad (*)$$

First, consider the set of all edges $(u_i, v_j) \in \tilde{M}^*$ for firm u_i, for which (*) holds. Then, each such v_j is assigned to some other firm $u_{i'}$ that makes a better offer. Hence, $(u_{i'}, v_j) \in M^{\mathcal{A}}$ with $w(u_{i'}, v_j) \geq w(u_i, v_j) \geq \tilde{w}(u_i, v_j)$, and we can charge (u_i, v_j) to $(u_{i'}, v_j)$.

Now consider the set of all edges $(u_i, v_j) \in \tilde{M}^*$ for firm u_i, for which (*) does not hold. We denote their number by k_i. More generally, consider the superset $E_i^{t_i}$ of all edges (u_i, v_j) (in or not in M^*) with $w(u_i, v_j) \geq t_i$ such that (*) does not hold. These are all edges with $\tilde{w}(u_i, v_j) > 0$ that firm u_i can obtain *uncontested by other firms* – we want to bound the value of the independent set in $\mathcal{S}_i$ that u_i chooses among them. To this end, consider a maximum-cardinality independent set in matroid $\mathcal{S}_i$ that uses only edges from $E_i^{t_i}$. Denote this maximum cardinality by ℓ_i. Obviously, $\ell_i \geq k_i$, since $\tilde{M}^*$ is feasible for firm i. Moreover, firm u_i accepts in $M^{\mathcal{A}}$ at least ℓ_i applicants, irrespective of the arrival order. This is a simple consequence of the exchange property of matroid $\mathcal{S}_i$. Thus, for each of the k_i edges $(u_i, v_j) \in \tilde{M}^*$, for which (*) does not hold, there is some edge $(u_i, v_{j'}) \in M^{\mathcal{A}}$ with $w(u_i, v_{j'}) \geq \tilde{w}(u_i, v_{j'}) = \tilde{w}(u_i, v_j)$ to which we can charge it. This step can be done in such a way that it defines an *injective* mapping f_i that maps j to j' iff. $(u_i, v_j) \in \tilde{M}^* \cap E_i^{t_i}$ is charged to $(u_i, v_{j'}) \in M^{\mathcal{A}}$.

Finally, consider an arbitrary edge $(u_{i'}, v_{j'}) \in M^{\mathcal{A}}$. In the above accounting scheme, this edge can only be charged by edge $(u_i, v_j) \in \tilde{M}^*$ if either $v_j = v_{j'}$, in which case i is uniquely defined since j can only be matched to one firm, or $u_i = u_{i'}$, in which case j is uniquely defined since the mapping f_i constructed above is injective. Thus, $(u_{i'}, v_{j'})$ is used at most twice. This proves $\tilde{w}(\tilde{M}^*) \leq 2 \cdot w(M^{\mathcal{A}})$. □

Combining the insights of Lemmas 1 and 2, we see that that $w(M^*) \leq 32(\lceil \log_2 b \rceil + 2) \cdot \mathbb{E}\left[w(M^{\mathcal{A}})\right]$, which proves the theorem. □

In Appendix B we extend the result of this theorem to a general sampling model that includes the secretary model, the prophet-inequality model and other models that combine stochastic and worst-case adversarial elements.

3.2 Lower Bounds

Our general upper bound results from a *thresholding-based* algorithm. We contrast this result with a lower bound for thresholding-based algorithms in the basic model (in which every firm wants to hire only a single applicant). This shows that among all thresholding-based approaches, Algorithm 2 is close to optimal. Furthermore, this demonstrates a major obstacle to obtain significantly sublogarithmic competitive ratios, as the lower bound carefully constructs a challenge to determine information about the number of "relevant" firms. Learning such information in our decentralized settings with restricted feedback appears hardly possible: A firm only obtains information about the presence of other firms if it makes an offer to an applicant who rejects in favor of another firm.

Formally, an algorithm $\mathcal{A}$ is called *thresholding-based* if during its execution $\mathcal{A}$ rejects applicants for some number of rounds τ, then determines a threshold T and afterwards enters an *acceptance phase*. In the acceptance phase, it makes an offer to exactly those applicants whose weight exceeds threshold T. Note that the number of rejecting rounds τ in the beginning and the threshold T can be chosen arbitrarily at random on the basis of the observed information. More formally, when running $\mathcal{A}$ for firm u we require that (1) τ is a stopping time – also called Markov time – with respect to the sequence $w(u, v_1), w(u, v_2), \ldots$, and (2) the threshold T is a random variable depending only on $w(u, v_1), \ldots w(u, v_\tau)$.

In Section 3.2.1, we consider the case in which the number of firms is unknown to an algorithm. Here, we show a lower bound even for a class of *identical-firm* instances, in which for each applicant v_j all firms have the same weight, i.e., there is $w(v_j) \geq 0$ such that $w(u_i, v_j) = w(v_j)$ for every firm u_i. The main challenge here is to guess the right number of firms m in order to concentrate on the most profitable class of applicants.

Note that in this class of instances we always have $m \leq n$. Still, we manage to show a lower bound that depends on n, since m remains unknown and the best upper bound on $|M^*|$ that is known to the firms is n.

In Section 3.2.2, we consider the alternative case in which the number of firms present is public knowledge. More formally, for any fixed m, we consider an arbitrary collection of thresholding-based algorithms for these m firms and bound the competetive ratio of the allocation obtained by this collection. To do so, we adjust the instance for oblivious algorithms to contain an unknown number of firms that contribute only negligibly to the social welfare. In fact, here we ensure that our instance satisfies $m = n$, and hence the lower bound can be expressed as $\Omega(\log m/(\log\log m))$. This almost matches our upper bound in the basic model for the case that firms know m. More generally, one can always add more dummy applicants with weight 0 for all firms. This shows that the lower bound in m continues to hold even more generally when $m \leq n$ for the case that m is known to the firms.

3.2.1 Oblivious Algorithms

For algorithms oblivious to m, we prove the lower bound in the (identical-firm) *iid model*, which is a special case of the prophet-inequality model, and effectively can be interpreted a special case of the secretary model. Here, we consider instances restricted to identical firms: each applicants has a global weight $w(u_i)$ and all firms u_i have equal preference $w(u_i, v_j) = w(v_j)$ to this applicant. In the identical-firm iid model, we draw the weight $w(v_j)$ for each v_j independently at random from a single distribution D. Since each applicant value is drawn independently from the same distribution, a random arrival order is equivalent to an adversarial arrival order. Note that an instance is completely specified by the number of applicants n, the applicants' weight distribution D and the number of firms m.

As usual, since the optimal allocation M^* becomes a random variable, we relate the expected quality of the algorithm's allocation to the *expected* optimum: An algorithm $\mathcal{A}$ has competitive ratio α, if for all instances $I = (n, m, D)$, we have $\mathbb{E}[w(M^*)]/\mathbb{E}[w(M^{\mathcal{A}})] \leq \alpha$, where M^* is the optimal offline solution, $M^{\mathcal{A}}$ is the allocation returned when all firms run algorithm $\mathcal{A}$ and the expectation is taken over the randomly generated preferences and $\mathcal{A}$'s internal randomness.

We give a lower bound of $\Omega(\log n/(\log\log n))$ using an idea similar to the one underlying the proof of Proposition 1. Recall that in this proof, there are two classes of 'good' and 'bad' applicants. Instead, here we construct an instance with logarithmically many classes of applicants. The weights of applicants in larger classes is decreasing, but the total number of applicants in these classes is increasing. In total, the applicants in larger classes can generate more welfare – if there are enough firms to hire them. However, if the number of firms is too small, then offering to applicants from higher classes with lower weights is a bad strategy. These applicants generate small total weight, and since their numbers are significantly larger, the probability to hire an applicant from a lower class with high weight is very small. Thus, guessing the right order of magnitude of m represents the inherent difficulty on which the lower bound is based.

Theorem 2. *Let $\mathcal{A}$ be a thresholding-based algorithm oblivious to the number of firms m. Then there is an infinite family of instances $I = (n, m, D)$ in the identical-firm iid model in which $m \leq n$ and the allocation obtained by every firm running algorithm $\mathcal{A}$ has a competitive ratio of $\Omega(\log n/(\log\log n))$. This lower bound also holds in the secretary model.*

Proof. For every $t \in \mathbb{N}, t \geq 3$, we construct an instance $I = (n, m, D)$ on $n = \sum_{j=1}^{t} t^{2j} = (t^{2t}-1)\cdot t^2/(t^2-1) = (1 + o(1)) \cdot t^{2t}$ applicants. Since $t^{2t} < n < t^{2t}(1 + o(1))$, this yields $t = \Theta(\log n/(\log\log n))$. To define the distribution D over the applicants' weights $w(v_j)$, we set $\Pr_{w\leftarrow D}[w = t^{-j}] = t^{2j}/n$ for $j = 1, \ldots, t$. Observe that by definition of n, this indeed yields a probability distribution.

The crucial part is to define the number of firms m. To obtain the desired number via an averaging argument, we analyze the threshold-setting behavior of $\mathcal{A}$ – note that $\mathcal{A}$'s threshold is not influenced by other firms, as $\mathcal{A}$ makes no offers (and learns no feedback about other firms) before determining a threshold. In particular, consider the threshold that $\mathcal{A}$ sets when presented with n applicants with weights drawn iid from D. We may assume without loss of generality that it chooses a threshold among $\{t^{-1}, \ldots, t^{-t}\}$, since all other choices are equivalent to one of these with regard to the set of applicants receiving an offer. Let p_j be the probability (over the randomness of the applicants' weights and the random choices of $\mathcal{A}$) that

$\mathcal{A}$ picks threshold t^{-j}. Clearly, there is some $1 \le k \le t$ with $p_k \le 1/t$. Setting the number of firms to $m = \sum_{j=1}^{k} t^{2j} < (9/8)t^{2k}$ finally concludes the definition of the instance I. Intuitively, to obtain a competitive solution in this instance, most firms should choose a threshold of t^{-k}, yet by our choice of k, few firms do so (in expectation).

We now analyze the social welfare obtained when all firms run $\mathcal{A}$ in the above constructed instance. Let F_j be the number of firms with threshold t^{-j}. Note that while the thresholds chosen by different firms may be correlated, we have $\mathbb{E}[F_j] = mp_j$ by linearity of expectation. Thus, $\mathbb{E}[F_k] \le m/t$ by choice of k, which is the main property we exploit below.

We define classes $C_1, \dots, C_t$, where class C_j consists of all applicants $v_{j'}$ with value $w(v_{j'}) = t^{-j}$. Let us furthermore denote by T_i the threshold chosen by firm u_i. We categorize the contribution of each firm-applicant pair (u_i, v_j) (of the algorithm's allocation) to the social welfare into one of three types: (T1) u_i has a threshold of $T_i \ge t^{-k}$ and v_j belongs to an applicant class $C_1, \dots, C_{k-1}$, (T2) u_i has the threshold $T_i = t^{-k}$ and v_j belongs to applicant class C_k and (T3) u_i has a threshold of $T_i \le t^{-(k+1)}$. Observe that each matched pair corresponds to exactly one category, since each firm u_i with threshold $T_i = t^{-k'}$ only accepts candidates among $C_1, \dots, C_{k'}$.

Consider the contribution of all pairs of type (T1): trivially, the total contribution is at most $\sum_{j=1}^{k-1} |C_j| t^{-j}$, since each applicant in $C_1, \dots, C_{k-1}$ can be matched at most once. To bound the expectation of this term, consider iid random variables $Y_{j'}$ that take the value of the j'-th candidate if it falls into classes $C_1, \dots, C_{k-1}$ and 0 otherwise. Formally, $Y_{j'} \in [0,1]$ are iid with $\Pr[Y_{j'} = t^{-j}] = t^{2j}/n$ for $j = 1, \dots, k-1$ and $\Pr[Y_{j'} = t^{-j}] = 0$ for $j = k, \dots, t$. Then, since $t \ge 3$,

$$\mathbb{E}\left[\sum_{j=1}^{k-1} |C_j| t^{-j}\right] = \mathbb{E}\left[\sum_{j'=1}^{n} Y_{j'}\right] = \sum_{j=1}^{k-1} t^j < (3/2)t^{k-1} \ . \tag{1}$$

We turn to the type-(T2) pairs. Their total contribution is bounded by $F_k \cdot t^{-k}$, since only firms with threshold t^{-k} can participate in such a pair (contributing a value of t^{-k} in this case) and each such firm can be matched to at most one applicant. By the choice of k, we have

$$\mathbb{E}[F_k]\, t^{-k} \le m/(t^{k+1}) < (9/8)t^{k-1}. \tag{2}$$

Finally for type (T3), consider any firm with threshold $T_i = t^{-k'}$ with $k' \ge k+1$. Let W_i be the value of an applicant matched to firm u_i (or zero, if u_i remains unmatched). We argue that W_i is stochastically dominated by the random variable $w_{\text{cond}} \leftarrow D$ *conditioned on* $w_{\text{cond}} \in \{t^{-1}, \dots, t^{-k'}\}$, using the principle of deferred decisions: to condition on u_i having threshold $t^{-k'}$, we first only need to reveal the value of all applicants in the sample. For all remaining applicants, we draw their weight at the moment they arrive. Whenever an applicant v_j arrives, we draw its value $w(v_j)$ from D, resulting either in $w(v_j) < t^{-k'}$ in which case u_i makes no offer, or $w(v_j) \in \{t^{-1}, \dots, t^{-k'}\}$ and u_i makes an offer. If an offer is made and v_j accepts it, then the value contributed by (u_i, v_j) is distributed exactly as w_{cond}, as desired. Otherwise, no offer is made or an offer is made but declined by v_j in favor of another firm, and we turn to the next applicant (if available). This process terminates either with an applicant matched to u_i, contributing a value distributed as w_{cond}, or with no applicant matched to u_i, yielding a contribution of $0 \le w_{\text{cond}}$, yielding the claim.

We define $S = \sum_{j=1}^{k'} t^{2j}$. Since the expectation of W_i is bounded from above by the expectation of w_{cond}, we obtain

$$\mathbb{E}[W_i] \le \sum_{j=1}^{k'} \frac{t^{2j}}{S} \cdot t^{-j} < \frac{3t^{k'}}{2t^{2k'}} \le (3/2)t^{-(k+1)}. \tag{3}$$

Using (1), (2) and (3), we can thus bound the expected contribution of all types to the social welfare by

$$\mathbb{E}_{I,\mathcal{A}}[w(M^{\mathcal{A}})] \quad \le \quad \mathbb{E}\left[\sum_{j=2}^{k-1} |C_j| t^{-j}\right] + \mathbb{E}[F_k]\, t^{-k} + \mathbb{E}\left[\sum_{i=1}^{m} W_i\right]$$

$$
\begin{aligned}
&\leq \quad (3/2)t^{k-1} + (9/8)t^{k-1} + (3/2)mt^{-(k+1)} \\
&< \quad (3/2)t^{k-1} + (9/8)t^{k-1} + (27/16)t^{k-1} \\
&= \quad (69/16)t^{k-1}.
\end{aligned}
$$

It remains to give a lower bound of $\Omega(t^k)$ on the expected weight of the optimal solution. To this end, we bound $|C_k|$ using a standard Chernoff bound. We have, over I,

$$\Pr[|C_k| < (1/2)t^{2k}] \quad \leq \quad \exp(-t^{2k}/8) \ .$$

Hence, with probability at least $(1 - \exp(-t^{2k}/8))$ there exists a matching that includes at least $t^{2k}/2$ applicants of class C_k. Since $t \geq 3$ and $k \geq 1$, the optimum solution M^* has expected value at least

$$\mathbb{E}_I[w(M^*)] \quad \geq \quad t^{-k} \cdot \frac{t^{2k}}{2} \cdot \left(1 - e^{-\frac{t^{2k}}{8}}\right) \quad > \quad t^k/3 \ .$$

Recall that $t = \Omega(\log n / \log\log n)$. Thus, in the iid model, the ratio of expectations is at least

$$\frac{\mathbb{E}_I[w(M^*)]}{\mathbb{E}_{I,\mathcal{A}}[w(M^{\mathcal{A}})]} \ > \ \frac{(1/3)t^k}{(69/16)t^{k-1}} \ > \ t/13 \ = \ \Omega\left(\frac{\log n}{\log\log n}\right).$$

Note that we did not optimize any constants, since we only intend to show that the asymptotics do not hide large constants. In fact, since the constants result from bounding exponential series and tail bounds, they can be expressed as functions of t that approach 1 quickly as t grows large.

Finally, let us connect the result to the secretary model: observe that $\mathbb{E}_I[w(M^*)]$ is the average of $w(M^*)$ over all randomly generated random weights, while $\mathbb{E}_{I,\mathcal{A}}[w(M^{\mathcal{A}})]$ is the average over $\mathbb{E}_{\mathcal{A}}[w(M^{\mathcal{A}})]$ (where the randomness is taken over $\mathcal{A}$'s internal randomness and the uniform random arrival order) weighted over all randomly generated weights. As a simple consequence of standard calculus there must be a choice of weights with competitive ratio

$$\frac{w(M^*)}{\mathbb{E}_{\mathcal{A}}[w(M^{\mathcal{A}})]} \ \geq \ \frac{\mathbb{E}_{\mathcal{I}}[w(M^*)]}{\mathbb{E}_{\mathcal{I},\mathcal{A}}[w(M^{\mathcal{A}})]} \ = \ \Omega\left(\frac{\log n}{\log\log n}\right).$$

□

3.2.2 Algorithms with Knowledge of m

We extend the previous result to the case in which each firm knows m. In this case, we do not give an identical-firm instance, but introduce a second category of firms that we call *non-valuable*. Formally, we use the following, slightly more general *2-category iid model*: Partition the set of firms U into two disjoint subsets U_1 and U_2. Analogously to the identical-firm case, there are two distributions D_1 and D_2 over applicant weights. For every applicant v_j, we draw his applicant values $w_\ell(v_j) \leftarrow D_\ell, \ell \in \{1, 2\}$ independently and set $w(u_i, v_j) = w_\ell(u_i, v_j)$ for each $u_i \in U_\ell, \ell \in \{1, 2\}$. Thus, any instance in the 2-category iid model is completely specified as $I = (n, U_1, U_2, D_1, D_2)$.

Consider any number of firms and fix, for each firm u_i, a thresholding-based algorithm $\mathcal{A}_i$. We aim to show that already in the 2-category iid model, the collection of algorithms $\boldsymbol{\mathcal{A}} := (\mathcal{A}_i)_{i\in[m]}$ is only $\Omega(\log m/(\log\log m))$-competitive. Note that we call the collection $(\mathcal{A}_i)_{i\in[m]}$ α-competitive, if for all instances, the competitive ratio $\mathbb{E}[w(M^*)]/\mathbb{E}[w(M^{\boldsymbol{\mathcal{A}}})]$ is bounded by α, where M^* is the optimal offline solution and $M^{\boldsymbol{\mathcal{A}}}$ is the allocation returned when each firm u_i runs its corresponding thresholding-based algorithm $\mathcal{A}_i$. For our lower bound, we may even allow the firms to use shared random bits.

We first sketch the difference to the previous lower bound: Consider any number of firms m and let $\mathcal{A}_i$ be a thresholding-based algorithm for firm u_i. We create an instance with $n = m$ applicants. Here, the adversary picks a number m' as the number of *valuable* firms which depends on the set of algorithms $\boldsymbol{\mathcal{A}}$ used. For every non-valuable firm, every applicant value is multiplied with $\epsilon \ll t^{-t}$. Hence, the non-valuable firms contribute negligibly to the value of any matching. As before, m' is chosen such that the few valuable firms have to pick the desired threshold to obtain a good allocation. Thus, the lower bound continues to hold in this setting. Since $m = n$, this also implies a lower bound of $\Omega(\log m/(\log\log m))$.

The following theorem makes the above proof sketch formal.

Theorem 3. *There is some $c > 0$ such that for any sufficiently large m and any collection of thresholding-based algorithms $\boldsymbol{\mathcal{A}} := (\mathcal{A}_i)_{i\in[m]}$, there is an instance I in the 2-category iid model such that $\boldsymbol{\mathcal{A}}$ has a competitive ratio of at least $c \cdot \log m/(\log\log m)$. This lower bound also holds in the secretary model.*

Proof. We first construct a distribution over instances $I = (n, U_1, U_2, D_1, D_2)$ with $n = m = |U_1| + |U_2|$ as follows. Let t be the largest integer with $\sum_{j=1}^{t} t^{2j} \le m$. Since $t^{2t} < m < (t+1)^{2(t+1)}(1 + o(1))$, this yields $t = \Theta(\log m/(\log\log m))$. At the moment, we only define the valuable-firms distribution D_1 over the applicants' weights $w(v_j)$ by setting $\Pr_{w\leftarrow D_1}[w = t^{-j}] = t^{2j}/n$ for $j = 1, \ldots, t$.

Analogously to the proof of Theorem 2, consider the thresholds determined by the firms when presented with applicants with weights sampled from D_1. Without loss of generality, we may assume that each $\mathcal{A}_i$ picks a threshold among $\{t^{-1}, \ldots, t^{-t}\}$. Let X_{ij} be an indicator variable that equals 1 if and only if firm u_i picks threshold t^{-j}. Let $F_j = \sum_{i=1}^{m} X_{ij}$ be the number of firms picking threshold t^{-j}. Clearly, there is some $1 \le k \le t$ such that $\mathbb{E}[F_k] \le m/t$. This determines the number of *desired valuable firms* $m' = \sum_{j=1}^{k} t^{2j}$.

This allows us to define a distribution $\mathcal{I}$ over instances I as follows. We sample $U_1 \subseteq [m]$ uniformly at random from the set of all subsets of size m' of $[m]$. We obtain the distribution D_2 by scaling each value sampled from D_1 by a sufficiently small constant $\varepsilon > 0$, i.e., $\Pr_{w\leftarrow D_2}[w = \varepsilon w'] = \Pr_{w\leftarrow D_1}[w = w']$. It remains to show that $\mathbb{E}_{\mathcal{I}}[w(M^*)]/\mathbb{E}_{\mathcal{I},\boldsymbol{\mathcal{A}}}[w(M^{\boldsymbol{\mathcal{A}}})] \ge c \log m/(\log\log m)$ – it is straightforward to conclude then that there is an instance I in the support of $\mathcal{I}$ with $\mathbb{E}_I[w(M^*)]/\mathbb{E}_{I,\boldsymbol{\mathcal{A}}}[w(M^{\boldsymbol{\mathcal{A}}})] \ge c \log m/(\log\log m)$.

In particular, for every I in the support of $\mathcal{I}$, let $I_{\text{iid}} = (n, m', D_1)$ be the identical-firm iid instance obtained by ignoring all non-valuable firms. We have $\mathbb{E}_{I\leftarrow\mathcal{I}}[\mathbb{E}_{I_{\text{iid}}}[w(M^{\boldsymbol{\mathcal{A}}})]] \le (69/16)t^{k-1}$ by reworking the analysis of the proof of Theorem 2: The bounds on the contribution of all type-(T1) and type-(T3) applicant-firms pairs, given in (1) and (3), hold pointwise for all I_{iid} with I in the support of $\mathcal{I}$. For the bound (2) on all type-(T2) pairs, we no longer necessarily have $\mathbb{E}_{I_{\text{iid}}}[F_k] \le m/t$, however, $\mathbb{E}_{I\leftarrow\mathcal{I}}[\mathbb{E}_{I_{\text{iid}}}[F_k]] \le m/t$ still holds. Thus, by combining the bounds on the expected contribution (over I_{iid} with $I \leftarrow \mathcal{I}$) for all three types, we obtain the claim verbatim as in the proof of Theorem 2.

There is a negligible loss by ignoring all non-valuable firms: By choosing a sufficiently small $\varepsilon > 0$, we can ensure that each weight in the support of D_2 is strictly smaller than the weights in the support of D_1. Thus for every I in the support of $\mathcal{I}$, the contribution of all valuable firms to the social welfare is at most $\mathbb{E}_{I_{\text{iid}}}[w(M^{\boldsymbol{\mathcal{A}}})]$. Indeed, whenever a valuable firm makes an offer, the applicant never rejects the offer in favor of a non-valuable firm. Thus, the allocation of valuable firms is correctly simulated by the identical-firm iid instance I_{iid}. Furthermore, the total contribution of non-valuable firms to the social welfare is bounded by $m\varepsilon/t$, which can be made arbitrarily small by choosing ε sufficiently small.

Finally, we observe that the lower bound $\mathbb{E}_{\mathcal{I}}[w(M^*)] \ge \mathbb{E}_{I\leftarrow\mathcal{I}}[\mathbb{E}_{I_{\text{iid}}}[w(M^*)]] \ge t^k/3$ can be shown verbatim as in the proof of Theorem 2. Combining the arguments above, we obtain

$$\frac{\mathbb{E}_{\mathcal{I}}[w(M^*)]}{\mathbb{E}_{\mathcal{I},\boldsymbol{\mathcal{A}}}[w(M^{\boldsymbol{\mathcal{A}}})]} \;\ge\; \frac{(1/3)t^k}{(69/16)t^{k-1} + m\varepsilon/t} \;>\; t/13 \;=\; \Omega\left(\frac{\log m}{\log\log m}\right),$$

for sufficiently small $\varepsilon > 0$, yielding the desired claim for $m = n$. We sketch how to show that this lower bound in m also holds for $n \ge m$: Simply modify the distribution D_1 to $\Pr_{w\leftarrow D_1}[w = t^{-j}] = t^{2j}/n$ for $j = 1, \ldots, t$, where the remaining probability mass is assigned to $\Pr_{w\leftarrow D_1}[w = 0] = 1 - \sum_{j=1}^{t} t^{2j}/n$. It is straightforward to adapt the proof to this case.

Finally, the connection to the secretary model follows analogously to the proof of Theorem 2.

□

4 Independent Preferences

In this section, we show a constant competitive ratio for decentralized matching in the secretary model when the preferences are independent among firms. In particular, for each firm the preference over applicants can be adversarial, but the preference profiles for different firms are composed independently. This contrasts our lower bound from the previous section, where for each firm the preferences over applicants were iid, but the preference profiles for different firms were composed in an adversarial way.

Independent Preferences in the Basic Model. The basic model with independent preferences is a special case of the basic model (which is characterized by adversarial weights and uniform random arrival order). In this special case, the adversary specifies a separate set $\mathcal{P}_i$ of n *applicant values* for each firm u_i. In round t, when a new applicant v_t arrives, we pick one remaining value $p_{it} \in \mathcal{P}_i$ for each firm $u_i \in U$ independently and uniformly at random. The weight for firm u_i is then given by $w(u_i, v_t) = p_{it}$. We pick values from $\mathcal{P}_i$ uniformly at random without replacement.

Special cases of this model are, e.g., when all weights for firms are independently sampled from a certain distribution. In particular, let each firm u_i have some weight distribution D_i, and sample, upon arrival of v_t, all weights $w(u_i, v_t)$ independently from D_i; then this yields a special case of the basic model where for each applicant the preferences over each firm are independent.

Firms with Multiple Positions. We conduct the analysis in a more general domain when each firm has several positions. We generalize the basic model to a decentralized secretary matching model: Instead of maintaining a single position (or, as in Section 3.1, independent sets in a matroid $\mathcal{S}_i$), firms may have more than a single position available and each applicant has different qualifications, i.e., weights for each individual position. More formally, we assume firm u_i has a set U_i of k_i positions available. Whenever an applicant arrives, each firm must make an immediate decision whether or not to make an offer to the applicant, and if so, to which position $u_{ij} \in U_i$. If an applicant accepts the offer, position u_{ij} is irrevocably filled by the applicant.

Consider first the case with multiple positions per firm, adversarially chosen weights $w(u_{ij}, v_t)$ for all positions $u_{ij} \in U_i$ for each firm u_i and each applicant v_t, and random-order arrival of applicants. This is a generalization of the basic model studied in the previous section, so the logarithmic lower bound of Theorem 2 applies. Moreover, a straightforward $O(\log n)$-competitive algorithm is to run Algorithm 2 separately for each position of each firm. View every position of a firm as an independent *subfirm* simulating Algorithm 2: Whenever an applicant arrives, each regular firm u_i collects all offers of its subfirms, forwards only the highest-weight offer to the applicant and gives feedback about rejection/acceptance to the subfirms. Since applicants always choose the highest offer they receive, the allocation returned is the same as in an instance of the basic model consisting of all independent subfirms. Since in such an instance the social welfare for all subfirms is identical to the social welfare for all firms in the original instance, the competitive ratio of $O(\log n)$ of Algorithm 2 transfers directly, and the claim follows.

Independent Preferences and Multiple Positions. Instead of completely adversarial weights $w(u_{ij}, v_t)$ for all positions $u_{ij} \in U_i$ for each firm u_i and each applicant v_t, we consider the following case of independently chosen preferences: An adversary specifies a separate set $\mathcal{P}_i$ of n *applicant profiles* for each firm u_i. An applicant profile $p \in \mathcal{P}_i$ is a function $p : U_i \to \mathbb{R}^+$. In round t, when a new applicant v_t arrives, we pick one remaining profile $p_{it} \in \mathcal{P}_i$ for each firm $u_i \in U$ independently and uniformly at random. The weight for position $u_{ij} \in U_i$ is then given by $w(u_{ij}, v_t) = p_{it}(u_{ij})$. We pick profiles from $\mathcal{P}_i$ uniformly at random without replacement.

In this case, when $n \geq \sum_{i=1}^{m} k_i$ and $k_i \leq \alpha n$ for all $i \in [m]$ and some constant $\alpha \in (0, 1)$, we can achieve a constant competitive ratio using Algorithm 3. This algorithm resembles an optimal algorithm for secretary matching with a single firm [28]. Each firm rejects a number of applicants and enters an acceptance phase. In this phase, it maintains two virtual solutions: (1) an individual virtual optimum $M^*_{i,t}$ with respect to applicants that arrived up to and including round t, and (2) a virtual solution M'_i where all applicants are assumed to accept the offers of u_i. If the newly arrived applicant v_t is matched in $M^*_{i,t}$, it is offered the same position unless this position is already filled in M'_i.

Note that for a single firm in the basic model, this algorithm reduces to the standard e-competitive algorithm discussed in the introduction. As such, our approach here is exactly the one we outlined above

Algorithm 3: Matching algorithm for firm u_i for independent weights

Reject the first $r-1$ applicants
$M_i, M_i' \leftarrow \emptyset$
for *applicant* v_t *arriving in round* $t = r, \ldots, n$ **do**
  Let $M_{i,t}^*$ be optimum matching for firm u_i and applicants $\{v_1, \ldots, v_t\}$
  if v_t *is matched to position* u_{ij} *in* M_i^* *and* u_{ij} *is unmatched in* M_i' **then**
    Make an offer for position u_{ij} to v_t
    $M_i' \leftarrow M_i' \cup \{(u_{ij}, v_t)\}$
    if v_t *accepts* **then**
      $M_i \leftarrow M_i \cup \{(u_{ij}, v_t)\}$

in Proposition 2. The main point is to show that under the conditions on n and k_i, the properties of this specific algorithm combined with the independence among firms avoid the increase of a factor of m in the competitive ratio.

Theorem 4. *Algorithm 3 achieves a constant competitive ratio for firms with multiple positions and independent preferences.*

Proof. Fix a firm u_i. The matching M_i' is constructed by assuming that u_i is the only firm in the market, i.e., every applicant accepts the offer of firm u_i. Consider the individual optimum $M_{i,n}^*$ in hindsight. Then, by repeating the analysis of [28, Section 2] and replacing the sampling size of $\lceil n/e \rceil$ by $r-1$, the expected value of M_i' becomes

$$\mathbb{E}\left[w(M_i')\right] \quad \geq \quad \sum_{\ell=r}^{n} \frac{r-1}{\ell-1} \cdot \frac{w(M_{i,n}^*)}{n} \quad \geq \quad \frac{r-1}{n} \ln\left(\frac{n}{r-1}\right) \cdot w(M_{i,n}^*) \quad = \quad f(r) \cdot w(M_{i,n}^*) \ ,$$

where we denote the ratio by $f(r)$. Recall that $k_i \leq \alpha n$ and set $r = \beta n$ for some constant $\beta \in (0,1)$ such that $\beta > \alpha$. This ensures that $f(r)$ is bounded by a constant.

Let us now analyze the performance of the algorithm in the presence of competition. Consider applicant v_t in round t and the following events: (1) $P(u_i, v_t)$ is the event that u_i sends an offer to v_t, and (2) $A(u_i, v_t)$ is the event that u_i sends an offer to v_t and he accepts it. u_i's decision to offer depends only on $M_{i,t}^*$ and M_i', but not on the acceptance decisions of earlier applicants. v_t for sure accepts an offer from u_i if u_i offers and no other firm offers. Offers from other firms $u_{i'}$ occur only if $u_{i'}$ is matched in $M_{i',t}^*$. More formally, $A(u_i, v_t)$ occurs (at least) if $P(u_i, v_t)$ and none of the $P(u_{i'}, v_t)$ occur. Since the profiles for different firms are combined independently

$$\mathbf{Pr}\left[A(u_i, v_t) \mid P(u_i, v_t)\right] \quad \geq \quad \prod_{i \neq i'} (1 - \mathbf{Pr}\left[P(u_{i'}, v_t)\right]) \ .$$

Consider the probability that v_t is matched in $M_{i',t}^*$. Since the order of profiles for $u_{i'}$ is independent of the order for u_i, we can imagine again choosing t profiles uniformly at random. After choosing these t profiles, we pick the one for v_t uniformly at random. The t profiles determine $M_{i',t}^*$, which matches $\min(t, k_{i'})$ profiles. Since the profile of the last applicant is determined at random, the probability that v_t is matched in $M_{i',t}^*$ is at most $\min(1, k_{i'}/t)$. As $t \geq r = \beta n > \alpha n \geq k_{i'}$, we have

$$\mathbf{Pr}\left[P(u_{i'}, v_t)\right] \leq \begin{cases} 0 & \text{if } t \leq r-1, \\ k_{i'}/(\beta n) & \text{otherwise.} \end{cases}$$

Thus, for $t \geq r$

$$\mathbf{Pr}\left[A(u_i, v_t) \mid P(u_i, v_t)\right] \quad \geq \quad \prod_{i \neq i'} (1 - \mathbf{Pr}\left[P(u_{i'}, v_t)\right]) \quad \geq \quad \exp\left(\sum_{i=1}^{m} \ln\left(1 - \frac{k_i}{\beta n}\right)\right)$$

$$\geq \quad \exp\left(-\sum_{i=1}^{m} \frac{1}{1-(\alpha/\beta)} \cdot \frac{k_i}{\beta n}\right) \quad \geq \quad \exp\left(-\frac{1}{\beta - \alpha}\right).$$

The third inequality follows since $k_i \leq \alpha n$ implies $(1 - k_i/(\beta n)) \geq 1 - \alpha/\beta$. Furthermore, it holds that $\ln(1-x) \geq -\frac{x}{1-x}$ for all $x \in (0,1)$ (see Fact 1 in the Appendix). The last inequality is due to $n \geq \sum_j k_j$.

Consequently, $\mathbb{E}[w(M_i)]$ recovers at least a constant fraction of $\mathbb{E}[w(M_i')]$, which represents a constant factor approximation to the individual optimum $M_{i,n}^*$ for i in hindsight. By linearity of expectation, the algorithm achieves a constant competitive ratio for the expected weight of the optimum matching. □

5 Conclusion and Open Problems

In this paper, we study online secretary problems with n applicants, m firms, and local information. We analyze thresholding-based algorithms and show how to obtain a competitive ratio of $O(\log n)$, even if firms can accept sets of applicants based on a local matroid. Moreover, we show a lower bound of $O(\log n/(\log \log n))$ if all firms use thresholding-based algorithms. These bounds continue to hold in terms of m if every firm knows the number of firms in the market. For a more structured domain, we show that a constant competitive ratio can be achieved.

It is an interesting open problem if our bounds can be improved, in general and for other meaningful special cases. For the general case, a crucial issue is to determine the right order of magnitude of firms that contribute significantly to social welfare. In the basic model, if a firm makes no offer, then the firm does not obtain feedback that allows it to learn the number of firms competing in the market. Feedback about the number of (better) firms is generated only by accepted and rejected offers. Crucially, to circumvent the lower bound that thresholding-based algorithms face, we would need to use this feedback to learn information about the number of (better) firms or deduce other market parameters. Such information might then be helpful in designing better algorithms.

References

[1] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proc. 13th Conf. Electronic Commerce (EC)*, pages 18–35, 2012.

[2] Moshe Babaioff, Michael Dinitz, Anupam Gupta, Nicole Immorlica, and Kunal Talwar. Secretary problems: Weights and discounts. In *Proc. 20th Symp. Discrete Algorithms (SODA)*, pages 1245–1254, 2009.

[3] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exchanges*, 7(2), 2008.

[4] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proc. 18th Symp. Discrete Algorithms (SODA)*, pages 434–443, 2007.

[5] Yakov Babichenko, Yuval Emek, Michal Feldman, Boaz Patt-Shamir, Ron Peretz, and Rann Smorodinsky. Stable secretaries. In *Proc. 18th Conf. Economics and Computation (EC)*, pages 243–244, 2017.

[6] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Trans. Algorithms*, 9(4):32, 2013.

[7] Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary problems via linear programming. In *Proc. 14th Intl. Conf. Integer Programming and Combinatorial Optimization (IPCO)*, pages 163–176, 2010.

[8] Ning Chen, Martin Hoefer, Marvin Künnemann, Chengyu Lin, and Peihan Miao. Secretary markets with local information. In *Proc. 42nd Intl. Coll. Automata, Languages and Programming (ICALP)*, volume 2, pages 552–563, 2015.

[9] Daniel Cownden and David Steinsaltz. Effects of competition in a secretary problem. *Oper. Res.*, 62(1):104–113, 2014.

[10] Nikhil Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proc. 12th Conf. Electronic Commerce (EC)*, pages 29–38, 2011.

[11] Nedialko Dimitrov and Greg Plaxton. Competitive weighted matching in transversal matroids. *Algorithmica*, 62(1-2):333–348, 2012.

[12] Michael Dinitz and Guy Kortsarz. Matroid secretary for regular and decomposable matroids. *SIAM J. Comput.*, 43(5):1807–1830, 2014.

[13] Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing non-stochastic inputs. In *Proc. 27th Symp. Foundations of Computer Science (FOCS)*, pages 540–551, 2017.

[14] Paul Dütting and Robert Kleinberg. Polymatroid prophet inequalities. In *Proc. 23rd European Symp. Algorithms (ESA)*, pages 437–449, 2015.

[15] Eugene Dynkin. The optimum choice of the instant for stopping a Markov process. In *Sov. Math. Dokl*, volume 4, pages 627–629, 1963.

[16] Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $O(\log\log(\text{rank}))$-competitive algorithm for the matroid secretary problem. In *Proc. 26th Symp. Discrete Algorithms (SODA)*, pages 1189–1201, 2015.

[17] Moran Feldman and Moshe Tennenholtz. Interviewing secretaries in parallel. In *Proc. 13th Conf. Electronic Commerce (EC)*, pages 550–567, 2012.

[18] Moran Feldman and Rico Zenklusen. The submodular secretary problem goes linear. In *Proc. 56th Symp. Foundations of Computer Science (FOCS)*, pages 486–505, 2015.

[19] Thomas Ferguson. Who solved the secretary problem? *Statistical Science*, 4(3):282–289, 1989.

[20] Oliver Göbel, Martin Hoefer, Thomas Kesselheim, Thomas Schleiden, and Berthold Vöcking. Online independent set beyond the worst-case: Secretaries, prophets and periods. In *Proc. 41st Intl. Coll. Automata, Languages and Programming (ICALP)*, volume 2, pages 508–519, 2014.

[21] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *Proc. 6th Workshop Internet & Network Economics (WINE)*, pages 246–257, 2010.

[22] Martin Hoefer and Bojana Kodric. Combinatorial secretary problems with ordinal information. In *Proc. 44th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 133:1–133:14, 2017.

[23] Sungjin Im and Yajun Wang. Secretary problems: Laminar matroid and interval scheduling. In *Proc. 22nd Symp. Discrete Algorithms (SODA)*, pages 1265–1274, 2011.

[24] Nicole Immorlica, Adam Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. Dueling algorithms. In *Proc. 43rd Symp. Theory of Computing (STOC)*, pages 215–224, 2011.

[25] Nicole Immorlica, Robert Kleinberg, and Mohammad Mahdian. Secretary problems with competing employers. In *Proc. 2nd Workshop Internet & Network Economics (WINE)*, pages 389–400, 2006.

[26] Patrick Jaillet, José Soto, and Rico Zenklusen. Advances on matroid secretary problems: Free order model and laminar case. In *Proc. 16th Intl. Conf. Integer Programming and Combinatorial Optimization (IPCO)*, pages 254–265, 2013.

[27] Anna Karlin and Eric Lei. On a competitive secretary problem. In *Proc. 29th Conf. Artif. Intell. (AAAI)*, pages 944–950, 2015.

[28] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Proc. 21st European Symp. Algorithms (ESA)*, pages 589–600, 2013.

[29] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal beats dual on online packing LPs in the random-order model. In *Proc. 46th Symp. Theory of Computing (STOC)*, pages 303–312, 2014.

[30] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proc. 16th Symp. Discrete Algorithms (SODA)*, pages 630–631, 2005.

[31] Robert Kleinberg and Matthew Weinberg. Matroid prophet inequalities. In *Proc. 44th Symp. Theory of Computing (STOC)*, pages 123–136, 2012.

[32] Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *Proc. 36th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 508–520, 2009.

[33] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bull. Amer. Math. Soc*, 83:745–747, 1977.

[34] Ulrich Krengel and Louis Sucheston. On semiamarts, amarts and processes with finite value. *Advances in Prob.*, 4:197–266, 1978.

[35] Oded Lachish. O(log log rank) competitive ratio for the matroid secretary problem. In *Proc. 55th Symp. Foundations of Computer Science (FOCS)*, pages 326–335, 2014.

[36] Dennis Lindley. Dynamic programming and decision theory. *Applied Statistics*, 10:39–51, 1961.

[37] Marco Molinaro and R. Ravi. Geometry of online packing linear programs. *Math. Oper. Res.*, 39(1):46–59, 2014.

[38] José Soto. Matroid secretary problem in the random-assignment model. *SIAM J. Comput.*, 42(1):178–211, 2013.

A Useful Facts

Fact 1. *For all $x \in [0,1)$ it holds that*

$$\ln(1-x) \quad \geq \quad -\frac{x}{1-x} \ .$$

Proof. For $x = 0$ we have equality. The derivative of the left and right hand sides are $-(1-x)^{-1}$ and $-(1-x)^{-2}$, respectively. Hence, the right-hand side drops faster when $x > 0$ grows towards 1, so the inequality holds for the entire interval. □

B Extension to the Sampling Model

In this section, we extend our logarithmic approximation to a general sampling model presented in [20]. This model extends the secretary model (adversarial values, random-order arrival), the prophet-inequality model (stochastic values from known distributions, adversarial arrival) as well as other mixtures of stochastic and worst-case aspects.

Formally, in the sampling model we have two values for each firm-applicant pair (u_i, v_j), a non-negative *sample value* $w^S(u_i, v_j)$ and a non-negative *input value* $w^I(u_i, v_j)$. The sample and input values are both drawn from possibly different, unknown distributions. For a single applicant v_j the sample and input distributions can be arbitrarily correlated among different firms and among each other. However, there is no correlation among distributions of different applicants. This defines a probability space over a class of instances $\mathcal{I}$.

The arrival process proceeds as follows. First, the adversary draws all values $w^S(u_i, v_j)$ and $w^I(u_i, v_j)$ for all pairs (u_i, v_j). It then reveals to firm u_i all drawn sample values $w^S(u_i, v_j)$, for all applicants v_j. Subsequently, depending on the drawn values w^I it chooses a worst-case arrival order of applicants. Upon arrival, an applicant v_j reveals its "real" value $w^I(u_i, v_j)$ to firm u_i. The algorithm $\mathcal{A}$ for firm u_i decides whether to make an offer to v_j, and applicant v_j accepts an offer that maximizes $w^I(u_i, v_j)$. Then the next applicant arrives. Decisions made in earlier rounds cannot be revoked. The goal of the algorithm is to maximize the social welfare, i.e., to generate an assignment $M^{\mathcal{A}}$ that minimizes the competitive ratio $\mathbb{E}_{\mathcal{I}}[w^I(M^*)]/\mathbb{E}_{\mathcal{I},\mathcal{A}}[w^I(M^{\mathcal{A}})]$.

Clearly, if sample values are completely unrelated to input values, no algorithm $\mathcal{A}$ can obtain a bounded competitive ratio. Towards this end, we assume that for each value k, there is a similar probability that w^I and w^S have value k for pair (u_i, v_j). We here restrict attention to discrete distributions over integers. It is straightforward to show that our results hold for general distributions, but this minor extension does not justify the notational and technical overhead it will add to the presentation. More formally, we assume

- *Stochastic similarity*: Suppose $c > 1$ is a fixed constant. For every pair (u_i, v_j) and every integer $k > 0$, we assume that $\mathbf{Pr}\left[w^I(u_i, v_j) = k\right] \leq c \cdot \mathbf{Pr}\left[w^S(u_i, v_j) = k\right]$ and $\mathbf{Pr}\left[w^S(u_i, v_j) = k\right] \leq c \cdot \mathbf{Pr}\left[w^I(u_i, v_j) = k\right]$.
- *Stochastic independence*: For every pair (u_i, v_j), the weights $w^I(u_i, v_j)$ and $w^S(u_i, v_j)$ do not depend on the weights w^S and w^I of other candidates $v_{j'} \neq v_j$.

For further discussion of the sampling model and an exposition how to formulate the secretary and prophet-inequality models within this framework, see [20].

Consider Algorithm 4, which is an extension of Algorithm 2 to the sampling model. It can be applied when every firm has a local matroid $\mathcal{S}_i$ that determines the set of applicants the firm can hire simultaneously. It is executed in parallel by all firms u_i. The algorithm first simplifies the structure of the input and sample values by assuming that no candidate has $w^S(u_i, v_j) > 0$ and $w^I(u_i, v_j) > 0$. This loses a factor of at most 2 in the expected value of the solution. Analogous to our proof in the secretary model, we assume that every firm knows an upper bound on the maximum cardinality of optimal solutions. More precisely, define $n_{\max}$ as the maximum cardinality of a legal assignment of applicants to firms (i.e., an assignment

Algorithm 4: Thresholding algorithm for firm u_i for general weights and matroids.

1 For each v_j flip a fair coin: if heads $w^I(u_i, v_j) \leftarrow 0$, if tails $w^S(u_i, v_j) \leftarrow 0$
2 $m_i \leftarrow \arg\max_{v_j} w^S(u_i, v_j)$
3 $X_i \leftarrow \text{Uniform}(0, 1, \ldots, \lceil \log_2 b \rceil + 1)$, where $b \geq n_{\max}$
4 $t_i \leftarrow w^S(u_i, m_i)/2^{X_i}$
5 $M_i \leftarrow \emptyset$
6 **for** *all* v_j *over time* **do**
7   **if** $w^I(u_i, v_j) \geq t_i$ *and* $M_i \cup \{v_j\}$ *is independent set in* $\mathcal{S}_i$ **then**
8   make an offer to v_j
9   **if** v_j *accepts* **then**
10             $M_i \leftarrow M_i \cup \{v_j\}$

such that the set of hired applicant's of firm u_i is an independent set in u_i's matroid $\mathcal{S}_i$). Note that in general, $n_{\max} \leq \min\{n, \sum_{i=1}^{m} k_i\}$, where k_i denotes the rank of $\mathcal{S}_i$, but $n_{\max}$ can also be significantly smaller depending on the structure of the matroids. Using the parameter $n_{\max}$, we determine a random threshold based on the maximum weight seen by firm u_i in its simplified sample. Then the algorithm greedily makes an offer only to those applicants whose simplified input values are above the threshold.

Theorem 5. *Algorithm 4 is* $16(c+1)^2(\lceil \log_2 b \rceil + 2)$*-competitive in the sampling model.*

Proof. The proof follows largely the one presented for the secretary model in Section 3 above. At first, however, we use arguments similar to [20] to capture the relation between sample and input values and to transform the scenario into a simpler domain.

The first line of our algorithm implements an adjustment of weights, so that at most one of the two weights for an applicant and a firm is positive. Let us assume w.l.o.g. that this condition holds already for the initial weights w^I and w^S. Formally, we denote

$$\hat{w}(u_i, v_j) = \max\{w^I(u_i, v_j), w^S(u_i, v_j)\}$$

and assume that $(w^I(u_i, v_j), w^S(u_i, v_j)) \in \{(0, \hat{w}(u_i, v_j)), (\hat{w}(u_i, v_j), 0)\}$. This preserves stochastic independence and similarity properties of the sampling model. Moreover, it lowers the expected value of the optimum solution by at most a factor of 2, i.e.,

$$\mathbb{E}_{\mathcal{I}}[w^I(M^*)] \quad \leq \quad 2\mathbb{E}_{\mathcal{I}}[\hat{w}(M^*)] \quad \leq \quad 2\mathbb{E}_{\mathcal{I}}[\hat{w}(\hat{M}^*)] \ ,$$

where M^* and $\hat{M}^*$ are optimal solutions for w^I and $\hat{w}$, respectively.

We condition on properties of the applicant with the largest and second largest value for firm u_i. To cope with the resulting correlations, we introduce a conditional probability space. For each applicant v_j we assume that $\hat{w}(u_i, v_j)$ is fixed arbitrarily. For simplicity, we drop applicants from consideration for which $\hat{w}(u_i, v_j) = 0$. Let $V_i^I = \{v_j \mid w^I(u_i, v_j) > 0\}$ and $V_i^S = \{v_j \mid w^S(u_i, v_j) > 0\}$. Stochastic similarity implies

$$\mathbf{Pr}\left[w^I(u_i, v_j) = \hat{w}(u_i, v_j)\right] \quad \geq \quad (1/c) \cdot \mathbf{Pr}\left[w^S(u_i, v_j) = \hat{w}(u_i, v_j)\right]$$

and

$$\mathbf{Pr}\left[w^S(u_i, v_j) = \hat{w}(u_i, v_j)\right] \quad \geq \quad (1/c) \cdot \mathbf{Pr}\left[w^I(u_i, v_j) = \hat{w}(u_i, v_j)\right] \ .$$

Since $V_i^I \cap V_i^S = \emptyset$, we have

$$\mathbf{Pr}\left[v_j \in V_i^I\right] \geq \frac{1}{c+1} \qquad \text{and} \qquad \mathbf{Pr}\left[v_j \in V_i^S\right] \geq \frac{1}{c+1} \tag{4}$$

for each applicant v_j, independent of the outcome of weights of other applicants. In particular, (4) holds for every v_j, independently of $v_{j'} \in V_i^S$ or not for all other applicants $j' \neq j$.

We now execute the proof of the theorem, which proceeds very similarly to the proof of Theorem 1 above. We make two assumptions that make the analysis easier but do not hurt the overall result.

1. Based on our reformulation on a conditional probability space, we assume all $\hat{w}(u_i, v_j)$ are fixed arbitrarily. Furthermore, we assume $\hat{M}^*$ is an optimum solution when all applicants are in V_i^I for all firms u_i. As such, we assume that both $\hat{w}$ and $\hat{M}^*$ are deterministic. Our analysis is based only on the randomization expressed by the sampling inequalities (4) and the randomized choice of t_i in Algorithm 4.
2. To avoid technicalities, we again assume that for each firm u_i, the values $\hat{w}(u_i, v_j)$ of all applicants are mutually disjoint.

Let $v_i^{\max} = \text{argmax}_j \hat{w}(u_i, v_j)$ and $v_i^{\text{2nd}} = \text{argmax}_{j \neq v_i^{\max}} \hat{w}(u_i, v_j)$ be the best and second best applicant for firm u_i, respectively. Let $w_i^{\max} = \hat{w}(u_i, v_i^{\max})$ and $w_i^{\text{2nd}} = \hat{w}(u_i, v_i^{\text{2nd}})$ denote the corresponding weights. For most of the analysis, we again work with *capped weights* $\tilde{w}(u_i, v_j)$, based on thresholds t_i set by the algorithm as follows

$$\tilde{w}(u_i, v_j) = \begin{cases} w_i^{\max} & \text{if } v_j \in V_i^I, t_i = w_i^{\text{2nd}}, \text{ and } \hat{w}(u_i, v_j) > w_i^{\text{2nd}}, \\ t_i & \text{else, if } v_j \in V_i^I \text{ and } \hat{w}(u_i, v_j) \geq t_i, \\ 0 & \text{otherwise.} \end{cases}$$

The definition of $\tilde{w}$ relies on random events, i.e., $v_j \in V_i^I$ and the choice of thresholds t_i. For any outcome of these events, however, $\tilde{w}(u_i, v_j) \leq \hat{w}(u_i, v_j)$ for all pairs (u_i, v_j). The following lemma adapts Lemma 1 and shows that, in expectation over all the correlated random events, an optimal offline solution with respect to $\tilde{w}$ gives a logarithmic approximation to the optimal offline solution with respect to $\hat{w}$.

Lemma 3. *Denote by $\hat{w}(M)$ and $\tilde{w}(M)$ the weight and capped weight of a solution M. Let $\tilde{M}^*$ and $\hat{M}^*$ be optimal solutions for $\tilde{w}$ and $\hat{w}$, respectively. Then*

$$\mathbb{E}\left[\tilde{w}(\tilde{M}^*)\right] \geq \frac{1}{4(c+1)^2(\lceil \log_2 b \rceil + 2)} \cdot \hat{w}(\hat{M}^*).$$

Proof. Let $(u_i, v_j) \in \hat{M}^*$ be an arbitrary pair. First, assume that v_j maximizes $\hat{w}(u_i, v_j)$, i.e., $v_j = v_i^{\max}$. By (4) with probability at least $1/(c+1)^2$, we have $v_j \in V_i^I$ and $v_i^{\text{2nd}} \in V_i^S$. For any such outcome, we have with probability $1/(\lceil \log_2 b \rceil + 2)$ that $t_i = w_i^{\text{2nd}}$ and $\tilde{w}(u_i, v_j) = w_i^{\max}$. This yields $\mathbb{E}\left[\tilde{w}(u_i, v_j)\right] \geq \hat{w}(u_i, v_j)/((c+1)^2(\lceil \log_2 b \rceil + 2))$.

Second, for any $v_j \neq v_i^{\max}$ with $w_i^{\max}/(2b) < \hat{w}(u_i, v_j) \leq w_i^{\max}$, by (4) we know $v_i^{\max} \in V_i^S$ is an independent event which happens with probability at least $1/(c+1)$. Then, there is some $0 \leq k' \leq \lceil \log_2 b \rceil + 1$, with $\hat{w}(u_i, v_j) > w_i^{\max}/2^{k'} \geq \hat{w}(u_i, v_j)/2$. With probability $1/(\lceil \log_2 b \rceil + 2)$, we have that $X_i = k'$ and $\tilde{w}(u_i, v_j) = t_i \geq \hat{w}(u_i, v_j)/2$. This yields $\mathbb{E}\left[\tilde{w}(u_i, v_j)\right] \geq \hat{w}(u_i, v_j)/(2(c+1)^2(\lceil \log_2 b \rceil + 2))$, since $v_j \in V_i^I$ with probability at least $1/(c+1)$ by (4).

Finally, we denote by $\hat{M}^>$ the set of pairs $(u_i, v_j) \in \hat{M}^*$ for which $\hat{w}(u_i, v_j) > w_i^{\max}/(2b)$. The expected weight of the best assignment with respect to the threshold values is thus

$$\begin{aligned} \mathbb{E}\left[\tilde{w}(\tilde{M}^*)\right] &\geq \sum_{(u_i, v_j) \in \hat{M}^*} \mathbb{E}\left[\tilde{w}(u_i, v_j)\right] \geq \sum_{(u_i, v_j) \in \hat{M}^>} \frac{\hat{w}(u_i, v_j)}{2(c+1)^2(\lceil \log_2 b \rceil + 2)} \\ &= \frac{1}{2(c+1)^2(\lceil \log_2 b \rceil + 2)} \cdot (\hat{w}(\hat{M}^*) - \hat{w}(\hat{M}^* \setminus \hat{M}^>)) \\ &\geq \frac{1}{4(c+1)^2(\lceil \log_2 b \rceil + 2)} \cdot \hat{w}(\hat{M}^*), \end{aligned}$$

since $\sum_{(u_i, v_j) \in \hat{M}^* \setminus \hat{M}^>} w_i^{\max}/(2b) \leq \max_i w_i^{\max}/2 \leq \hat{w}(\hat{M}^*)/2$. Here we use $b \geq n_{\max} \geq |\hat{M}^*|$, which holds since $\hat{M}^*$ is a legal assignment and consequently, its cardinality is bounded by the maximum cardinality $n_{\max}$ of any legal assignment. □

The previous lemma bounds the weight loss due to (1) all random choices inherent in the process of input generation and threshold selection and (2) using the capped weights. The next lemma is essentially identical to Lemma 2 and bounds the remaining loss due to adversarial arrival of elements in V_i^I, exploiting that $\tilde{w}$ equalizes equal-threshold firms. Note that in Lemma 2 we already prove the result for arbitrary arrival, arbitrary weights w, and arbitrary thresholds based on w. Moreover, we define thresholds t_i based on $\hat{w}$ in exactly the same way as they we did based on w for Lemma 2. Hence, the lemma and its proof can be applied literally when using $\hat{w}$ instead of w.

Lemma 4. *Suppose subsets V_i^I and thresholds t_i are fixed arbitrarily and consider the resulting weight function $\tilde{w}$. Let $M^{\mathcal{A}}$ be the feasible solution resulting from Algorithm 4 using the thresholds t_i, for any arbitrary arrival order of applicants in $\bigcup V_i^I$. Then $\hat{w}(M^{\mathcal{A}}) \geq \tilde{w}(\tilde{M}^*)/2$.*

Combining these insights we see that that

$$\begin{aligned}
\mathbb{E}_{\mathcal{I}}[w^I(M^*)] &\leq 2\mathbb{E}_{\mathcal{I}}[\hat{w}(\hat{M}^*)] \\
&\leq 8(c+1)^2(\lceil \log_2 b \rceil + 2)\mathbb{E}_{\mathcal{I},\mathcal{A}}[\tilde{w}(\tilde{M}^*)] \\
&\leq 16(c+1)^2(\lceil \log_2 b \rceil + 2)\mathbb{E}_{\mathcal{I},\mathcal{A}}[\hat{w}(M^{\mathcal{A}})] \\
&\leq 16(c+1)^2(\lceil \log_2 b \rceil + 2)\mathbb{E}_{\mathcal{I},\mathcal{A}}[w^I(M^{\mathcal{A}})] \ .
\end{aligned}$$

This proves the theorem. □