Interviewing Secretaries in Parallel

Moran Feldman*and Moshe Tennenholtz†

Abstract

Motivated by the parallel nature of on-line internet help-desks and human inspections, we introduce the study of interviewing secretaries in parallel, extending upon the study of the classical secretary problem. In our setting secretaries arrive into multiple queues, and are interviewed in parallel, with the aim of recruiting several secretaries in a timely manner. We consider a variety of new problems that fit this setting, and provide both upper and lower bounds on the efficiency of the corresponding interviewing policies, contrasting them with the classical single queue setting.

1 Introduction

On-line help desks have become a central issue in the service industry. For example, companies such as LivePerson conduct on-line chat support in order to solve problems on-line in a timely manner. In such settings, multiple human service providers face, in parallel, a stream of clients. A major constraint is that only a small number of the clients can be transitioned to a higher authority treatment. Thus, the service providers must decide on line, and in parallel, which clients need such a treatment the most.

A similar setting arise from inspection procedures in airports. Here, each inspector is associated with her queue of people. The inspector has to interview the people in her queue serially, and refer the suspected ones to further inquiry. Again, only a limited number of people can be referred to such an inquiry, and therefore, the inspectors have to select on line, and in parallel, the most suspected people.

In both the above examples, there are two types of agents: interviewers and experts. Each interviewer gets a portion of the clients, and interview them serially. Clients that require more than a short interview are then refereed to further treatment by an expert. A similar setting with only one type of agents exists too. Again, each interviewer gets a portion of the clients, and interview them serially. However, if a client needs further treatment, the interviewer herself provides this treatment, and ignores the rest of the clients from this point on (till she is done with the current one).

To illustrate the above consider a situation of two inspectors working in parallel, each interviewing half of the population serially (due to time constraints, it might not be possible to interview serially more people). Assume that each interview takes one time unit, and each inspector is expected to choose one person for detailed inspection immediately after interviewing him, and release all others. We face here a novel secretary problem in which the secretaries randomly partition into two queues, and a single secretary is to be selected from each queue. Comparison is possible only between secretaries belonging to the same queue (because they are interviewed by the same inspector).

*Microsoft Research
†Microsoft Research

The intuitive approach for the above problem is for each inspector to apply the classical secretary algorithm[1] to its queue. However, that turns out to be suboptimal. Also, notice that the parallelism is the factor preventing the inspectors from selecting two secretaries of the same queue. A naïve attempt to improve the system, and still comply with the time constraints, might be as follows. Let one inspector interview half the secretaries, and hire *two* of the secretaries interviewed. The rest of the secretaries are released immediately without being interviewed. It turns out that such a modification strictly decreases the probability of the inspectors to select the right secretaries.

1.1 The Model

In the classical secretary problem [16, 7], the input consists of a set of secretaries and a strict total order among them. The secretaries arrive online in a random order. Each time a secretary arrives, the algorithm can compare him to the previously seen secretaries, and then it must either hire or dismiss the secretary. Both decisions are irreversible. The algorithm can hire only a single secretary, and its objective is to hire the best secretary.

Our model is a generalization of the classical secretary problem. Here the set of secretaries is evenly and randomly partitioned into multiple queues. The order of the secretaries in each queue is also random. Only the first D secretaries of each queue can be interviewed before the deadline, the others are never considered by the algorithm. Each time a secretary arrives, the algorithm can compare him to the previously seen secretaries of that queue (secretaries of different queues can never be compared), and then it must either hire or dismiss the secretary. The algorithm can hire k secretaries, and its objective is to hire as many top k secretaries as possible, where k is a parameter of the problem.

Two sub-models correspond to the two kinds of settings discussed above. In the first sub-model, the algorithm can hire at most one secretary from each queue. This sub-model corresponds to the case where the interviewer has to select one client to whom she will provide further treatment on her own. We say that this sub-model has "exclusive" positions, because each queue is associated with a single secretary position that can be manned from this queue only.

In the other sub-model we consider, the algorithm can hire secretaries from all queues, as long as less than k secretaries have been hired so far. This sub-model corresponds to the case where clients selected for further treatment are transferred from the interviewer to an expert; leaving the interviewer to continue inspecting clients.

Information shared between policies of different queues. In the above description of the model, we assume that secretaries of different queues cannot be compared. The idea behind this assumption is that scores given by different interviewers are not comparable (*e.g.*, some interviewers might be harsher than others). Let us argue that once we make this assumption, the only meaningful information that can be shared between policies is the number of secretaries hired so far by each policy. From the viewpoint of every single policy, its input is simply a random permutation. Moreover, these random permutations are independent. Thus, information about the relative value of secretaries in the input of one policy is of no use to the other policies. Despite of this argument, our hardness results assume nothing beside the inability to compare secretaries of different queues.

[1] The classical secretary algorithm skips the first $1/e$ fraction of the secretaries, and then hire the first secretary better than any previous one [16, 7].

Table 1: Subcases for $k = 2$ and $D = n/2$

Sub Case	**Algorithm**		**Hardness**	
One Queue	0.192	[Cor 4.6]	0.266	[Thm 4.3]
Two queues with "exclusive" positions	0.319	[Cor 4.10]	0.321	[Thm 4.12]
Two queues with "shared" positions	0.356	[Cor 4.17]	-	

Table 2: Subcases for large k and $D = n/k$

Sub Case	**Algorithm**		**Hardness**	
One Queue	$1/k - o(1)$	[Thm 6.1]	$1/k$	[Thm 6.1]
k queues with "exclusive" positions	0.288	[Thm 6.6]	0.816	[Thm 6.2]
k queues with "shared" positions	0.288	[Thm 6.6]	0.301	[Thm 6.4]

1.2 Our Results

We first consider the case of $k = 1$ and $D = n/d$, *i.e.*, we need to hire a single secretary, and each interviewer has time to interview only a fraction of $1/d$ of the secretaries. This is a simple setting for which we can give tight results. We compare the two extreme cases of a single queue and d queues (having more than d queues clearly does not help the algorithm). For a single queue we show that the best possible competitive ratio is $(de)^{-1}$. On the other hand, d queues allow a competitive ratio of $d^{-d/(d-1)} - o(1)$. Notice that for large d, the improvement in the competitive ratio approaches a factor of e. The last result is tight up to low order terms.

Next, we consider the case of $k = 2$ and $D = n/2$. This setting allows us to compare "exclusive" and "shared" positions, and still is simple enough to produce strong results. We consider three subcases which are summarized in Table 1.

Table 1 shows that the three sub cases considered have strictly different optimal competitive ratios. As one might expect, two queues with "shared" positions is the strongest mechanism. However, the relation between the two other mechanisms is not that easy to predict. Two queues with "exclusive" positions gets to interview twice as many secretaries as the one queue mechanism, but enjoys less freedom since it must hire at most one secretary from each queue.

We next shift our attention to hiring a large number of secretaries, *i.e.*, $n \gg k \gg 1$. We consider two extreme values for D: $n/2$ and n/k. If $D = n/2$, then we get a competitive ratio of $1 - o(1)$ for two queues and $1/2 - o(1)$ for one queue. Both ratios are tight up to the $o(1)$ term. The results for $D = n/k$ are summarized in Table 2.

There is one important result which does not appear in Table 2. An intuitive strategy for the case of k queues with "exclusive" positions is to apply the classical secretary algorithm to each queue independently. We give evidence that the approximation ratio of this policy is only about 0.274, *i.e.*, suboptimal.

1.3 Related Work

The Classical Secretary Problem was introduced during the 60's of the 20th century, nobody is sure exactly when [16, 7, 9]. Since its introduction, many variants of the problem have been proposed and researched. We survey here only the most relevant ones.

One of the most common extensions of the classical problem allows the algorithm to hire up to k secretaries. For the case where every subset of k secretaries can be hired, two incomparable competitive ratios of e^{-1} and $1-O(k^{-0.5})$ were obtained by Buchbinder et al. [2] and Kleinberg [13], respectively. Related problems impose either knapsack [2] or matroid constraints on the set of secretaries hired [3, 6, 12, 14].

Babaioff et al. [1] consider time constraints imposed on the interviewing process. In their model secretaries lose value over time, making it more profitable to hire a somewhat inferior but early secretary than a better secretary that arrives late. The use of soft time constraints in this model make it very different from ours.

Another interesting line of work consider ground sets with a partial order [10, 15] (as opposed to the full order assumed in the classical problem). Partial order implies that the algorithm cannot compare every pair of two secretaries. This resembles the incomparability of secretaries of different queues in our model. However, under partial order, a pair of secretaries is always either comparable or incomparable, whereas, every pair of secretaries in our model is comparable with some probability.

Buchbinder et al. [5] show an interesting relation between linear programs (LPs) and many variants of the secretary problem. An LP represents a problem $\mathcal{P}$ if any feasible solution x of the LP with value $V(x)$ implies an algorithm for $\mathcal{P}$ with a competitive ratio of $V(x)$, and vice versa. Buchbinder et al. [5] show how to construct an LP representing some variants of the secretary problem, including the classical one. Such LPs are useful for two reasons: finding an optimal solution for a given n (the number of secretaries), and proving hardness results using dual-fitting.

Fekdnab et ak, [8] suggest using an alternative view of the arrival process. Instead of assuming the secretaries arrive at a random order, [8] assumes the secretaries arrive at random times during the interval $[0, 1]$. Under most variants of the secretary problem, both arrival processes are equivalent; however, the alternative view is often the easier one to analyze. Feldman et al. [8] demonstrate this observation by improving the competitive ratio known for a few submodular secretarty problems (*i.e.*, secretary problems with a submodular objective function). Submodular secretary problems were also considered in other works [4, 11].

2 Preliminaries

In all problems considered in this paper, the input includes of a set $\mathcal{S}$ of n secretaries, and a strict total order $\prec$ on the secretaries of $\mathcal{S}$. We say that secretary s is *better* than s' if $s \prec s'$. The objective of the algorithm is to hire up to k secretaries. For every secretary s hired, the algorithm gets a point if s is one of the top k secretaries in $\mathcal{S}$.[2]

In every single problem there are two additional parameters Q and D. The parameter Q determines the number of input queues (we assume for simplicity that Q divides n), and is considered to be a constant. A distinct random subset of n/Q secretaries of $\mathcal{S}$ arrive to each queue in random order.[3] The parameter D determines the number of secretaries of each queue that can be interviewed (before the deadline).

An algorithm ALG for our problems describes a policy for each one of the queues. A policy P of a queue q considers the secretaries of q sequentially. For secretary s considered, P must decide,

[2] A similar model associates an adversary chosen distinct value with every secretary. Under this model, the algorithm observes the value of each arriving secretary, and its revenue from hiring the secretary is equal to the observed value. Our algorithms work for this model as well, and yield the same competitive ratios.

[3] The following is an alternative view of the input model. The n secretaries arrive at random order. The secretaries at positions $1, Q+1, 2Q+2, \ldots$ are sent to the first queue, the secretaries at positions $2, Q+2, 2Q+2, \ldots$ are sent to the second queue, and so on.

irreversibly, whether to hire s. When deciding about s, the policy P has access only to the following information.

I1 The total number of secretaries.

I2 The relation $\prec$ between the secretaries of q already seen.

I3 The number of secretaries already hired from each one of the queues.

The k positions available for secretaries can be either "exclusive" or "shared". If the positions are "exclusive", then every position is associated with a single queue; and only the policy of this queue can hire for that position. On the other hand, if the positions are "shared", every policy of every queue can hire to each one of the positions.

For "shared" positions, we need to assume that all policies of all queues advance at equal rates (*i.e.*, they all consider the secretary at position i of their respective queues before any of them consider the secretary at position $i+1$ of their respective queue). The order in which the secretaries at position i are considered does not affect our results. For simplicity, we assume this order agrees with the indexes of the queues. Observe that for "exclusive" positions, the assumption of equal rates and **I3** are redundant.

The competitive ratio of ALG is defined as the ratio between the expected value of the algorithm (*i.e.*, the expected number of points gained by the algorithm), and k, where the expectation is over the randomness of the arrival model.

2.1 Random Arrival Times

Recall that in the classical secretary problems, it is assumed that the secretaries arrive in a random order. This arrival model is used by most of the results mentioned above. Following [8], our algorithms are described in terms of a somewhat different arrival model. In this model, each queue q gets a random disjoint subset of D secretaries. Each of these D secretaries arrive at a random time from the range $[0, 1]$. Notice that if $D < n/Q$, then some of the secretaries never arrive to any of the queues. Alternatively, we can think that each queue gets n/Q secretaries, but $n/Q - D$ secretaries of each queue arrive too late to be interviewed. For "exclusive" positions the arrival model described above reduces to this model as described in Algorithm 1.[4]

ALGORITHM 1: Reduction

```
for each queue q do
    Choose a set T_q of D random arrival times.
    Sort the times of T_q.
    Assign the times of T_q sequentially to the first D secretaries of q upon arrival.
end
```

For "shared" positions, the two models are not equivalent because the assumption that all queues advance at equal rates cannot be stated in terms of the random arrival times model. To bypass this problem, we use the idea of "well-representation". Given an input $\mathcal{I}_P$ for the random arrival order model (random permutation), Algorithm 1 produces a random input $\mathcal{I}_T$ for the random arrival times model. Let $i(s, \mathcal{I}_P)$ denote the index of the position at which secretary s appears, in its queue, in input $\mathcal{I}_P$, and let $t(s, \mathcal{I}_T)$ denote the arrival time of secretary s in input $\mathcal{I}_T$. We say

[4] In fact the two models are equivalent.

that $\mathcal{I}_T$ is a *well-representation* of $\mathcal{I}_P$ if for every two secretaries $s, s' \in \mathcal{S}$: $i(s, \mathcal{I}_P) \leq i(s', \mathcal{I}_P) \Rightarrow t(s, \mathcal{I}_T) < t(s', \mathcal{I}_T) + D^{-1/3}$.

Well-representation gives additional relation between the positions and times which allows us to take the assumption that the queues advance at equal rates into consideration when analyzing our algorithms. Moreover, the following lemma shows that well-representation happens with high probability.

Lemma 2.1. *Given an input $\mathcal{I}_P$ for the random arrival order model, Algorithm 1 produces a random input $\mathcal{I}_T$ for the random arrival times model which is a well representation of $\mathcal{I}_P$ with probability $1 - o(1)$.*

Proof. Algorithm 1 uses n random time variables. Let X_1 and X_2 be two such random variables, and let s_1 and s_2 be the random secretaries assigned the random arrival times X_1 and X_2. If X_1 and X_2 are associated with the same queue, then clearly s_1 and s_2 will not violate the well-representation property. Hence, we can concentrate from now on on variables X_1 and X_2 associated with different queues.

Fix the random times X_1 and X_2. We would like to upper bound the the probability that s_1 and s_2 violate the well-representation property. Assume without loss of generality $X_1 \leq X_2$. s_1 and s_2 violate the well-representation property if and only if the two following conditions hold:

C1 $X_1 \leq X_2 - D^{-1/3}$.

C2 $i(s_1, \mathcal{I}_P) \geq i(s_2, \mathcal{I}_P)$.

Assume **C1**, *i.e.*, $X_1 \leq X_2 - D^{-1/3}$, and let us upper bound the probability that **C2** holds as well. Let q_1 and q_2 be the two queues corresponding to X_1 and X_2, respectively. For the purpose of this proof we allow a queue to have less than D times. If this is the case, the secretaries are assigned times sequentially upon arrival, till no more times are available. Secretaries that arrive after all times have been exhausted get no time.

Consider the following process. The process has D steps. Let us denote the values of $i(s_1, \mathcal{I}_P)$ and $i(s_2, \mathcal{I}_P)$ at step j by $i_j(s_1, \mathcal{I}_P)$ and $i_J(s_2, \mathcal{I}_P)$, respectively. At step 0, both queues have a single time allocated to each one of them: X_1 to q_1 and X_2 to q_2. The first secretary of q_1 gets time X_1 and becomes s_1. Similarly, the first secretary of q_2 becomes X_2. Hence, $i_0(s_1, \mathcal{I}_P) = i_0(s_2, \mathcal{I}_P) = 1$. In each step we add two additional random times, one for q_1 and one for q_2, and update the choice of s_1 and s_2. Observe that for every $1 \leq j \leq D - 1$,

$$i_j(s_1, \mathcal{I}_P) - i_{j-1}(s_1, \mathcal{I}_P) = \begin{cases} 1 & \text{with probability } X_1 \\ 0 & \text{otherwise} \end{cases} .$$

A similar observation holds for $i_j(s_2, \mathcal{I}_P) - i_{j-1}(s_2, \mathcal{I}_P)$ also, with X_2 replacing X_1. Moreover, both quantities are independent (because $q_1 \neq q_2$). Let us define for every $0 \leq j \leq D - 1$, $Y_j = i_j(s_1, \mathcal{I}_P) - i_j(s_2, \mathcal{I}_P) + j \cdot (X_2 - X_1)$. Notice that $Y_0 = i_0(s_1, \mathcal{I}_P) - i_0(s_2, \mathcal{I}_P) = 0$. Consider the quantity $Y_j - Y_{j-1}$ for some $1 \leq j \leq D - 1$. It can be observed that:

$$Y_j - Y_{j-1} = \begin{cases} 1 + X_2 - X_1 & \text{with probabilitiy } X_1(1 - X_2) \\ X_2 - X_1 & \text{with probabilitiy } X_1X_2 + (1 - X_1)(1 - X_2) \\ -1 + X_2 - X_1 & \text{with probabilitiy } X_2(1 - X_1) \end{cases}$$

Hence, $\mathbb{E}[Y_j - Y_{j-1}] = 0$, and the series $\{Y_j\}_{j=0}^{D-1}$ is a martingale. Using Azuma's inequality, we

get:

$$\Pr[i_{D-1}(s_1,\mathcal{I}_P) \geq i_{D-1}(s_1,\mathcal{I}_P)] = \Pr[Y_{D-1} \geq (D-1)(X_2 - X_1)] \\ \leq e^{-\frac{(D-1)^2(X_2-X_1)^2}{2(D-1)(1+X_2-X_1)^2}} \overset{(*)}{\leq} e^{-\frac{(D-1)D^{2/3}}{8}} \leq e^{-\frac{D^{1/3}}{9}} ,$$

where (*) holds since we assumed **C1**, which implies: $1 \geq X_2 - X_1 \geq D^{1/3}$. Notice that at step $D-1$, both q_1 and q_2 already have D random times, and therefore, $i_{D-1}(s_1,\mathcal{I}_P) = i(s_1,\mathcal{I}_P)$ and $i_{D-1}(s_2,\mathcal{I}_P) = i(s_2,\mathcal{I}_P)$. Thus, the last inequality implies that given **C1**, with probability at least $1 - e^{-\frac{(n/k)^{1/3}}{9}}$, **C2** does not hold. Hence, with probability $1 - e^{-\frac{(n/k)^{1/3}}{9}}$, s_1 and s_2 does not violate the well-representation property.

The number of pairs of random time variables is less than n^2, and therefore, by the union bound, with probability at least $1 - n^2 e^{-\frac{(n/k)^{1/3}}{9}} = 1 - o(1)$, no pair of random times are assigned to two secretaries violating the well-representation property, *i.e.*, $\mathcal{I}_T$ is a well-representation of $\mathcal{I}_P$. □

2.2 Hardness Results

Some of the hardness results we describe are based on the method of [5]. This method works as following. For a given number n of secretaries, one constructs a primal maximization LP with the following properties.

- The variables of the primal LP represent the probabilities of different events assuming an arbitrary algorithm ALG is applied to the problem.
- The objective function of the primal LP is the competitive ratio of ALG, resulting from the above probabilities.
- The constraints of the primal LP are inequalities that the above probabilities must obey regardless of the algorithm ALG considered.

Clearly, any algorithm ALG induces a solution for the primal LP whose value is the competitive ratio of this algorithm for n secretaries. Hence, any solution for the dual LP provides an upper bound on the best achievable competitive ratio for n secretaries. In this paper we get hardness results using this machinery via two methods:

- Upper bounding the value of the dual solution for every n.
- Finding a dual solution of value at most α for some n_0 secretaries, and proving that any algorithm with a competitive ratio $\beta > \alpha$ for some $n > n_0$ implies a β-competitive ratio for n_0 as well.

3 A Single Position

In this section we consider the case that we want to hire only a single secretary, *i.e.*, $k = 1$. The single position available is of course "shared" among the queues (as opposed to being "exclusive" for one queue). This is a relatively simple case on which we can demonstrate many of our techniques. Assume the deadline $D = n/d$ for some positive integer d, *i.e.*, we can interview at most n/d secretaries of each queue. We compare two extreme cases $Q = 1$ and $Q = d$. Due to space constraints, some proofs of this section are deferred to Appendix A.

3.1 A Single Queue

Let us start with the case $Q = 1$. In this case we have a single queue, but we can interview only the first n/d secretaries of this queue.

Theorem 3.1. *There is a $(de)^{-1}$-competitive algorithm for the case $Q = 1$ and $D = n/d$, and this is the best possible up low order terms.*

3.2 d Queues

We now consider the case of $Q = d$, *i.e.*, each queue gets n/d secretaries, and all secretaries can be interviewed. Observe that since we cannot compare secretaries of different queues, it is not possible even after interviewing all secretaries to determine which queue contained the best secretary. Hence, we might guess that this case is equivalent to the previous one (here we must guess the right queue, whereas in the previous case the input was a randomly chosen queue). However, this intuition is not correct, as we prove in this section.[5]

ALGORITHM 2: Algorithm for $k = 1$, $Q = d$ and $D = n/d$

```
for each queue independently do
    Wait till time t = d^{-1/(d-1)}.
    Let s be the first secretary after time t which is better than any previously seen secretary.
    if no secretary was hired before (by another queue) then
        Hire s.
    end
end
```

We suggest Algorithm 2 for the problem. Let s_1 be the best secretary, and let q_1 be the queue it arrives to. We now define a set C as following. Given an input $\mathcal{I}_P$ of the random order model (such an input is simply a random permutation of the secretaries of $\mathcal{S}$), let $\mathcal{I}_T$ be any input of the random arrival times model that might be produced from $\mathcal{I}_P$ using Algorithm 1. The pair $(\mathcal{I}_P, \mathcal{I}_T)$ is in C if all the following conditions hold:

D1 $t(s_1, \mathcal{I}_T) > t$.

D2 The best secretary of q_1 in the range $[0, t(s_1, \mathcal{I}_T))$ arrives before time t, or this range is empty.

D3 In any queue other than q_1, the best secretary in the range $[0, \min\{1, t(s_1, \mathcal{I}_T) + D^{-1/3}\}]$ arrives before time t, or this range is empty.

A random pair $(\mathcal{I}_P, \mathcal{I}_T)$ is a pair constructed as following. A random input $\mathcal{I}_P$ for the random order model is selected. Algorithm 1 is them used to produce from $\mathcal{I}_P$ an input $\mathcal{I}_T$ for the random arrival times model.

Observation 3.2. *The arrival time of every secretary in a random pair $(\mathcal{I}_P, \mathcal{I}_T)$ is uniformly random and independent of the arrival times of the other secretaries.*

Proof. The same distribution of random pairs can also be constructed as following. Construct $\mathcal{I}_T$ by choosing a random partitioning of the secretaries to queues, and an independent arrival time for each secretary. Next, the order induced by the arrival times of the secretaries in each queue becomes their order in $\mathcal{I}_P$. □

[5] For off-line algorithms, which can interview all secretaries before making their decisions, this intuition turns out to be correct. For such algorithms it can be easily shown that the best approximation ratio under both cases is d^{-1}.

Lemma 3.3. *For a random pair* $(\mathcal{I}_P, \mathcal{I}_T)$, $\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C] \geq d^{-d/(d-1)} - o(1)$.

Proof. Let A_x be the event that s_1 arrives at time x. Given that A_x occurs for some $x > t$, we know that:

- **D1** is guaranteed to hold.
- Due to symmetry arguments, **D2** holds with probability t/x.
- Due to similar symmetry arguments, **D3** holds with probability $[t/(\min\{x + D^{-1/3}, 1\})]^{Q-1}$.
- **D2** and **D3** are independent.

Thus,

$$\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C | A_x] = \frac{t}{x} \cdot \left(\frac{t}{\min\{x + D^{-1/3}, 1\}} \right)^{Q-1} .$$

The probability that s_1 arrives in an interval of size ℓ is ℓ. Hence, the probability it arrives in an infinitesimal interval of size dx is dx. Therefore, by the law of total probability, the probability $Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C]$ is lower bounded by:

$$\int_t^1 \frac{t}{x} \cdot \left(\frac{t}{\min\{1, x + D^{-1/3}\}} \right)^{Q-1} dx = \int_t^1 \frac{t}{x} \cdot \left(\frac{t}{x} \right)^{Q-1} dx - o(1) = \int_t^1 \left(\frac{t}{x} \right)^{Q} dx - o(1)$$

$$\begin{aligned}
&= -\frac{t^Q}{(Q-1)x^{Q-1}}\Big|_t^1 - o(1) \\
&= \frac{t^Q}{(Q-1)t^{Q-1}} - \frac{t^Q}{Q-1} - o(1) \\
&= \frac{t(1-t^{Q-1})}{Q-1} - o(1) = \frac{d^{-1/(d-1)}(1-1/d)}{d-1} - o(1) \\
&= d^{-d/(d-1)} - o(1) \ . \quad \square
\end{aligned}$$

$\square$

Corollary 3.4. *For a random pair* $(\mathcal{I}_P, \mathcal{I}_T)$*, with probability at least* $d^{-d/(d-1)} - o(1)$, $(\mathcal{I}_P, \mathcal{I}_T) \in C$ *and* $\mathcal{I}_T$ *is well-representation of* $\mathcal{I}_P$.

Proof. Follows from Lemmata 2.1 and 3.3 and the union bound. $\square$

We say that Algorithm 2 got a pair $(\mathcal{I}_P, \mathcal{I}_T)$ if the original input for the random order model was $\mathcal{I}_P$, and it was converted to the input $\mathcal{I}_T$ of the random arrival times model by Algorithm 1.

Lemma 3.5. *Assuming Algorithm 2 gets a pair* $(\mathcal{I}_P, \mathcal{I}_T) \in C$ *as input. Then, if* $\mathcal{I}_T$ *is a well-representation of* $\mathcal{I}_P$*, then Algorithm 2 hires* s_1.

Proof. We say that a queue q attempts to hire a secretary s if s is a secretary of queue q, and Algorithm 2 gets to Line 4 with this secretary. Since $(\mathcal{I}_P, \mathcal{I}_T) \in C$, we know that **D1** and **D2** hold. Thus, queue q_1 attempts to hire s_1. In order to prove that Algorithm 2 hires s_1, we are left to show that no other queue attempts to hire a secretary s with $i(s, \mathcal{I}_P) \leq i(s_1, \mathcal{I}_P)$.

Assume some queue $q \neq q_1$ attempts to hire a secretary s. Since **D3** holds, the secretary s that the queue attempts to hire must arrive after time $t_T(s_1) + (k/n)^{1/3}$. However, since $\mathcal{I}_T$ is a well-representation of $\mathcal{I}_P$, this this implies $i(s_1, \mathcal{I}_P) < i(s, \mathcal{I}_P)$. $\square$

Corollary 3.6. *Algorithm 2 is a $(d^{-d/(d+1)} - o(1))$-competitive algorithm.*

Proof. Corollary 3.4 and Lemma 3.5 imply that with probability $d^{-d/(d+1)}$, Algorithm 2 hires s_1 when given a random pair. □

Next, we use the method of [5] to prove that Algorithm 2 is optimal up to low order terms. Consider the following linear program.

$$\begin{array}{lllll} \text{(LP1)} & \max & \frac{1}{n} \cdot \sum_{i=1}^{d} \sum_{j=1}^{n/d} f_{i,j} & & \\ & \text{s.t.} & f_{i,j} + \sum_{h=1}^{d} \sum_{\ell=1}^{j-1} \frac{f_{h,\ell}}{\ell} + \sum_{h=1}^{i-1} \frac{f_{h,j}}{j} & \leq 1 & \forall\, 1 \leq i \leq d, 1 \leq j \leq n/d \\ & & f_{i,j} & \geq 0 & \forall\, 1 \leq i \leq d, 1 \leq j \leq n/d \end{array}$$

Lemma 3.7. *The optimal value of (LP1) is an upper bound on the competitive ratio of any algorithm for the case $k = 1$, $Q = d$ and $D = n/d$.*

Finding the optimal assignment to (LP1) is difficult. Instead, we upper bound the optimal value of (LP1) using its dual (LP2).

$$\begin{array}{lllll} \text{(LP2)} & \min & \sum_{i=1}^{d} \sum_{j=1}^{n/d} y_{i,j} & & \\ & \text{s.t.} & y_{i,j} + \sum_{h=1}^{d} \sum_{\ell=j+1}^{n/d} \frac{y_{h,\ell}}{j} + \sum_{h=i+1}^{d} \frac{y_{h,j}}{j} & \geq \frac{1}{n} & \forall\, 1 \leq i \leq d, 1 \leq j \leq n/d \\ & & y_{i,j} & \geq 0 & \forall\, 1 \leq i \leq d, 1 \leq j \leq n/d \end{array}$$

Lemma 3.8. *(LP2) has a solution of value at most $d^{-d/(d-1)} + O(n^{-0.5})$.*

Corollary 3.9. *No algorithm for the case $k = 1$, $Q = d$ and $D = n/d$ has a better competitive ratio than $d^{-d/(d-1)} + O(n^{-n})$.*

Proof. Follows from Lemmata 3.7 and 3.8, and the observation that any solution of (LP2) provides an upper bound on the optimal solution of (LP1). □

4 Two Positions

In this section we consider the case of $k = 2$ and $D = n/2$, *i.e.*, two positions need to be manned and at most half the secretaries can be interviewed in each queue. We devote a section for this case because on the one hand it is complex enough so that new concepts such as "exclusive" positions can be presented, and on the other hand it simple enough to produce many (almost) tight results. We consider the following subcases. Due to space constraints some of the proofs of this section are deferred to Appendix B.

- One queue, *i.e.*, $Q = 1$.
- Two queues with "exclusive" positions, *i.e.*, $Q = 2$ and at most one secretary is hired from each queue.
- Two queue with "shared" positions, *i.e.*, $Q = 2$ and secretaries can be hired from both queues as long as there is still an unmanned position.

Our objective in this section is to prove the following claim.

Claim 4.1. *The above cases are in a strictly increasing competitive ratios order.*

4.1 One Queue

In this section we consider the sub case where there is only one queue, *i.e.*, $Q = 1$. Notice that under these settings, only the first half of the secretaries can be interviewed. We begin our study by showing a hardness result for this case. Consider the following linear program.

$$
\begin{array}{lll}
\text{(LP3)} \quad \max & \frac{1}{2}\left[\frac{1}{n}\cdot\sum_{i=1}^{n/2} f_i + \frac{1}{n}\cdot\sum_{i=1}^{n/2}\left(\frac{n-i}{n-1}\cdot f_i + \frac{i-1}{n-1}s_i\right)\right] & \\
\text{s.t.} & f_i^1 + f_i^2 & = f_i \quad \forall\, 1 \le i \le n/2 \\
& s_i^1 + s_i^2 & = s_i \quad \forall\, 1 \le i \le n/2 \\
& f_i^1 + f_1^1 + \sum_{j=2}^{i-1}\frac{f_j^1+s_j^1}{j} & \le 1 \quad \forall\, 2 \le i \le n/2 \\
& s_i^1 + f_1^1 + \sum_{j=2}^{i-1}\frac{f_j^1+s_j^1}{j} & \le 1 \quad \forall\, 2 \le i \le n/2 \\
& f_i^2 - f_1^1 + \sum_{j=2}^{i-1}\frac{f_j^2+s_j^2-f_j^1-s_j^1}{j} & \le 0 \quad \forall\, 2 \le i \le n/2 \\
& s_i^2 - f_1^1 + \sum_{j=2}^{i-1}\frac{f_j^2+s_j^2-f_j^1-s_j^1}{j} & \le 0 \quad \forall\, 2 \le i \le n/2 \\
& f_1^2 & = 0 \\
& f_i^1, s_i^1, f_i^2, s_i^2 & \ge 0 \quad \forall\, 1 \le i \le n/2
\end{array}
$$

Lemma 4.2. *The optimal value of (LP3) is an upper bound on the competitive ratio of any algorithm for the case $k = 2$, $Q = 1$ and $D = n/2$.*

Theorem 4.3. *No algorithm is better than* 0.266*-competitive for the case $k = 2$, $Q = 1$ and $D = n/2$.*

Proof. By numerically solving (LP3) for $n = 1000$, it can be shown that no algorithm with competitive ratio better than 0.2652 exists for 1000 secretaries. Assume for the sake of contradiction that there exists an algorithm ALG with a competitive ratio $\alpha > 0.266$ for n' secretaries, where $n' > n$. Let us use ALG to get an algorithm for n secretaries with a competitive ratio better than 0.2652.

From the view point of ALG it gets a random permutation of a random set of $n'/2$ secretaries that can contain at most two secretaries that give a point to ALG when hired. Let E_i be the expected number of points ALG gets when given a random permutation of a set containing i secretaries that give a point when hired (note that this expectation depends only on i). Our assumption that ALG is α competitive implies:

$$\frac{E_1}{2}\cdot\frac{n'/2}{n'-1} + \frac{E_2}{2}\cdot\frac{n'/2-1}{n'-1} \ge \alpha \ .$$

We now construct an algorithm for n secretaries using ALG. Our algorithm feeds ALG with the input it gets plus additional $(n'-n)/2$ dummy secretaries that are worse than any secretary of $\mathcal{S}$. The positions of the dummy secretaries are chosen at random from the $n'/2$ positions in the input of ALG. The competitive ratio of this algorithm is:

$$
\begin{aligned}
&\frac{E_1}{2}\cdot\frac{n/2}{n-1} + \frac{E_2}{2}\cdot\frac{n/2-1}{n-1} \ge \frac{E_1}{2}\cdot\frac{n'/2}{n'-1} + \frac{E_2}{2}\cdot\left[\frac{n'/2-1}{n'-1} - \frac{n}{2(n-1)^2}\right] \\
&\ge \alpha - \frac{n}{2(n-1)^2} > 0.2652 \ . \quad \square
\end{aligned}
$$

$\square$

ALGORITHM 3: Algorithm for $k = 2$, $Q = 1$ and $D = n/2$

1 Wait till time $t = 1 - \sqrt{1/3}$.
2 Let s_t be the second best secretary before time t (if less than 2 secretaries arrive before time t, let s_t be a dummy secretary worse than any other).
3 After time t: hire the first two secretaries better than s_t.

Next, we give a positive result for the above case. This result is probably far from being tight, however, we prove it for completeness. Notice that this result is not required for the proof of Claim 4.1.

Let s_1 and s_2 be the best and second best secretaries, respectively.

Lemma 4.4. *Algorithm 3 hires s_1 with probability at least $t(t-2)(t-1)/2$.*

Proof. Let A_x be the event that s_1 arrives at time x. If A_x occurs for some $x > t$, then s_1 is hired if at least two of the top 3 secretaries in the range $[0, x)$ arrive before time t (or there are less than 3 secretaries in this range, and at most 2 of them arrive after time t). Hence, given A_x, the probability of the s_1 to be hired is:

$$\left(\frac{t}{x}\right)^3 + 3 \cdot \left(\frac{t}{x}\right)^2 \cdot \left(1 - \frac{t}{x}\right) \quad .$$

The probability that A_x occurs for some x in a range R is proportional to half the size of R (because with probability $1/2$, s_1 is never interviewed). Hence, by the low of total probability, the probability of s_1 to be hired is:

$$\int_t^1 \left[\left(\frac{t}{x}\right)^3 + 3 \cdot \left(\frac{t}{x}\right)^2 \left(1 - \frac{t}{x}\right)\right] \cdot \frac{1}{2} \cdot dx = \frac{t^2}{2} \cdot \int_t^1 \left[\frac{t}{x^3} + \frac{3}{x^2} - \frac{3t}{x^3}\right] dx = \frac{t^2}{2} \cdot \left[\frac{t}{x^2} - \frac{3}{x}\right]_t^1$$
$$= \frac{t^2}{2} \cdot \left[t - 3 + \frac{2}{t}\right] = \frac{t(t-2)(t-1)}{2} \quad . \quad \square$$

$\square$

Lemma 4.5. *Algorithm 3 hires s_2 with probability at least $t(t-2)(t-1)/2$.*

Proof. Let us pair inputs for Algorithm 3 as following. Given some input, its pair is the same input with the rolls of s_1 and s_2 switched (*i.e.*, if s_1 arrives in the original input at time t_1, then s_2 will now arrive at time t_1, and vice versa). Notice that this is a one-to-one pairing between the set of inputs for Algorithm 3 and itself.

Notice that s_t is never s_1 or s_2, unless both s_1 and s_2 arrive before time t, in which case neither of them is hired. Hence, if s_1 is hired in a given input I of Algorithm 3, then s_2 is hired in the input paired to I, and vice versa. Thus, s_1 and s_2 are hired with the same probability by Algorithm 3. The lemma now follows from Lemma 4.4. $\square$

Corollary 4.6. *Algorithm 3 is at least $t(t-2)(t-1)/2 \geq 0.192$-competitive.*

Proof. Follows from the linearity of the expectation and Lemmata 4.4 and 4.5 $\square$

ALGORITHM 4: Algorithm for $k = 2$, $Q = 2$ and $D = n/2$ with "exclusive" positions

for *each queue independently* **do**
  Wait till time $t_1 = 0.35$.
  Let s_{t_1} be the best secretary before time t_1 (if no secretaries arrive before time t_1, let s_{t_1} be a dummy secretary worse than any other).
  Between time t_1 and time $t_2 = 0.8$: hire the first secretary better than s_{t_1}.
  if *No secretary is hired till time* t_2 **then**
    Let s_{t_2} be the second best secretary before time t_2 (if less than 2 secretaries arrive before time t_2, let s_{t_2} be a dummy secretary worse than any real secretary).
    After time t_2: hire the first secretary better than s_{t_2}.
  end
end

4.2 Two Queue with Exclusive Positions

In this section we consider the sub case where there are two queues, *i.e.*, $Q = 2$, and each queue has a single "exclusive" position to man. We suggest Algorithm 4 for this sub case.

Lemma 4.7. *Algorithm 4 hires s_1 with probability at least $t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2$.*

Lemma 4.8. *Assuming s_1 and s_2 arrive at the same queue, then Algorithm 4 hires s_1 with probability at least $t_1 \ln t_2 - 2t_1 t_2 - t_1 \ln t_1 + t_1^2 + t_1$.*

Lemma 4.9. *Algorithm 4 hires s_1 with probability at least $t_1 \ln t_2 - 1.5 t_1 t_2 - t_1 \ln t_1 + 0.5 t_1^2 + t_1$.*

Proof. Let H be the event that s_2 is hired, and let A be the event that s_1 and s_2 arrive both to the same queue. Using a proof identical to the one of Lemma 4.7, we get $\Pr[H|\bar{A}] \geq t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2$. On the other hand, Lemma 4.8 can be formally written as: $\Pr[H|A] \geq t_1 \ln t_2 - 2t_1 t_2 - t_1 \ln t_1 + t_1^2 + t_1$. Hence, by the law of total expectation expectation, we can lower bound the probability that s_2 is hired by:

$$\begin{aligned}\Pr[H] \geq\ & \Pr[A] \cdot (t_1 \ln t_2 - 2t_1 t_2 - t_1 \ln t_1 + t_1^2 + t_1) + \Pr[\bar{A}] \cdot (t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2) \\ &= t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2 - \Pr[A] \cdot t_1 \cdot (t_2 - t_1)\end{aligned}$$

For large number of secretaries, it is clear that $\Pr[A] \approx 1/2$. However, we need something a bit stronger than that. Given that s_1 arrives to some queue q, clearly, the probability that s_2 also arrive to q somewhat decreases. Hence, we always have $\Pr[A] < 0.5$. Plugging this inequality into the previous one completes the proof of the lemma. □

Corollary 4.10. *The competitive ratio of Algorithm 4 is at least* 0.319.

Proof. From Lemmata 4.7 and 4.9, the competitive ratio of Algorithm 4 is at least:

$$\begin{aligned}&0.5 \left[t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2\right] + 0.5 \left[t_1 \ln t_2 - 1.5 t_1 t_2 - t_1 \ln t_1 + 0.5 t_1^2 + t_1\right] \\ &= t_1 \ln t_2 - t_1 \ln t_1 + t_1 - 1.25 t_1 t_2 + 0.25 t_1^2 \ .\end{aligned}$$

The corollary follows by plugging the values of t_1 and t_2 into the above expression. □

Next, we give an almost matching hardness. Consider the following linear program.

$$
\begin{array}{lll}
\text{(LP4)} \quad \max & \frac{1}{n} \cdot \sum_{i=1}^{n/2} f_i + \frac{1}{n} \cdot \sum_{i=1}^{n/2} \left(\frac{n-i}{n-1} \cdot f_i + \frac{i-1}{n-1} s_i \right) & \\
\text{s.t.} & f_i + f_1 + \sum_{j=2}^{i-1} \frac{f_j + s_j}{j} & \leq 1 \quad \forall\, 2 \leq i \leq n \\
& s_i + f_1 + \sum_{j=2}^{i-1} \frac{f_j + s_j}{j} & \leq 1 \quad \forall\, 2 \leq i \leq n \\
& f_i, s_i & \geq 0 \quad \forall\, 1 \leq i \leq n
\end{array}
$$

Lemma 4.11. *The optimal value of (LP4) is an upper bound on the competitive ratio of any algorithm for the case of "exclusive" positions with $k = 2$, $Q = 2$ and $D = n/2$.*

The following theorem provides an upper bound on the best possible competitive ratio for the case of "exclusive" positions with $k = 2$, $Q = 2$ and $D = n/2$. The proof of the theorem is similar to the one of Theorem 4.3, with Lemma 4.11 taking the role of Lemma 4.2.

Theorem 4.12. *No algorithm is better than* 0.321*-competitive for the case of "exclusive" positions with $k = 2$, $Q = 2$ and $D = n/2$.*

4.3 Two Queue with Shared Positions

In this section we consider the sub case where there are two queues, *i.e.*, $Q = 2$, and the two available positions are "shared" between the queues (*i.e.*, each queue can man up to 2 positions). We suggest Algorithm 5 for this sub case.

ALGORITHM 5: Algorithm for $k = 2$, $Q = 2$ and $D = n/2$ with "shared" positions

```
for each queue independently do
    Wait till time t_1 = 0.348.
    while no secretary was hired (from any of the queues), for every secretary s arriving do
        if s is better than any previously seen secretary then
            Hire s.
        end
    end
    Wait till time time t_2 = 0.563 (if we did not reach this time yet).
    while one position is still unmanned, for every secretary s arriving do
        if s is better than any previously seen secretary then
            Hire s.
        end
    end
end
```

Let s_1 (s_2) be the (second) best secretary, and let q_1 (q_2) be the queue he arrives to. We also denote by $\bar{q}_1$ ($\bar{q}_2$) the queue which is not q_1 (q_2). We now define two sets C_1 and C_2 as following. Given an input $\mathcal{I}_P$ of the random order model (such an input is simply a random permutation of the secretaries of $\mathcal{S}$), let $\mathcal{I}_T$ be any input of the random arrival times model that might be produced from $\mathcal{I}_P$ using Algorithm 1. The pair $(\mathcal{I}_P, \mathcal{I}_T)$ is in C_1 if one of the following conditions holds:

E1 $t(s_1, \mathcal{I}_T) \in (t_1, t_2 - D^{-1/3})$ and the best secretary in the range $[0, t(s_1, \mathcal{I}_T))$ arrives before time t_1.

E2 $t(s_1, \mathcal{I}_T) > t_2$, the best secretary of the range $[0, t(s_1, \mathcal{I}_T) + D^{-1/3}]$ in $\bar{q}_1$ arrives before time t_1, and at least one of the two following happens:

- The best secretary in the range $[0, t(s_1, \mathcal{I}_T))$ in q_1 arrives before time t_2.

- The second best secretary in the range $[0, t(s_1, \mathcal{I}_T))$ in q_1 arrives before time t_1.

E3 $t(s_1, \mathcal{I}_T) > t_2$, the best secretary of the range $[0, t_T(s_1))$ in q_1 arrives before time t_1, and at least one of the following happens:

- The best secretary in the range $[0, t(s_1, \mathcal{I}_T) + D^{-1/3}]$ in $\bar{q}_1$ arrives during $[t_1, t_2)$.
- The best secretary in the range $[0, t(s_1, \mathcal{I}_T) + D^{-1/3}]$ in $\bar{q}_1$ arrives after time t_2, and the second best secretary in the same range and queue arrives before t_1.

Regardless the membership of $(\mathcal{I}_P, \mathcal{I}_T)$ in C_1. The pair $(\mathcal{I}_P, \mathcal{I}_T)$ is in C_2 if one of the following conditions holds:

F1 The pair satisfies one of the conditions to be included in C_1 with s_1 replaced by s_2 and q_1 replaced by q_2, and also one of the following holds.

- $q_1 \neq q_2$
- s_1 appears after s_2 in one queue.

F2 $t(s_2, \mathcal{I}_T) > t(s_1, \mathcal{I}_T) > t_2$, $q_1 = q_2$, the best secretary in the range $[0, t(s_2, \mathcal{I}_T) + D^{-1/3})$ in $\bar{q}_2$ appears before time t_1, and the best secretary in the range $[0, t(s_2, \mathcal{I}_T))$ in q_1 other than s_1 and s_2 also appears before time t_1.

Lemma 4.13. *For a random pair* $(\mathcal{I}_P, \mathcal{I}_T)$, $\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_1] \geq t_1 \ln(t_2/t_1) + t_1^2 t_2 - t_1^2 - 2t_1 t_2 + 2t_1 - o(1) \geq 0.4186$.

Lemma 4.14. *For a random pair* $(\mathcal{I}_P, \mathcal{I}_T)$, $\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_2] \geq t_1 \ln(t_2/t_1) + t_1 t_2 \ln t_2 + 0.5 t_1^2 t_2 - 2.5 t_1 t_2 + 2t_1 - o(1) \geq 0.2951$.

Corollary 4.15. *For a random pair* $(\mathcal{I}_P, \mathcal{I}_T)$*, with probability at least* 0.418*,* $(\mathcal{I}_P, \mathcal{I}_T) \in C_1$ *and* $\mathcal{I}_T$ *is well-representation of* $\mathcal{I}_P$*, and with probability at least* 0.295*,* $(\mathcal{I}_P, \mathcal{I}_T) \in C_2$ *and* $\mathcal{I}_T$ *is well-representation of* $\mathcal{I}_P$*.*

Proof. Follows from Lemmata 2.1, 4.13 and 4.14 and the union bound. □

We say that Algorithm 5 got a pair $(\mathcal{I}_P, \mathcal{I}_T)$ if the original input for the random order model was $\mathcal{I}_P$, and it was converted to the input $\mathcal{I}_T$ of the random arrival times model by Algorithm 1.

Lemma 4.16. *Assuming Algorithm 5 gets a pair* $(\mathcal{I}_P, \mathcal{I}_T)$ *in which* $\mathcal{I}_T$ *is a well-representation of* $\mathcal{I}_P$*. Then, if* $(\mathcal{I}_P, \mathcal{I}_T) \in C_1$*, Algorithm 5 hires* s_1*, and if* $(\mathcal{I}_P, \mathcal{I}_T) \in C_2$*, Algorithm 5 hires* s_2

Proof. The proof of this lemma is technical and uses the same kind of arguments as the proof of Lemma 3.5. Thus, we omit it. □

Corollary 4.17. *Algorithm 5 is a* 0.356*-competitive algorithm.*

Proof. Corollary 4.15 and Lemma 4.16 imply that when given a random pair: with probability 0.418 Algorithm 2 hires s_1, and with probability 0.295 Algorithm 2 hires s_2. Hence, by the linearity of the expectation, Algorithm 2 earns in expectation at least 0.713 points. □

5 k Positions and Half the Required Time

In this section we start the analysis of the case of k positions, where k is assumed to be large. Here we assume that there is enough time to interview half of the secretaries (*i.e.*, $D = n/2$), and therefore, the number of queues is either 1 or 2. This case is interesting on its own right, and will also serve us as a warm up for the next section.

5.1 One Queue

Let us begin with the case of $Q = 1$.

Theorem 5.1. *There is a $1/2 - o(1)$ competitive algorithm for the case of $D = n/2$, $Q = 1$ and large k.*

Proof. We apply the algorithm of [13] for hiring k secretaries to the $n/2$ secretaries that we interview. Let us analyze this algorithm.

The last algorithm has a competitive ratio of $1 - o(1)$. That means that if we fix the set of secretaries that are interviewed, then our algorithm earns a value of $[1 - o(1)] \cdot N$, where N is the number of top k secretaries that are interviewed. Clearly $\mathbb{E}[N] = k/2$, and therefore, by the law of total expectation, the expected value that our algorithm earns is at least: $\mathbb{E}[[1 - o(1)] \cdot N] = [1 - o(1)] \cdot \mathbb{E}[N] = [1/2 - o(1)] \cdot k$. □

Theorem 5.2. *No better than $1/2$-competitive algorithm exists for the case $D = n/2$, $Q = 1$.*

Proof. The expected number of top k secretaries that are interviewed is only $k/2$, and clearly no algorithm can hire more top k secretaries than it interviews. □

5.2 Two Queues

We now consider the case of $Q = 2$. Notice that in this case, it is possible to interview all secretaries. Algorithm 6 is our algorithm for this case. Algorithm 6 was designed for the "exclusive" positions model, however, it works also for "shared" positions.

ALGORITHM 6: Algorithm for $Q = 2$, $D = n/2$ and large k

```
1 for each queue independently do
2 |  Use the algorithm of [13] to hire k/2 secretaries from this queue.
3 end
```

Let $\mathcal{S}_k$ be the set of the top k secretaries. For every secretary $s \in \mathcal{S}_k$, let us denote by X_s an indicator for the event that s arrives to the first queue.

Lemma 5.3. *For every subset $S \subseteq \mathcal{S}_k$, $\Pr[\prod_{s \in S} X_s = 1] \leq 2^{-|S|} = \prod_{s \in S} \Pr[X_s = 1]$.*

Proof. If $|S| > n/2$, then clearly, $\Pr[\prod_{s \in S} X_s = 1] = 0 \leq 2^{-|S|}$. If $|S| \leq n/2$, then:

$$\Pr\left[\prod_{s \in S} X_s = 1\right] = \frac{\binom{n-|S|}{n/2-|S|}}{\binom{n}{n/2}} = \frac{\frac{(n-|S|)!}{(n/2-|S|)!(n/2)!}}{\frac{n!}{(n/2)!(n/2)!}} = \frac{(n-|S|)!(n/2)!}{(n/2-|S|)!n!}$$
$$= \frac{\prod_{i=n/2-|S|+1}^{n/2} i}{\prod_{i=n-|S|+1}^{n} i} = \prod_{i=n-|S|+1}^{n} \frac{i - n/2}{i} \leq 2^{-|S|} \quad . \quad \square$$

□

Lemma 5.3 implies that we can apply the generalized Chernoff bound of [17] to sums of the form $\sum_{s \in S} X_s$ where S is a subset of $\mathcal{S}_k$. We use this observation in the proof of the following theorem.

Theorem 5.4. *Algorithm 6 is a $1-o(1)$ competitive algorithm for the case $Q=2$, $D=n/2$ and large k.*

Proof. Lemma 5.3 implies that we can apply the generalized Chernoff bound of [17] to sums of the form $\sum_{s\in S} X_s$ where S is a subset of $\mathcal{S}_k$. Let $\mathcal{S}'$ be the set of top $k-k^{2/3}$ secretaries. We can upper bound $\Pr[\sum_{s\in\mathcal{S}'} X_s \geq k/2]$ as following.

$$\Pr\left[\sum_{s\in\mathcal{S}'} X_s \geq k/2\right] \leq e^{-2(k-k^{2/3})\left(\frac{k/2}{k-k^{2/3}}-0.5\right)^2} = e^{-0.5\cdot\frac{k^{4/3}}{k-k^{2/3}}} \leq e^{-0.5\cdot k^{1/3}} = o(1) \ .$$

Hence, with probability $1-o(1)$, the first queue contains no more than $k/2$ secretaries of $\mathcal{S}'$. Since the two queues are symmetric, we can use the union bound to show that this property holds in both queues at the same time with probability $1-o(1)$. Let us denote this event by E.

Notice that the event E depends only the distribution of secretaries between the queues. Fix some distribution P for which E holds. Given P, the queues get a random permutation of two sets S_1 and S_2 of secretaries with the following property. The sets $S_1\cap\mathcal{S}'$ and $S_2\cap\mathcal{S}'$ both contain at most $k/2$ secretaries. The competitive ratio of the algorithm of [13] is $1-o(1)$. Hence, given P, it must collect an expected values of $[1-o(1)]\cdot|S_1\cap\mathcal{S}'|$ and $[1-o(1)]\cdot|S_2\cap\mathcal{S}'|$ from the two queues, respectively. Combing the value from both queues, we get that Algorithm 6 collects a total value of:

$$[1-o(1)]\cdot|S_1\cap\mathcal{S}'|+[1-o(1)]\cdot|S_2\cap\mathcal{S}'| = [1-o(1)]\cdot|\mathcal{S}'| = [1-o(1)]\cdot k \ . \tag{1}$$

The above calculation was done assuming a fixed distribution P of the secretaries among the queues which respects E. However, (1) is independent of the distribution, and therefore, it is also the expected value of Algorithm 6 given just E. Hence, by the law of total expectation, without any assumptions, the value Algorithm 6 collects is at least: $\Pr[E]\cdot[1-o(1)]\cdot k \geq [1-o(1)]^2\cdot k = [1-o(1)]\cdot k$. □

6 k Positions and $1/k$ of the Required Time

In this section we continue the analysis of the case of k positions. Here we assume that there is enough time to interview only $1/k$ of the secretaries (*i.e.*, $D=n/k$). This is the shortest time that still allows us to interview all secretaries under the "exclusive" positions model (because we cannot have more queues than positions under this model). We consider the two extreme cases in terms of the number of queues: $Q=1$ and $Q=k$. Due to space limitations, some of the proofs of this section are deferred to Appendix C.

Theorem 6.1. *For the case $Q=1$, there is a $1/k-o(1)$ competitive algorithm, and no algorithm can be better than $1/k$ competitive for this case.*

Proof. Follows from a slight modification of the proofs of Theorems 5.1 and 5.2. □

In the rest of this section we consider the case $Q=k$.

Theorem 6.2. *No better than $1-1/(2e)+o(1)\approx 0.816$-competitive algorithm exists for the case $Q=k$, $D=n/k$ for large k and "shared" positions.*

Consider the following auxiliary problem. n secretaries arrive at random order to a single queue. The algorithm for the problem can hire at most one secretary. The algorithm gets a point if the secretary hired is a top i secretary.

Lemma 6.3. *If there exists an α-competitive algorithm for the auxiliary problem with n secretary, then there is a α-competitive algorithm for the auxiliary problem with n' secretaries for every $i \leq n' < n$.*

Proof. Let ALG be the α-competitive algorithm for the auxiliary problem with n secretaries. The algorithm we suggest for n' secretaries feeds ALG with an input constructed as following.

- Randomly select $n - n'$ positions for dummy secretaries worse than any real secretary.
- The real secretaries fill in the other positions in the same order that they arrive.

Since ALG is α competitive, it will hire a top i secretary with probability at least α. □

Lemma 6.3 allows us to prove hardness results for the auxiliary problem (for a given value of i) by numerically solving a LP defined in [5]. The proof of the following theorem shows how to convert these hardness results into a hardness for the case $Q = k$, $D = n/k$ for large k and "exclusive" positions.

Theorem 6.4. *No better than* 0.301*-competitive algorithm exists for the case $Q = k$, $D = n/k$ for large k and "exclusive" positions.*

We now shift our attention to positive results. In the case of $Q = k$, $D = n/k$ and "exclusive" positions, at most one secretary is hired from each queue. The natural intuition for this case is to use the classical secretary algorithm for every queue independently, *i.e.*, use Algorithm 7 with $t = e^{-1}$. The next theorem analyzes this algorithm. The theorem is given using the exponential integral function $\mathsf{Ei}(x)$, which is defined by $\mathsf{Ei}(x) = \int_{-\infty}^{x}(e^t/t)dt$.

ALGORITHM 7: Algorithm for $Q = k$ and $D = n/k$

```
for each queue independently do
    Wait till time t.
    while no secretary was hired from the current queue, for every secretary s arriving do
        if s is better than any previously seen secretary then
            Hire s.
        end
    end
end
```

Theorem 6.5. *Algorithm 7 is a $t \cdot \left[\mathsf{Ei}(-1) + e^{-1} - 1 - \mathsf{Ei}(-t) - \frac{e^{-t}}{t} + t^{-1}\right]$ competitive algorithm for the case $Q = k$ and $D = n/k$. Hence, for $t = 0.323$, Algorithm 7 is a* 0.276*-competitive algorithm.*

Numerically, it can be shown that the value of t maximizing the competitive ratio given by Theorem 6.5 is about 0.323. Notice that for $t = e^{-1} \approx 0.368$, this analysis provides only an inferior competitive ratio of 0.274. We believe this analysis is tight, *i.e.*, the best value for t is not e^{-1}.

Based on our results for the case $Q = k = 2$, $D = n/2$ and "exclusive" positions, we suspect that an optimal algorithm for the case $Q = k$, $D = n/k$ and "exclusive" positions should use k different time threshold $t_1 < t_2 < \cdots < t_k$. A policy of such an algorithm will hire a secretary s that arrives between time t_i and t_{i+1} if no secretary was hired from the queue of this policy before, and s is better than all previously seen secretaries except $i - 1$. Unfortunately, analyzing such an algorithm seems to be too complicated. Instead, we analyze Algorithm 8 which uses only two time thresholds. This gives some idea about the improvement that can achieved by additional time thresholds.

ALGORITHM 8: Algorithm for $Q = k$ and $D = n/k$ with "exclusive" positions

```
for each queue independently do
    Wait till time t_1 = 0.34.
    while time t_2 = 0.748 was not reached yet, for every secretary s arriving do
        if no secretary was hired before, and s is better than any previously seen secretary then
            Hire s.
        end
    end
    Let s' be the best secretary seen up to this point.
    for every secretary s arriving do
        if no secretary was hired before, and s is better than any previously seen secretary other
          than s' then
            Hire s.
        end
    end
end
```

Theorem 6.6. *For large k, Algorithm 8 is a 0.288-competitive algorithm for the case $Q = k$ and $D = n/k$.*

Acknowledgment

The authors would like to thank Ron Lavi for many useful discussions.

References

[1] M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar. Secretary problems: Weights and discounts. In *20th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1245–1254, 2009.

[2] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. *A knapsack secretary problem with applications*, volume 4628 of *LNCS*, pages 16–28. Springer, Heidelberg, 2007.

[3] M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 434–443, 2007.

[4] M. H. Bateni, M. T. Hajiaghayi, and M. Zadimoghaddam. Submodular secretary problem and extensions. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 6302:39–52, 2010.

[5] Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary problems via linear programming. In *IPCO*, pages 163–176, 2010.

[6] N. B. Dimitrov and C. G. Plaxton. Competitive weighted matching in transversal matroids. *Automata, Languages and Programming*, 5125:397–408, 2008.

[7] E. B. Dynkin. The optimum choice of the instant for stopping a markov process. *Soviet Math. Dokl.*, 4:627–629, 1963.

[8] Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. Improved competitive ratios for submodular secretary problems. In *14th international workshop and 15th international conference on Approximation, randomization, and combinatorial optimization: algorithms and techniques (APPROX)*, pages 218–229, 2011.

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

[10] Nicholas Georgiou, Malgorzata Kuchta, Michal Morayne, and Jaroslaw Niemiec. On a universal best choice algorithm for partially ordered sets. *Random Struct. Algorithms*, 32:263–273, May 2008.

[11] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar. Constrainted non-monotone submodular maximization: Online and secretary algorithms. *Internet and Network Economics*, 6484:246–257, 2010.

[12] S. Im and Y. Wang. Secretary problems: Laminar matroid and interval scheduling. In *22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1096–1116, 2011.

[13] R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 630–631, 2005.

[14] N. Korula and M. Pál. Algorithms for secretary problems on graphs and hypergraphs. *Automata, Languages and Programming*, 5556:508–520, 2009.

[15] Ravi Kumar, Silvio Lattanzi, Sergei Vassilvitskii, and Andrea Vattani. Hiring a secretary from a poset. In *12th ACM conference on Electronic Commerce (EC)*, pages 39–48, 2011.

[16] D. V. Libdley. Dynamic programming and decision theory. *Applied Statistics*, 10:39–51, 1961.

[17] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM J. Comput.*, 26:350–368, April 1997.

APPENDIX

A Omitted Proofs of Section 3

THEOREM 3.1. There is a $(de)^{-1}$-competitive algorithm for the case $Q = 1$ and $D = n/d$, and this is the best possible up low order terms.

Proof. Let $\mathcal{S}' \subseteq \mathcal{S}$ be the set of the first n/d, and let s_1 be the best secretary in $\mathcal{S}$. The algorithm we suggest is to apply the algorithm for the classical secretary problem to the first n/d secretaries. Let us analyze this algorithm. Fix $\mathcal{S}'$, then the input to the algorithm for the classical secretary problem is a random permutation of $\mathcal{S}'$. If $s_1 \notin \mathcal{S}'$ then clearly the algorithm cannot hire s_1. Otherwise, s_1 is also the best secretary in $\mathcal{S}'$, and therefore, the algorithm hires him with probability at least e^{-1}. Thus, the expected value of the algorithm is at least:

$$\sum_{A \subseteq \mathcal{S} \;|\; |A|=n/d, s_1 \in A} \Pr[\mathcal{S}' = A] \cdot e^{-1} = e^{-1} \cdot \Pr[s_1 \in \mathcal{S}] = (ed)^{-1} \ .$$

We are left to prove that no algorithm can do better. Consider some algorithm ALG. We can use ALG to solve the classical secretary problem with n/d secretaries as following. Given an input

$\mathcal{I}$ for the classical secretary problem, we give ALG the n/d secretaries of $\mathcal{I}$ and after them $n - n/d$ arbitrary other secretaries. Since ALG can interview only n/d secretaries, it must return a feasible solution for the classical secretary problem. Due to the hardness known for the classical secretary problem, ALG can hire the best secretary of $\mathcal{I}$ with probability at most $e^{-1} + o(1)$.

Let us now return to our problem. Fix the subset $\mathcal{S}' \subseteq \mathcal{S}$ of the n/d first secretaries input of ALG. Since ALG must hire a secretary of $\mathcal{S}'$ (or no secretary at all), it follows that ALG gets no value when $s_1 \notin \mathcal{S}'$. From the above reduction to the classical secretary problem, we learn that even when $s_1 \in \mathcal{S}$, ALG hires him with probability at most $e^{-1} + o(1)$. Thus, the competitive ratio of ALG can be upper bounded by:

$$\sum_{A \subseteq \mathcal{S} \mid |A| = n/d, s_1 \in A} \Pr[\mathcal{S}' = A] \cdot (e^{-1} + o(1)) = (e^{-1} + o(1)) \cdot \Pr[s_1 \in \mathcal{S}] = (ed)^{-1} + o(1) \ . \quad \square$$

$\square$

LEMMA 3.7. The optimal value of (LP1) is an upper bound on the competitive ratio of any algorithm for the case $k = 1$, $Q = d$ and $D = n/d$.

Proof. Consider some arbitrary algorithm ALG for the problem. We assume ALG never hires a secretary if a better secretary from the same queue was already interviewed. Clearly, any algorithm can be made to act this way without deteriorating its competitive ratio. Let us construct an assignment for (LP1) using ALG. Let $f_{i,j}$ be the probability that ALG hires the best secretary, given that he arrives at position j of queue i.

First, notice that the best secretary has equal probability of $1/n$ to get to every one of the positions in every one of the queues. Hence, given the above assignment, the objective function of (LP1) becomes equal to the competitive ratio of ALG.

Second, we need to prove that the above assignment is feasible. Let us calculate the probability that the algorithm hires the secretary of position j in queue i. With probability $(j-1)/j$, the secretary at this position is not better than all previously seen secretaries of this queue. In this case ALG clearly does not hire the secretary since it has 0 probability of being the best secretary. On the other hand, with probability $1/j$, the secretary at the above position is the best that ALG interviewed up to this from queue i. Notice that ALG cannot distinguish this case from the case that the best secretary is at position j of queue i, and therefore, it must hire the secretary of this position with probability $f_{i,j}$. In conclusion, ALG hires the secretary at position j of queue i with probability:

$$\frac{j-1}{j} \cdot 0 + \frac{1}{j} \cdot f_{i,j} = \frac{f_{i,j}}{j} \ .$$

Let us now prove that the constraint associated with position j of queue i holds. Fix the best secretary to this position. By definition, ALG hires the best secretary with probability $f_{i,j}$. For every position ℓ of queue h for which either $\ell < j$ or $\ell = j$ and $h < i$, it can easily seen that the above discussion holds even though we fixed the position of the best secretary, *i.e.*, the secretary at position ℓ of queue h is hired with probability $f_{h,\ell}/\ell$. The constraint now follows, since ALG can hire at most one secretary.

In conclusion, any algorithm, including the optimal one, induces a feasible solution for (LP1) whose value is equal to the competitive ratio of the algorithm. Hence, the optimal value of (LP1) upper bounds the optimal competitive ratio. $\square$

LEMMA 3.8. (LP2) has a solution of value at most $d^{-d/(d-1)} + O(n^{-0.5})$.

Proof. Consider the following solution.

$$y_{i,j} = \begin{cases} 0 & \text{if } j \le nd^{-d/(d-1)} \\ \frac{d^d(j/n)^{d-1}-1}{n(d-1)} & \text{otherwise} \end{cases}$$

We first need to prove that this solution is feasible. Consider a constraint corresponding to some $1 \le i \le d$ and $nd^{-d/(d-1)} \le j \le n$. Let us lower bound the left hand side of this constraint.

$$\begin{aligned} y_{i,j} + \sum_{h=1}^{d}\sum_{\ell=j+1}^{n/d}\frac{y_{h,\ell}}{j} + \sum_{h=i+1}^{d}\frac{y_{h,j}}{j} &\ge y_{i,j} + \sum_{h=1}^{d}\sum_{\ell=j+1}^{n/d}\frac{y_{h,\ell}}{j} \\ &= \frac{d^d(j/n)^{d-1}-1}{n(d-1)} + \sum_{\ell=j+1}^{n/d}\frac{d^{d+1}(\ell/n)^{d-1}-d}{jn(d-1)} \\ &\ge \frac{d^d(j/n)^{d-1}-1}{n(d-1)} + \int_{j}^{n/d}\frac{d^{d+1}(x/n)^{d-1}-d}{jn(d-1)}dx \\ &= \frac{d^d(j/n)^{d-1}-1}{n(d-1)} - \frac{d(n/d-j)}{jn(d-1)} + \left.\frac{d^d(x/n)^d}{j(d-1)}\right|_{j}^{n/d} \\ &= \frac{[d^d j^d/n^{d-1}-j]+[jd-n]+[n-d^d j^d/n^{d-1}]}{jn(d-1)} = \frac{1}{n}\ . \end{aligned}$$

Consider now a constraint corresponding to $1 \le i \le d$ and $1 \le j \le nd^{-d/(d-1)}$, let us lower bound the left hand side of this constraint.

$$\begin{aligned} y_{i,j} + \sum_{h=1}^{d}\sum_{\ell=j+1}^{n/d}\frac{y_{h,\ell}}{j} + \sum_{h=i+1}^{d}\frac{y_{h,j}}{j} &\ge \sum_{h=1}^{d}\sum_{\ell=j+1}^{n/d}\frac{y_{h,\ell}}{j} = d\cdot\int_{j}^{n/d}\frac{y_{h,\lceil x\rceil}}{j}dx \\ &\ge \frac{d}{j}\cdot\int_{nd^{-d/(d-1)}}^{n/d}\frac{d^d(x/n)^{d-1}-1}{n(d-1)}dx \\ &\overset{(*)}{=} \frac{1}{j}\cdot\int_{d^{-1/(d-1)}}^{1}\frac{dz^{d-1}-1}{d-1}dz \\ &= \frac{1}{j}\cdot\left.\frac{z^d-z}{d-1}\right|_{d^{-1/(d-1)}}^{1} \\ &= \frac{1}{j}\cdot\frac{d^{-1/(d-1)}-d^{-d/(d-1)}}{d-1} = \frac{d^{-d/(d-1)}}{j} \\ &\ge \frac{d^{-d/(d-1)}}{nd^{-d/(d-1)}} = \frac{1}{n}\ , \end{aligned}$$

where (*) follows from substituting $z = dx/n$. This completes the proof that the above solution is

feasible. To complete the proof of the lemma, we need to upper bound the value of this solution.

$$\begin{aligned}
\sum_{h=1}^{d}\sum_{\ell=1}^{n/d} y_{h,\ell} &= d\cdot\int_{1}^{n/d+1} y_{h,\lfloor x\rfloor}dx = d\cdot\int_{nd^{-d/(d-1)}}^{n/d+1} y_{h,\lfloor x\rfloor}dx \\
&\le d\cdot\int_{nd^{-d/(d-1)}}^{n/d+1}\frac{d^d(x/n)^{d-1}-1}{n(d-1)}dx \\
&\overset{(**)}{=} \int_{d^{-1/(d-1)}}^{1+d/n}\frac{dz^{d-1}-1}{d-1}dz = \left.\frac{z^d-z}{d-1}\right|_{d^{-1/(d-1)}}^{1+d/n} \\
&= \frac{(1+d/n)^d-1-d/n}{d-1} - \frac{d^{-d/(d-1)}-d^{-1/(d-1)}}{d-1} \le \frac{e^{d^2/n}-1}{d-1} + d^{-d/(d-1)} \\
&\overset{(***)}{\le} d^{-d/(d-1)} + \frac{d/\sqrt{n}}{(d-1)} = k^{-d/(d-1)} + O(n^{-0.5}) \ ,
\end{aligned}$$

where (**), again, follows from substituting $z = dx/n$, and (***) follows since $e^x - 1 \le \sqrt{x}$ for $0 \le x \le 16^{-1}$. □

B Omitted Proofs of Section 4

LEMMA 4.2. The optimal value of (LP3) is an upper bound on the competitive ratio of any algorithm for the case $k = 2$, $Q = 1$ and $D = n/2$.

Proof. Consider some arbitrary algorithm ALG for the problem. We assume ALG never hires a secretary if two better secretaries were already interviewed. Clearly, any algorithm can be made to act this way without deteriorating its competitive ratio. Let us construct an assignment for (LP3) using ALG. Let f_i (s_i) be the probability that ALG hires the secretary at position i given that he is the (second) best secretary seen so far. We also need f_i^1 (f_i^2) which is the same as f_i with the additional requirement that the secretary at position i is the first (second) secretary hired. Similarly, we also define s_i^1 and s_i^2. Observe that since ALG never hires more than two secretaries, the above solution obeys the two types of equalities in (LP3).

The best secretary appears in every one of the first $n/2$ positions in the queue with equal probability of $1/n$. When this secretary appears in position i, it is hired with probability f_i (ALG cannot distinguish the best secretary from every other secretary in position i which is better than all previous secretaries). Hence, the probability that ALG hires the best secretary is:

$$\frac{1}{n}\cdot\sum_{i=1}^{n/2} f_i \ . \tag{2}$$

The second best secretary also appears in each one of the first $n/2$ positions with equal probability. When it appears in position i, it is the best secretary up to this point with probability $(i-1)/(n-1)$, and the second best secretary up to this point otherwise (depending on the position of the single better secretary). ALG accepts the secretary in question with probability f_i in the first case, and probability s_i in the second case. Thus, the probability that ALG hires the second best secretary is:

$$\frac{1}{n}\cdot\sum_{i=1}^{n/2}\left(\frac{n-i}{n-1}\cdot f_i + \frac{i-1}{n-1}s_i\right) \ . \tag{3}$$

By combining (2) and (3), we get the value of the solution suggested is equal to the competitive ratio of ALG.

We say that ALG hires a secretary as the first secretary if no secretary was hired by ALG before, and as the second secretary otherwise.

Consider the case that the secretary in position i is the best secretary up to that point. ALG hires this secretary as the first secretary probability f_i^1. One the other hand, consider some position $j < i$. If the secretary at this position is the best up to this point (which happens with probability $1/j$), ALG hires him with probability f_j^1 as the first secretary. Similarly, if the secretary at this position is the second best up to this point (which happens with probability $1/j$ for $j \geq 2$), ALG hires him with probability s_j^1 as the first secretary. Since ALG can hire at most one secretary as the first secretary, the above discussion implies that the inequalities of the first type are satisfied by the above solution. Inequalities of the second type are satisfied because of a similar argument for the case that the secretary in position i is the second best secretary up to this point.

Still considering the case that the secretary in position i is the best secretary up to that point. The above discussion implies that the probability that some secretary gets hired as the first secretary before ALG gets to position i is: $f_1^1 + \sum_{j=2}^{i-1} \frac{f_j^1 + s_j^1}{j}$. Similar arguments show that the probability that some secretary gets hired as the second secretary before ALG gets to position i is: $\sum_{j=2}^{i-1} \frac{f_j^2 + s_j^2}{j}$. Hence, the probability that exactly one secretary is hired till ALG gets to position i is the difference between the above expressions, *i.e.*,

$$f_1^1 + \sum_{j=2}^{i-1} \frac{f_j^1 + s_j^1 - f_j^2 - s_j^2}{j} \quad .$$

Clearly, the probability of the event that the secretary at position i is hired as the second secretary is upper bounded by this difference. On the other hand, this probability is exactly f_i^2. From this argument we get that the inequalities of the third type hold for the solution constructed. This kind of argument also prove that inequalities of the fourth type are satisfied when applied to the case that the secretary in position i is the second best secretary up to this point.

In conclusion, any algorithm, including the optimal one, induces a feasible solution for (LP3) whose value is equal to the competitive ratio of the algorithm. Hence, the optimal value of (LP3) upper bounds the optimal competitive ratio. □

LEMMA 4.7. Algorithm 4 hires s_1 with probability at least $t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2$.

Proof. Let A_x be the event that s_1 arrives at time x. If A_x occurs for some $x \in (t_1, t_2]$, then s_1 is hired if the best secretary in the range $[0, x)$ arrives before time t_1 (or the range is empty), which occurs with probability t_1/x. On the other hand, if A_x occurs for some $x > t_2$, then s_2 is hire if the best secretary in the range $[0, x)$ arrives before time t_1 and the second best secretary in the same range arrives before time t_2 (or the range is empty), which occurs with probability $t_1 t_2 / x^2$.

The probability that A_x occurs for some x in a range R is proportional to the size of R (because s_1 arrives at a random time of the range $[0, 1]$). Hence, by the law of total probability, the probability of s_1 to be hired is:

$$\int_{t_1}^{t_2} \frac{t_1}{x} dx + \int_{t_2}^{1} \frac{t_1 t_2}{x^2} dx = t_1 \ln x \Big|_{t_1}^{t_2} - \frac{t_1 t_2}{x} \Big|_{t_2}^{1} = t_1 \ln t_2 - t_1 \ln t_1 + t_1 - t_1 t_2 \quad . \quad \square$$

□

LEMMA 4.8. Assuming s_1 and s_2 arrive at the same queue, then Algorithm 4 hires s_1 with probability at least $t_1 \ln t_2 - 2t_1 t_2 - t_1 \ln t_1 + t_1^2 + t_1$.

Proof. Let A_x be the event that s_2 arrives at time x. If A_x occurs for some $x \in (t_1, t_2]$, then s_2 is hired if s_1 arrives after it, and the best secretary in the range $[0, x)$ arrives before time t_1. Both events occur with probability $(1-x) \cdot t_1/x$. If A_x occurs for some $x > t_2$, then s_2 is hired if one of the following happens:

- s_1 arrives before time t_1, and the best secretary in the range $[0, x)$ (excluding s_1) arrives before time t_2 (or the range is empty except for s_1).
- s_1 arrives after time x, the best secretary in the range $[0, x)$ arrives before time t_1, and the second best secretary in the range $[0, x)$ arrives before time t_2 (or the range is empty).

The probability that one of these events occur is:

$$t_1 \cdot \frac{t_2}{x} + (1-x) \cdot \frac{t_1 t_2}{x^2} = \frac{t_1 t_2}{x} \ .$$

The probability that A_x occurs for some x in a range R is proportional to the size of R (because s_1 arrives at a random time of the range $[0, 1]$). Hence, by the law of total probability, the probability of s_2 to be hired is:

$$\begin{aligned}\int_{t_1}^{t_2} \left((1-x) \cdot \frac{t_1}{x}\right) dx + \int_{t_2}^{1} \frac{t_1 t_2}{x^2} &= \int_{t_1}^{t_2} \left(\frac{t_1}{x} - t_1\right) dx + \int_{t_2}^{1} \frac{t_1 t_2}{x^2} dx \\ &= [t_1 \ln x - t_1 x]_{t_1}^{t_2} - \left.\frac{t_1 t_2}{x}\right|_{t_2}^{1} \\ &= t_1 \ln t_2 - 2t_1 t_2 - t_1 \ln t_1 + t_1^2 + t_1 \ . \quad \square\end{aligned}$$

$\square$

LEMMA 4.11. The optimal value of (LP4) is an upper bound on the competitive ratio of any algorithm for the case of "exclusive" positions with $k = 2$, $Q = 2$ and $D = n/2$.

Proof. Consider some arbitrary algorithm ALG for the problem. We assume each queue policy of ALG never hires a secretary if it has already interviewed two better secretaries. Clearly, any algorithm can be made to act this way without deteriorating its competitive ratio. Let us construct an assignment for (LP4) using ALG. Let f_i (s_i) be the probability that ALG hires the secretary at position i given that he is the (second) best secretary seen so far.

The best secretary appears in position i of some queue with probability $2/n$ for every position $1 \le i \le n/2$. When this secretary appears in position i, it is hired with probability f_i (ALG cannot distinguish the best secretary from every other secretary in position i given that it is better than all previous secretaries). Hence, the probability that ALG hires the best secretary is:

$$\frac{2}{n} \cdot \sum_{i=1}^{n/2} f_i \ . \tag{4}$$

The second best secretary also appears in each one of the $n/2$ possible positions with equal probability of $n/2$. When it appears in position i, it is the best secretary up to this point with probability $(i-1)/(n-1)$, and the second best secretary up to this point otherwise (depending on the position

of the single better secretary). ALG accepts the secretary in question with probability f_i in the first case, and probability s_i in the second case. Thus, the probability that ALG hires the second best secretary is:

$$\frac{2}{n} \cdot \sum_{i=1}^{n/2} \left(\frac{n-i}{n-1} \cdot f_i + \frac{i-1}{n-1} s_i \right) \ . \tag{5}$$

By combining (4) and (5), we get the value of the solution suggested is equal to the competitive ratio of ALG.

¿From now on, let us fix a single queue policy of ALG. Consider the case that the secretary in position i is the best secretary seen by the policy up to that point. The policy hires this secretary with probability f_i. One the other hand, consider some position $j < i$. If the secretary at this position is the best up to this point in this queue (which happens with probability $1/j$), the policy hires him with probability f_j. Similarly, if the secretary at this position is the second best up to this point (which happens with probability $1/j$ for $j \geq 2$), the secretary hires him with probability s_j. Since each policy can hire at most one secretary, the above discussion implies that the inequalities of the first type in (LP4) are satisfied by the above solution. Similarly, it is possible to prove that inequalities of the second type are also satisfied by this solution using the same argument for the case that the secretary in position i is the second best secretary up to this point.

In conclusion, any algorithm, including the optimal one, induces a feasible solution for (LP4) whose value is equal to the competitive ratio of the algorithm. Hence, the optimal value of (LP4) upper bounds the optimal competitive ratio. □

THEOREM 4.12. No algorithm is better than 0.321-competitive for the case of "exclusive" positions with $k = 2$, $Q = 2$ and $D = n/2$.

Proof. By numerically solving (LP4) for $n = 1500$, it can be shown that no algorithm with competitive ratio better than 0.3205 exists for 1500 secretaries. Assume for the sake of contradiction that there exists an algorithm ALG with a competitive ratio $\alpha > 0.321$ for n' secretaries, where $n' > n$. Let us use ALG to get an algorithm for n secretaries with a competitive ratio better than 0.3205.

ALG is composed of two policies, one for each queue. From the view point of the policy of the first queue, it gets a random permutation of a random set of $n'/2$ secretaries that can contain at most two secretaries that give a point to the policy when hired. Let $E_{i,1}$ be the expected number of points this policy gets when given a random permutation of a set containing i secretaries that give a point when hired (note that this expectation depends only on i). The expected number of points gained by this policy is:

$$E_{1,1} \cdot \frac{n'/2}{n'-1} + E_{2,1} \cdot \frac{n'/2-1}{n'-1} \ .$$

Let us we define $E_{i,2}$ as the expected number of points gained by the policy of the second queue when given a random permutation of a set containing i secretaries that give a point when hired. Using an argument similar to the above, we can prove that the expected number of points gained by this policy is:

$$E_{1,2} \cdot \frac{n'/2}{n'-1} + E_{2,2} \cdot \frac{n'/2-1}{n'-1} \ .$$

At this point, we can use the linearity of the expectation to calculate the expected number of points gained by ALG as a whole. Using this notation, our assumption that ALG is α-competitive can be stated as:

$$\frac{E_{1,1}+E_{1,2}}{2} \cdot \frac{n'/2}{n'-1} + \frac{E_{2,1}+E_{2,2}}{2} \cdot \frac{n'/2-1}{n'-1} \geq \alpha \ .$$

We now construct an algorithm for n secretaries using ALG. Each policy of our algorithm feeds the corresponding policy of ALG with the input it gets plus additional $(n'-n)/2$ dummy secretaries that are worse than any secretary of $\mathcal{S}$. The positions of the dummy secretaries are chosen at random from the $n'/2$ positions in the input of ALG. The competitive ratio of this algorithm is:

$$\begin{aligned}
&\frac{E_{1,1}+E_{1,2}}{2}\cdot\frac{n/2}{n-1}+\frac{E_{2,1}+E_{2,2}}{2}\cdot\frac{n/2-1}{n-1}\\
&\geq \frac{E_{1,1}+E_{1,2}}{2}\cdot\frac{n'/2}{n'-1}+\frac{E_{2,1}+E_{2,2}}{2}\cdot\left[\frac{n'/2-1}{n'-1}-\frac{n}{2(n-1)^2}\right]\\
&\geq \alpha-\frac{n}{2(n-1)^2}>0.3205\ .\quad\square
\end{aligned}$$

□

LEMMA 4.13. For a random pair $(\mathcal{I}_P,\mathcal{I}_T)$, $\Pr[(\mathcal{I}_P,\mathcal{I}_T)\in C_1]\geq t_1\ln(t_2/t_1)+t_1^2t_2-t_1^2-2t_1t_2+2t_1-o(1)\geq 0.4186$.

Proof. Let A_x be the event that s_1 arrives at time x. Given that A_x occurs for some $x\in(t_1,t_2-D^{1/3})$, **E1** holds with probability t_1/x. If $x>t_2$, than:

- **E2** holds with probability:

$$\frac{t_1}{x+D^{-1/3}}\cdot\left[\frac{t_2}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]\ .$$

- **E3** holds with probability:

$$\frac{t_1}{x}\cdot\left[\frac{t_2-t_1}{x+D^{-1/3}}+\left(1-\frac{t_2}{x+D^{-1/3}}\right)\cdot\frac{t_1}{x+D^{-1/3}}\right]\ .$$

Moreover, **E1**, **E2** and **E3** are disjoint events. Thus for $x\in(t_1,t_2-D^{1/3})$,

$$\Pr[(\mathcal{I}_P,\mathcal{I}_T)\in C_1|A_x]=\frac{t_1}{x+D^{-1/3}}\ ,$$

and for $x>t_2$,

$$\begin{aligned}
\Pr[(\mathcal{I}_P,\mathcal{I}_T)\in C_1|A_x]=&\frac{t_1}{x+D^{-1/3}}\cdot\left[\frac{t_2}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]\\
&+\frac{t_1}{x}\cdot\left[\frac{t_2-t_1}{x+D^{-1/3}}+\left(1-\frac{t_2}{x+D^{-1/3}}\right)\cdot\frac{t_1}{x+D^{-1/3}}\right]\ .
\end{aligned}$$

The probability that s_1 arrives in an interval of size ℓ is ℓ. Hence, the probability it arrives in an infinitesimal interval of size dx is dx. Therefore, by the law of total probability, the probability

$Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_1]$ is lower bounded by:

$$\begin{aligned}
&\int_{t_1}^{t_2 - D^{-1/3}} \frac{t_1}{x} dx + \int_{t_2}^{1} \frac{t_1}{x + D^{-1/3}} \cdot \left[\frac{t_2}{x} + \left(1 - \frac{t_2}{x}\right) \cdot \frac{t_1}{x} \right] dx \\
&+ \int_{t_2}^{1} \frac{t_1}{x} \cdot \left[\frac{t_2 - t_1}{x + D^{-1/3}} + \left(1 - \frac{t_2}{x + D^{-1/3}}\right) \cdot \frac{t_1}{x + D^{-1/3}} \right] dx \\
\geq &\int_{t_1}^{t_2} \frac{t_1}{x} dx + \int_{t_2}^{1} \frac{t_1}{x} \cdot \left[\frac{t_2}{x} + \left(1 - \frac{t_2}{x}\right) \cdot \frac{t_1}{x} \right] dx \\
&+ \int_{t_2}^{1} \frac{t_1}{x} \cdot \left[\frac{t_2 - t_1}{x} + \left(1 - \frac{t_2}{x}\right) \cdot \frac{t_1}{x} \right] dx - o(1) \\
\geq &\int_{t_1}^{t_2} \frac{t_1}{x} dx + \int_{t_2}^{1} \frac{t_1}{x} \cdot \left[\frac{t_1 + 2t_2}{x} - \frac{2t_1 t_2}{x^2} \right] dx - o(1)
\end{aligned}$$

$$\begin{aligned}
&= t_1 \ln x \Big|_{t_1}^{t_2} + \left[\frac{t_1^2 t_2}{x^2} - \frac{t_1(t_1 + 2t_2)}{x} \right]_{t_2}^{1} - o(1) \\
&= t_1 \ln(t_2/t_1) + [t_1^2 t_2 - t_1^2 - 2t_1 t_2] - [t_1^2/t_2 - t_1^2/t_2 - 2t_1] - o(1) \\
&= t_1 \ln(t_2/t_1) + t_1^2 t_2 - t_1^2 - 2t_1 t_2 + 2t_1 - o(1) \ . \quad \square
\end{aligned}$$

□

LEMMA 4.14. For a random pair $(\mathcal{I}_P, \mathcal{I}_T)$, $\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_2] \geq t_1 \ln(t_2/t_1) + t_1 t_2 \ln t_2 + 0.5 t_1^2 t_2 - 2.5 t_1 t_2 + 2t_1 - o(1) \geq 0.2951$.

Proof. Let A_x be the event that s_2 arrives at time x. Given that A_x, the probability that $q_1 \neq q_2$ or s_1 arrives after s_2 is $(2-x)/2$. Hence, the probability that **F1** occurs is equal to the probability that **E1**, **E2** or **E3** occurs times $(2-x)/x$. For $x > t_2$, we also have that the probability that **F2** occurs is:

$$\frac{x - t_2}{2} \cdot \frac{t_1}{t(s_2, \mathcal{I}_T) + D^{-1/3}} \cdot \frac{t_1}{t(s_2, \mathcal{I}_T)} \ .$$

Moreover, **F1**, and **F2** are disjoint events. Thus for $x \in (t_1, t_2 - D^{1/3})$,

$$\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_2 | A_x] = \frac{2 - x}{2} \cdot \frac{t_1}{x + D^{-1/3}} \ ,$$

and for $x > t_2$,

$$\begin{aligned}
\Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_1 | A_x] = &\frac{t_1}{x + D^{-1/3}} \cdot \left[\frac{t_2}{x} + \left(1 - \frac{t_2}{x}\right) \cdot \frac{t_1}{x} \right] \\
&+ \frac{t_1}{x} \cdot \left[\frac{t_2 - t_1}{x + D^{-1/3}} + \left(1 - \frac{t_2}{x + D^{-1/3}}\right) \cdot \frac{t_1}{x + D^{-1/3}} \right] \\
&+ \frac{x - t_2}{2} \cdot \frac{t_1}{t(s_2, \mathcal{I}_T) + D^{-1/3}} \cdot \frac{t_1}{t(s_2, \mathcal{I}_T)} \ .
\end{aligned}$$

The probability that s_2 arrives in an interval of size ℓ is ℓ. Hence, the probability it arrives in an infinitesimal interval of size dx is dx. Therefore, by the law of total probability, the probability

$Pr[(\mathcal{I}_P, \mathcal{I}_T) \in C_2]$ is lower bounded by:

$$\begin{aligned}
&\int_{t_1}^{t_2-D^{-1/3}} \frac{2-x}{2}\cdot\frac{t_1}{x}dx + \int_{t_2}^{1}\frac{2-x}{2}\cdot\frac{t_1}{x+D^{-1/3}}\cdot\left[\frac{t_2}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]dx\\
&+\int_{t_2}^{1}\frac{2-x}{2}\cdot\frac{t_1}{x}\cdot\left[\frac{t_2-t_1}{x+D^{-1/3}}+\left(1-\frac{t_2}{x+D^{-1/3}}\right)\cdot\frac{t_1}{x+D^{-1/3}}\right]dx\\
&+\int_{t_2}^{1}\frac{x-t_2}{2}\cdot\frac{t_1}{x+D^{-1/3}}\cdot\frac{t_1}{x}dx\\
\geq\ &\int_{t_1}^{t_2}\frac{2-x}{2}\cdot\frac{t_1}{x}dx+\int_{t_2}^{1}\frac{2-x}{2}\cdot\frac{t_1}{x}\cdot\left[\frac{t_2}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]dx\\
&+\int_{t_2}^{1}\frac{2-x}{2}\cdot\frac{t_1}{x}\cdot\left[\frac{t_2-t_1}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]dx+\int_{t_2}^{1}\frac{x-t_2}{2}\left(\frac{t_1}{x}\right)^2dx-o(1)\\
\overset{(*)}{=}\ &[t1\ln(t_2/t_1)+t_1^2t_2-t_1^2-2t_1t_2+2t_1]\\
&-\int_{t_1}^{t_2}\frac{t_1}{2}dx-\int_{t_2}^{1}\frac{t_1}{2}\cdot\left[\frac{t_2}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]dx
\end{aligned}$$

$$\begin{aligned}
&-\int_{t_2}^{1}\frac{t_1}{2}\cdot\left[\frac{t_2-t_1}{x}+\left(1-\frac{t_2}{x}\right)\cdot\frac{t_1}{x}\right]dx+\int_{t_2}^{1}\frac{x-t_2}{2}\left(\frac{t_1}{x}\right)^2dx-o(1)\\
=\ &[t_1\ln(t_2/t_1)+t_1^2t_2-t_1^2-2t_1t_2+2t_1]-t_1(t_2-t_1)/2-\int_{t_2}^{1}\frac{t_1}{2}\cdot\left[\frac{2t_2}{x}-\frac{t_1t_2}{x^2}\right]dx-o(1)\\
=\ &[t_1\ln(t_2/t_1)+t_1^2t_2-t_1^2/2-2.5t_1t_2+2t_1]-\frac{t_1}{2}\cdot\left[2t_2\ln x+\frac{t_1t_2}{x}\right]_{t_2}^{1}-o(1)\\
=\ &[t_1\ln(t_2/t_1)+t_1^2t_2-t_1^2/2-2.5t_1t_2+2t_1]-\frac{t_1}{2}\cdot[t_1t_2-2t_2\ln t_2-t1]-o(1)\\
=\ &t_1\ln(t_2/t_1)+t_1t_2\ln t_2+0.5t_1^2t_2-2.5t_1t_2+2t_1-o(1)\ ,
\end{aligned}$$

where (*) follows from the proof of Lemma 4.13. □

C Omitted Proofs of Section 6

THEOREM 6.2. No better than $1-1/(2e)+o(1)\approx 0.816$-competitive algorithm exists for the case $Q=k$, $D=n/k$ for large k and "shared" positions.

Proof. Consider some algorithm ALG for the problem. Fix some distribution P of the secretaries among the queue. After fixing P the randomness of the input means that each queue gets a random permutation of a constant set of n/k secretaries. Each queue has some probability p_i that at least one secretary is hired from this queue. Notice that the probabilities p_i are independent of the distribution P because the algorithm can only compare the secretaries of one queue to each other.

Let us now lower bound the total number of secretaries that ALG man with non-top k secretaries. Each queue q_i contains no top k secretary with probability:

$$(1-1/k)^k \geq e^{-1}\cdot(1-1/k) = e^{-1}-o(1)\ .$$

On the other hand, ALG hires from q_i with probability at least p_i, regardless of D, and therefore, ALG hires from q_i a non-top k secretary with probability at least:

$$[e^{-1} - o(1)] \cdot p_i \ .$$

Summing over all top k secretaries, we can lower bound the total number of non-top k secretaries hired by:

$$[e^{-1} - o(1)] \cdot \sum_{i=1}^{k} p_i \ . \tag{6}$$

Next, we lower bound the total number of top k secretaries that ALG does not hire. Each top k secretary s arrives to each queue q_i with equal probability. If no secretary is hired from q_i (which happens with probability $1 - p_i$), then clearly ALG does not hire s. Hence, the probability that s is not hired is at least:

$$\frac{\sum_{i=1}^{k} 1 - p_i}{k} \ .$$

Summing over all top k secretaries, we can lower bound the total number of top k secretaries not hired by:

$$\sum_{i=1}^{k} 1 - p_i \tag{7}$$

Combining (6) and (7), we get that the competitive ratio of ALG is at most:

$$1 - \frac{\max\left\{[e^{-1} - o(1)] \cdot \sum_{i=1}^{k} p_i, \sum_{i=1}^{k} 1 - p_i\right\}}{k} \leq 1 - \frac{e^{-1}}{2} + o(1) \ . \quad \square$$

$\square$

THEOREM 6.4. No better than 0.301-competitive algorithm exists for the case $Q = k$, $D = n/k$ for large k and "exclusive" positions.

Proof. Consider some arbitrary algorithm ALG for the problem. Notice that ALG can hire at most one secretary from each queue. Let us upper bound the probability of the event E_q that ALG hires a top k secretary from some queue q.

First let us lower bound the probability that q contains i secretaries for some constant i. For $i \geq 1$, we can use the following lower bound:

$$\binom{k}{i} \cdot k^{-i}(1 - 1/k)^{k-i} = \frac{k! \cdot (1 - 1/k)^{k-i}}{k^i(k-i)! \cdot i!} = \frac{(1 - 1/k)^{k-i}}{i!} \geq \frac{e^{-1}}{i!} \ .$$

For $i = 0$, we use the lower bound:

$$(1 - 1/k)^k \geq e^{-1} \cdot (1 - 1/k) = e^{-1} - o(1) \ .$$

Assume now that we have an upper bound B_i on the probability that ALG manages to hire a top k secretary from q assuming there are i top k secretaries there. Using the previous lower bounds on the probability that there are exactly i top k secretaries in q, we get the following lower bound on $\Pr[E_q]$.

$$\Pr[E_q] \leq 1 - \sum_{i=0}^{\ell} (1 - B_i) \cdot \frac{e^{-1}}{i!} \ , \tag{8}$$

Table 3: B_i values

i	B_i	n **used**
2	0.575	1100
3	0.709	1000
4	0.800	900
5	0.862	800

where ℓ is an arbitrary constant (the larger is ℓ the better is the bound).

Notice that by the linearity of the expectation, the expected value that ALG collects is given by $\sum_{j=1}^{k} \Pr[E_{q_j}]$, where E_{q_j} is the event that ALG hires a top k secretary from queue q_j. Since (8) holds for all queues q_j, we get that the competitive ratio of ALG is also upper bounded by (8).

Hence, it all boils down to providing good upper bounds B_i. For $i = 0$, there is no top k secretary in q, and therefore, ALG has zero probability of hiring such a secretary from q, *i.e.*, $B_i = 0$. For $i = 1$, there is a single top k secretary in q, *i.e.*, we get the classical secretary problem in q. No algorithm can hire this single secretary with probability over $B_1 = e + o(1)$.

Plugging the two above values of B_i into (8) results in an upper bound of ≈ 0.632. To strengthen this result we use the method of [5]. Notice that B_i is upper bounded by the achievable competitive ratio for the auxiliary problem. Buchbinder et al. [5] describe an LP whose value is the competitive ratio of the auxiliary problem for n secretaries. Due to Lemma 6.3, such a solution is also an upper bound on the competitive ratio of the auxiliary problem for a general number of secretaries, *i.e.*, it is a possible value for B_i. Table 3 give some B_i values achieved this way. The theorem follows by plugging these values of B_i into (8). □

THEOREM 6.5. Algorithm 7 is a $t \cdot \left[\mathsf{Ei}(-1) + e^{-1} - 1 - \mathsf{Ei}(-t) - \frac{e^{-t}}{t} + t^{-1}\right]$ competitive algorithm for the case $Q = k$ and $D = n/k$. Hence, for $t = 0.323$, Algorithm 7 is a 0.276-competitive algorithm.

Proof. Let s_j be the j^{th} best secretary in $\mathcal{S}$. Assuming that s_j arrives at time $x > t$, he is hired if:

- No better secretary appears before him.
- The best secretary in the range $[0, x)$ arrives before time t, or this range is empty.

And both conditions occur with probability at least $(t/x) \cdot (1 - x/k)^{j-1}$. Dropping the assumption that s_j arrives at time time x, we get that s_j is hired with at least the following probability.

$$\int_t^1 \frac{t}{x} \cdot \left(1 - \frac{x}{k}\right)^{j-1} dx \ .$$

Summing over all top k secretaries, we get that the expected number of top k secretaries that are

hired by Algorithm 7 is at least:

$$\begin{aligned}
\sum_{j=1}^{k}\left(\int_t^1 \frac{t}{x}\cdot\left(1-\frac{x}{k}\right)^{j-1}dx\right) &= \int_t^1\left(\frac{t}{x}\cdot\sum_{j=1}^{k}\left(1-\frac{x}{k}\right)^{j-1}\right)dx \\
&= \int_t^1\left(\frac{t}{x}\cdot\frac{(1-x/k)^k-1}{(1-x/k)-1}\right)dx \\
&= -tk\cdot\int_t^1\left(\frac{(1-x/k)^k-1}{x^2}\right)dx \geq -tk\cdot\int_t^1\left(\frac{e^{-x}-1}{x^2}\right)dx \\
&= tk\cdot\left[\mathsf{Ei}(-x)+\frac{e^{-x}}{x}-x^{-1}\right]_t^1 \\
&= tk\cdot\left[\mathsf{Ei}(-1)+e^{-1}-1-\mathsf{Ei}(-t)-\frac{e^{-t}}{t}+t^{-1}\right] \quad . \quad \square
\end{aligned}$$

□

THEOREM 6.6. For large k, Algorithm 8 is a 0.288-competitive algorithm for the case $Q = k$ and $D = n/k$.

Proof. Let s_j be the j^{th} best secretary in $\mathcal{S}$. Let us calculate the probability s_j is hired. Using argument from the proof of Theorem 6.5, we get that if s_j arrives at some time $x \in (t_1, t_2)$, he is hired with probability:

$$\frac{t_1}{x}\cdot\left(1-\frac{x}{k}\right)^{j-1} \quad .$$

Consider now the case that s_j arrives at time $x > t_2$. There are two options to consider. The first option is that no better secretary arrives before s_j. This options occurs with probability $(1-x/k^{j-1}$. If this option occurs, then s_j is hired if the best secretary of the range $[0, x)$ arrives before time t_1 (or this range is empty), and the second best secretary of this range arrives before time t_2 (or there are less than 2 secretaries in this range). Hence, a secretary s_j is hired due to this option with probability:

$$\frac{t_2}{x}\cdot\frac{t_1}{x}\cdot\left(1-\frac{x}{k}\right)^{j-1} \quad .$$

The other option is that exactly one better secretary s_b arrives before s_j, and it arrives before time t_1. This option occurs with probability $(j-1)\cdot(t_1/k)\cdot(1-x/k)^{j-2}$ For s_j to be hired under this option, the second best secretary of the range $[0, x)$ must arrive before time t_2 (or this range contains s_b alone). Hence, a secretary s_j is hired due to this option with probability:

$$(j-1)\cdot\frac{t_1}{k}\cdot\left(1-\frac{x}{k}\right)^{j-2}\cdot\frac{t_2}{x} \quad .$$

Since the two above options are disjoint, the sum of the above two probabilities is the probability that s_j is hired by Algorithm 8 given that $x > t_2$. Removing the assumption that s_j arrives on time x, we get he is hired with probability:

$$\int_{t_1}^{t_2}\frac{t_1}{x}\cdot\left(1-\frac{x}{k}\right)^{j-1}dx+\int_{t_2}^{1}\frac{t_2}{x}\cdot\frac{t_1}{x}\cdot\left(1-\frac{x}{k}\right)^{j-1}dx+\int_{t_2}^{1}(j-1)\cdot\frac{t_1}{k}\cdot\left(1-\frac{x}{k}\right)^{j-2}\cdot\frac{t_2}{x}dx \quad .$$

Summing over all top k secretaries, we get that the expected number of top k secretaries hired by Algorithm 8 is at least:

$$\sum_{j=1}^{k} \left(\int_{t_1}^{t_2} \frac{t_1}{x} \cdot \left(1 - \frac{x}{k}\right)^{j-1} dx + \int_{t_2}^{1} \frac{t_2}{x} \cdot \frac{t_1}{x} \cdot \left(1 - \frac{x}{k}\right)^{j-1} dx \right. \tag{9}$$
$$\left. + \int_{t_2}^{1} (j-1) \cdot \frac{t_1}{k} \cdot \left(1 - \frac{x}{k}\right)^{j-2} \cdot \frac{t_2}{x} dx \right) \ .$$

In the proof of Theorem 6.5, the sum of the first terms of (9) was lower bounded by:

$$t_1 k \cdot \left[\mathsf{Ei}(-t_2) + \frac{e^{-t_2}}{t_2} - t_2^{-1} - \mathsf{Ei}(-t_1) - \frac{e^{-t_1}}{t_1} + t_1^{-1} \right] \ .$$

Next, let us now lower bound the sum of the second terms of (9).

$$\begin{aligned}
\sum_{j=1}^{k} \left(\int_{t_2}^{1} \frac{t_2}{x} \cdot \frac{t_1}{x} \cdot \left(1 - \frac{x}{k}\right)^{j-1} dx \right) &= t_1 t_2 \cdot \int_{t_2}^{1} x^{-2} \cdot \left(\sum_{j=1}^{k} \left(1 - \frac{x}{k}\right)^{j-1} \right) dx \\
&= t_1 t_2 \cdot \int_{t_2}^{1} x^{-2} \cdot \left(\frac{(1 - x/k)^k - 1}{(1 - x/k) - 1} \right) dx \\
&= -t_1 t_2 k \cdot \int_{t_2}^{1} \frac{(1 - x/k)^k - 1}{x^3} dx \\
&\geq -t_1 t_2 k \cdot \int_{t_2}^{1} \frac{e^{-x} - 1}{x^3} dx \\
&= -\frac{t_1 t_2}{2} \cdot \left[\mathsf{Ei}(-x) + \frac{1 + e^{-x}(x-1)}{x^2} \right]_{t_2}^{1} \\
&= \frac{t_1 t_2}{2} \cdot \left[\mathsf{Ei}(-t_2) + \frac{1 + e^{-t_2}(t_2 - 1)}{{t_2}^2} - \mathsf{Ei}(-1) - 1 \right] \ .
\end{aligned}$$

Finally, let us lower bound the sum of the third terms of (9).

$$
\begin{aligned}
&\sum_{j=1}^{k}\left(\int_{t_2}^{1}(j-1)\cdot\frac{t_1}{k}\cdot\left(1-\frac{x}{k}\right)^{j-2}\cdot\frac{t_2}{x}dx\right)\\
&=\int_{t_2}^{1}\left(\frac{t_1t_2}{kx}\cdot\sum_{j=2}^{k}(j-1)\cdot\left(1-\frac{x}{k}\right)^{j-2}\right)dx\\
&=-\int_{t_2}^{1}\left(\frac{t_1t_2}{x}\cdot\frac{d}{dx}\left[\sum_{j=2}^{k}\left(1-\frac{x}{k}\right)^{j-1}\right]\right)dx\\
&=-\int_{t_2}^{1}\left(\frac{t_1t_2}{x}\cdot\frac{d}{dx}\left[\left(1-\frac{x}{k}\right)\cdot\frac{(1-x/k)^{k-1}-1}{(1-x/k)-1}\right]\right)dx\\
&=\int_{t_2}^{1}\left(\frac{t_1t_2}{x}\cdot\frac{d}{dx}\left[\left(\frac{k}{x}-1\right)\cdot[(1-x/k)^{k-1}-1]\right]\right)dx\\
&=-\int_{t_2}^{1}\left(\frac{t_1t_2}{x}\cdot\left[\frac{k\cdot[(1-x/k)^{k-1}-1]}{x^2}+\left(\frac{k}{x}-1\right)\cdot\frac{(k-1)(1-x/k)^{k-2}}{k}\right]\right)dx\\
&=-\int_{t_2}^{1}\left(\frac{t_1t_2}{x}\cdot\left[\frac{k\cdot[(1-x/k)^{k-1}-1]}{x^2}+\frac{(k-1)(1-x/k)^{k-1}}{x}\right]\right)dx\\
&\geq-\int_{t_2}^{1}\left(\frac{t_1t_2}{x}\cdot\left[\frac{k\cdot[e^{-x}/(1-1/k)-1]}{x^2}+\frac{(k-1)e^{-x}/(1-1/k)}{x}\right]\right)dx\\
&=\frac{t_1t_2}{1-1/k}\cdot\int_{t_2}^{1}\left(\frac{k\cdot[1-1/k-e^{-x}]}{x^3}-\frac{(k-1)e^{-x}}{x^2}\right)dx\\
&=\frac{t_1t_2}{2(1-1/k)}\cdot\left[(k-2)\mathsf{Ei}(-x)+\frac{1-k+e^{-x}[kx+k-2x]}{x^2}\right]_{t_2}^{1}\\
&=\frac{t_1t_2}{2(1-1/k)}\cdot\left[(k-2)\mathsf{Ei}(-1)+1-k+e^{-1}[2k-2]-(k-2)\mathsf{Ei}(-t_2)\right.\\
&\left.-\frac{1-k+e^{-t_2}[kt_2+k-2t_2]}{{t_2}^2}\right]\ .
\end{aligned}
$$

Plugging all the above lower bounds into (9), and diviing by k, we get the following lower bound for the competitive ratio of Algorithm 8.

$$
\begin{aligned}
&t_1\cdot\left[\mathsf{Ei}(-t_2)+\frac{e^{-t_2}}{t_2}-t_2^{-1}-\mathsf{Ei}(-t_1)-\frac{e^{-t_1}}{t_1}+t_1^{-1}\right]\\
&+\frac{t_1t_2}{2}\cdot\left[\mathsf{Ei}(-t_2)+\frac{1+e^{-t_2}(t_2-1)}{{t_2}^2}-\mathsf{Ei}(-1)-1\right]\\
&+\frac{t_1t_2}{2k(1-1/k)}\cdot\left[(k-2)\mathsf{Ei}(-1)+1-k+e^{-1}[2k-2]-(k-2)\mathsf{Ei}(-t_2)\right.\\
&\left.-\frac{1-k+e^{-t_2}[kt_2+k-2t_2]}{{t_2}^2}\right]\ .
\end{aligned}
$$

As k increases, this competitive ratio goes to:

$$\begin{aligned}
&t_1 \cdot \left[\mathsf{Ei}(-t_2) + \frac{e^{-t_2}}{t_2} - t_2^{-1} - \mathsf{Ei}(-t_1) - \frac{e^{-t_1}}{t_1} + t_1^{-1}\right] \\
&+\frac{t_1 t_2}{2} \cdot \left[\mathsf{Ei}(-t_2) + \frac{1 + e^{-t_2}(t_2 - 1)}{{t_2}^2} - \mathsf{Ei}(-1) - 1\right] \\
&+\frac{t_1 t_2}{2} \cdot \left[\mathsf{Ei}(-1) + e^{-1}(2 - e) - \mathsf{Ei}(-t_2) + \frac{1 - e^{-t_2}(t_2 + 1)}{{t_2}^2}\right] \\
&= t_1 \cdot \left[\mathsf{Ei}(-t_2) - \mathsf{Ei}(-t_1) - \frac{e^{-t_1}}{t_1} + t_1^{-1} + t_2(e^{-1} - 1)\right] \quad .
\end{aligned}$$

The theorem now follows by plugging the values of t_1 and t_2 into the above expression. □