The Public University Secretary Problem*

Benjamin Moseley† Heather Newman‡ Kirk Pruhs§

Abstract

We consider a variation of the classic secretary problem where the goal is to hire the k cheapest secretaries. We use a beyond-worst-case approach where we assume that the online algorithm knows a priori an upper bound on the optimal cost, and we show this assumption is necessary. The main result is that the optimal competitive ratio is $\Theta(\log k)$. The upper bound holds even when candidates are interviewed in adversarial order, and is attained by a simple deterministic algorithm. The lower bound of $\Omega(\log k)$ holds against randomized algorithms and even when candidates are interviewed in random order.

1 Introduction

1.1 Motivation Credit for initiating the wildly popular line of research on "secretary problems" is generally given to Merrill Flood (as well as credit for initiating research on the prisoner's dilemma problem and much credit for popularizing the traveling salesman problem) [14, 16]. The setting for secretary problems involves sequentially interviewing candidate employees, with secretaries being the quintessential type of employee, with the goal of hiring the best employees, subject to some constraints. It seems Flood formulated the initial secretary problem in the late 1940s while working for the RAND Corporation, which was formed in the immediate aftermath of World War II by a collaboration of the War Department, the Office of Scientific Research and Development, and industry. Given the emphasis on maximizing quality in the formulation of secretary problems, without concern as to costs, one might reasonably infer that this reflected the culture of RAND at that time, and this culture grew from RAND being well-funded.

Our research was spurred by an observation: many institutions, particularly often underfunded public universities, tend to prioritize cost efficiency over quality during their hiring process. This phenomenon was strikingly apparent at the University of Pittsburgh during its early recovery stages from an economic crisis that resulted in the loss of nearly 100,000 manufacturing jobs from a labor force of a million within a condensed time frame [12].

During this period, it was a widely accepted notion that the Pennsylvania state legislature expected the University of Pittsburgh, as a public institution, to play an active role in the economic rebound. The expectation was that a portion of the university's state funding would be channeled towards creating new employment opportunities.

Consequently, the hiring strategy primarily revolved around attracting candidates, especially administrative staff, who were prepared to accept lower salaries. The aim was clear: to boost employment figures rather than to focus on the quality of hires.

Problem Definition: This paper introduces the Public University Secretary problem. A priori, the online algorithm learns two positive integers k and B. We remark that the algorithm does not need to know n, the number of secretaries who are to be interviewed. Here k is the minimum number of secretaries that need to be hired. Each secretary has a positive integer cost x_i and the goal is to hire a set C of *no less than* k secretaries to minimize $\sum_{i \in C} x_i$. The number B is assumed to be an upper bound on the cost of the optimal solution and the algorithm seeks to bound $\sum_{i \in C} x_i$ within some factor of B. We point out that, without knowledge of B, an online algorithm can accrue unbounded cost, as we show in subsection 1.2. In practice, B is the given target budget and would typically be known.

*Moseley and Newman are supported in part by a Google Research Award, an Inform Research Award, a Carnegie Bosch Junior Faculty Chair, and NSF grants CCF-2121744 and CCF-1845146. Pruhs is supported in part by NSF grants CCF-1907673, CCF-2036077, CCF-2209654 and an IBM Faculty Award.

†Tepper School of Business. Carnegie Mellon University.

‡Department of Mathematical Sciences. Carnegie Mellon University.

§Computer Science Department. University of Pittsburgh.

The n candidate secretaries are interviewed over time. Candidate $i \in [n]$ is the i^{th} candidate to be interviewed, and the online algorithm only learns x_i when candidate i is interviewed. The online algorithm must decide immediately whether to irrevocably hire candidate i as a secretary, or to irrevocably reject candidate i, without seeing the future.

We will consider interviewing in both adversarial and random order. The adversarial order model assumes a worst-case order. In the random order model, the candidates are interviewed in uniformly random order over all possible $n!$ permutations.

1.2 Contrasting the Standard Secretary Problem with the Public University Secretary Problem

Recall that in the standard secretary problem, each x_i represents the quality of secretary i (rather than cost), and the objective is to maximize the aggregate quality of the hired secretaries subject to the constraint that at most j secretaries can be hired. We call this the RAND Secretary problem to contrast it with the Public University Secretary problem.

Here we illustrate the algorithmic difference between hiring secretaries to minimize aggregate cost and hiring secretaries to maximize aggregate quality. For this discussion, we assume $k = 1$. Along with several observations, we emphasize that the standard algorithmic approach of *sample-and-price* [7] will not be of use in the Public University Secretary problem. By sample-and-price, we are referring to a stopping rule for sampling a certain number of candidates, and then computing a hiring threshold. Intuitively, this approach will not work because the Public University Secretary problem is a minimization problem rather than a maximization problem.

Let us consider the RAND Secretary problem. The first observation is that no deterministic algorithm can achieve any reasonable competitiveness in the RAND Secretary problem if the candidates are interviewed in adversarial order [4]. To see this, assume that $n = 2$ and that the first candidate has quality $x_1 = q$. If the online algorithm hires the first candidate, then the second candidate has quality $x_2 = q^2$, resulting in competitiveness q, as hiring the second candidate was optimal. Otherwise, if the online algorithm does *not* hire the first candidate, then the second candidate has quality $x_2 = 1$, again resulting in competitiveness q.

Due to the above example, most research on secretary problems assumes some beyond-worst-case analysis approach. Most commonly the algorithm is assumed to know a priori the number n of candidates to be interviewed and that they are interviewed in random order. The insurmountable issue faced by the online algorithm with adversarial ordering is that the algorithm does not know whether q is high quality or low quality. Random ordering allows the online algorithm to learn the scale of secretary quality. Indeed, the optimal e-competitive online algorithm is a threshold algorithm that hires the first candidate among the last $(1 - \frac{1}{e})n$ candidates that is of higher quality than a threshold. The threshold is set to be the quality of the best candidate from the first $\frac{n}{e}$ candidates. The first $\frac{n}{e}$ candidates are used to learn the scale of secretary quality in the applicant pool.

Now consider the Public University Secretary problem. We first show no bounded competitiveness is achievable even when secretaries are interviewed in random order. To see this, consider the following two instances in which $k = 1$ and $n = 2$. In the first instance, the secretaries have cost 1 and q, and in the second instance, the secretaries have cost q and q^2.

Consider what happens when the first candidate has cost q, which happens with probability $\frac{1}{2}$ in both instances. If the online algorithm hires this first candidate, then it is q-competitive on the first instance as the optimal solution was to hire the candidate of cost 1. And if the online algorithm does not hire this first candidate, then it is q-competitive on the second instance as the optimal solution was to hire the candidate of cost q. Thus any online algorithm has competitiveness $\Omega(q)$ in expectation.

Due to the Public University Secretary problem being a minimization problem, random order does not allow the online algorithm to learn the scale of secretary costs in the applicant pool. This is because the algorithm cannot simply reject secretaries (e.g., as in a sampling phase) as all remaining secretaries may have enormous cost and we must hire k secretaries. In contrast, algorithms for the maximization version only care that there is some chance of hiring the best secretary, and can effectively ignore cases where low-value secretaries are hired.

In order to have any hope of achieving reasonable competitiveness, the online algorithm needs some a priori understanding of the scale of secretary costs—even in the random order model, as we observe above. After some reflection, it seems natural to assume that the online algorithm knows a priori a reasonable upper bound B of the aggregate cost of the k cheapest secretaries in the applicant pool, or equivalently a reasonable estimate $\frac{B}{k}$ of the average cost of a secretary in the optimal solution. This can be viewed as a target budget, which is often known in practice.

1.3 Our Results

Our main result is that the best possible online algorithm for the Public University Secretary problem has cost $\Theta(B \log k)$ in both adversarial and random order models.

The upper bound of $O(B \log k)$ is given in Section 2, and holds even with adversarial ordering. This upper bound is attained by a simple deterministic algorithm, which we call the Cautious algorithm. Intuitively, the Cautious algorithm will hire candidate i if and only if candidate i is in the best solution up until that time. The best solution is defined as the set containing the maximum number of candidates among those that have been interviewed so far, subject to the constraint that the aggregate cost of these candidates is at most B.

The lower bound of $\Omega(B \log k)$ is given in Section 3, and holds against randomized algorithms and even in the random order model. The key step of our lower bound argument is to show that any reasonably competitive algorithm must hire each candidate hired by the Cautious algorithm. Otherwise, a proof shows there is some instance for which the algorithm's cost is effectively unbounded. Thus the Cautious algorithm is cautious in that it only hires when not hiring could be calamitous.

A key takeaway is that randomization (either internal to the algorithm or in the interviewing order) is of negligible benefit to the online algorithm. This is in contrast to the importance of random order for the maximization version.

1.4 Related Results

As far as the authors are aware, there is no prior literature on minimizing cost in the class of secretary problems considered in this paper. In contrast, the algorithmic literature on problems involving sequential decision-making, that can be conceptualized as involving interviewing and hiring high quality candidates (i.e., maximization-based problems), is vast. Thus we can at best just scratch the surface, and will only explicitly mention those results that we believe are the most relevant. We point readers interested in a deeper dive to four relatively recent surveys [7, 4, 6, 16].

The objective in the most classic version of the secretary problem is to maximize the probability that the unique best secretary is hired (where only one hire is allowed). The classic (essentially optimal e-competitive) algorithm is a threshold algorithm that hires the first candidate among the last $(1 - \frac{1}{e})n$ candidates that is of higher quality than a threshold, which is set to be the quality of the best candidate from the first $\frac{n}{e}$ candidates. This algorithm is obviously also e-competitive for the objective of maximizing the expected quality of the hired secretary (where only one hire is allowed). A similar threshold algorithm is $\left(1 - O\left(\frac{1}{\sqrt{k}}\right)\right)$-competitive for the objective of maximizing the aggregate quality of k hired secretaries [11]. The constraint that at most k secretaries can be hired is a type of matroidal constraint. There is a significant literature on hiring the best secretary subject to matroidal constraints (for a survey see [7]). There is a type of threshold algorithm that is $O(\log k)$-competitive for any type of matroidal constraint, where k is the rank of the matroid [3]. Further, there is a complicated algorithm that improve to $O(\sqrt{\log k})$-competitive for any type of matroidal constraint [5], and algorithms that improves to $O(\log \log k)$-competitive [9, 13]. Constant competitive algorithms exist for many common types of combinatorial matroidal constraints [7]. There is an e-competitive threshold algorithm for the problem of maximizing the number of secretaries (i.e. unweighted value) hired subject to the constraint that their aggregate salary cannot exceed some a priori bound [2]. This can be viewed as a knapsack constraint on the aggregate salary. Further this can be extended to an $O(1)$-competitive algorithm for the problem of maximizing the aggregate value (i.e. weighted value) of the hired secretaries subject to the constraint their aggregate salary cannot exceed some a priori bound [2].

There is also an extensive literature on prophet inequality problems, which can be viewed as trying to hire the best secretaries in the case that the quality of each candidate is drawn from some distribution that is known a priori by the online algorithm [6].

Lastly, there are various works that consider the maximization variant with advice. A commonly considered form of advice is a prediction on the secretaries' quality [1, 10]. Additionally, there are alternatives to the random order model. For example, models where candidate qualities are drawn from unknown distributions, or a sample of secretaries is given in advance [8, 15].

We remark that the model considered in this paper is related to, but does not fall within, the typical form of an algorithms with predictions model. For, the number B received as input is a known upper bound on optimal.

2 Adversarial Order Upper Bound

In subsection 2.1, we describe the Cautious algorithm restricted to the special case that all costs are integer powers of two. Then, in subsection 2.2, we generalize this to a description of the Cautious algorithm for general instances. Finally, in subsection 2.3, we analyze the Cautious algorithm.

2.1 Simplified Version of Cautious Algorithm Before describing the simplified version of the Cautious algorithm for these instances, we need one definition.

Definition 1. *Let $x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(i)}$ be the costs of the first i candidates sorted in nondecreasing order σ, with ties broken in favor of earlier interviewed candidates. Notice that for each $j \in [1, i-1]$, either $x_{\sigma(j)} < x_{\sigma(j+1)}$, or $x_{\sigma(j)} = x_{\sigma(j+1)}$ and $\sigma(j) < \sigma(j+1)$. Let h be maximum such that $h \leqslant k$ and $\sum_{j=1}^{h} x_{\sigma(j)} \leqslant B$. We then define candidates $\sigma(1), \sigma(2), \dots, \sigma(h)$ to be the* best solution after i interviews.

Simplified Version of the Cautious Algorithm Description: Candidate i is hired if and only if candidate i is in the best solution after i interviews.

As a corollary to our later analysis of the Cautious algorithm for general instances, one can conclude that this simplified version of the algorithm incurs cost $O(B \log k)$ for instances where each cost x_i is of the form $2^j \cdot \frac{B}{k}$ for some integer j such that $0 \leqslant j \leqslant h$. But this simplified algorithm can have high cost for general instances. To see this consider the instance where $B = n^2$, $k = \frac{n}{2} + 1$, for each $i \in [1, \frac{n}{2}]$ it is the case that $x_i = n^2 - i$, and for each $i \in [\frac{n}{2} + 1, n]$ it is the case that $x_i = 1$. The simplified Cautious algorithm will hire each of the first $n/2$ candidates, incurring a cost of $\Theta(n^3)$. But the optimal solution hires the last $\frac{n}{2} + 1$ candidates, for a total cost of $B = \Theta(n^2)$.

2.2 The General Cautious Algorithm Intuitively, the mistake that the simplified version of the Cautious algorithm makes in this instance is to treat candidates of similar costs differently. The natural remedy is to round all costs to integer powers of two times $\frac{B}{k}$, and then use the simplified version of the Cautious algorithm. We start with some preliminary definitions.

Definition 2.

- *The cost class c_i of candidate i is 0 if $x_i < \frac{2B}{k}$, infinite if $x_i > B$, and the maximum integer j such that $2^j \frac{B}{k} \leqslant x_i$ otherwise.*
- *The pseudo-cost y_i of candidate i is $2^{c_i} \frac{B}{k}$. Note that all costs are rounded down except the candidates of cost at most $\frac{B}{k}$.*
- *Let $y_{\sigma(1)}, y_{\sigma(2)}, \dots, y_{\sigma(i)}$ be the pseudo-costs of the first i candidates sorted in nondecreasing order, with ties broken in favor of earlier interviewed candidates. Let h be maximal such that $h \leqslant k$ and $\sum_{j=1}^{h} y_{\sigma(j)} \leqslant 2B$. We then define candidates $\sigma(1), \sigma(2), \dots, \sigma(h)$ to be the* best pseudo-cost solution after i interviews.

Cautious Algorithm Description: Candidate i is hired if and only if candidate i is in the best pseudo-cost solution after i interviews.

2.3 Algorithm Analysis In Observation 1, we observe that pseudo-costs lower bound actual costs. In Lemma 2.1, we show that the Cautious algorithm produces a feasible solution. In Lemma 2.2, we upper bound the cost incurred by the Cautious algorithm in each cost class. Finally, in Theorem 2.1, we show that the Cautious algorithm is $O(\log k)$-competitive on instances where B is an upper bound to optimal.

Observation 1. *If $\frac{B}{k} \leqslant x_i \leqslant B$, then $\frac{x_i}{2} \leqslant y_i \leqslant x_i$.*

Lemma 2.1. *For instances of the Public University Secretary problem, where the cost of the optimal solution is at most B, the Cautious algorithm will hire at least k secretaries.*

Proof. By our assumption, the aggregate cost of the k cheapest secretaries is at most B. By Observation 1, the pseudo-cost lower bounds the actual cost of candidates of original cost at least $\frac{B}{k}$. Let δ denote the permutation of candidates sorted in non-decreasing order of their pseudo-cost. Consider the k candidates $\delta(1) < \cdots < \delta(k)$ with minimum pseudo-cost. It must be the case that $\sum_{j=1}^{k} y_{\delta(j)} \leqslant 2B$ by the earlier observation and the fact that at most k candidates are chosen with pseudo-cost $\frac{B}{k}$. When candidate $\delta(i)$ is interviewed, the i candidates that have arrived with minimum pseudo-cost are $\delta(1), \dots \delta(i)$, and $\sum_{j=1}^{i} y_{\delta(j)} \leqslant 2B$. Thus, by the definition of the Cautious algorithm, candidate $\delta(i)$ will be hired. ∎

Lemma 2.2. *For instances of the Public University Secretary problem, where the cost of the optimal solution is at most B, the Cautious algorithm will hire at most $\frac{k}{2^{j-1}}$ secretaries in cost class j, for $0 \leqslant j \leqslant \lceil \log k \rceil$.*

Proof. The aggregate pseudo-cost of $\frac{k}{2^{j-1}} + 1$ secretaries of cost class j is $(2^j \frac{B}{k})(\frac{k}{2^{j-1}} + 1)$, which is strictly greater than B. Thus the only candidates of cost class j that can possibly be hired are the first $\frac{k}{2^{j-1}}$ ones that are interviewed. ∎

Theorem 2.1. *For instances of the Public University Secretary problem, where the cost of the optimal solution is at most B, the Cautious algorithm incurs a cost of at most $O(B \log k)$ (even with adversarial ordering).*

Proof. By Lemma 2.1, the Cautious algorithm outputs a feasible solution. By Lemma 2.2, the aggregate pseudo-cost that the Cautious algorithm incurs on candidates in cost class j is at most $(2^j \frac{B}{k})(\frac{k}{2^{j-1}}) = 2B$. The aggregate actual cost is at most $4B$ because actual costs are rounded down by at most a factor 2 to obtain pseudo-costs. As there are only $O(\log k)$ cost classes, the result follows. ∎

3 Lower Bounds

In this section, we give a lower bound on the competitiveness of any (deterministic or randomized) algorithm for the Public University Secretary problem. We start with adversarial ordering in subsection 3.1 and then in subsection 3.2 we build on these ideas for random ordering. In each case, we first show that any reasonably competitive algorithm must have a property that we call the acceptance property, and then show that every algorithm with the acceptance property has to be $\Omega(\log k)$-competitive. But we first need to define these terms.

Definition 3.

- *For a particular instance, let Opt be the aggregate cost of the k cheapest secretaries.*
- *We define a randomized algorithm to be* reasonably competitive with adversarial ordering *if on all instances with Opt $\leqslant B$, it is the case that there exists a function $f(n, k)$ such that the expected cost incurred by the algorithm is at most $f(n, k)B$ with adversarial ordering.*
- *We define a randomized algorithm to be* reasonably competitive with random ordering *if on all instances with Opt $\leqslant B$, it is the case that there exists a function $f(n, k)$ such that the expected cost incurred by the algorithm is at most $f(n, k)B$ with random ordering.*
- *Define $s(i)$ to be the maximum number of candidates that can be hired among the first i candidates interviewed subject to the constraint that the aggregate weight is at most B.*
- *We say that an algorithm has the* acceptance property *if for all instances where Opt $= B$, and all interview orderings, and for all candidates i in that ordering, with probability one the algorithm has hired $s(i)$ or more candidates among the first i candidates.*

3.1 Adversarial Ordering

Lemma 3.1. *Every randomized algorithm that is reasonably competitive with adversarial ordering has the acceptance property.*

Proof. Assume to reach a contradiction that there is an instance I, interview ordering σ, and a candidate i, such that with some positive probability the algorithm has not hired $s(i)$ candidates after considering candidate i. By taking the smallest i where this occurs, we can assume that the algorithm did not hire candidate i. Let C be the maximum cardinality subcollection of the first i candidates with aggregate cost at most B. So $s(i)$ is the cardinality of C.

We now construct an instance I' and an ordering σ' on which the algorithm is not reasonably competitive. This new instance I' and its ordering σ' agree with I and σ up through candidate i. The remaining portion of σ' consists of $k - s(i)$ candidates, where $k - s(i)$ of these candidates have cost $(B - \sum_{j \in C} x_j)/(k - s(i))$, followed by one candidate that has some large cost L. The optimal solution is to hire the candidates in C (which includes i) and the $k - s(i)$ candidates interviewed immediately after candidate i. But because the algorithm did not hire candidate i with some positive probability, it has to hire the cost L candidate with positive probability. The algorithm's expected cost can be made arbitrarily large by letting L approach infinity. ∎

Lemma 3.2. *For every randomized algorithm for the Public University Secretary problem there are instances where $B =$ Opt and the expected cost incurred by the algorithm with adversarial ordering is $\Omega(B \cdot \log k)$.*

Proof. For simplicity, take k to be a power of 2. Candidates will arrive in $1 + \log k$ batches, beginning with batch 0 and ending with batch $\log k$. For $i \in \{0, \ldots, \log k\}$, batch i contains 2^i secretaries, each with cost $\frac{\text{Opt}}{2^i}$. Each batch has total cost Opt. Observe that the last batch contains k secretaries of cost $\frac{\text{Opt}}{k}$ each, so the optimal offline solution is this batch.

The single candidate in batch 0 must be hired, by Lemma 3.1. More generally, after all the candidates in batch i have been interviewed, the algorithm must have accepted at least 2^i secretaries, by Lemma 3.1. The cheapest way to maintain this invariant is to hire 2^{i-1} secretaries from batch i, for $i = 1, 2, \ldots, \log k$. Thus any algorithm must incur a cost of at least $\sum_{i=1}^{\log k} 2^{i-1} \cdot \frac{\text{Opt}}{2^i} = \frac{1}{2} \cdot \log k \cdot \text{Opt} = \Omega(\text{Opt} \cdot \log k)$ cost. ∎

3.2 Random Ordering

LEMMA 3.3. *Every randomized algorithm that is reasonably competitive in the random order model has the acceptance property.*

Proof. Assume to reach a contradiction that there is an instance I, and interview ordering σ, and a candidate i, such that with some positive probability the algorithm has not hired $s(i)$ candidates after considering candidate i. By taking the smallest i where this occurs, we can assume that the algorithm did not hire candidate i with positive probability when candidates arrive in this order. Let C be the maximum cardinality subcollection of the first i candidates with aggregate cost at most Opt. So $s(i)$ is the cardinality of C.

We now construct an instance I' on which the algorithm is not reasonably competitive in the random order model. This new instance I' consists of the candidates in I that are one of the first i candidates in the σ order, and $k-s(i)+1$ additional candidates, where $k-s(i)$ of these candidates have cost $(B-\sum_{j\in C} x_j)/(k-s(i))$, and one candidate that has large cost L. With probability at least $\frac{1}{n!}$ the candidates I arrive before all candidates in $I'\backslash I$ in the order σ. Here n is the number of candidates. The optimal solution will hire the candidates in C and the $k-s(i)$ candidates that do not have cost L. But because the algorithm did not hire candidate i with some positive probability when the initial portion of the random order up through candidate i is consistent with σ, it has to hire the cost L candidate with positive probability. Even though this order has probability $\frac{1}{n!}$ of occurring, the algorithm's expected cost can be made arbitrarily large by letting L approach infinity. ∎

THEOREM 3.1. *For every randomized algorithm for the Public University Secretary problem there are instances where $B = Opt$ and the expected cost incurred by the algorithm with random ordering is $\Omega(B \cdot \log k)$.*

Proof. We will construct an instance such that, with constant probability, a random permutation of the candidates intuitively looks like the adversarial instance.

Consider the adversarial instance from the proof of Lemma 3.2. We create a new instance with several copies of each candidate in the adversarial instance. The number of candidates of batch i is significantly greater than the number of secretaries of batch $i+1$. Set $C = 8k$. For $i \in \{0, \dots, \log k\}$, there will be $C^{\log k - i}$ copies of each of the 2^i candidates in batch i, each of which has cost $\frac{\text{Opt}}{2^i}$. This new instance contains $n = \sum_{i=0}^{\log k} 2^i C^{\log k - i}$ candidates, and the k cheapest candidates have total cost Opt.

By Lemma 3.3, it suffices to show that with constant probability, it is the case that for all batches i, at least 2^i candidates in batch i arrive before any candidate of batch $i+1, \dots \log k$. This is because of the following. Let candidate j be the 2^{i^*}th candidate of batch i^* to arrive. Assume no candidates of batch larger than i^* arrive before j. Then any algorithm satisfying the acceptance property accepts at least 2^{i^*} candidates from batch i^* or less at the time j arrives. Note that if this arrival condition holds for all batches then any algorithm satisfying the acceptance property has cost $\Omega(B \log k)$.

Formally, let A_i be the event that at least 2^i candidates of batch i are interviewed before any candidates in batches $i+1, \dots, \log k$ for $i = 0, \dots, \log k - 1$. We will show that $\mathbb{P}\left(\bigcap_{i=0}^{\log k - 1} A_i\right) \geqslant 1/2$.

We bound the probability of each event $\overline{A_i}$ and then we apply a union bound to $\mathbb{P}\left(\bigcup_{i=0}^{\log k - 1} \overline{A_i}\right)$. Let r_i denote the ratio of the total number of candidates in batches $i+1, \dots, \log k$ to the total number of elements in batches $i, \dots, \log k$. Then

$$\mathbb{P}(\overline{A_i}) \leqslant 2^i \cdot r_i = 2^i \cdot \frac{\sum_{j=i+1}^{\log k} 2^j C^{\log k - j}}{\sum_{j=i}^{\log k} 2^j C^{\log k - j}} \leqslant 2^i \cdot \frac{2 \cdot 2^{i+1} C^{\log k - (i+1)}}{2^i C^{\log k - i}} = \frac{2^{i+2}}{C}$$

To see why the first inequality holds, recall that A_i is the event that, in the ordering of candidates from batches $i, i+1, \dots, \log k$, the first 2^i candidates are from batch i. So to bound $\overline{A_i}$ we may take a union bound over $B_1, \dots, B_{2^i}$, where B_j is the event that the jth candidate in this ordering is *not* from batch i; so $\mathbb{P}(B_j) = r_i$. Thus we have

$$\mathbb{P}\left(\bigcup_{i=0}^{\log k - 1} \overline{A_i}\right) \leqslant \sum_{i=0}^{\log k - 1} \frac{2^{i+2}}{C} \leqslant \frac{4k}{C} = \frac{1}{2}$$

as desired. This completes the proof as the expected cost of any algorithm satisfying the acceptance property is $\Omega(B \log k)$.

∎

REMARK 1. *The proof of Theorem 3.1 also gives a lower bound of $\Omega(B\sqrt{\log n})$ on the expected cost incurred by the algorithm.*

References

[1] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. "Secretary and online matching problems with machine learned advice". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7933–7944.

[2] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. "A Knapsack Secretary Problem with Applications". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings.* Ed. by Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim. Vol. 4627. Lecture Notes in Computer Science. Springer, 2007, pp. 16–28.

[3] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. "Matroids, secretary problems, and online mechanisms". In: *ACM-SIAM Symposium on Discrete Algorithms, SODA*. SIAM, 2007, pp. 434–443.

[4] Peter Borosan and Mudassir Shabbir. *A Survey of Secretary Problem and its Extensions.* https://www.researchgate.net/profile/Mudassir-Shabbir/publication/242097841_A_Survey_of_Secretary_Problem_and_its_Extensions/links/00b7d53482af6d7909000000/A-Survey-of-Secretary-Problem-and-its-Extensions.pdf.

[5] Sourav Chakraborty and Oded Lachish. "Improved Competitive Ratio for the Matroid Secretary Problem". In: *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms.* 2012, pp. 1702–1712.

[6] Jose Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. "Recent Developments in Prophet Inequalities". In: *ACM SIGecom Exchanges* 17.1 (May 2019), pp. 61–70.

[7] Michael Dinitz. "Recent Advances on the Matroid Secretary Problem". In: *ACM SIGACT News* 44.2 (June 2013), pp. 126–142.

[8] Paul Dütting, Silvio Lattanzi, Renato Paes Leme, and Sergei Vassilvitskii. "Secretaries with advice". In: *Proceedings of the 22nd ACM Conference on Economics and Computation.* 2021, pp. 409–429.

[9] Moran Feldman, Ola Svensson, and Rico Zenklusen. "A simple O(log log (rank))-competitive algorithm for the matroid secretary problem". In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM. 2014, pp. 1189–1201.

[10] Kaito Fujii and Yuichi Yoshida. "The Secretary Problem with Predictions". In: *Mathematics of Operations Research* 0.0 (0), null. DOI: 10.1287/moor.2022.0031. eprint: https://doi.org/10.1287/moor.2022.0031. URL: https://doi.org/10.1287/moor.2022.0031.

[11] Robert D. Kleinberg. "A multiple-choice secretary algorithm with applications to online auctions". In: *ACM-SIAM Symposium on Discrete Algorithms, SODA*. SIAM, 2005, pp. 630–631.

[12] Eleanor Klibanoff. *How to close a steel mill: Lessons from Pittsburgh.* https://whyy.org/articles/how-to-close-a-steel-mill-lessons-from-pittsburgh/. PBS station WHYY. 2016.

[13] Oded Lachish. "O (log log rank) competitive ratio for the matroid secretary problem". In: *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE. 2014, pp. 326–335.

[14] *Merrill M. Flood.* https://en.wikipedia.org/wiki/Merrill_M._Flood.

[15] Pranav Nuti and Jan Vondrák. "Secretary problems: The power of a single sample". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2023, pp. 2015–2029.

[16] *Secretary Problem.* https://en.wikipedia.org/wiki/Secretary_problem.