Improved Competitive Ratios for Submodular Secretary Problems

(Extended Abstract)

Moran Feldman* Joseph (Seffi) Naor† Roy Schwartz‡

Abstract

The Classical Secretary Problem was introduced during the 60's of the 20^{th} century (nobody is sure exactly when). Since its introduction, many variants of the problem have been proposed and researched. In the classical secretary problem, and many of its variant, the input (which is a set of secretaries, or elements) arrives in a random order. In this paper we apply to the secretary problem a simple observation which states that the random order of the input can be generated by independently choosing a random continuous arrival time for each secretary. Surprisingly, this simple observation enables us to improve the competitive ratio of several known and studied variants of the secretary problem. In addition, in some cases the proofs we provide assuming random arrival times are shorter and simpler in comparison to existing proofs. The three variants of the secretary problem we consider are the following. First, we are allowed to hire any set of up to k secretaries, and the objective is to hire a set of secretaries maximizing a given monotone submodular function f. The second variant has the same objective, however, now we can hire a set of secretaries only if it is an independent set of a given partition matroid. The last variant also has the same objective, but this time we can hire any set of secretaries agreeing with a given knapsack constraint.

1 Introduction

In the (classical) secretary problem (CS), a set of n secretaries arrives in a random order for an interview. Each secretary is associated with a distinct non-negative value which is revealed upon arrival, and the objective of the interviewer is to choose the best secretary (the one having maximum value). The interviewer must decide after the interview whether to choose the candidate or not. This decision is irrevocable and cannot be altered later. The goal is to maximize the probability of choosing the best secretary.[1] It is known that the optimal algorithm for CS is to reject the first n/e secretaries, and then choose the first secretary that is better than any of the first n/e secretaries. This algorithm succeeds in finding the best secretary with probability of e^{-1} [10, 23].

Throughout the years, many variants of CS have been considered. In this work we focus on variants where a subset of the secretaries can be chosen and the goal is to maximize some function of this chosen subset. Such variants of CS have attracted much attention (see, *e.g.*, [3, 4, 5, 16, 18]). Some examples of these variants include: limiting the size of the subset to at most k, requiring the

*Computer Science Department, Technion, Haifa 32000, Israel. E-mail: *moranfe@cs.technion.ac.il*

†Computer Science Department, Technion, Haifa 32000, Israel. E-mail: *naor@cs.technion.ac.il*.

‡Computer Science Department, Technion, Haifa 32000, Israel. E-mail: *schwartz@cs.technion.ac.il*.

[1] Note that the above definition is slightly different from the standard definition of CS, though both definitions are equivalent. In the standard definition of the problem the secretaries have ranks instead of values, and the relative rank of each secretary is revealed upon arrival.

subset of chosen secretaries to form an independent set in a given matroid and requiring the chosen secretaries to satisfy some knapsack constraints. All these variants have been studied with both linear and submodular objective functions. More details on previous results can be found in Section 1.2.

In this work we use a simple observation which has been employed in problems other than CS. The observation states that the random order in which the secretaries arrive can be modeled by assigning each secretary an independent uniform random variable in the range $[0, 1)$. This continuous random variable determines the time in which the secretary arrives. Obviously, this modeling is equivalent to a random arrival order.[2] Though this modeling of the arrival times as continuous random variables is very simple, it has several advantages when applied to variants of the secretary problem.

First, this modeling of arrival times enables us to achieve better competitive ratios for several variants of CS. Second, in this modeling the proofs of the performance guarantees of the algorithms is much simpler. This can be seen by the following example.

A difficulty encountered while designing algorithms for variants of the secretary problem is that the position of a secretary in the random arrival order depends on the positions of all other secretaries. For example, if a set S contains many secretaries that have arrived early, then a secretary outside of S is likely to arrive late, since many of the early positions are already taken by members of S. This difficulty complicates both the algorithms and their analysis. To get around this dependence [3, 18], for example, partition the secretaries into two sets: one containing the first m secretaries and the other containing the last $n - m$ secretaries. The value of m is binomially distributed $\mathsf{Bin}(n, 1/2)$. It can be shown that this partition, together with the randomness of the input, guarantees that every secretary is uniformly and independently assigned to one of the two sets. The reason that such an elaborate argument is needed to create a set containing each secretary independently, with probability $1/2$, is the dependencies between positions of secretaries in the arrival order.

In contrast, using the modeling of the arrival times as continuous random variables, creating a subset of secretaries where each secretary independently belongs to it with probability $1/2$ is simple. One just has to choose all secretaries that arrive before time $t = 1/2$. This simplifies the above argument, designed for a random arrival order, considerably. This simple example shows how continuous arrival times can be used to simplify both the algorithm and its analysis. In the variants of CS considered in this paper, this simplification enables us to obtain improved competitive ratios.

For some variants, the algorithms presented in this paper can be viewed as the continuous "counterparts" of known algorithms (see the uniform matroid and knapsack variants), but this is not always the case. For the partition matroid variant, the algorithms we define in this work using continuous arrival times use different techniques than previous known algorithms for this case.

It is important to note that the continuous time "counterparts" of algorithms designed using a random arrival order are *not* equivalent to the original algorithms. For example, the optimal algorithm for CS assuming continuous random arrival times is to inspect secretaries up to time e^{-1}, and hire the first secretary after this time which is better than any previously seen secretary (see Appendix A for an analysis of this algorithm using continuous arrival times). Observe that this algorithm inspects n/e secretaries in *expectation*, while the classical algorithm inspects that number of secretaries *exactly*. This subtle difference is what enables us to improve and simplify previous results.

Formally, all problems considered in this paper are online problems in which the input consists of a set of secretaries (elements) arriving in a random order. Consider an algorithm $\mathcal{A}$ for such a problem, and denote by opt an optimal offline algorithm for the problem. Let I be an instance of the problem, and let $\mathcal{A}(I)$ and $\mathsf{opt}(I)$ be the values of the outputs of $\mathcal{A}$ and opt, respectively,

[2] Given a random arrival order (without times), we can sample n independent uniformly random arrival times, sort them and assign them sequentially to the secretaries upon arrival.

given I. We say that A is α-competitive (or has an α-competitive ratio) if $\inf_I \frac{\mathbb{E}[\mathcal{A}(I)]}{\mathsf{opt}(I)} \geq \alpha$, where the expectation is over the random arrival order of the secretaries of I and the randomness of $\mathcal{A}$ (unless $\mathcal{A}$ is deterministic). The competitive ratio is a standard measure for the quality of an online algorithm.

1.1 Our Results

In this paper we consider variants of CS where the objective function is normalized, monotone and submodular.[3] There are three variants for which we provide improved competitive ratios. The first is the *submodular partition matroid secretary* problem (SPMS) in which the secretaries are partitioned into subsets, and at most one secretary from each subset can be chosen. The second is the *submodular cardinality secretary* problem (SCS) in which up to k secretaries can be chosen. The third and last variant is the *submodular knapsack secretary* problem (SKS), in which each secretary also has a cost (which is revealed upon arrival), and any subset of secretaries is feasible as long as the total cost of the subset does not exceed a given budget.

For SPMS we present a competitive ratio of $(1 - \ln 2)/2 \approx 0.153$, which improves on the current best result of $O(1)$ by Gupta et al. [16]. We note that the exact competitive ratio given by [16] is not stated explicitly, however, by inspecting their result carefully it seems that the competitive ratio they achieve is at most $71/1280000$. We note that for SPMS the algorithm we provide is not a continuous time "counterpart" of the algorithm of [16]. This demonstrates the fact that modeling the arrival times as continuous random variables helps in designing and analyzing algorithms for submodular variants of CS.

For SCS we present a competitive ratio of $(e - 1)/(e^2 + e) \approx 0.170$, and the current best result for this problem is due to Bateni et al. [5] who provided a $(1 - e^{-1})/7 \approx 0.0903$ competitive ratio. There are two points to notice when comparing our result and that of [5]. First, [5] were not careful when calculating their exact competitive ratio. In fact, their algorithms provide better ratios then stated. However, in this paper we still obtain improved competitive ratios compared to their true competitive ratios, though the improvement is smaller than stated above. Second, the algorithm presented in this paper for SCS can be seen as a continuous time "counterpart" of their algorithm. However, our analysis is simpler than the analysis presented in [5] which enables us to provide improved competitive ratios.

For SKS we provide a competitive ratio of $(20e)^{-1} \approx 0.0184$. The current best result is due to Bateni et al. [5] which provide a ratio of $O(1)$. The exact competitive ratio is not stated in [5], but careful inspection of their algorithm shows that it is at most $96^{-1} \approx 0.0104$. Notice that the best known competitive ratio for the *linear* version of SKS is only $10e^{-1} \approx 0.0368$ [3]. As before, the algorithm presented in this paper for SKS can be seen as a continuous time "counterpart" of [5]'s algorithm. However, our analysis is simpler than the analysis presented in [5], enabling us to provide improved competitive ratios. Table 1 summarizes the above results.

1.2 Related Work

Many variants of CS have been considered throughout the years and we shall mention here only those most relevant to this work. Babaioff et al. [4] considered the case where the chosen subset of secretaries needs to be an independent set of a given matroid. They provided a competitive ratio of $O(\log r)$ for this problem, where r is the rank of the matroid. For several specific matroids, better constant competitive ratios are known [4, 9, 17, 19]. The special variant of SCS where the objective

[3]Given a groundset $\mathcal{S}$, a function $f : 2^{\mathcal{S}} \to \mathbb{R}$ is called *submodular* if for every $A, B \subseteq \mathcal{S}$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. Additionally, f is called *normalized* if $f(\varnothing) = 0$ and *monotone* if for every $A \subseteq B \subseteq \mathcal{S}$, $f(A) \leq f(B)$.

Table 1: Comparison of our results with the known results for the monotone submodular and linear variants of the problems we consider.

Problem	Our Result	Previous Result	Best Result for Linear Variant
SPMS	0.153	0.0000555 [16]	0.368[1]
SCS	0.170	0.0903 [5]	0.368 [3]
SKS	0.0184	0.0104 [5]	0.0368 [3]

[1] For linear objective functions one can apply the algorithm for the classical secretary problem to each subset of secretaries independently.

function f is linear has also been studied. Two incomparable competitive ratios were obtained by Babaioff et al. [3] and Kleinberg [18], achieving competitive ratios of e^{-1} and $1 - O(1/\sqrt{k})$, respectively. An interesting variant of SCS with a linear objective function gives each of the k possible slots of secretaries a different weight. The value of the objective function in this case is the sum of the products of a slots' weights with the values of the secretaries assigned to them. Babaioff et al. [2] provide a competitive ratio of 1/4 for this special variant. Additional variants of CS can found in [1, 6, 12, 13, 15].

Another rich field of study is that of submodular optimization, namely optimization problems in which the objective function is submodular. In recent years many new results in this field have been achieved. The most basic problem, in this field, is that of unconstrained maximization of a nonmonotone submodular function [11, 14]. Other works consider the maximization of a nonmonotone submodular function under various combinatorial constraints [16, 24, 21]. The maximization of a monotone submodular function under various combinatorial constraints (such as a matroid, the intersection of several matroids and knapsack constraints) has also been widely studied [7, 8, 20, 22, 24].

Thus, it comes as no surprise that recent works have combined the secretary problem with submodular optimization. Gupta et al. [16] were the first to consider this combination. For the variant where the goal is to maximize a submodular function of the chosen subset of secretaries under the constraint that this subset is independent in a given matroid, Gupta et al. [16] provide a competitive ratio of $O(\log r)$ (r is the rank of the matroid). If the constraint is that the chosen subset of secretaries belongs to the intersection of ℓ matroids, Bateni et al. [5] provide a competitive ratio of $O(\ell \log^2 r)$. The special case of a partition matroid is exactly SPMS, and the special case of a uniform matroid is exactly SCS. As mentioned before, for SPMS Gupta et al. [16] provide a competitive ratio of $O(1)$ for SPMS which is at most 71/1280000 (the exact constant is not explicitly stated in their work). They also get a similar competitive ratio for a variant of SPMS with a non-monotone submodular objective function. For SCS, Gupta et al. [16] provide a competitive ratio of $O(1)$ which is at most 1/1417 (again, the exact constant is not explicitly stated in their work). This was improved by Bateni et al. [5] who provided a competitive ratio of $(1 - e^{-1})/7 \approx 0.0903$ for this problem, a $e^{-2}/8 \approx 0.0169$-competitive algorithm for a variant of SCS with a non-monotone submodular objective function. For SKS the current best result is due to Bateni et al. [5] who provide a competitive ratio of at most $96^{-1} \approx 0.0104$ (again, the exact constant is not explicitly stated in their work). Another extension considered by [5] is a generalization of SKS where every secretary has a ℓ dimensional cost, and the total cost of the secretaries in each dimension should not exceed the budget given for this dimension (*i.e.*, the hired secretaries should obey ℓ knapsack constraints). For this problem [5] gives an $O(\ell)$ competitive algorithm.

Organization. Section 2 contains formal definitions and several technical lemmata. Sections 3, 4 and 5 provide the improved competitive ratio for SPMS, SCS and SKS, respectively.

2 Preliminaries

An instance of a constrained secretary problem consists of three components $\mathcal{S}, \mathcal{I}$ and f.

- $\mathcal{S}$ is a set of n secretaries arriving in a random order.[4]
- $\mathcal{I} \subseteq 2^{\mathcal{S}}$ is a collection of independent sets of secretaries. The sets in $\mathcal{I}$ are known in advance in some settings (*e.g.*, SCS), and are revealed over time in other settings (*e.g.*, SKS). However, at any given time, we know all independent sets containing only secretaries that already arrived.
- $f : 2^{\mathcal{S}} \to \mathbb{R}$ is a function over the set of secretaries accessed using an oracle which given a set S of secretaries that have already arrived, returns $f(S)$. Specific problems restrict f to be of specific types: linear, monotone submodular, etc.

The goal is to maintain an independent set R of secretaries, and maximize the final value of $f(R)$ (*i.e.*, its value after all secretaries have arrived). Upon arrival of a secretary s, the algorithm has to either add it to R (assuming $R \cup \{s\} \in \mathcal{I}$), or reject it. Either way, the decision is irrevocable.

Given a submodular function $f : \mathcal{S} \to \mathbb{R}^+$, the discrete derivative of f with respect to s is $f_s(R) = f(R \cup \{s\}) - f(R)$. We use this shorthand throughout the paper.

Most algorithms for secretary problems with linear objective functions require every secretary to have a distinct value. This requirement does not make sense for submodular objective functions, and therefore, we work around it by introducing a total order over the secretaries, which is a standard practice (see, *e.g.*, [9]). Formally, we assume the existence of an arbitrary fixed order Z over the secretaries. If such an order does not exist, it can be mimicked by starting with an empty ordering, and placing every secretary at a random place in this ordering upon arrival. The resulting order is independent of the arrival order of the secretaries, and therefore, can be used instead of a fixed order. Let s_1, s_2 be two secretaries, and let S be a set of secretaries. Using order Z we define $s_1 \succ_S s_2$ to denote "$f_{s_1}(S) > f_{s_2}(S)$, or $f_{s_1}(S) = f_{s_2}(S)$ and s_1 precedes s_2 in Z". Notice that $\succ_S$ is defined using f and Z. Whenever we use $\succ_S$, we assume f is understood from context and Z is the order defined above.

Remark: The probability that two secretaries arrive at the same time is 0, thus we ignore this.

3 $(1 - \ln 2)/2 \approx 0.153$-Competitive Algorithm for SPMS

The Submodular Partition Matroid Secretary Problem (SPMS) is a secretary problem with a normalized monotone and submodular objective function f. The set $\mathcal{I}$ of independent sets is $G_1 \times \ldots \times G_k$ (where the G_i's are a partition of $\mathcal{S}$). For every secretary s, the index of the set G_i containing s is revealed when s arrives.

When designing an algorithm for SPMS, we want to select the best secretary from every set G_i. If f was linear, we could apply the algorithm for the classical secretary problem to each G_i separately. The following algorithm is based on a similar idea.

SPMS Algorithm(f, k)**:**

1. Initialize $R \leftarrow \varnothing$.
2. Observe the secretaries arriving till time t.[a]
3. After time t, for every secretary s arriving, let G_i be the set of s. Accept s into R if:
   (a) no previous secretary of G_i was accepted,
   (b) and for every previously seen $s' \in G_i$, $s' \prec_R s$.
4. Return R.

[a] t is a constant to be determined later.

[4] If the input is given as a random permutation, the size n of $\mathcal{S}$ is assumed to be known. Note that in order to generate the random arrival times of the secretaries and assign them upon arrival, n has to be known in advance. On the other hand, if the arrival times are part of the input, the algorithms in this work need not know n.

In this subsection we prove the following theorem.

Theorem 3.1. *The above algorithm is a* $(1 - \ln 2)/2 \approx 0.153$*-competitive algorithm for SPMS.*

The algorithm clearly maintains R as a feasible set of secretaries, hence, we only need to show that, in expectation, it finds a good set of secretaries.

Observation 3.2. *We can assume there is at least one secretary in every set* G_i*.*

Proof. The behavior of the algorithm is not effect by adding or removing empty sets G_i. □

3.1 Analysis of a single G_i

In this subsection we focus on a single G_i. Let E_i be an event determining the arrival times of all secretaries in $\mathcal{S} - G_i$, we assume throughout this subsection that some fixed event E_i occurred. Let R_x be the set of secretaries from $\mathcal{S} - G_i$ collected by the algorithm up to time x assuming no secretary of G_i arrives (observe that R_x is not random because we fixed E_i). We define $\hat{s}_x$ as the maximum secretary in G_i with respect to $\succ_{R_x}$. The analysis requires two additional event types: A_x is the event that $\hat{s}_x$ arrives at time x, and B_x is the same event with the additional requirement that the algorithm collected $\hat{s}_x$.

Lemma 3.3. $\Pr[B_x | A_x] \geq 1 - \ln x + \ln t$*.*

Proof. Event A_x states that $\hat{s}_x$ arrived at time x. If no other secretary of G_i is collected till time x, $\hat{s}_x$ is collected by the definition of the algorithm. Hence, it is enough to bound the probability that no secretary of G_i is collected till time x, given A_x.

Observe that R_x takes at most $k - 1$ values for $x \in [0, 1)$. Hence, the range $[0, 1)$ can be divided into k intervals $\mathcal{I}_1, \ldots, \mathcal{I}_k$ such that the set R_x is identical for all times within one interval. Divide the range $[0, x)$ into small steps of size Δy such that Δy divides t and x, and every step is entirely included in an interval (this is guaranteed to happen if Δy also divides the start time and the length of every interval $\mathcal{I}_i$). Since each step is entirely included in a single interval, for every time x in step j, $R_x = R_{(j-1)\cdot\Delta y}$.

A secretary cannot be collected in step j if $j \cdot \Delta y \leq t$. If this is not the case, a secretary is collected in step j if the maximum secretary of G_i in the range $[0, j \cdot \Delta y)$ with respect to $\succ_{R_{(j-1)\cdot\Delta y}}$ arrives at time $(j - 1) \cdot \Delta y$ or later. The probability that this happens is $\Delta y/(j \cdot \Delta y) = j^{-1}$. We can now use the union bound to upper bound the probability that any secretary is accepted in any of the steps before time x:

$$\Pr[B_x | A_x] \geq 1 - \sum_{j=t/\Delta y+1}^{x/\Delta y} j^{-1} \geq 1 - \int_{t/\Delta y}^{x/\Delta y} \frac{dj}{j} = 1 - [\ln j]_{t/\Delta y}^{x/\Delta y} = 1 - \ln x + \ln t. \qquad \square$$

Let s_i^* denote the single secretary of $G_i \cap OPT$, and let a_i be the secretary of G_i collected by the algorithm. If no secretary is collected from G_i, assume a_i is a dummy secretary of value 0 (*i.e.*, f is oblivious to the existence of this dummy secretary in a set). We also define R_i to be the set R immediately before the algorithm collects a_i (if the algorithms collects no secretary of G_i, R_i is an arbitrary set).

Observation 3.4. *If* B_x *occurs for some* x*,* $f_{a_i}(R_i) \geq f_{s_i^*}(R)$*, where* R *is the set returned by the algorithm.*

Proof.
$$f_{a_i}(R_i) \overset{(1)}{=} f_{a_i}(R_x) \overset{(2)}{\geq} f_{s_i^*}(R_x) \overset{(3)}{\geq} f_{s_i^*}(R).$$
Where (1) and (2) follow from the fact that B_x occurred, and therefore, a_i was collected at time x and $a_i = \hat{s}_x$. Also, B_x implies $R_x \subseteq R$, hence, the submodularity of f implies (3). □

Let $B_i = \cup_{x \in (t,1)} B_x$, and let P_i be the set $\{s_i^*\}$ if B_i occurred, and $\varnothing$, otherwise.

Corollary 3.5. $f_{a_i}(R_i) \geq f(R \cup P_i) - f(R)$.

Proof. If $P_i = \varnothing$, the claim follows because $f_{a_i}(R_i)$ is nonnegative by the monotonicity of f. If $P_i = \{s_i^*\}$, we know that B_i occurred, and therefore, there must be an x for which B_x occurred also. The corollary now follows immediately from Observation 3.4. □

Lemma 3.6. $\Pr[B_i] \geq 2 + \ln t - 2t$.

Proof. Observe that the events B_x are disjoint, hence, $\Pr[B_i]$ is the sum of the probabilities of the events B_x. Since B_x implies A_x, $\Pr[B_x] = \Pr[B_x | A_x] \cdot \Pr[A_x]$. The event A_x requires that the maximum secretary with respect to $\succ_{R_x}$ arrives in time x. The probability that this secretary arrives in an interval of size Δx is Δx. Hence, the probability that it arrives in an infinitesimal interval of size dx is $\Pr[A_x] = dx$. Therefore,
$$\Pr[B_i] = \int_t^1 \Pr[B_x | A_x] dx \geq \int_t^1 (1 - \ln x + \ln t) dx = [x - x(\ln x - 1) + x \ln t]_t^1 = [2 + \ln t] - 2t. \quad \square$$

The last bound on $\Pr[B_i]$ is maximized when $t = 0.5$, thus, we assume $t = 0.5$ from now on.

3.2 Analysis of the Entire Output

Throughout the previous subsection we assumed some fixed event E_i occurred. Hence, Corollary 3.5 and Lemma 3.6 were proven given this assumption, however, they are also true without it.

Lemma 3.7. *Corollary 3.5 and Lemma 3.6 also hold without fixing an event E_i.*

Proof. Corollary 3.5 states that for every fixed E_i, if B_i occurs then $f_{a_i}(R_i) \geq f(R \cup P_i) - f(R)$. Since some event E_i must occur (the secretaries of $\mathcal{S} - G_i$ must arrive at some times), this is also true without fixing some E_i.

Let us rephrase Lemma 3.6 to explicitly present the assumption that some fixed E_i occurred: $\Pr[B_i | E_i] \geq 1 - \ln 2$ (recall that we chose $t = 0.5$). Therefore,
$$\Pr[B_i] = \sum_{E_i} \Pr[E_i] \cdot \Pr[B_i | E_i] \geq (1 - \ln 2) \cdot \sum_{E_i} \Pr[E_i] = 1 - \ln 2. \quad \square$$

Let $P = \cup_{i=1}^k P_i$, and notice that $P \subseteq OPT$. The following lemma lower bounds $f(P)$.

Lemma 3.8. $\mathbb{E}[f(P)] \geq (1 - \ln 2) f(OPT)$.

Proof. Let X_i be an indicator for the event B_i. Then,
$$\begin{aligned}
\mathbb{E}[f(P)] &= \mathbb{E}\left[\sum_{i=1}^k X_i \cdot f_{s_i^*}\left(\cup_{j=1}^{i-1} P_j\right)\right] = \sum_{i=1}^k \Pr[X_i = 1] \cdot \mathbb{E}\left[f_{s_i^*}\left(\cup_{j=1}^{i-1} P_j\right) | X_i = 1\right] \\
&\geq \sum_{i=1}^k \Pr[B_i] \cdot f_{s_i^*}\left(\{s_j^* | j < i\}\right) \geq (1 - \ln 2) \cdot \sum_{i=1}^k f_{s_i^*}\left(\{s_j^* | j < i\}\right) \\
&= (1 - \ln 2) \cdot f(OPT).
\end{aligned}$$
Where the last inequality follows from Lemma 3.6 and the value we chose for t. □

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Observe the following.

$$\mathbb{E}[f(R)] = \mathbb{E}\left[\sum_{i=1}^{k} f_{a_i}(R_i)\right] \geq \mathbb{E}\left[\sum_{i=1}^{k} f(R_i \cup P_i) - f(R_i)\right] \geq \mathbb{E}[f(R \cup P)] - \mathbb{E}[f(R)].$$

Rearranging terms, we get: $\mathbb{E}[f(R)] \geq \frac{f(R\cup P)}{2} \geq \frac{f(P)}{2} \geq \frac{1-\ln 2}{2} \cdot f(OPT)$. □

4 The Submodular Cardinality Secretary Problem

The Submodular Cardinality Secretary Problem (SCS) is a secretary problem in which the objective function f is a normalized monotone submodular function, and we are allowed to hire up to k secretaries (in other words, the set $\mathcal{I}$ of independent sets contains every set of up to k secretaries).

The idea behind the following algorithm is to divide the time range $[0, 1)$ into k intervals of equal length $I_1, \ldots, I_k$. Each interval contains one secretary of OPT in expectation. A natural way to look for this secretary is using the algorithm for the classical secretary problem.

SCS Algorithm(f):

1. Initialize $R \leftarrow \varnothing$.
2. For every $i = 0$ to $k - 1$ do:
   (a) Observe the secretaries arriving within the time range $\left[\frac{i}{k}, \frac{i+e^{-1}}{k}\right)$.
   (b) Let $\hat{s}$ be the maximum secretary in the above range with respect to $\succ_R$.[a]
   (c) During the time range $\left[\frac{i+e^{-1}}{k}, \frac{i+1}{k}\right)$, add to R the first secretary s' such that $s' \succ_R \hat{s}$.
3. Return R.

[a] If no secretary arrives during the above range, we assume $s \succ_R \hat{s}$ for every secretary s.

In this section we prove the following theorem.

Theorem 4.1. *There above algorithm is a* $(e-1)/(e^2+e) \approx 0.170$*-competitive algorithm for SCS.*

Let $k' \leq k$ be the number of secretaries in OPT. We select random representative secretaries for the intervals. Let S_I be the set of secretaries of OPT that ended up in interval I, then the representative secretary of interval I is randomly and uniformly selected from S_I. If $S_I = \varnothing$, then no representative secretary is selected for interval I. Let $s_1^*, \ldots, s_{k'}^*$ be the k' secretaries of OPT, and let X_i be an indicator for the event that s_i^* was selected as the representative of some interval (regardless of which interval it was). Define P to be the set of representative secretaries, *i.e.*, $P = \{s_i^* \in OPT | X_i = 1\}$. The following two lemmata show that P has a large expected value.

Lemma 4.2. *For every* $1 \leq i \leq k'$, $\mathbb{E}[X_i] \geq 1 - e^{-1}$.

Proof. Due to symmetry, the variables X_i are identically distributed for every $1 \leq i \leq k'$. Let us denote by p their probability of getting the value 1. By the linearity of the expectation, $\mathbb{E}\left[\sum_{i=1}^{k'} X_i\right] = k'p$. However, $\sum_{i=1}^{k'} X_i$ is also the number of intervals that got any representative, hence, it is equal to k minus the number of intervals that have no representative.

Consider some interval I. The probability that no secretary of OPT arrived during I (and therefore, I has no representative) is $(1-1/k)^{k'} \leq e^{-k'/k}$. Hence, the expected number of intervals with no representative is at most $ke^{-k'/k}$. Combining this with the earlier observations, we get:

$$k'p = \mathbb{E}\left[\sum_{i=1}^{k'} X_i\right] \geq k - ke^{-k'/k} \Rightarrow p \geq \frac{k}{k'}(1 - e^{-k'/k}) \geq 1 - e^{-1}.$$

Where the last inequality follows since $(1-e^{-x})/x$ is a decreasing function, and its value for $x=1$ is $1-e^{-1}$. □

Lemma 4.3. $\mathbb{E}[f(P)] \geq (1-e^{-1}) \cdot f(OPT)$.

Proof.

$$\begin{aligned}\mathbb{E}[f(P)] &= \mathbb{E}\left[\sum_{i=1}^{k'} X_i \cdot f_{s_i^*}\left(\{s_j^* | j < i, X_j = 1\}\right)\right] = \sum_{i=1}^{k'} \mathbb{E}\left[X_i \cdot f_{s_i^*}\left(\{s_j^* | j < i, X_j = 1\}\right)\right] \\ &= \sum_{i=1}^{k'} \Pr[X_i = 1] \cdot \mathbb{E}\left[f_{s_i^*}\left(\{s_j^* | j < i, X_j = 1\}\right) | X_i = 1\right] \\ &\geq \sum_{i=1}^{k'} \Pr[X_i = 1] \cdot f_{s_i^*}\left(\{s_j^* | j < i\}\right) = (1-e^{-1}) \cdot f(OPT).\end{aligned}$$ □

Let E_i be an event determining the set of secretaries arriving during each interval, and the arrival times of all secretaries in intervals other than the one in which s_i^* arrives. We say that event E_i is *consistent* with a set $A \subseteq OPT$ of secretaries (and denote it by $E_i \bowtie A$), if $\Pr[P = A | E_i] > 0$.

Let a_i be the secretary that the algorithm selects from the interval in which s_i^* arrived. If no secretary is selected from this interval, assume a_i is a dummy secretary of value 0 (*i.e.*, f is oblivious to the existence of this dummy secretary in a set). Also, let R_i represent the value of R before the algorithm considers the interval in which s_i^* falls. The following lemma shows that with good probability the secretary selected from the interval of s_i^* is at least as good as s_i^*.

Lemma 4.4. *Given an event E_i,* $\Pr[f_{a_i}(R_i) \geq f_{s_i^*}(R_i) | E_i] \geq e^{-1}$.

Proof. Denote the interval in which s_i^* arrives by $I = \left[\frac{i}{k}, \frac{i+1}{k}\right)$, and let S' be the fixed set of secretaries that arrive in this interval. Notice, the arrival time of every secretary in S' is uniformly distributed along I. Let $\tilde{s}$ be the maximum secretary with respect to $\succ_{R_i}$ among the secretaries of S'. Clearly, $f_{\tilde{s}}(R_i) \geq f_{s_i^*}(R_i)$, hence, it is enough to show that the algorithm selects $\tilde{s}$ with probability of at least e^{-1}.

Let x be the time in which secretary $\tilde{s}$ arrived. The probability of x having some value from a range of length ℓ is $k \cdot \ell$, therefore, the probability x have a value from an infinitesimal range of size dx is $k \cdot dx$. Assuming $x \in \left[\frac{i+e^{-1}}{k}, \frac{i+1}{k}\right)$, $\tilde{s}$ is selected if one of the following holds:

- There are no secretaries in the range $\left[\frac{i}{k}, x\right)$.
- The maximum secretary s with respect to $\succ_{R_i}$ in the range $\left[\frac{i}{k}, x\right)$ appears before time $\frac{i+e^{-1}}{k}$.

With probability at least $\frac{(i+e^{-1})/k - i/k}{x - i/k} = \frac{e^{-1}}{kx-i}$ one of these conditions occurs. Hence, by the law of total probability, the probability that the algorithm selects secretary $\tilde{s}$ is at least:

$$\int_{\frac{i+e^{-1}}{k}}^{\frac{i+1}{k}} \frac{e^{-1}}{kx-i} \cdot k \cdot dx = ke^{-1} \cdot \left[\frac{\ln(kx-i)}{k}\right]_{\frac{i+e^{-1}}{k}}^{\frac{i+1}{k}} = e^{-1} \cdot \left[\ln 1 - \ln e^{-1}\right] = e^{-1}.$$ □

Corollary 4.5. *Given a set $A \subseteq OPT$, for every $s_i^* \in A$,*

$$\mathbb{E}[f_{a_i}(R_i) | P = A] \geq e^{-1} \cdot \mathbb{E}[f_{s_i^*}(R_i) | P = A].$$

Proof. Let E_i be an event consistent with A. By Lemma 4.4, $\Pr[f_{a_i}(R_i) \geq f_{s_i^*}(R_i) | E_i] \geq e^{-1}$. Observe that the distribution of the arrival times of the secretaries is the same given $E_i \cap \{P = A\}$ or E_i, hence, we also get $\Pr[f_{a_i}(R_i) \geq f_{s_i^*}(R_i) | E_i \cap \{P = A\}] \geq e^{-1}$. $f_{a_i}(R_i)$ is nonnegative, and $f_{s_i^*}(R_i)$

is constant given E_i (or $E_i \cap \{P = A\}$), therefore, the last inequality implies $\mathbb{E}[f_{a_i}(R_i)|E_i \cap \{P = A\}] \geq e^{-1} \cdot \mathbb{E}[f_{s_i^*}(R_i)|E_i \cap \{P = A\}]$. Using the law of total probability and Lemma 4.2, we get:

$$\begin{aligned}\mathbb{E}[f_{a_i}(R_i)|P = A] &= \sum_{E_i \bowtie P} \Pr[E_i|P = A] \cdot \mathbb{E}[f_{a_i}(R_i)|E_i \cap \{P = A\}] \\ &\geq e^{-1} \cdot \sum_{E_i \bowtie P} \Pr[E_i|P = A] \cdot \mathbb{E}[f_{s_i^*}(R_i)|E_i \cap \{P = A\}] \\ &= e^{-1} \cdot \mathbb{E}[f_{s_i^*}(S_i)|P = A]. \end{aligned}$$ □

We can now lower bound $f(R)$ (the final solution returned by the algorithm) given $P = A$.

Lemma 4.6. *For every set $A \subseteq OPT$,*

$$\mathbb{E}[f(R)|P = A] \geq \mathbb{E}[f(R \cup P)|P = A]/(e + 1).$$

Proof. Notice that a_i is identical for all secretaries $s_i^* \in OPT$ that arrived during a single interval, however, at most one of these secretaries is in P. Therefore, the value gained by the algorithm is $\sum_{i \in P} f_{a_i}(R_i)$. Using Corollary 4.5, we get:

$$\begin{aligned}\mathbb{E}[f(R)|P = A] = \mathbb{E}\left[\sum_{i \in P} f_{a_i}(R_i)|P = A\right] &\geq \sum_{i \in A} e^{-1} \cdot \mathbb{E}\left[f_{s_i^*}(R_i)|P = A\right] \\ \geq e^{-1} \cdot \mathbb{E}\left[\sum_{i \in P} f_{s_i^*}(R)|P = A\right] &\geq e^{-1} \cdot \mathbb{E}\left[f(R \cup P) - f(R)|P = A\right].\end{aligned}$$

And after rearranging terms,

$$(1 + e^{-1}) \cdot \mathbb{E}[f(R)|P = A] \geq e^{-1} \cdot \mathbb{E}[f(R \cup P)|P = A] \Rightarrow \mathbb{E}[f(R)|P = A] \geq \frac{\mathbb{E}[f(R \cup P)|P = A]}{e + 1}.$$ □

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Recall that by Lemma 4.3, $\mathbb{E}[f(P)] \geq (1 - e^{-1}) \cdot f(OPT)$. Therefore,

$$\begin{aligned}\mathbb{E}[f(R)] &= \sum_{A \subseteq OPT} \Pr[P = A] \cdot \mathbb{E}[f(R)|P = A] \geq \sum_{A \subseteq OPT} \Pr[P = A] \cdot \frac{\mathbb{E}[f(R \cup P)|P = A]}{e + 1} \\ &= \frac{\mathbb{E}[f(R \cup P)]}{e + 1} \geq \frac{\mathbb{E}[f(P)]}{e + 1} \geq \frac{e - 1}{e^2 + e}.\end{aligned}$$ □

5 The Submodular Knapsack Secretary Problem

The Submodular Knapsack Secretary Problem (SKS) is a secretary problem in which the objective function f is a normalized monotone submodular function and every secretary s has a cost $c(s)$ (revealed upon arrival). A budget B is also given as part of the input, and the algorithm is allowed to hire secretaries as long as it does not exceed the budget. In other words, the collection $\mathcal{I}$ of allowed sets contains every set of secretaries whose total cost is at most B.

Theorem 5.1. *There is a $1/(20e)$-competitive algorithm for SKS.*

Due to space limitations, the proof of Theorem 5.1 appears in Appendix B.

References

[1] Miklos Ajtai, Nimrod Megiddo, and Orli Waarts. Improved algorithms and analysis for secretary problems and generalizations. *SIAM J. Discrete Math*, 14:1–27, 1995.

[2] Moshe Babaioff, Michael Dinitz, Anupam Gupta, Nicole Immorlica, and Kunal Talwar. Secretary problems: Weights and discounts. In *SODA*, pages 1245–1254, 2009.

[3] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In Moses Charikar, Klaus Jansen, Omer Reingold, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Lecture Notes in Computer Science, pages 16–28. Springer Berlin / Heidelberg, 2007.

[4] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA*, pages 434–443, 2007.

[5] Mohammad Hossein Bateni, Mohammad Taghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. In Maria Serna, Ronen Shaltiel, Klaus Jansen, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Lecture Notes in Computer Science, pages 39–52. Springer Berlin / Heidelberg, 2010.

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

[7] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrak. Maximizing a monotone submodular function subject to a matroid constraint. To appear in SIAM Journal on Computing.

[8] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *to appear in STOC '11*, 2011.

[9] Nedialko B. Dimitrov and C. Greg Plaxton. Competitive weighted matching in transversal matroids. In Luca Aceto, Ivan Damgărd, Leslie Goldberg, Magnús Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 397–408. Springer Berlin / Heidelberg, 2008.

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

[11] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. In *FOCS*, pages 461–471, 2007.

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

[13] P. R. Freeman. The secretary problem and its extensions: A review. *International Statistical Review*, 51(2):189–206, 1983.

[14] Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. To appear in SODA 2011.

[15] John Gilbert and Frederick Mosteller. Recognizing the maximum of a sequence. In Stephen Fienberg and David Hoaglin, editors, *Selected Papers of Frederick Mosteller*, Springer Series in Statistics, pages 355–398. Springer New York, 2006.

[16] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrainted non-monotone submodular maximization: Offline and secretary algorithms, 2010. To appear in WINE 2010.

[17] Sungjin Im and Yajun Wang. Secretary problems: Laminar matroid and interval scheduling. In *SODA*, 2011.

[18] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA*, pages 630–631, 2005.

[19] Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *ICALP*, pages 508–520, 2009.

[20] Ariel Kulik, Hadas Shachnai, and Tami Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *SODA*, pages 545–554, 2009.

[21] Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Maximizing non-monotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, (4):2053–2078, 2010.

[22] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. In *APPROX*, pages 244–257, 2009.

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

[24] Jan Vondrák. Symmetry and approximability of submodular maximization problems. In *FOCS*, pages 651–670, 2009.

A Example - the Classical Secretary Problem

The Classical Secretary Problem (CS) is a secretary problem with the set $\mathcal{I}$ of independent sets consisting of all singletons. We demonstrate the usefulness of the continuous model by analyzing an algorithm for this problem.

CS Algorithm(f):

1. Observe the secretaries arriving till time $t = e^{-1}$, and let L be the set of secretaries arriving until this time.
2. Let $\hat{s}$ be the maximum secretary in L with respect to $\succ_{\varnothing}$.[a]
3. After time t, accept the first secretary s such that $f(s) \succ_{\varnothing} f(\hat{s})$.

[a] If no secretary arrives till time t, we assume $s \succ_{\varnothing} \hat{s}$ for every secretary s.

Theorem A.1. *The above algorithm for CS is e^{-1}-competitive.*

Proof. Let s^* be the secretary of the optimal solution (breaking ties in favor of the earlier secretary according to $\succ_{\varnothing}$). Given that s^* arrives at some time $x \in (t, 1)$, s^* is accepted if one of the following conditions hold:

- No secretary arrives before time x.
- The best secretary arriving in the time range $[0, x)$ arrives before time t.

Since the secretaries are independent, with probability at least t/x, at least one of these conditions holds. The probability that s^* 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 that the above algorithm accept s^* is at least

$$\int_t^1 \frac{t}{x} dx = t[\ln x]_t^1 = -t \ln t = e^{-1}. \qquad \square$$

B $(20e)^{-1} \approx 0.018$ - Competitive Algorithm for SKS

Recall that the Submodular Knapsack Secretary Problem (SKS) is a secretary problem in which the objective function f is a normalized monotone submodular function and every secretary s has a cost $c(s)$ (revealed upon arrival). A budget B is also given as part of the input, and the algorithm is allowed to hire secretaries as long as it does not exceed the budget. In other words, the collection $\mathcal{I}$ of allowed sets contains every set of secretaries whose total cost is at most B.

Given a set $S \subseteq \mathcal{S}$ of secretaries, we denote by $c(S)$ the total cost of the secretaries in S. Using this notation, a set S is feasible if and only if $c(S) \leq B$. We assume that for every secretary s, $c(s) \leq B$, *i.e.*, every singleton is a feasible solution. If this is not true, we simply ignore secretaries whose cost exceeds B.

B.1 Algorithm for SKS with no Extra Ordinary Secretaries

In this subsection we give an algorithm for SKS that works assuming no single secretary is significant on its own. Formally, we assume that there is no secretary s such that $f(\{s\}) \geq \beta \cdot f(OPT)$, for some $\beta \in (0, 0.1]$. Consider the following algorithm.

Simple-SKS Algorithm (SA)(f, B)**:**

1. Wait till time $t = 1/2$, and let L be the set of secretaries arriving before this time.
2. Let $L' \subseteq L$ be the feasible set maximizing $f(L')$.[a]
3. Initialize $R' \leftarrow \emptyset$.
4. Add to R' every secretary s if $\frac{f_s(R')}{c(s)} \geq \alpha \cdot f(L')/B$[b], and $c(R') + c(s) \leq B$.
5. Return R'.

[a] Finding L' requires an exponential number of oracle queries. If polynomial time is required, L' has to be approximated, which results in a slightly reduced competitive ratio.

[b] $\alpha \in (0, 1]$ is a constant to be determined later.

We call the quantity $\frac{f_s(R')}{c(s)}$ the *marginal density* of secretary s. Analogously, we also define the marginal density of a set S of secretaries to be $\frac{f(R' \cup S) - f(R')}{c(S)}$. Notice that if s has marginal density d, then adding s to R' increases the value of $f(R')$ by $d \cdot c(s)$. Let R be the set of secretaries arriving after time t. By definition, R' is a subset of R.

Observation B.1. *If SA manages to collect secretaries of total cost at least $0.5B$, then $f(R') \geq 0.5\alpha \cdot f(OPT \cap L)$.*

Proof. Every secretary s collected had a marginal density of at least $\alpha \cdot f(L')/B$. Since SA collected secretaries of total cost at least $0.5B$, the total value it collected must be at least: $0.5B \cdot (\alpha \cdot f(L')/B) = 0.5\alpha \cdot f(L') \geq 0.5\alpha \cdot f(OPT \cap L)$. $\square$

Lemma B.2. *If SA does not manage to collect secretaries of total cost at least $0.5B$, $f(R') \geq f(OPT \cap R) - (\alpha + \beta) \cdot f(OPT)$.*

Proof. Let $OPT' = \{s \in OPT | c(s) \leq 0.5B\}$. Since SA failed to collect secretaries of total cost at least $0.5B$, no secretary of OPT' was rejected due to budget constraints. Hence, every secretary in $OPT' \cap (R - R')$ had a marginal density of less than $\alpha \cdot f(L')/B \leq \alpha \cdot f(OPT)/B$ when considered by SA. Since f is submodular, this also true when SA terminates. The final marginal value of $OPT' \cap (R - R')$ can be lower bounded by:

$$\begin{aligned} f[(OPT' \cap (R - R')) \cup R'] - f(R') &= f[(OPT' \cap R) \cup R'] - f(R') \\ &\geq f(OPT' \cap R) - f(R'). \end{aligned}$$

However, the total cost of the secretaries in $OPT' \cap (R - R')$ is at most $c(OPT')$, hence, there must be a secretary s in this set with a marginal density of at least $(f(OPT' \cap R) - f(R'))/c(OPT')$. Combining this with our previous observation that every secretary in $OPT' \cap (R - R')$ has a marginal density of less than $\alpha \cdot f(OPT)/B$, we get:

$$\frac{f(OPT' \cap R) - f(R')}{c(OPT')} < \frac{\alpha \cdot f(OPT)}{B}. \tag{1}$$

Let $A = OPT - OPT'$. Notice that $|A| \leq 1$ because $c(OPT) \leq B$ and $c(s) > B/2$ for every $s \in A$.

$$\frac{f(OPT')}{c(OPT')} \geq \frac{f(OPT) + f(\varnothing) - f(A)}{c(OPT) - |A| \cdot 0.5B} \overset{(*)}{\geq} \frac{1 - |A|\beta}{1 - 0.5|A|} \cdot \frac{f(OPT)}{B} \overset{(**)}{\geq} \frac{f(OPT)}{B}.$$

Inequality (*) follows from our assumption that no single secretary is too good, and inequality (**) follows from the fact that $\beta \leq 0.1$. Plugging this inequality into Equation (1) implies:

$$\frac{f(OPT' \cap R) - f(R')}{c(OPT')} < \frac{\alpha \cdot f(OPT')}{c(OPT')} \Rightarrow f(R') > f(OPT' \cap R) - \alpha \cdot f(OPT').$$

The lemma now follows from the following observation:

$$f(OPT' \cap R) \geq f(OPT \cap R) + f(\varnothing) - f(A) \geq f(OPT \cap R) - \beta \cdot f(OPT).$$

□

Observation B.1 and Lemma B.2 imply immediately the following corollary.

Corollary B.3. $f(R') \geq \min\{0.5\alpha \cdot f(OPT \cap L), f(OPT \cap R) - (\alpha + \beta) \cdot f(OPT)\}$.

Let us denote the secretaries of OPT by $s_1^*, \ldots, s_m^*$, and let w_i be the marginal contribution of s_i^* to OPT, *i.e.*, $w_i = f_{s_i^*}(\{s_j^* | 1 \leq j < i\})$. Let X_i be a random indicator for the event that secretary s_i arrived before time t, and let $W = \sum_{i=1}^m w_i \cdot X_i$.

Lemma B.4. *Let p be the ratio $W/f(OPT)$, then $f(R') \geq f(OPT) \cdot \min\{0.5\alpha p, 1 - p - \alpha - \beta\}$. Choosing $\alpha = 8/25$, and recalling $\beta \leq 0.1$ makes the right hand side at least $f(OPT) \cdot \min\{0.5p \cdot (8/25), 1 - p - 8/25 - 0.1\} = f(OPT) \cdot \min\{8p, 29 - 50p\}/50$.*

Proof. By the submodularity of OPT, $f(OPT \cap L) \geq W$ and $f(OPT \cap R) \geq \sum_{i=1}^m w_i - W = f(OPT) - W$. By Corollary B.3, the value $f(R')$ is at least:

$$\begin{aligned} &\min\{0.5\alpha \cdot f(OPT \cap L), f(OPT \cap R) - (\alpha + \beta) \cdot f(OPT)\} \\ &\geq \min\{0.5\alpha \cdot W, f(OPT) - W - (\alpha + \beta) \cdot f(OPT))\} \\ &= f(OPT) \cdot \min\{0.5\alpha p, 1 - p - \alpha - \beta\}. \end{aligned}$$

□

Since $\mathbb{E}[X_i] = 1/2$ for every $1 \leq i \leq m$, we also get $\mathbb{E}[p] = 1/2$. If p is close to its expectation, Lemma B.4 gives a strong lower bound on the value SA collects. Let us bound the probability that p is far away from $1/2$.

Lemma B.5. $\Pr[|p - 1/2| \geq c] \leq \beta/(4c^2)$*, and since* p *is a symmetric random variable with respect to* $1/2$*,* $\Pr[p < 1/2 - c] = \Pr[p > 1/2 + c] \leq \beta/(4c^2)$*.*

Proof. The variance of p is:

$$\begin{aligned}\mathrm{Var}[p] &= \frac{\mathrm{Var}[W]}{f^2(OPT)} = \frac{\mathrm{Var}\left[\sum_{i=1}^m w_i \cdot X_i\right]}{f^2(OPT)} = \frac{\sum_{i=1}^m w_i^2 \cdot \mathrm{Var}[X_i]}{f^2(OPT)} = \frac{\sum_{i=1}^m w_i^2}{4f^2(OPT)} \\ &\leq \frac{\max_{1\leq i\leq m} w_i \cdot \sum_{i=1}^m w_i}{4f^2(OPT)} = \frac{\max_{1\leq i\leq m} w_i \cdot f(OPT)}{4f^2(OPT)} \leq \frac{\beta}{4}.\end{aligned}$$

Let us now use Chebyshev's inequality to bound the distance of p from its expectation.

$$\Pr[|p - 1/2| \geq c] \leq \frac{1}{(c/\sqrt{\mathrm{Var}[p]})^2} = \frac{\mathrm{Var}[p]}{c^2} = \frac{\beta/4}{c^2} = \frac{\beta}{4c^2}.$$

□

Lemma B.6. *The expected value of* $f(R')$ *is at least:*

- *For* $0.1 \geq \beta \geq 0.0256$*:* $\frac{f(OPT)}{25} \cdot (1 - \sqrt{\beta})^2$.
- *For* $0.0256 \geq \beta$*:* $\frac{f(OPT)}{400} \cdot \left[32 + 641\beta - 232\sqrt{\beta}\right]$.

Proof. We are looking for the worst distribution of p from the point of view of SA. The lower bound given by Lemma B.3 on $f(R')$ is an increasing function of p for $p \in [0, 1/2]$. Hence, we should assume that when $p \leq 1/2$, it is distributed in such a way that the bound in Lemma B.5 is tight when its value is at most $1/2$, and p never get values which make the bound exceed $1/2$. This implies that p never gets values from the range $(1/2 - \sqrt{\beta/4}, 0.5]$, and for lower values, its distribution is defined by, $\Pr[p < c] = \beta/(8(1/2 - c)^2)$. The density function of p in the range $[0, 1/2 - \sqrt{\beta/4}]$ is, therefore,

$$\left[\frac{\beta}{8(1/2 - p)^2}\right]' = \frac{2\beta}{8(1/2 - p)^3} = \frac{\beta}{4(1/2 - p)^3}.$$

Hence, the part of the expected value of $f(R')$ resulting from the case $p \leq 1/2$ is:

$$\begin{aligned}&\int_0^{1/2-\sqrt{\beta/4}} \frac{\beta}{4(1/2 - p)^3} \cdot \frac{8p \cdot f(OPT)}{50} dp \\ &= \frac{\beta \cdot f(OPT)}{25} \cdot \int_0^{1/2-\sqrt{\beta/4}} \frac{p}{(1/2 - p)^3} dp \\ &= \frac{\beta \cdot f(OPT)}{25} \cdot \left[\frac{4p - 1}{(1 - 2p)^2}\right]_0^{1/2-\sqrt{\beta/4}} \\ &= \frac{f(OPT)}{25} \cdot \left[1 - 2\sqrt{\beta} + \beta\right] \\ &= \frac{f(OPT)}{25} \cdot (1 - \sqrt{\beta})^2.\end{aligned}$$

We now do a similar analysis for the case $p \geq 1/2$. The lower bound given by Lemma B.3 on the value of $f(R')$ is a decreasing function of p for $p \in [1/2, 29/50]$ (it is decreasing for larger p values as well, but it becomes negative for such values). Hence, we should assume that when $p \geq 1/2$, it is distributed in such a way that the bound in Lemma B.5 is tight when it is at most $1/2$, and p never gets values which make the bound exceed $1/2$. This implies that p never gets values from the range $[0.5, 1/2 + \sqrt{\beta/4})$, and for larger values, its distribution is defined by, $\Pr[p > c] = \beta/(8(c - 1/2)^2)$. The density function of p in the range $[1/2 + \sqrt{\beta/4}, 29/50]$ is, therefore,

$$\left[\frac{\beta}{8(p-1/2)^2}\right]' = -\frac{2\beta}{8(p-1/2)^3} = -\frac{\beta}{4(p-1/2)^3}.$$

Hence, the part of the expected value of $f(R')$ resulting from the case $p \geq 1/2$ is:

$$\begin{aligned}
&\int_{1/2+\sqrt{\beta/4}}^{29/50} \frac{\beta}{4(p-1/2)^3} \cdot \frac{(29-50p)\cdot f(OPT)}{50} dp \\
&= \frac{\beta \cdot f(OPT)}{200} \cdot \int_{1/2+\sqrt{\beta/4}}^{29/50} \frac{29-50p}{(p-1/2)^3} dp \\
&= \frac{\beta \cdot f(OPT)}{50} \cdot \left[\frac{50x-27}{(1-2p)^2}\right]_{1/2+\sqrt{\beta/4}}^{29/50} \\
&= \frac{f(OPT)}{400} \cdot \left[625\beta + 16 - 200\sqrt{\beta}\right].
\end{aligned}$$

The bound for the case $p \geq 1/2$ is nonnegative only for $\beta \geq 0.0256$, therefore, for $\beta \leq 0.0256$ we use the bound of the case $p \leq 1/2$ alone (0 is always a lower bound on the contribution of the case $p \geq 1/2$). For $\beta \leq 0.0256$, we get an improved bound by accumulating the bounds of the two cases. This results in the following improved bound:

$$\begin{aligned}
&\frac{f(OPT)}{25} \cdot (1-\sqrt{\beta})^2 + \frac{f(OPT)}{400} \cdot \left[625\beta + 16 - 200\sqrt{\beta}\right] \\
&= \frac{f(OPT)}{400} \cdot \left[32 + 641\beta - 232\sqrt{\beta}\right].
\end{aligned}$$

□

B.2 Algorithm for SKS

In this subsection we present an algorithm for SKS that works without any assumptions on the input.

SKS Algorithm(f, B)**:**

1. With probability $1/2$ use SA.
2. Otherwise, use the algorithm for CS from Subsection A, where the value of secretary s is defined by $f(\{s\})$.

Let $\beta = \max_{s \in \mathcal{S}} f(\{s\})/f(OPT)$. Observe that β is a property of the input, and if $\beta \leq 0.1$, then the assumption we used in Subsection B.1 holds with this β value.

Theorem B.7. *The algorithm returns a solution of expected value at least:*

- *For* $\beta \geq 0.1$: $\frac{f(OPT)}{20e}$.
- *For* $1/4 \geq \beta \geq 1/100$: $\frac{f(OPT)}{50} \cdot \left[(1-\sqrt{\beta})^2 + \frac{25\beta}{e}\right]$.
- *For* $1/100 \geq \beta$: $\frac{f(OPT)}{800} \cdot \left[32 + 641\beta - 232\sqrt{\beta} + \frac{400\beta}{e}\right]$.

Proof. If $\beta \geq 0.1$, there is a secretary with value of at least $OPT/10$. The algorithm for CS is used with probability $1/2$, and it returns the best secretary with probability e^{-1}.

If $0.1 \geq \beta \geq 0.0256$ then by Lemma B.6, SA returns a solution with expected value of at least $\frac{f(OPT)}{25} \cdot (1-\sqrt{\beta})^2$. On the other hand, the algorithm for CS finds the best single secretary with probability $1/e$, hence, its expected value is at least $\beta \cdot f(OPT)/e$. Therefore, our algorithm returns a solution of expected value at least:

$$\frac{1}{2} \cdot \left[\frac{f(OPT)}{25} \cdot (1-\sqrt{\beta})^2\right] + \frac{1}{2} \cdot \left[\frac{\beta \cdot f(OPT)}{e}\right] = \frac{f(OPT)}{50} \cdot \left[(1-\sqrt{\beta})^2 + \frac{25\beta}{e}\right].$$

Similarly, if $\beta \leq 0.0256$, our algorithm returns a solution of expected value at least:

$$\begin{aligned} &\frac{1}{2} \cdot \left[\frac{f(OPT)}{400} \cdot \left(32 + 641\beta - 232\sqrt{\beta}\right)\right] + \frac{1}{2} \cdot \left[\frac{\beta \cdot f(OPT)}{e}\right] \\ &= \frac{f(OPT)}{800} \cdot \left[32 + 641\beta - 232\sqrt{\beta} + \frac{400\beta}{e}\right]. \end{aligned}$$

□

Corollary B.8. *For any input, the expected value of the solution of our algorithm is at least $f(OPT)/(20e)$, hence, it is a $1/(20e)$-competitive algorithm for SKS.*

Proof. Let us find the derivative (by β) of the lower bound on the expected value of the algorithm given by Theorem B.7 for $0.1 \geq \beta \geq 0.0256$.

$$\begin{aligned} \left[\frac{f(OPT)}{50} \cdot \left[(1-\sqrt{\beta})^2 + \frac{25\beta}{e}\right]\right]' &= \frac{f(OPT)}{50} \cdot \left[2(1-\sqrt{\beta})\left(-\frac{1}{2\sqrt{\beta}}\right) + \frac{25}{e}\right] \\ &= \frac{f(OPT)}{50} \cdot \left[\frac{25}{e} + 1 - \frac{1}{\sqrt{\beta}}\right]. \end{aligned}$$

The derivative implies that the expected value is minimal for $\beta = (25/e+1)^{-2} \approx 0.00962$. We are interested in the range $0.1 \geq \beta \geq 0.0256$, and we get that the minimal expected value for this range is achieved for $\beta = 0.0256$. The competitive ratio associated with this β value is:

$$\begin{aligned} &\frac{f(OPT)}{50} \cdot \left[(1-\sqrt{0.0256})^2 + \frac{25 \cdot 0.0256}{e}\right] \\ &\geq \frac{f(OPT)}{50} \cdot [0.705 + 0.235] \\ &= \frac{f(OPT)}{50} \cdot 0.94 \\ &\geq \frac{f(OPT)}{54}. \end{aligned}$$

Similarly, we now find the derivative of the lower bound on the expected value of the algorithm given by Theorem B.7 for $\beta \leq 0.0256$.

$$\begin{aligned} &\left[\frac{f(OPT)}{800} \cdot \left[32 + 641\beta - 232\sqrt{\beta} + \frac{400\beta}{e}\right]\right]' \\ &= \frac{f(OPT)}{800} \cdot \left[641 - 232 \cdot \frac{1}{2\sqrt{\beta}} + \frac{400}{e}\right] \\ &= \frac{f(OPT)}{800} \cdot \left[641 - \frac{116}{\sqrt{\beta}} + \frac{400}{e}\right]. \end{aligned}$$

This derivative implies that the expected value is minimal for $\beta = 116^2(641 + 400/e)^{-2} \approx 0.0216$. We are interested in the range $\beta \leq 0.0256$, and we get that the minimal expected value for this range is also achieved for $\beta = 0.0216$. The competitive ratio associated with this β value is:

$$\begin{aligned}
&\frac{f(OPT)}{800} \cdot \left[32 + 641 \cdot 0.0216 - 232\sqrt{0.0216} + \frac{400 \cdot 0.0216}{e}\right] \\
&\geq \frac{f(OPT)}{800} \cdot [32 + 13.845 - 34.097 + 3.178] \\
&= \frac{f(OPT)}{800} \cdot 14.926 \geq \frac{f(OPT)}{54}.
\end{aligned}$$

The corollary now follows since $20e > 54$. $\square$