HIRING MULTIPLE TEMPORARY SECRETARIES

Abstract. Similar ideas and strategies emerge from the classic secretary problem, the multiple choice problem, and the temporary secretary problem. We review two unique approaches to hiring k secretaries and two further approaches to the generalized problem with temporal constraints. We explain how the algorithms build upon each other, which leads us to an overview of recent best-in-class results as well as future research directions.

1. Introduction

The most famous variant of the classic problem was first posed and solved in the 1960s. Ferguson wrote a neat historical overview of the secretary problem in [1], noting that the problem was considered as early as 1611 when Kepler methodically set about finding a second wife. The multi-choice and temp secretary problems are more recent, the latter formally posed by Fiat et al. in a 2015 paper [7]. Important applications abound, for example choosing a set of measurements or images through a sample trajectory in the context of robotic vision [4] or using web sites such as Airbnb and oDesk to find short term lodging or freelance employees [7].

In this paper we compare approaches to two variations of the *classic secretary problem*. The goal of this problem is to select the best secretary out of a fixed candidate pool in an online setting. Candidates arrive sequentially so we only know about previous candidates and the current candidate we are interviewing. We must accept or reject a candidate immediately after her interview, and once the choice is made, it is final.

Optimal stopping theory yields a solution to the classic problem with constant success probability and competitive ratio $1/e$ in the limit as the number of candidates n goes to infinity. The strategy is to observe and unilaterally reject some number of candidates r, after which point we accept the next one which is better than all the previous. In this way, the best previous candidate provides a threshold comparison.

The first variation, called the *multiple choice secretary problem* (multi-choice), extends the classic secretary problem by asking what to do when we need to hire k secretaries. Both the multi-choice and classic problem assume we keep the same secretaries employed forever once they are hired.

For the multi-choice problem, there are a few natural criteria one may optimize for [4], all of which make use of the solution method to the classic secretary problem. In this paper, we explore the difference between optimizing for the probability of choosing the exact k best secretaries and optimizing for the expected value of the hired secretaries. Approaches have been proposed for both of these criteria: Girdhar and Dudek's *fixed-threshold algorithm* in [4] and *Kleinberg's recursive algorithm* in [2], respectively. The fixed-threshold algorithm finds the optimal value of r in much the same way as the classic secretary problem, and we hire all candidates after r that pass our threshold. Kleinberg's algorithm recursively breaks the candidates

Date: January 5, 2017.

into two groups, using the first group to create a threshold for selecting from the second. In the base case, Kleinberg makes use of the classic algorithm.

The second variation, called the *temporary secretary problem* (temp), extends the multiple-choice problem further by asking what to do if the secretaries' contracts are of fixed duration. In this setting, we imagine the sequence of candidates mapped to various arrival times on the unit interval, and we employ those selected for an employment term of constant duration γ which is much smaller than one.

For the temp secretary problem, Fiat et al. propose a Kleinberg-inspired recursive approach called the *Charter algorithm*. Recently, improved results have been found in [8] by Kesselheim and Tönnis with their *linear scaling algorithm*, which more closely resembles the fixed-threshold algorithm.

In this paper, we begin by reviewing the proof of the classic secretary problem. We proceed with a description of Kleinberg's algorithm and the fixed-threshold algorithm, with an examination of the relationship between the two. We then describe the Charter algorithm and the improvements of the linear scaling algorithm, with a focus on how these arise from previous methods. We conclude with areas of future research.

Beyond the scope of this paper are other variations such as the matroid and knapsack secretary problems. A neat review is contained in [3] by Babaioff (the Tarjan of secretary problems) et al.

2. Classic Secretary Problem

In the classic secretary problem, we have a set of n applicants each with value v_i and we want to hire the best one. We interview one at a time in random order, and we must either hire or reject the candidate immediately after the interview. We are not allowed to go back and hire a previously rejected candidate, and we assume candidate values are distinct (or equivalently, that we have a consistent tie-breaking scheme).

The optimal strategy is to review the first r candidates without accepting any, find the highest score among them, then hire the first candidate with score higher than that. As we will see, this is the fundamental stopping theory which arises again and again in approaches to extensions of the classic problem.

To choose r, consider the probability of successfully finding the candidate of highest value:

$$\mathbb{P}(\text{found best}) = \sum_{i=r+1}^{n} \mathbb{P}(i \text{ is best and previous candidates not selected}) \tag{2.1}$$

$$= \sum_{i=r+1}^{n} \frac{1}{n}\frac{r}{i-1} \tag{2.2}$$

$$= \frac{r}{n}\sum_{i=r}^{n-1}\frac{1}{i}. \tag{2.3}$$

The above is true since $\frac{1}{n}$ is the probability that the i^{th} candidate has the highest value, and $\frac{r}{i-1}$ is the probability none of the previous candidates were selected. We optimize the probability by optimizing the integral for which the above sum is the Riemann approximation, which yields $r = \frac{n}{e}$, as $n \to \infty$. A detailed proof can be found in [5].

We can generalize the classic secretary problem by choosing k secretaries instead of one, and this introduces new questions [4]. Should we maximize the probability that one of the selected secretaries is the best one, should we maximize the expected sum of the values, or should we maximize the probability that all k of the chosen secretaries are in fact the best?

3. Multiple Choice Secretary Problem

In the multi-choice secretary problem, we extend the classic secretary problem by considering the hiring of k candidates from n applicants. In this domain, we have two natural choices of functions to optimize for. We can either directly extend the classic approach and maximize the probability of choosing the k best candidates exactly, or we can maximize the expected value of the chosen candidates. Interestingly, these two goals yield two very different algorithms and bounds which we explore in the next couple subsections.

3.1. Maximizing Best-Case Scenario.

The classic secretary algorithm already maximizes the probability of selecting the k best candidates for $k = 1$. It does this by interviewing the first $r = n/e$ candidates and using the max as the threshold for selecting candidates. Girdhar and Dudek extend this to multi-choice with their fixed-threshold algorithm in [4], by making r a function of k as well as a function of n. Intuitively, as k increases, r should decrease. They strongly suggest but do not formally prove that the optimal r value is

$$r(k) = \frac{n}{ke^{1/k}}. \tag{3.1}$$

Analysis. Let success be defined by the event that all of the top k highest scoring candidates have been selected. Let S_i^k be the event that with the selection of the ith candidate, we have succeeded. Note that this is not possible for the first $r+k-1$ candidates. Thus,

$$P(success) = \sum_{i=r+k}^{n} P(S_i^k) = \sum_{i=r+k}^{n} \frac{k}{n}\frac{r}{i-1}\frac{\binom{i-r-1}{k-1}}{\binom{n}{k-1}}. \tag{3.2}$$

The first term is the probability the ith candidate is one of the top k candidates. The second term is the probability that none of the previous candidates were the last of the top k candidates. The last term is the probability that all the remaining $k-1$ candidates have been selected.

Evaluating the expression analytically is extremely difficult, but for large n, we can numerically approximate the value of r for which this probability of success is maximized, given k. Girdhar and Dudek demonstrate that equation 3.1 fits the data remarkably well [4].

3.2. Maximizing Expected Value.

Although maximizing probability is interesting, far more applicable to real world problems is an algorithm that maximizes the expected value of the chosen candidates.

Consider for a moment the classic secretary problem where $k = 1$ and the probability of choosing the best candidate tends to $1/e$ as $n \to \infty$. This means expected value of the chosen candidate also tends to a competitive ratio of at least e.

This raises the question, what is the competitive ratio for larger k? A plausible algorithm is to generalize the classic solution by using a simple sliding threshold.

(1) Observe the first $\lfloor n/e \rfloor$ candidates.
(2) Remember the best k candidates and call this set S.
(3) Hire every candidate that is better than the worst candidate in S and replace the worst candidate.

Babaioff et al. showed in [3] that this is similar enough to the classic algorithm to obtain a competitive ratio of no worse than e for all k. Kleinberg, however, introduced a recursive algorithm that achieves a competitive ratio of $1 - O(\sqrt{1/k})$ [2]. Remarkably, the competitive ratio approaches 1 as $k \to \infty$. Since we are already familiar with simple threshold approaches, we focus on Kleinberg's more challenging algorithm in this section.

3.2.1. *Kleinberg's Algorithm.* The algorithm is recursively defined as follows.

(1) If $k = 1$, use the classic secretary algorithm to select one candidate.
(2) For $k > 1$, split the n candidates into two groups. See details below.
(3) Select $l = \lfloor k/2 \rfloor$ candidates from the first group recursively using this algorithm.
(4) Select the rest of the candidates from the second group. Accept if the candidate is better than the l^{th} best candidate seen in the first group. Note that this is different from the worst candidate accepted in the first group.

To split n candidates into two groups, let m be a random sample from the binomial distribution $B(n, 1/2)$ or the number of heads in n coin flips. Let the first group contain the first m candidates and the second group contain the rest.

Analysis. We will outline how Kleinberg's algorithm gives a selection with expected value within a $(1-\frac{5}{\sqrt{k}})$ factor of the optimal value OPT^*. This analysis will develop intuition and tools which guide later analysis of the more complex cases.

We begin by defining the set S_k to contain the k highest-valued candidates. View $y_1, \ldots, y_m \in Y$ and $z_1, \ldots, z_{n-m} \in Z$ as the left m candidates and the right $n - m$ candidates, respectively. These are the two groups for the recursion described in the algorithm.

We will combine the expected values obtained in each group to prove the result. We proceed with the left group by induction on k. We set the inductive hypothesis as the desired result, so for the induction step, the group should be within a $(1-\frac{5}{\sqrt{k}})$ factor of that group's optimal value OPT.

We immediately have three key facts.

(1) The first group Y has the probability distribution $B(k, \frac{1}{2})$ for containing some number of elements in S_k.
(2) If there are r elements from S_k in Y, $\mathbb{E}[Y] = \frac{r}{k} OPT^*$.
(3) The expected value of the best $l = \frac{k}{2}$ elements of Y is bounded below by

$$\sum_{r=1}^{k} \mathbb{P}\left(|Y \cap S_k| = r\right) \cdot \left(\frac{\min(r, l)}{k}\right) OPT^* \geq \frac{1}{2}\left(1 - \frac{1}{2\sqrt{k}}\right) OPT^*. \tag{3.3}$$

This is the probability Y contains r of the best elements times the proportion of how many we can select out of our total times the optimal value.

Now we are ready to induct using $l = \frac{k}{2}$ as stated in the algorithm. We take the competitive ratio hypothesis times the expected value of the top elements from Y

$$(1 - \frac{5}{\sqrt{l}}) \times \frac{1}{2}\left(1 - \frac{1}{2\sqrt{k}}\right) OPT^* \tag{3.4}$$

as the expected value the algorithm finds in the left group.

On the right, we count the number of elements q exceeding y_l, the threshold the algorithm chose. We call q_i the number of elements whose value lies between y_i and y_{i-1}. These variables are "stochastically dominated by i.i.d. geometrically distributed random variables each having mean 1 and variance 2" [2], so their sum from $i = 1$ to l, denoted q, satisfies $\mathbb{E}\left[|q - l|\right] \leq \sqrt{k}$.

Let $r = |q - l|$, then condition on r to find expected value of selected elements is at least

$$\left(\frac{1}{2} - \frac{r}{k}\right) OPT^* = (\frac{1}{2} - \frac{1}{\sqrt{k}})OPT^* \tag{3.5}$$

once we remove the conditioning and substitute for r.

Finally, we combine the left and right competitive ratios,

$$(1 - \frac{5}{\sqrt{l}}) \times \frac{1}{2}\left(1 - \frac{1}{2\sqrt{k}}\right) + (\frac{1}{2} - \frac{1}{\sqrt{k}}) \geq 1 + \frac{5\sqrt{2}}{4k} - \frac{5 + 10\sqrt{2}}{4\sqrt{k}} \geq 1 - \frac{5}{\sqrt{k}}, \tag{3.6}$$

as desired.

4. Temporary Secretary Problem

In the temp secretary problem, we extend the multi-choice secretary problem by considering the hiring of candidates who show up for interviews during the time interval $[0, 1]$. Arrival times are drawn from a probability distribution, typically uniform. We are constrained to simultaneous employment of up to B secretaries as our "bandwidth," and we can hire a total of up to k secretaries overall as our "budget." All secretaries are hired for a certain length of time $\gamma << 1$.

Just as in the previous section, we find two very different approaches which build on their multi-choice counterparts in interesting ways. Unlike the previous section, both approaches here have the same goal to maximize the expected value of the chosen candidates, so we can now have a clear winner in terms of competitive ratio.

We will explore the competitive ratios of the Charter and linear scaling algorithms in the following subsections, both of which are asymptotically tight for $\gamma \to 0$ [8]. We can compare the two algorithms by removing the budget constraint. The recursive Charter algorithm of Fiat et al. attains at least $\frac{1}{2}(1 - O(\sqrt{\gamma \ln(\frac{1}{\gamma})}))$ for $B = 1$ and $1 - O(\frac{\sqrt{\ln B}}{\sqrt{B}}) - O(\sqrt{\gamma \log(\frac{1}{\gamma})})$ for arbitrary B as competitive ratios [7]. The linear scaling algorithm of Kesselheim and Tönnis beats this with a competitive ratio of $\frac{1}{2} - O(\sqrt{\gamma})$ for all values of B, and $1 - O(\frac{1}{\sqrt{B}}) - O(\sqrt{\gamma})$ for large B [8]. These are the best known competitive ratios for the temp secretary problem.

4.1. Recursive Approach.

Recursive Approach. We are already familiar with the recursive approach from Kleinberg's algorithm. In fact, with contract length $\gamma = 0$ and budget k, we have the multi-choice problem, and the Charter algorithm behaves exactly like Kleinberg's algorithm.

The Charter algorithm is defined in full as follows.

(1) if $k = 1$, use the classic secretary algorithm to select one candidate.
(2) For $k > 1$, split the time interval in half and recursively process $[0, \frac{1}{2})$ with budget $\lfloor \frac{k}{2} \rfloor$ and contract length 2γ.
(3) Use the value of the candidate of rank $\frac{k}{2}$ that appeared during $[0, \frac{1}{2})$ as threshold T, or if no such rank let $T = 0$.
(4) Greedily accept all candidates that appear during $[\frac{1}{2}, 1)$ with value at least T which do not violate constraints B or k.

Interestingly, the temporal setting means that the algorithm does not rely on prior knowledge of the total candidate count, as Kleinberg's algorithm does.

Analysis. We will go over the proof idea for $B = 1$. For unconstrained budget $k = \frac{1}{\gamma}$, we want to obtain the competitive ratio $\frac{1}{2}\left(1 - O\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)}\right)$. We do this through a Kleinberg-style inductive approach. The important new concept here is that we must properly account for the feasibility of selecting elements within the constraints.

We take the competitive ratio from theorem 2 of [7],

$$\frac{1}{1+\gamma k}\left(1 - 7.4\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)} - \frac{5}{\sqrt{k}}\right), \tag{4.1}$$

as inductive hypothesis. For the inductive step, we use $\frac{k}{2}$ and 2γ per the algorithm.

The left part is easy since we just substitute to find our competitive ratio from the hypothesis,

$$\frac{1}{1+\gamma k}\left(1 - 7.4\sqrt{2\gamma \ln\left(\frac{1}{2\gamma}\right)} - \frac{5\sqrt{2}}{\sqrt{k}}\right). \tag{4.2}$$

This is the multiplier we will use on the expected value of top candidates in the left group, OPT, which is roughly $OPT^*/2$. More precisely, $OPT \geq \frac{1}{2}\left(1 - \frac{1}{2\sqrt{k}}\right) OPT^*$, as Fiat et al. show with lemma 2 in [7], but also just as we saw before in inequality 3.3 from the Kleinberg analysis! Asymptotically, we find competitive ratio

$$\frac{1}{1+\gamma k}\left(1 - 7.4\sqrt{2\gamma \ln\left(\frac{1}{2\gamma}\right)} - \frac{5\sqrt{2}}{\sqrt{k}}\right) \times \frac{1}{2}\left(1 - \frac{1}{2\sqrt{k}}\right) OPT^* \tag{4.3}$$

$$= \frac{1}{4}\left(1 - O\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)}\right) \tag{4.4}$$

for the left group, when we substitute $k = \frac{1}{\gamma}$.

The right side is harder, so we explain where the terms come from but leave the detailed derivations to [7]. The idea is to end up with expected value at least

$$\frac{1}{1+\gamma k}\left(1 - 6\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)} - 4.5\right)\left(1 - \frac{2\sqrt{1+\frac{1}{k}}}{\sqrt{k}} - \frac{1}{k}\right)\frac{1}{2}OPT^*, \tag{4.5}$$

which means the right group asymptotically yields competitive ratio

$$\frac{1}{4}\left(1 - O\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)}\right), \tag{4.6}$$

when $k = \frac{1}{\gamma}$. Adding this to the left competitive ratio clearly obtains the desired result.

The terms in expression 4.5 represent the expected size of a feasible subset chosen greedily by the algorithm,

$$\frac{1}{1+\gamma k}\left(1 - 6\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)} - 4.5\right)|OPT^*|, \tag{4.7}$$

times the summed probability that the chosen number of elements are larger than the threshold and the expected value of an average element,

$$\left(1 - \frac{2\sqrt{1+\frac{1}{k}}}{\sqrt{k}} - \frac{1}{k}\right)\frac{OPT^*}{2|OPT^*|}. \tag{4.8}$$

The expected size of a feasible subset in expression 4.7 is obtained from theorem 3 in [7]. The theorem makes use of n independently uniform samples to show that the expected size of a feasible selection by the greedy algorithm is about $\frac{n}{1+n\gamma}$, since the number of points that lie in a γ-interval starting at an arrival time is one for the chosen candidate plus $n\gamma$ for uniformly random points which fall into an interval of length γ. The $\left(1 - 6\sqrt{\gamma \ln\left(\frac{1}{\gamma}\right)} - 4.5\right)$ term comes from lemmas 6 and 9 in [7], which bound the expected error due to feasibility overlap in the interval $[\frac{1}{2}, \frac{1}{2} + \gamma]$ where we cannot select candidates who might interfere with those selected earlier, near the end of the left interval $[\frac{1}{2} - \gamma, \frac{1}{2}]$.

The final component, expression 4.8, is obtained from lemma 5 in [7], which uses the same idea as before in the Kleinberg analysis of the right group with "identical geometrical random variables" [7], combined with the fact that $2\lceil\frac{k}{2}\rceil \leq k+1$, to lower bound the sum of probabilities that the chosen elements are larger than the threshold. Of course, the last part is $\frac{OPT^*}{2|OPT^*|}$, the average expected value of the right interval's best candidates.

4.2. **Linear Scaling Approach.** Recall Babaioff et al.'s sliding threshold algorithm mentioned in the introduction to subsection 3.2 on Kleinberg. The linear scaling approach is strongly motivated by this remarkably straightfoward idea that we can take incoming candidates while incrementally adjusting a smart threshold criterion for acceptance. Kesselheim and Tönnis showed very recently in [8] that if we are smart enough, we can obtain even better results with this simple method than Fiat et al.'s relatively complex Charter algorithm for the temp secretary problem.

Kesselheim and Tönnis were certainly not the first to think along these lines. In fact, Fiat et al. describe a time-slicing algorithm in [7], where they first introduced the temp secretary problem! Their algorithm runs the classic secretary algorithm or Kleinberg's algorithm on slices of length 2γ, depending on the plurality of capacity B. The first half of the slice is where the candidates are chosen. The second half is simply left as work time for hirees from the first half, so feasibility conflicts are avoided by not even considering candidates there. Naturally, in expectation half the candidates are considered, so the time-slice algorithm achieves half of Kleinberg's $1 - 5/\sqrt{B}$ competitive ratio or $1/2e$ for the classic case where $B = 1$. Fiat et al.

suspected that time-slicing could be improved by making better use of information revealed over time, and the linear scaling algorithm confirms this suspicion.

Before we describe the algorithm, we make one final note that the linear scaling approach also resembles Girdhar and Dudek's probability-maximizing fixed-threshold algorithm, in that both alter the classic approach by changing the hiring criterion, which is a threshold function of some parameters. Namely, the fixed-threshold algorithm depends on the quantity of candidates n, hiring constraint k, and candidate arrival order. By the same token, the linear scaling algorithm depends on the quantity of time $[0, 1]$, hiring constraints B and γ, and candidate arrival time.

The linear scaling algorithm is defined as follows. For each candidate:

(1) Set $t =$ arrival time of the next candidate.
(2) Let S_t be the $\lfloor t\frac{B}{\gamma}\rfloor$ highest-valued candidates seen thus far.
(3) Accept the candidate if she is in S_t, without violating constraints B or k.

The key insight is that in expectation, a t fraction of the optimal secretaries have been seen at time t, and the expected value of the optimal solution can be upper bounded by the value of the $\frac{B}{\gamma}$ highest-valued elements. So S_t approximates this set by including the best $\lfloor t\frac{B}{\gamma}\rfloor$ up to time t.

Analysis. We will go over the proof idea for general B. This time, we do not have recursion, so we directly calculate the expected value of our algorithm's tentative selection and multiply by the probability that the selection is feasible. The full proof is extremely lengthy (spanning both papers [8] and [6]), so we state the key facts here for understanding how this differs from our other algorithms and leave the details to the papers.

For unconstrained budget $k = \frac{B}{\gamma}$, we want to obtain the competitive ratio $\frac{1}{2}\left(1 - O\left(\sqrt{\gamma}\right)\right)$. We divide the interval into N sub-intervals and iterate through rounds, so round l occurs at time t such that $l/N = t$, and we overload notation so $S_l = S_t$. We set $N >> n$ large enough such that the probability two items fall into the same interval is negligible and we can effectively assume that all items arrive at times which are multiples of $\frac{1}{N}$ because the value of $\lfloor t\frac{B}{\gamma}\rfloor$ stays constant in almost all intervals [8].

The idea is to end up with expected value at least

$$\sum_{k=2}^{\lfloor\frac{1}{\sqrt{\gamma}}\rfloor-1}\left(\frac{1}{2}-\frac{\sqrt{\gamma}}{4}-\frac{1}{4\sqrt{\gamma}N}\right)\left(1-9\sqrt{\frac{1}{\frac{l}{N}\frac{B}{\gamma}}}\right)\frac{l}{N}\frac{1-\frac{1}{2}\sqrt{\frac{\gamma}{B}}}{1+\gamma}OPT^* \tag{4.9}$$

$$\geq\left(\frac{1}{2}-\frac{7}{4}\sqrt{\gamma}-\frac{37}{4}\sqrt{\frac{\gamma}{B}}-\frac{\gamma}{2}-\frac{1}{4\sqrt{\gamma}N}\right)OPT^*, \tag{4.10}$$

from which we find the competitive ratio

$$\left(\frac{1}{2}-\frac{7}{4}\sqrt{\gamma}-\frac{37}{4}\sqrt{\frac{\gamma}{B}}-\frac{\gamma}{2}-\frac{1}{4\sqrt{\gamma}N}\right) \quad\longrightarrow\quad \frac{1}{2}-O(\sqrt{\gamma}) \tag{4.11}$$

as desired, since the square root of γ is asymptotically larger than γ. Now, where do these terms in expression 4.9 come from?

We first explain the expected value of the tentative selection, then we explain the probability that the selection is feasible.

The expression $\frac{l}{N}\frac{B}{\gamma}$ represents the approximate size of S_l at round l. The entire term

$$\left(1 - 9\sqrt{\frac{1}{\frac{l}{N}\frac{B}{\gamma}}}\right)\frac{l}{N}\frac{1-\frac{1}{2}\sqrt{\frac{\gamma}{B}}}{1+\gamma}OPT^* \tag{4.12}$$

comes from lemma 12 in [8] and serves to quantify how close we expect to be to $\frac{l}{N}\frac{B}{\gamma} \geq 1 - \frac{1}{2}\sqrt{\frac{\gamma}{B}}$. To derive expression 4.12, Kesselheim and Tönnis consider a more general online problem of packing with linear programs in a separate paper [6]. The idea is to maximize the profit obtained from a client's requests without violating resource constraints. Then results from [6] for the packing problem can be applied. One intuitive example of this is to view the secretaries' contracts as having some amount of "volume" γ, and we can imagine trying to pack as many valuable ones as possible into B unit bins as in the bin packing problem.

A further breakdown of the terms is instructive. We have the feasible proportion of the selection, $\left(1 - 9\sqrt{\frac{1}{\frac{l}{N}\frac{B}{\gamma}}}\right)$, times the fraction of items seen so far, $\frac{l}{N}$, times a factor which ensures full containment of items in the packing problem, $1 - \frac{1}{2}\sqrt{\frac{\gamma}{B}}$, times the size of the total optimal choices which we can pack, $\frac{1}{1+\gamma}|OPT^*|$, times the average value of an item from the optimal selection, $\frac{OPT^*}{|OPT^*|}$, for $l \geq 2\sqrt{\frac{\gamma}{B}}N$ so the math works out nicely.

Finally, the probability that the selection is feasible comes from lemma 3 of [8],

$$\left(\frac{1}{2} - \frac{\sqrt{\gamma}}{4} - \frac{1}{4\sqrt{\gamma}N}\right). \tag{4.13}$$

The probability is nearly $\frac{1}{2}$, which comes from the aforementioned fact that we can consider time-slices of length 2γ and make selections in the first half of the slice with certain feasibility, given that we do not select any candidates in the second half. This is precisely the reason that we recurse with new contract length $\gamma' = 2\gamma$ in the Charter algorithm, and it is also the reason that we obtain an asymptotic competitive ratio of $\frac{1}{2}$ as well! The lemma shows how to decouple random indicator variables—representing the choice of a candidate and that candidate's feasibility—to precisely quantify how nearly we approach $\frac{1}{2}$, as shown in expression 4.13.

Extension To Arbitrary Contract Durations. Kesselheim and Tönnis exhibit a number of extensions to the linear scaling algorithm. One of particular interest solves the temp secretary problem with competitive ratio at least $\frac{1}{4} - O(\sqrt{\gamma})$ for differing contract durations $\lambda_i \leq \gamma$ [8]. To define S_t for this extension, order the seen items by non-increasing metric $\frac{v_i}{\lambda_i}$ and take the minimal prefix such that the sum of element values in S_t is at least $\frac{tB}{2}$. We leave the analysis to [8]; essentially, this is due to the knapsack nature of the problem.

5. Concluding Remarks

There are very nice symmetries among the various secretary problems and the algorithms which solve them. In this paper, we have shown how the solution to the classic secretary problem is employed in the solutions to the multiple choice secretary problem are employed in the solutions to the temporary secretary problem.

In particular, we examined the classic algorithm to understand how it finds r and selects a threshold value. We showed how the fixed-threshold algorithm utilizes remarkably similar analysis in the multi-choice generalization, which is used numerically to obtain r, optimizing for the probability of selecting the best candidates. We discussed how Kleinberg's approach to the multi-choice problem optimizes for the expected value candidates provide, and it does so recursively to obtain an optimal 1-competitive algorithm as k goes to infinity. We demonstrated how Kleinberg's algorithm inspires the Charter algorithm for the temp secretary problem, specifically through the recursive structure which leads to a similar inductive proof method. We expounded on the motivations for the linear scaling algorithm, from threshold-sliding and time-slicing, as well as the similarities to the fixed-threshold algorithm. Through analysis of this simple yet powerful algorithm, we observed how the contract duration influences the recursive parameters in the Charter algorithm and limits the competitive ratio due to feasibility constraints. In addition to the algorithms' relationships, throughout the paper we elaborated on the similarities and differences between the problems themselves, for instance how the dependence on n no longer plays a vital role in the temp secretary problem as it does in the others.

There are a few areas of further research which would be of practical and theoretical interest to explore. For the multi-choice problem, a comparison of the expected value of selected secretaries from Kleinberg versus fixed-threshold would be very interesting. It is likely that Kleinberg is always better since it optimizes for expected value. However, for large n it may be that the running time for the simpler fixed-threshold algorithm is superior in practice and still yields acceptable results.

For the temp secretary problem, arrival times from a prior other than the uniform distribution would be very interesting to analyze since no work has been done for this particular setting without the uniformly i.i.d. assumption. It is already clear that the linear scaling algorithm obtains better results than the Charter algorithm for uniform arrival distributions, but perhaps they behave differently for other distributions or knapsack extensions, since the linear approach for scaling might become quite problematic, and the even interval divisions in Charter might require modification as well, though this would be easy to fix with knowledge of the prior.

6. Acknowledgments

We would like to thank Professors Karger and Madry for the lovely class in how to be optimally lazy. We would like to thank TA's Peebles and Wagner for their helpful office hours throughout the term. We would like to thank reviewers Bhargava and Zanger for their excellent comments, some of which we duplicated verbatim for the final paper. We thank all the giants on whose shoulders we comfortably recline.

References

[1] Thomas S. Ferguson. "Who Solved the Secretary Problem?" In: *Statistical Science* 4.3 (1989), pp. 282–289.

[2] Robert Kleinberg. "A Multiple-choice Secretary Algorithm with Applications to Online Auctions". In: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '05. Vancouver, British Columbia: Society for Industrial and Applied Mathematics, 2005, pp. 630–631.

[3] Moshe Babaioff et al. "Online auctions and generalized secretary problems". In: *SIGecom Exchanges* 7.2 (2008).

[4] Y. Girdhar and G. Dudek. "Optimal Online Data Sampling or How to Hire the Best Secretaries". In: *2009 Canadian Conference on Computer and Robot Vision*. May 2009, pp. 292–298.

[5] Mohammadhossein Bateni, Mohammadtaghi Hajiaghayi, and Morteza Zadimoghaddam. "Submodular Secretary Problem and Extensions". In: *ACM Trans. Algorithms* 9.4 (Oct. 2013), 32:1–32:23.

[6] Thomas Kesselheim et al. "Primal Beats Dual on Online Packing LPs in the Random-Order Model". In: *CoRR* (2013).

[7] Amos Fiat et al. "The Temp Secretary Problem". In: *Algorithms - ESA 2015: 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*. Ed. by Nikhil Bansal and Irene Finocchi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 631–642.

[8] Thomas Kesselheim and Andreas Tönnis. "Think Eternally: Improved Algorithms for the Temp Secretary Problem and Extensions". In: *CoRR* (2016).