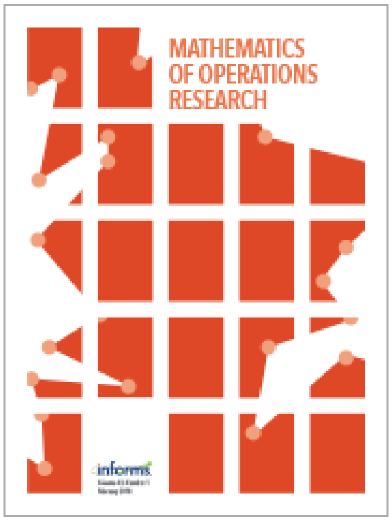

Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Secretary Problems via Linear Programming

Niv Buchbinder, Kamal Jain, Mohit Singh

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 39, No. 1, February 2014, pp. 190–206
ISSN 0364-765X (print) | ISSN 1526-5471 (online)
http://dx.doi.org/10.1287/moor.2013.0604

Secretary Problems via Linear Programming

Niv Buchbinder
Statistics and Operations Research Department, Tel Aviv University, Tel Aviv 69978, Israel, niv.buchbinder@gmail.com

Kamal Jain
eBay Research Labs, Redmond, Washington 98052, kamaljain@gmail.com

Mohit Singh
Microsoft Research, Redmond, Washington 98052, mohits@microsoft.com

In the classical secretary problem an employer would like to choose the best candidate among n competing candidates that arrive in a random order. In each iteration, one candidate's rank vis-a-vis previously arrived candidates is revealed and the employer makes an irrevocable decision about her selection. This basic concept of n elements arriving in a random order and irrevocable decisions made by an algorithm have been explored extensively over the years, and used for modeling the behavior of many processes. Our main contribution is a new linear programming technique that we introduce as a tool for obtaining and analyzing algorithms for the secretary problem and its variants. The linear program is formulated using judiciously chosen variables and constraints and we show a one-to-one correspondence between algorithms for the secretary problem and feasible solutions to the linear program. Capturing the set of algorithms as a linear polytope holds the following immediate advantages:

- Computing the optimal algorithm reduces to solving a linear program.
- Proving an upper bound on the performance of any algorithm reduces to finding a feasible solution to the dual program.
- Exploring variants of the problem is as simple as adding new constraints, or manipulating the objective function of the linear program.

We demonstrate these ideas by exploring some natural variants of the secretary problem. In particular, using our approach, we design optimal secretary algorithms in which the probability of selecting a candidate at any position is equal. We refer to such algorithms as *position independent* and these algorithms are motivated by the recent applications of secretary problems to online auctions. We also show a family of linear programs that characterize all algorithms that are allowed to choose J candidates and gain profit from the K best candidates. We believe that a linear programming based approach may be very helpful in the context of other variants of the secretary problem.

Keywords: secretary problem; linear programming
MSC2000 subject classification: Primary: 68W40 analysis of algorithms
OR/MS subject classification: Primary: linear programming; secondary: algorithms
History: Received February 29, 2012; revised January 13, 2013. Published online in *Articles in Advance* June 27, 2013.

1. Introduction. In the classical secretary problem an employer would like to choose the best candidate among n competing candidates. The candidates are assumed to arrive in a random order. After each interview, the position of the interviewee in the total order is revealed vis-á-vis already interviewed candidates. The interviewer has to decide, irrevocably, whether to accept the candidate for the position or to reject the candidate. The objective in the basic problem is to accept the best candidate with high probability. An algorithm used for choosing the best candidate is to interview the first n/e candidates for the purpose of evaluation, and then hire the first candidate that is better than all previous candidates. Analysis of the algorithm shows that it hires the best candidate with probability $1/e$ and that it is optimal (Dynkin [11], Lindley [24]).

This basic concept of n elements arriving in a random order and irrevocable decisions made by an algorithm have been explored extensively over the years. We refer the reader to the survey by Ferguson [12] on the historical and extensive work on different variants of the secretary problem. Recently, there has been an interest in the secretary problem with its application to the online auction problem (Hajiaghayi et al. [17], Babaioff et al. [2]). This has led to the study of variants of the secretary problem that are motivated by online auction scenarios. For example, Kleinberg [20] studied a setting in which the algorithm is allowed to select multiple candidates and the goal is to maximize the expected profit. Imposing other combinatorial structures on the set of selected candidates, for example, selecting elements that form an independent set of a matroid (Babaioff et al. [3]), selecting elements that satisfy a given knapsack constraint (Babaioff et al. [4]), selecting elements that form a matching in a graph or hypergraph (Korula and Pál [21]), have also been studied. Other variants include settings in which the profit of selecting a secretary is discounted with time (Babaioff et al. [5]). Therefore, finding new ways of abstracting, as well as analyzing and designing algorithms, for secretary type problems is of major interest.

1.1. Our contributions. Our main contribution is a new linear programming technique that we introduce as a tool for obtaining and analyzing algorithms for various secretary problems. We introduce a linear program with judiciously chosen variables and constraints and show a one-to-one correspondence between algorithms for the secretary problem and feasible solutions to the linear program. Obtaining an algorithm that maximizes a certain objective therefore reduces to finding an optimal solution to the linear program. We then use linear programming duality to give a simple proof of optimality. We illustrate our technique by applying it to the classical secretary problem and obtaining a simple proof of optimality of the famous thresholding algorithm (Dynkin [11]) in §2.

Our linear program for the classical secretary problem consists of a single constraint that bounds the probability that the algorithm may select the ith candidate. Despite its simplicity, we show that such a set of constraints suffices to correctly capture all possible algorithms. Thus, optimizing over this polytope results in the optimal algorithm. The simplicity and the tightness of the linear programming formulation makes it flexible and applicable to many other variants. Capturing the set of algorithms as a linear polytope holds the following immediate advantages:

- Computing the optimal algorithm reduces to solving a linear program.
- Proving an upper bound on the performance of any algorithm reduces to finding a feasible solution to the dual program.
- Exploring variants of the problem is as simple as adding new constraints, or manipulating the objective function of the linear program.

We next demonstrate these ideas by exploring some natural variants of the secretary problem.

1.1.1. The J-choice, K-best secretary problem. Our Linear programming (LP) formulation approach is able to capture a much broader class of secretary problems. We define a most general problem that we call the J-choice, K-best secretary problem, referred to as the (J, K)-secretary problem. Here, n candidates arrive randomly. The algorithm is allowed to pick up to J different candidates and the objective is to pick as many from the top K ranked candidates. Observe that the optimal offline algorithm for the (J, K)-secretary problem selects $\min(J, K)$ candidates among the best K. The $(1, 1)$-secretary problem is the classical secretary problem. For any J, K, we provide a linear program that characterizes all algorithms for the problem by generalizing the linear program for the classical secretary problem. To measure the performance of an algorithm, we use the standard notions from competitive analysis (Borodin and El-Yaniv [6]) and compare it to the best offline algorithm that knows all the information.

The case $J > K$ corresponds to a case where you may choose $J > K$ candidates, but only the best K candidates are of interest. If $J < K$ you have less choices than K, but gain if you manage to hire as many as possible from the best K candidates. A subclass that is especially interesting is the (K, K)-secretary problem, since it is closely related to the problem of maximizing the expected profit in a cardinal version of the problem. In the cardinal version of the problem, n elements that have arbitrary nonnegative values arrive in a random order. The algorithm is allowed to select at most K elements and its goal is to maximize the expected sum of the values of the items selected. We observe in the following that essentially a c-competitive algorithm for the (K, K) secretary corresponds to a c-competitive algorithm for the cardinal version of the problem, and vice versa.

We define a *monotone* algorithm for the (J, K)-secretary problem to be an algorithm that, at any iteration, does not select an element that is t best so far with probability higher than an element that is $t' < t$ best so far. We note that any reasonable algorithm (and in particular the optimal algorithm) is monotone. We also define an *oblivious* algorithm for the cardinal version of the problem to be an algorithm that makes its decisions solely based on the relative ranks so far and ignores the actual values. We remark that the best known algorithm for the cardinal problem is oblivious (Kleinberg [20]). The following is a simple observation whose proof appears in the appendix.

Observation 1. The following holds.

- Let $\mathcal{A}$ be a monotone algorithm for the (K, K)-secretary problem that is c-competitive. Then it corresponds to an algorithm for the cardinal version of the problem that is also c-competitive for maximizing the expected profit.
- Let $\mathcal{A}$ be an oblivious algorithm for the cardinal problem that is c-competitive. Then it corresponds to an algorithm for the (K, K)-secretary problem that is also c-competitive.

Kleinberg [20] gave an asymptotically tight algorithm for the cardinal version of the problem. However, this algorithm is randomized, and also not tight for small values of k. Better algorithms, even restricted to small values of k, are helpful not only for solving the original problem, but also for improving algorithms that are based upon them. For example, an algorithm for the secretary knapsack (Babaioff et al. [4]) uses an algorithm that is $1/e$ competitive for maximizing the expected profit for small values of k ($k \leq 27$). Analyzing the LP asymptotically for any value n is a challenge even for small value k. However, using our characterization we

Table 1. The ratio of the expected profit of the algorithm to the value obtained from the k highest candidates. The ratio was obtained by solving the linear formulation for $n = 100$. We do not know how to analyze the LP asymptotically for any value n.

Number of elements allowed to be picked by the algorithm	Competitive ratio
1	$1/e = 0.368$
2	0.474
3	0.565
4	0.613

solve the problem easily for small values k and n that gives an idea on how a competitive ratio behaves for small values of k. Our results appear in Table 1. We also give complete asymptotic analysis for the cases of $(1, 2)$, $(2, 1)$-secretary problems.

1.1.2. Position independent algorithms. As discussed earlier, the optimal algorithm for the classical secretary problem is the thresholding algorithm that interviews the first n/e candidates for the purpose of evaluation, and then hires the first candidate that is better than all previous candidates. Consider a scenario where the candidates have an option of changing their arrival position. The thresholding algorithm then suffers from a crucial drawback. The candidates arriving early may delay their interview and candidates arriving after the position $n/e + 1$ may advance their interview. Such a behavior challenges the main assumption of the model that interviewees arrive in a random order. This issue of delaying or advancing the interview position is of major importance especially since secretary problems have been used recently in the context of online auctions (Hajiaghayi et al. [17], Babaioff et al. [2]).

Using the linear programming technique, we study algorithms that are *position independent*. We call an algorithm for the secretary problem *position independent* if the probability of selecting a candidate at ith position is equal for each position $1 \le i \le n$. We show that there exists a position independent algorithm that selects the best candidate with probability $1 - 1/\sqrt{2} \approx 0.29$ and that this algorithm is optimal. Ensuring that a candidate is selected at each position with equal probability is ensured by introducing a set of very simple constraints in the linear program.

Surprisingly, we find that the optimal position independent algorithm sometimes selects a candidate who is worse than a previous candidate. To deal with this issue, we call an algorithm *regret free* if the algorithm only selects candidates that are better than all previous candidates. We show that the best position independent algorithm that is regret free accepts the best candidate with probability $\frac{1}{4}$. Another issue with the optimal position independent algorithm is that it does not always select a candidate. In the classical secretary problem, the algorithm can always pick the last candidate, but this solution is unacceptable when we want the probability of acceptance to be equal at each position. We call an algorithm *must hire* if it always hires a candidate. We show that there is a must-hire position independent algorithm that hires the best candidate with probability $\frac{1}{4}$ (and it is optimal). All the above results are optimal and we use the linear programming technique to derive the algorithms as well as to prove their optimality.

In subsequent work (Buchbinder et al. [8]), we further explore the importance of position independent algorithms in the context of online auctions. In this context, bidders are bidding for an item and may have an incentive to change their position if this may increase their utility. We show how to obtain truthful algorithms for such settings using position independent algorithms for the secretary problem.

1.2. Related work. The basic secretary problem was introduced in a puzzle by Gardner [14]. Dynkin [11] and Lindley [24] gave the optimal solution and showed that no other strategy can do better (see the historical survey by Ferguson [12] on the history of the problem). Subsequently, various variants of the secretary problem have been studied with different assumptions and requirements (Samuels [26]) (see the survey Freeman [13]).

More recently, there has been significant work using generalizations of secretary problems as a framework for online auctions (Babaioff et al. [2, 4, 3], Hajiaghayi et al. [17], Kleinberg [20], Chakraborty and Lachish [10]). For example, Babaioff et al. [3] proposed a very general framework in which elements of a matroid arrive one by one and the algorithm solution is restricted to be an independent set of the matroid. Incentive issues in online algorithms have been studied in several models (Awerbuch et al. [1], Hajiaghayi et al. [17], Lavi and Nisan [23]). These works designed algorithms where incentive issues were considered for both value and time strategies. For example, Hajiaghayi et al. [17] studied a limited supply online auction problem, in which an auctioneer has a limited supply of identical goods and bidders arrive and depart dynamically. In their problem bidders also have

(P) $\max \quad \frac{1}{n}\cdot\sum_{i=1}^{n} ip_i$	(D) $\min \quad \sum_{i=1}^{n} x_i$
s.t. $\forall\, 1\le i\le n \quad i\cdot p_i \le 1-\sum_{j=1}^{i-1} p_j$	s.t. $\forall\, 1\le i\le n \quad \sum_{j=i+1}^{n} x_j + ix_i \ge i/n$
$\forall\, 1\le i\le n \quad p_i\ge 0$	$\forall\, 1\le i\le n \quad x_i\ge 0$

FIGURE 1. Linear program and its dual for the secretary problem.

a time window that they can lie about. Recently, Immorlica et al. [18] considered an interesting variant of the secretary problem (and other problems) in which two employers compete, and the goal of the algorithm is to beat the other employer rather than hiring the best candidate.

Our linear programming technique is similar to the technique of *factor revealing* linear programs that have been used successfully in many different settings (Buchbinder et al. [7], Goemans and Kleinberg [16], Jain et al. [19], Mehta et al. [25]). Factor revealing linear program formulates the worst-case performance of an algorithm as optimizing a linear program. Our technique, in contrast, formulates finding the *best* algorithm as optimizing a linear program. In that sense, our linear program captures the information structure of the problem itself by a linear program and not just the performance of a given algorithm.

The conference version of this paper can be found in Buchbinder et al. [9]. Recently, our techniques were shown to be useful for more secretary variants. Gharan and Vondrák [15] obtained tight factors using our approach for a variant of the secretary problem in which the number of candidates in unknown to the algorithm (and only a bound is given). Kumar et al. [22] studied a generalization of the classical secretary problem in a setting where there is only a partial order between the elements and the goal of the algorithm is to return one of the maximal elements of the poset. They managed to obtain an almost tight upper bound for the problem using our approach.

2. Introducing the technique: Classical secretary (and variants). In this section, we give a simple linear program that we show characterizes *all* possible algorithms for the secretary problem. We stress that the linear program captures not only thresholding algorithms, but any algorithm, including probabilistic algorithms. Thus, finding the best algorithm for the secretary problem is equivalent to finding an optimal solution to the linear program. The linear program and its dual appear in Figure 1. The following two lemmas show that the linear program exactly characterizes all feasible algorithms for the secretary problem.

LEMMA 1 (ALGORITHM TO LP SOLUTION). *Let π be any algorithm for selecting the best candidate that only selects the best candidate so far.*[1] *Let p_i^π denote the probability of selecting the candidate at position i. Then p^π is a feasible solution to the linear program (P), i.e., it satisfies the constraints $p_i^\pi \le (1/i)(1-\sum_{j<i} p_j^\pi)$ for each $1\le i\le n$. Moreover the objective value $(1/n)\sum_{i=1}^{n} ip_i^\pi$ is at least the probability of selecting the best candidate by π.*

PROOF. Let p_i^π be the probability in which algorithm π selects candidate i. We now show that p^π satisfies the constraints of linear program. Observe that

$$\begin{aligned} p_i^\pi &= \Pr[\pi \text{ selects candidate } i \mid \text{candidate } i \text{ is best so far}] \\ &\qquad \cdot \Pr[\text{candidate } i \text{ is best so far}] \\ &\le \Pr[\pi \text{ did not select candidates } \{1,\dots,i-1\}] \mid \text{candidate } i \text{ is best so far}]\cdot\frac{1}{i}. \end{aligned}$$

However, the probability of selecting candidates 1 to $i-1$ depends only on the relative ranks of these candidates and is independent on whether candidate i is best so far (which can be determined after the algorithm has done its choices regarding candidates 1 to $i-1$). Therefore, we obtain $p_i^\pi \le (1/i)(1-\sum_{j<i} p_j^\pi)$, which proves our claim.

Now we show that the objective function of the linear program is at least the probability with which π accepts the best candidate. Let us consider the probability distribution over the relative ranks of candidates $1,\dots,i$

[1] Any algorithm cannot increase its chances of hiring the best candidate by selecting a candidate that is not the best so far, therefore we may consider only such algorithms. Formally, if an algorithm selects at any time a candidate that is not best so far, then consider an algorithm that in this case rejects the candidate and all subsequent candidates (even if they are the best so far). This algorithm only selects candidates that are best so far, and has the same probability of success as the original algorithm.

given the following two events. First, an event in which the best candidate over all candidates appears in the ith position, and secondly the event in which the best candidate over candidates $1, \ldots, i$ appears in the ith position (but it is not necessarily best over all candidates). It is easy to verify that the two distributions are the same, and are composed of a uniform random permutation over the relative ranks of candidates $1, \ldots, i-1$ while the rank 1 candidate appears at position i. The algorithm cannot distinguish between the cases in which the ith candidate is the best candidate so far or best candidate over all, and in both cases its input sequence is distributed the same. Thus, the probability that the algorithm selects candidate i given that the best candidate is in the ith position equals the probability that the algorithm selects candidate i given that the best candidate among candidates 1 to i is in the ith position. Since the ith candidate is best so far with probability $1/i$, the latter probability is at most ip_i^π. Summing over all n positions we get that π hires the best candidate with probability at most $(1/n)\sum_{i=1}^n ip_i^\pi$.

Lemma 1 shows that the optimal solution to (P) is an upper bound on the performance of the algorithm. The following lemma shows that every LP solution actually corresponds to an algorithm that performs as well as the objective value of the solution.

Lemma 2 (LP Solution to Algorithm). *Let p_i for $1 \le i \le n$ be any feasible solution to (P). Then consider the algorithm π that selects the candidate i with probability $ip_i/(1-\sum_{j<i} p_j)$ if candidate i is the best candidate so far and candidate $1, \ldots, i-1$ has not been selected, i.e., the algorithm reaches candidate i. Then π is an algorithm that selects the best candidate with probability $(1/n)\sum_{i=1}^n ip_i$.*

Proof. First, notice that the algorithm is well defined since by the LP inequalities for any i, $ip_i/(1-\sum_{j<i} p_j) \le 1$. We prove by induction that the probability that the algorithm selects candidate at position i is exactly p_i. The base case is trivial. Assume this is true until $i-1$. At step i, we claim that the probability we choose candidate i is $(1-\sum_{j<i} p_j)(1/i)ip_i/(1-\sum_{j<i} p_j) = p_i$. This is simply a multiplication of the probability of three events:

1. The probability that the algorithm did not choose candidates 1 to $i-1$.
2. The probability that the current candidate is best so far (better than candidates $1, \ldots, i-1$).
3. The probability that the algorithm selects candidate i given that events (1) and (2) happened.

It is easy to verify that by the properties of the algorithm and the input sequence, the first two events are independent.

By the definition of the algorithm, its behavior is the same if the candidate is best so far, or best overall. Thus, the probability of hiring the ith candidate given that the ith candidate is the best candidate equals the probability of hiring the ith candidate given the ith candidate is the best candidate among candidates 1 to i. Since the ith candidate is best among candidates $1, \ldots, i$ with probability $1/i$, the latter probability equals ip_i (the algorithm hires only the best candidate so far). Summing over all possible positions we get that the algorithm π hires the best candidate with probability $(1/n)\sum_{i=1}^n ip_i$.

Using the above equivalence between LP solutions and the algorithms, we construct the optimal solution by showing a feasible solution to the primal program (P). To prove that the solution is indeed optimal we construct a feasible solution to the dual (D) with the same value. We remark that, as expected, the algorithm is equivalent to the previously suggested thresholding algorithm (Dynkin [11]). Although we do not have a systematic way of constructing the primal and dual solutions, complementary slackness conditions, as well as guessing the set of tight constraints, were useful in obtaining the solutions. Verifying that the primal-dual pair are indeed optimal is then a simple task.

To exhibit the primal and dual solutions let τ be the integer number such that

$$\sum_{i=\tau}^{n-1} \frac{1}{i} < 1 \le \sum_{i=\tau-1}^{n-1} \frac{1}{i}. \tag{1}$$

Observe that for $n \ge 2$, $\tau \ge 2$ (we assume here that $n \ge 2$ since, otherwise, the solution is trivial).

Lemma 3 (Dynkin [11]). *For any $n \ge 2$, there exists an algorithm that can hire the best candidate with probability $((\tau-1)/n)\sum_{j=\tau-1}^{n-1}(1/j)$, and this algorithm is optimal.*

It is not hard to verify that $\lim_{n\to\infty}((\tau-1)/n)\sum_{j=\tau-1}^{n-1}(1/j) = 1/e$, and this value is, of course, equal the previously known optimal value.

$$\text{(P}') \quad \max \quad \frac{1}{n}\cdot\sum_{i=1}^{n} ip_i + q\left(1-\sum_{i=1}^{n} p_i\right) \qquad \text{(D}') \quad \min \quad \sum_{i=1}^{n} x_i + q$$

$$\text{s.t.} \quad \forall\, 1\le i\le n \quad i\cdot p_i \le 1-\sum_{j=1}^{i-1} p_j \qquad \text{s.t.} \quad \forall\, 1\le i\le n \quad \sum_{j=i+1}^{n} x_j + ix_i \ge i/n - q$$

$$\forall\, 1\le i\le n \quad p_i\ge 0 \qquad \forall\, 1\le i\le n \quad x_i \ge 0$$

FIGURE 2. Linear program and its dual for the rehiring secretary problem.

PROOF. By Lemma 1 and Lemma 2 we get that the optimal solution to (P) corresponds to the optimal algorithm. Any feasible solution (D) corresponds to an upper bound. We construct here feasible primal and dual solutions with the same objective value, and hence are both optimal.

The primal and dual solutions are the following.

Primal solution	Dual solution
$p_i = \begin{cases} 0 & 1\le i<\tau \\ (\tau-1)\left(\dfrac{1}{i-1}-\dfrac{1}{i}\right) & \tau\le i\le n \end{cases}$	$x_i = \begin{cases} 0 & 1\le i<\tau \\ \dfrac{1}{n}\left(1-\displaystyle\sum_{j=i}^{n-1}\dfrac{1}{j}\right) & \tau\le i\le n \end{cases}$

We next prove that the primal and dual solutions are feasible and have equal objective function.

Primal (P) is feasible. All p_i are nonnegative by definition (note that $n\ge 2$ and thus $\tau\ge 2$). We prove that the constraints of (P) hold. For $i<\tau$,

$$ip_i = 0 \le 1 = 1-\sum_{j=1}^{i-1} p_j.$$

For $i\ge\tau$,

$$ip_i = \frac{\tau-1}{i-1} = 1-\sum_{j=\tau}^{i-1}(\tau-1)\left(\frac{1}{j-1}-\frac{1}{j}\right) = 1-\sum_{j=1}^{i-1} p_j.$$

The second equality follows by a telescopic sum.

Dual solution is feasible. By our choice of τ, $\sum_{i=\tau}^{n-1}(1/i) < 1$ and thus all variable x_i are nonnegative. We next prove that the constraints of (D) are all maintained. First, observe that for all $i\ge\tau$, $x_{i-1} = x_i - 1/(n(i-1))$, and for $i<\tau$, $x_{i-1}\ge x_i - 1/(n(i-1))$.

For $i=n$, $nx_n = 1$, and hence the constraint is maintained with equality. Next, we prove feasibility for every $i<n$ by a backward induction:

$$(i-1)x_{i-1} + \sum_{j=i}^{n} x_j \ge (i-1)\left(x_i - \frac{1}{n(i-1)}\right) + \sum_{j=i}^{n} x_j = -\frac{1}{n} + ix_i + \sum_{j=i+1}^{n} x_j \ge -\frac{1}{n} + \frac{i}{n} = \frac{i-1}{n}.$$

Observe also that the constraint is tight for any $\tau\le i\le n$.

Objective function. The objective function of the primal is equal to

$$\frac{1}{n}\sum_{i=1}^{n} ip_i = \frac{1}{n}\sum_{i=\tau}^{n}\frac{\tau-1}{i-1} = \frac{\tau-1}{n}\sum_{i=\tau-1}^{n-1}\frac{1}{i}. \tag{2}$$

By our observation that the dual constraint of $i=\tau$ is tight, we get that the objective function of the dual equals

$$\begin{aligned}\sum_{j=\tau}^{n} x_j &= -(\tau-1)x_\tau + \left(\tau x_\tau + \sum_{j=\tau+1}^{n} x_j\right) = -(\tau-1)\left(\frac{1}{n}\left(1-\sum_{j=\tau}^{n-1}\frac{1}{j}\right)\right) + \frac{\tau}{n} \\ &= \frac{1}{n} + \frac{\tau-1}{n}\sum_{j=\tau}^{n-1}\frac{1}{j} = \frac{\tau-1}{n}\sum_{j=\tau-1}^{n-1}\frac{1}{j}.\end{aligned}$$

2.1. Allowed to rehire. To demonstrate the power of the linear programming approach, we consider a natural extension of the secretary problem. In this variant, one is allowed to rehire the best secretary at the end with certain probability. That is, after the interviewer has seen all n candidates, he is allowed to hire the best candidate with certain probability q if no other candidate has been hired. The probability q is exogenously given to the algorithm, and is known to the algorithm. Observe that if $q = 0$, the problem reduces to the classical secretary problem and if $q = 1$, then the optimal strategy is to wait till the end and then hire the best candidate. We give a tight description of optimal strategy as q changes. We achieve this by modifying the linear program: simply add in the objective function $q(1 - \sum_{i=1}^{n} p_i)$. That is, if the algorithm did not hire any candidate you may hire the best candidate with probability q. Solving the primal and the corresponding dual (see Figure 2) give the following tight result.

THEOREM 1. *Let τ be the integer number such that*

$$\sum_{i=\tau}^{n-1} \frac{1}{i} < 1 - q \le \sum_{i=\tau-1}^{n-1} \frac{1}{i}. \tag{3}$$

For any $n \ge 2$, there exists an algorithm that can hire the best candidate with probability $((\tau-1)/n) \cdot (q + \sum_{j=\tau-1}^{n-1}(1/j))$, and this algorithm is optimal.

It is not hard to verify that here $\lim_{n\to\infty}((\tau-1)/n)(q + \sum_{j=\tau-1}^{n-1}(1/j)) = 1/e^{1-q}$.

PROOF. The primal and dual solutions are the following (note that the definition of τ was changed).

Primal solution	Dual solution
$p_i = \begin{cases} 0 & 1 \le i < \tau \\ (\tau-1)\left(\dfrac{1}{i-1} - \dfrac{1}{i}\right) & \tau \le i \le n \end{cases}$	$x_i = \begin{cases} 0 & 1 \le i < \tau \\ \dfrac{1-q}{n} - \dfrac{1}{n}\displaystyle\sum_{j=i}^{n-1}\dfrac{1}{j} & \tau \le i \le n \end{cases}$

We next prove that the primal and dual solutions are feasible and have equal objective function.

Primal (P) is feasible: Feasibility follows by the same arguments as Lemma 3.

Dual solution is feasible: By our choice of τ, $\sum_{i=\tau}^{n-1}(1/i) < 1 - q$ and thus all variable x_i are nonnegative. We next prove that the constraints of (D) are all maintained. First, observe that for all $i \ge \tau$, $x_{i-1} = x_i - 1/(n(i-1))$, and for $i < \tau$, $x_{i-1} \ge x_i - 1/(n(i-1))$.

For $i = n$, $nx_n = 1 - q$, and hence the constraint is maintained with equality. Next, we prove feasibility for every $i < n$ by a backward induction:

$$(i-1)x_{i-1} + \sum_{j=i}^{n} x_j \ge (i-1)\left(x_i - \frac{1}{n(i-1)}\right) + \sum_{j=i}^{n} x_j = -\frac{1}{n} + ix_i + \sum_{j=i+1}^{n} x_j \ge -\frac{1}{n} + \frac{i}{n} - q = \frac{i-1}{n} - q.$$

Observe also that the constraint is tight for any $\tau \le i \le n$.

Objective function. First, note that,

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

Thus, the objective function of the primal solution is

$$\frac{1}{n}\sum_{i=1}^{n} ip_i + q\left(1 - \sum_{i=1}^{n} p_i\right) = \frac{\tau-1}{n}\sum_{i=\tau-1}^{n-1}\frac{1}{i} + q\frac{\tau-1}{n}. \tag{4}$$

By our observation that the dual constraint of $i = \tau$ is tight, we get that the objective function of the dual equals

$$\begin{aligned} q + \sum_{j=\tau}^{n} x_j &= q - (\tau-1)x_\tau + \left(\tau x_\tau + \sum_{j=\tau+1}^{n} x_j\right) = -(\tau-1)\left(\frac{1-q}{n} - \frac{1}{n}\sum_{j=\tau}^{n-1}\frac{1}{j}\right) + \frac{\tau}{n} \\ &= \frac{\tau-1}{n}\left(q + \sum_{j=\tau-1}^{n-1}\frac{1}{j}\right). \end{aligned}$$

3. The J-choice K-best secretary problem. In this section we study (J, K)-secretary problem of selecting as many of the top K ranked secretaries given J candidates to select. Formally, the algorithm is allowed to select at most J (out of n) candidates, and it gains one unit from selecting any of the first K ranked candidates. We compare the performance of the algorithm to the offline optimum that selects $\min(J, K)$ candidates. Recall that the classical secretary problem is simply the $(1, 1)$-secretary problem. Other interesting special cases include the $(J, 1)$-secretary problem in which the algorithm is allowed to choose J candidates, but gets profit only for the best candidate, or an orthogonal $(1, K)$-secretary problem in which the algorithm is allowed to choose a single candidate, but receives profit for any one of the best K candidates. In this section, we extend the results of §2 and devise a simple linear formulation that characterizes all algorithms for the (J, K)-secretary problem.

3.1. Linear program. To devise the linear formulation we introduce the following variables:

- p_i^j will be the variable denoting the probability that the algorithm accepts the candidate at the ith position as its jth choice, that is, the algorithm chooses the candidate at the ith position and it already chose *exactly* $j-1$ candidates among candidates $1, 2, \ldots, i-1$;
- $q_i^{j\,|\,k}$ will be the variable denoting the probability of accepting the candidate at the ith position as its jth choice given that the candidate at the ith position is the kth best candidate *among the first i candidates*.

We formulate the following linear program:

$$\text{(LP–JK)} \quad \max \ \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{\ell=1}^{k}\frac{\binom{n-i}{k-\ell}\binom{i-1}{\ell-1}}{\binom{n-1}{k-1}}q_i^{j\,|\,\ell}$$

$$p_i^j = \frac{1}{i}\sum_{k=1}^{\min\{i,K\}} q_i^{j\,|\,k} \quad \forall\, 1 \le i \le n, \ \ 1 \le j \le J \tag{5}$$

$$q_i^{1\,|\,k} \le 1 - \sum_{\ell<i} p_\ell^1 \quad \forall\, 1 \le i \le n, \ \ 1 \le k \le K \tag{6}$$

$$q_i^{j\,|\,k} \le \sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^j \quad \forall\, 1 \le i \le n, \ \ 1 \le k \le K, \ \ 2 \le j \le J \tag{7}$$

$$p_i^j, \, q_i^{j\,|\,k} \ge 0 \quad \forall\, 1 \le i \le n, \ \ 1 \le k \le K, \ \ 1 \le j \le J. \tag{8}$$

For ease of analysis, we also introduce auxiliary variables $f_i^{j\,|\,k}$, which will denote the probability of accepting the candidate at the ith position as its jth choice given that the candidate at the ith position is the kth best candidate *globally*. While this variable does not explicitly appear in the linear program, it is closely related to variables $q_i^{j\,|\,k}$ and simplifies the expression for the objective function as given by the following lemma. This lemma crucially uses the fact that the secretary algorithm makes its decision solely based on the relative order of the candidates it already observed.

LEMMA 4. *For any algorithm for the (J, K)-secretary problem, let $q_i^{j\,|\,k}$ and $f_i^{j\,|\,k}$ be defined as above. Then, for any $1 \le i \le n$, $1 \le j \le J$ and $1 \le k \le K$, we have that*

$$f_i^{j\,|\,k} = \sum_{\ell=1}^{k}\frac{\binom{i-1}{\ell-1}\binom{n-i}{k-\ell}}{\binom{n-1}{k-1}}q_i^{j\,|\,\ell}. \tag{9}$$

Thus the objective function of the (LP–JK) *can be restated as*

$$\max \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\sum_{k=1}^{K} f_i^{j\,|\,k}.$$

PROOF. Let us analyze the probability that any algorithm π selects candidate i as its jth choice. For ease of notation we drop i and j, and say from now that the algorithm selects the candidate, meaning it selects candidate i as its jth choice. We claim that

$$\begin{aligned} f_i^{j\,|\,k} &= \Pr[\pi \text{ selects the candidate} \mid \text{the candidate is } k\text{th best globally}] \\ &= \sum_{\ell=1}^{k}\Pr[\pi \text{ selects the candidate} \mid \text{the candidate is } k\text{th best globally, and } \ell\text{th best so far}] \\ &\qquad \cdot \Pr[\text{the candidate is } \ell\text{th best so far} \mid \text{the candidate is } k\text{th best globally}] \end{aligned}$$

$$= \sum_{\ell=1}^{k} \Pr[\pi \text{ selects the candidate} \mid \text{the candidate is } \ell\text{th best so far}]$$
$$\cdot \Pr[\text{the candidate is } \ell\text{th best so far} \mid \text{the candidate is } k\text{th best globally}]$$
$$= \sum_{\ell=1}^{k} q_i^{j\mid\ell} \cdot \Pr[\text{the candidate is } \ell\text{th best so far} \mid \text{the candidate is } k\text{th best globally}]. \tag{10}$$

Equality (10) claims that π selects candidate i given that it is kth best globally and ℓth best so far is equal to the probability π selects the candidate given that it is ℓ best so far. To see this, let ℓ be between 1 and k. We consider the two probability distributions over the random permutations. The first is a uniform random permutation in which the ith candidate is the ℓth best among the first i candidates. The second is a uniform random permutation in which the kth highest candidate always appears in position i, and it is the ℓth best among the first i candidates. Observe that in both probability distributions π that can observe only the relative ranks of the first i candidates, observes a random permutation over the first i candidates (ranks), such that the ith candidate is the ℓ best so far. Therefore, the probability of the event that π selects the ith candidate given each one of the two probability distributions is exactly the same.

Next, we only need to calculate the probability that the candidate is ℓth best so far given that the candidate is k best globally. There are $k-1$ candidates that are better than the kth best candidate. The candidate appears ℓth best so far if exactly $\ell-1$ out of these candidates appear before position i. Since the positions are chosen uniformly at random, we can calculate the probability simply as follows. We choose $\ell-1$ positions among positions 1 to $i-1$ and $k-\ell$ positions among positions $i+1$ to n. Overall, there are $\binom{i-1}{\ell-1}\binom{n-i}{k-\ell}$ options to choose the positions for the $k-1$ best candidates such that the kth best candidate is ℓth best so far. The total number of options to choose positions for these $k-1$ candidates is $\binom{n-1}{k-1}$. Therefore, we get the desired bound.

3.2. Algorithms and linear programming solutions. In this section, we show that linear programming solutions to (LP–JK) exactly correspond to algorithms for the (J,K)-secretary problem, and vice versa. The following lemma states that the corresponding marginal probabilities of any algorithm for the (J,K)-secretary satisfy the requirements of (LP–JK).

Lemma 5 (Algorithm to LP Solution). *Let π be any algorithm for the (J,K)-secretary problem. Let $p_i^j(\pi)$ be the probability π selects the candidate at the ith position in the jth round. Let $q_i^{j\mid k}(\pi)$ be the probability π selects the candidate at the ith position in the jth round given that the candidate is the kth best among the i first candidates. Then, $(p(\pi), q(\pi))$ is a feasible solution to* (LP–JK) *and the expected number of the top K candidates π selects is at most the objective function of* (LP–JK).

Proof. First, any algorithm cannot increase its chances of selecting one of the K best candidates by selecting a candidate that is not one of the best K candidates so far. Thus, we may restrict our attention to such algorithms. Let us prove that the values $(p(\pi), q(\pi))$ satisfy the constraints of type (5). The probability that the candidate in the ith position is kth best among the first i candidates is $1/i$, we get that the variables satisfy the constraints of type (5).

We now show that constraints (7) hold. Consider the algorithm behavior at some position i and some choice j. We have that,

$$q_i^{j\mid k} = \Pr[\pi \text{ selects candidate } i \text{ as its } j\text{th choice} \mid \text{candidate } i \text{ is } k\text{th best so far}]$$
$$\leq \Pr[\pi \text{ selects exactly } j-1 \text{ candidates out of candidates } \{1,\ldots,i-1\}$$
$$\mid \text{candidate } i \text{ is } k\text{th best so far}] \tag{11}$$
$$= \Pr[\pi \text{ selects exactly } j-1 \text{ candidates out of candidates } \{1,\ldots,i-1\}] \tag{12}$$
$$= \sum_{\ell<i} p_\ell^{j-1}(\pi) - \sum_{\ell<i} p_\ell^{j}(\pi.) \tag{13}$$

Inequality (11) follows since whenever the algorithm selects candidate i as its jth choice, the algorithm must have selected exactly $j-1$ candidates out of the previous $i-1$ candidates. Equality (12) follows since the decisions made by the algorithm with respect to the first $i-1$ candidates depend only on the relative ranks of the first $i-1$ candidates, and is independent of the rank of the ith candidate. Finally, Equality (13) follows since the probability of selecting exactly $j-1$ candidates among candidates $1,\ldots,i-1$ equals the probability that the algorithm selects $j-1$ or more candidates among the first $i-1$ candidates minus the probability the algorithm selects j or more candidates among the first $i-1$ candidates.

Constraints (6) follow by the same arguments, where the probability that π selects exactly 0 candidates out of candidates $\{1, \ldots, i-1\}$ is $1-\sum_{\ell<i} p_\ell^1(\pi)$.

Finally, let us consider the objective function and prove that it upper bounds the performance of the algorithm. In terms of the probabilities $f_i^{j|k}$ and using the fact that the probability the k best candidate globally is in the ith position we get that the expected performance of the algorithm is

$$\max \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{J}\sum_{k=1}^{K} f_i^{j|k},$$

which from Lemma 4 equals the objective function.

Lemma 5 shows that the optimal solution to (LP–JK) is an upper bound on the performance of the algorithm. The following lemma shows that every solution to (LP–JK) corresponds to an algorithm that performs as well as the objective value of the solution.

Lemma 6 (LP Solution to Algorithm). *Let (p, q) be any feasible solution to* (LP–JK). *Then consider the algorithm π defined as follows. For each position $1 \le i \le n$,*

1. *if the candidate at the ith position is kth best so far for $1 \le k \le K$ and the algorithm has already selected $j < J$ candidates, select the candidate with probability*;
   - $q_i^{1|k}/(1-\sum_{\ell<i} p_\ell^1)$ *if the algorithm did not select any candidate*;
   - $q_i^{j|k}/(\sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^{j})$ *if the algorithm already selected $1 \le j < J$ candidates*;
2. *else reject the candidate.*

Then the expected number of top k candidates selected by π is exactly the objective function of (LP–JK).

Proof. First, notice that the algorithm is well defined since by the LP inequalities for any i, j, k, $q_i^{1|k}/(1-\sum_{\ell<i} p_\ell^1) \le 1$ and $q_i^{j|k}/(\sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^{j}) \le 1$. Also, by definition the algorithm always selects at most J candidates.

We prove by induction on the position i that the probability that the algorithm selects candidate at position i as its jth choice is exactly p_i^j. The base case is trivial. Assume this is true until $i-1$. At step i, we prove the claim for $1 < j \le J$. The proof for $j=1$ is similar. We claim that the probability we choose candidate i as the jth choice is

$$\sum_{k=1}^{\min\{i,K\}} \frac{1}{i}\left(\sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^{j}\right)\frac{q_i^{j|k}}{\sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^{j}} = \frac{1}{i}\sum_{k=1}^{\min\{i,K\}} q_i^{j|k} = p_i^j.$$

This is simply a multiplication of the probability of three events:

1. The probability that the algorithm chooses exactly j candidates out of candidates 1 to $i-1$.
2. The probability that the candidate i is kth best so far.
3. The probability that the algorithm selects candidate i as its jth choice given that events (1) and (2) happened.

It is easy to verify that by the properties of the algorithm and the input sequence, the first two events are independent. Next, we get that

$$\begin{aligned}
&\Pr[\pi \text{ selects candidate } i \text{ as its } j\text{th choice} \mid \text{candidate } i \text{ is } k\text{th best so far}]\\
&\quad = \Pr[\pi \text{ selects } i \text{ as its } j\text{th choice} \mid i \text{ is } k\text{th best so far and the algorithm chose exactly } j-1 \text{ candidates}]\\
&\qquad \cdot \Pr[\pi \text{ chose exactly } j-1 \text{ candidates so far} \mid \text{candidate } i \text{ is } k\text{th best so far}]\\
&\quad = \Pr[\pi \text{ selects } i \text{ as its } j\text{th choice} \mid i \text{ is } k\text{th best so far and the algorithm chose exactly } j-1 \text{ candidates}]\\
&\qquad \cdot \Pr[\pi \text{ chose exactly } j-1 \text{ candidates so far}]\\
&\quad = \frac{q_i^{j|k}}{\sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^{j}} \cdot \left(\sum_{\ell<i} p_\ell^{j-1} - \sum_{\ell<i} p_\ell^{j}\right) = q_i^{j|k}.
\end{aligned}$$

Thus, the probability π selects candidate i as its jth choice given that candidate i is k best so far is exactly $q_i^{j|k}$. By Lemma 4 we now get that the performance of the algorithm π is exactly the objective function of (LP–JK).

$$\text{(P21)}\quad \max \quad \frac{1}{n}\cdot\sum_{i=1}^{n} ip_i^1 + ip_i^2$$
$$\text{s.t.}\quad \forall\, 1\le i\le n \quad ip_i^1 \le 1-\sum_{j=1}^{i-1} p_j^1$$
$$\forall\, 1\le i\le n \quad ip_i^2 \le \sum_{j=1}^{i-1} p_j^1 - \sum_{j=1}^{i-1} p_j^2$$
$$\forall\, 1\le i\le n \quad p_i^1,\, p_i^2 \ge 0$$

$$\text{(D21)}\quad \min \quad \sum_{i=1}^{n} y_i$$
$$\text{s.t.}\quad \forall\, 1\le i\le n \quad \sum_{j=i+1}^{n} y_j + iy_i - \sum_{j=i+1}^{n} z_j \ge i/n$$
$$\forall\, 1\le i\le n \quad \sum_{j=i+1}^{n} z_j + iz_i \ge i/n$$
$$\forall\, 1\le i\le n \quad y_i,\, z_i \ge 0$$

FIGURE 3. Linear program and its dual for the $(2,1)$-secretary problem.

3.3. Analyzing special cases asymptotically. The linear formulation (LP–JK) enables easily to design the optimal algorithm for the (J, K)-secretary problem for any value of n. However, to devise upper bounds and algorithms for any value n one needs to provide an asymptotic solution to (LP–JK) and its dual. This seems a hard task. In this section we only provide such solutions to very few special cases. We provide an optimal algorithm for the $(1,2)$ and $(2,1)$-secretary problem. Observe that the $(1,1)$-secretary problem is the traditional secretary problem. The optimal solutions to these two variants turn out to be "threshold algorithms." Intuitively, for the $(2,1)$ case since the algorithm has two options for selecting the best candidate, it selects a candidate that is best so far even before the usual threshold of $1/e$. However, it never selects more than a single candidate at this early stage. For the $(1,2)$ case we have one choice for selecting either the best or the second best candidate. In this case the optimal solution is intuitively to select the best candidate from a certain threshold (which turns out to be slightly smaller than $1/e$). Then, from a later threshold the algorithm selects a candidate even if it is only the second best so far.

THEOREM 2. *There exists algorithms that achieve a performance of*
1. $1/e+1/e^{1.5}\simeq 0.591$ *for* $(2,1)$-*secretary problem*;
2. $\simeq 0.5736$ *for the* $(1,2)$ *secretary problem.*

Moreover all these algorithms are (*nearly*) *optimal.*

PROOF. We first give an algorithm and the corresponding primal solution to LP(J–K). The optimality is shown by exhibiting a dual solution of the same value:
$(2,1)$-*secretary*. Consider the following algorithm. Here τ_1 is defined as the smallest integer i for which

$$1-\sum_{j=i}^{n-1}\frac{1}{j}+\sum_{j=i}^{n}\frac{1}{j-1}\left(1-\sum_{k=j}^{n-1}\frac{1}{k}\right)\ge 0$$

and τ_2 be the smallest integer i such that

$$1-\sum_{j=i}^{n-1}\frac{1}{j}\ge 0.$$

Algorithm for $(2,1)$-secretary problem
- For each $1\le i\le n$, while two candidate have not been selected, do
 —If $\tau_1\le i<\tau_2$ and ith candidate is best so far and no other candidate has been selected then select the ith candidate.
 —If $\tau_2\le i\le n$ and ith candidate is best so far and at most one candidate has been selected, then select the ith candidate.

For any fixed n, we give the optimal primal and dual solutions. The primal solution exactly corresponds to the algorithm given in the figure above. We first simplify the primal linear program by eliminating the $q_i^{j|k}$ variables using the first set of constraints in (LP–JK). The simplified linear program and its dual is given in Figure 3.

Primal solution

$$p_i^1 = \begin{cases} 0 & 1 \le i < \tau_1 \\ (\tau_1 - 1)\left(\dfrac{1}{i-1} - \dfrac{1}{i}\right) & \tau_1 \le i \le n \end{cases}$$

$$p_i^2 = \begin{cases} 0 & 1 \le i < \tau_2 \\ (\tau_2 - 1)\left(\dfrac{1}{i-1} - \dfrac{1}{i}\right) - \dfrac{\tau_1 - 1}{i(i-1)}\left(1 - \displaystyle\sum_{j=\tau_2}^{i-1} \dfrac{1}{j-1}\right) & \tau_2 \le i \le n \end{cases}$$

Dual solution

$$z_i = \begin{cases} 0 & 1 \le i < \tau_2 \\ \dfrac{1}{n}\left(1 - \displaystyle\sum_{j=i}^{n-1} \dfrac{1}{j}\right) & \tau_2 \le i \le n \end{cases}$$

$$y_i = \begin{cases} 0 & 1 \le i < \tau_1 \\ \dfrac{1}{n}\left(1 - \displaystyle\sum_{j=i}^{n-1} \dfrac{1}{j}\right) + \dfrac{1}{n}\displaystyle\sum_{j=\tau_2}^{n} \dfrac{1}{j-1}\left(1 - \sum_{k=j}^{n-1} \dfrac{1}{k}\right) & \tau_1 \le i < \tau_2 \\ \dfrac{1}{n}\left(1 - \displaystyle\sum_{j=i}^{n-1} \dfrac{1}{j}\right) + \dfrac{1}{n}\displaystyle\sum_{j=i+1}^{n} \dfrac{1}{j-1}\left(1 - \sum_{k=j}^{n-1} \dfrac{1}{k}\right) & \tau_2 \le i \le n \end{cases}$$

A simple calculation shows that both primal and dual are feasible and have the same value of

$$\frac{1}{n}\left(\sum_{i=\tau_1}^{\tau_2 - 1} \frac{\tau_1 - 1}{i-1} + \sum_{i=\tau_2}^{n} \frac{\tau_2 - 1}{i-1} - (\tau_1 - 1)\sum_{i=\tau_2}^{n}\sum_{j=\tau_2}^{i-1} \frac{1}{(i-1)(j-1)}\right).$$

As $n \to \infty$, $\tau_1/n \to 1/e^{1.5}$, $\tau_2/n \to 1/e$ and the objective value is $1/e + 1/e^{1.5}$.

(1,2)-*secretary.* Consider the following algorithm. Here $\tau_2 = \lceil (2n)/3 \rceil$ and τ_1 is defined to be the smallest integer i such that

$$\frac{3\tau_2 - 2i - 4}{n-1} - 2\sum_{j=i}^{\tau_2} \frac{1}{i} \ge 0.$$

Algorithm for $(1, 2)$-secretary problem

- For each $1 \le i \le n$, while no candidate has been selected, do

  —If $\tau_1 \le i < \tau_2$ and ith candidate is best so far and no other candidate has been selected then select the ith candidate.

  —If $\tau_2 \le i \le n$ and ith candidate is among the best two candidates so far and no candidate has been selected, then select the ith candidate.

To show optimality of the algorithm, we first considered a simplified linear program for the problem in just the q variables. The primal LP and its dual appear in Figure 4.

$$
\begin{array}{lll}
\text{(P12)} & \max & \dfrac{1}{n}\cdot\sum_{i=1}^{n} q_i^{1|1}\left(1+\dfrac{n-i}{n-1}\right)+q_i^{1|2}\dfrac{i-1}{n-1}\\
 & \text{s.t.} & \forall\, 1\le i\le n \quad q_i^{1|1}+\sum_{l<i}\dfrac{1}{l}\left(q_l^{1|1}+q_l^{1|2}\right)\le 1\\
 & & \forall\, 1\le i\le n \quad q_i^{1|2}+\sum_{l<i}\dfrac{1}{l}\left(q_l^{1|1}+q_l^{1|2}\right)\le 1\\
 & & \forall\, 1\le i\le n \quad q_i^{1|1},\, q_i^{1|2}\ge 0
\end{array}
$$

$$
\begin{array}{lll}
\text{(D12)} & \min & \sum_{i=1}^{n} y_i+\sum_{i=1}^{n} z_i\\
 & \text{s.t.} & \forall\, 1\le i\le n \quad \dfrac{1}{i}\sum_{j=i+1}^{n}(y_j+z_j)+y_i\ge \dfrac{1}{n}+\dfrac{n-i}{n(n-1)}\\
 & & \forall\, 1\le i\le n \quad \dfrac{1}{i}\sum_{j=i+1}^{n}(y_j+z_j)+z_i\ge \dfrac{i-1}{n(n-1)}\\
 & & \forall\, 1\le i\le n \quad y_i,\, z_i\ge 0
\end{array}
$$

FIGURE 4. Linear program and its dual for the $(1,2)$-secretary problem.

Primal solution

$$
q_i^{1|1}=\begin{cases}0 & 1\le i<\tau_1\\[4pt] \dfrac{\tau_1-1}{i-1} & \tau_1\le i<\tau_2\\[4pt] \dfrac{(\tau_1-1)(\tau_2-2)}{(i-1)(i-2)} & \tau_2\le i\le n\end{cases}\qquad
q_i^{1|2}=\begin{cases}0 & 1\le i<\tau_2\\[4pt] \dfrac{(\tau_1-1)(\tau_2-2)}{(i-1)(i-2)} & \tau_2\le i\le n\end{cases}
$$

Dual solution

$$
y_i=\begin{cases}0 & 1\le i<\tau_1\\[4pt] \dfrac{3\tau_2-2i-4}{n(n-1)}-\dfrac{2}{n}\sum_{j=i}^{\tau_2-2}\dfrac{1}{i} & \tau_1\le i<\tau_2\\[4pt] \dfrac{i-1}{n(n-1)} & \tau_2\le i\le n\end{cases}
$$

$$
z_i=\begin{cases}0 & 1\le i<\tau_2\\[4pt] \dfrac{3i-2n+1}{n(n-1)} & \tau_2\le i\le n\end{cases}
$$

A simple but tedious calculation shows that both primal and dual are feasible and have the same objective value of

$$
\frac{2(\tau_1-1)}{n}\sum_{i=\tau_1}^{\tau_2-1}\frac{1}{i-1}+\frac{(\tau_1-1)(2n-3\tau_2+\tau_1+2)}{n(n-1)}.
$$

As $n\to\infty$, we have that $\tau_1/n\to -W(-2/(3e))\simeq 0.347$, $\tau_2/n\to 2/3$ and the objective is $-2W(-2/(3e))+W(-2/(3e))^2\simeq 0.5736$. Here $W(z)$ is the Lambert-W function defined to be the solution to equation $z=W(z)e^{W(z)}$.

4. Position independent algorithms. In this section we study position independent algorithms for the secretary problem. Recall that an algorithm is called position independent if it selects a candidate at every position with equal probability. We again use the linear programming framework to obtain the optimal algorithms and to show that no other algorithm can do better.

To formulate the linear programs we define a variable p that represents the probability that the algorithm selects the candidate in the ith position. Observe that this probability is independent of position i since we consider position independent algorithms. We also define variable f_i that represents the probability the algorithm selects the candidate in position i, given that the candidate in position i is the best candidate overall. The linear formulations that characterizes *all* position independent algorithms is (P1) and appears in Figure 5.

We also give two other linear programs that characterize regret-free algorithms and must-hire algorithms. An algorithm is regret free if it only selects candidates that are better ranked than all previous candidates. Although this requirement is trivial to satisfy for an arbitrary scenario without affecting the performance, the

(P1)	(P2)	(P3)
$\max \ \frac{1}{n}\sum_{i=1}^{n} f_i$	$\max \ \frac{1}{n}\sum_{i=1}^{n} f_i$	$\max \ \frac{1}{n}\sum_{i=1}^{n} f_i$
s.t. $p \le 1/n$	s.t. $p \le 1/n$	s.t. $p = 1/n$
$\forall i \quad f_i + (i-1)\cdot p \le 1$	$\forall i \quad f_i + (i-1)\cdot p \le 1$	$\forall i \quad f_i + (i-1)\cdot p \le 1$
$\forall i \quad f_i \le i\cdot p$	$\forall i \quad f_i = i\cdot p$	$\forall i \quad f_i \le i\cdot p$
$\forall i \quad p, f_i \ge 0$	$\forall i \quad p, f_i \ge 0$	$\forall i \quad p, f_i \ge 0$
(*Position independent*)	(*Regret free*)	(*Must-hire*)

FIGURE 5. (P1): Characterizes any position independent algorithm. (P2) characterizes algorithms that are regret free. (P3) characterizes algorithms that are must-hire algorithms.

optimal position independent algorithm is not regret free. Indeed the formulation (P2) characterizes all position independent algorithms that are also regret free. The second requirement is of must hire where we demand that an algorithm must always hire a candidate. Again this is trivial to satisfy in the basic secretary problem without affecting the performance by selecting the last candidate. But it is not trivial to do so for position independent algorithms, and (P3) characterizes algorithms that are must-hire position independent algorithms. This is formalized in the following two lemmas.

LEMMA 7 (ALGORITHM TO LP SOLUTION). *Let π be any algorithm for selecting the best candidate that is position independent. Let p^π denote the probability the algorithm selects a candidate at each position i, and let f_i^π be the probability the algorithm selects the candidate at position i given that the candidate at position i is the best candidate. Then,*

- *p^π, f_i^π is a feasible solution to the linear program* (P1);
- *if the algorithm is regret free then p^π, f_i^π is a feasible solution to the linear program* (P2) (*which are more restrict than linear program* (P1));
- *if the algorithm is must hire then p^π, f_i^π is a feasible solution to the linear program* (P3) (*which are more restrict than linear program* (P1));
- *the objective value $(1/n)\sum_{i=1}^{n} f_i^\pi$ is at least the probability of selecting the best candidate by π.*

PROOF. The proof follows the same lines as the proof of Lemma 1 and Lemma 5. The condition of position independence implies that $p_i^\pi = p_j^\pi = p^\pi$ for any two positions i and j. We show that p^π, f_i^π satisfy the constraints. First, $p^\pi \le 1/n$ since $\sum_{i=1}^{n} p_i^\pi = \sum_{i=1}^{n} p^\pi = np^\pi \le 1$. Also, if the algorithm is must hire then this holds with equality, so $p^\pi = 1/n$.

Note that by Lemma 4, for any i, f_i^π equals the probability of selecting a candidate at position i given that he is better than all previously arrived candidates. By this property the constraint $f_i^\pi = q_i^\pi \le 1-(i-1)p^\pi$, where the last inequality follows as a special case of constraint (6). Finally, the constraint $p_i^\pi \ge q_i^\pi/i = f_i^\pi/i$ follows as a special case of constraint (5). Moreover, if the algorithm is regret free, i.e., it only selects the best candidate so far, then the constraint holds with equality as given in $(LP2)$. Finally, the objective function $(1/n)\sum_{i=1}^{n} f_i^\pi$ is exactly the probability that π selects the best candidate.

Lemma 7 shows that the optimal solution to the linear formulations is an upper bound on the performance of the algorithm. To show the converse we define a family of algorithms that are defined by their probability of selecting a candidate at each position $0 \le p \le 1/n$ and we show that the set of feasible solutions to (P1) corresponds to the set of algorithms $\mathcal{A}_p$ defined here.

Position independent algorithm $\mathcal{A}_p$:

- Let $0 \le p \le 1/n$. For each $1 \le i \le n$, while no candidate is selected, do

  —If $1 \le i \le 1/(2p)$, select the ith candidate with probability $i/(1/p - i + 1)$ if she is the best candidate so far.

  —If $1/(2p) < i \le n$, let $r = i/(1/p - i + 1)$. Select the ith candidate with probability 1 if her rank is in top $\lfloor r \rfloor$ and with probability $r - \lfloor r \rfloor$ if her rank is $\lfloor r \rfloor + 1$.

The following lemma shows that every LP solution to (P1) corresponds to an algorithm that performs as well as the objective value of the solution.

LEMMA 8 (LP SOLUTION TO ALGORITHM). *Let p, f_i for $1 \le i \le n$ be a feasible LP solution to (P1). Then the algorithm $\mathcal{A}_p$ selects the best candidate with probability that is at least $(1/n)\sum_{i=1}^n f_i$.*

PROOF. For any p, the optimal values of f_i are given by the following: $f_i = ip$ for $1 \le i \le 1/(2p)$ and $f_i = 1-(i-1)p$ for $i > 1/(2p)$. These are exactly the values achieved by the algorithm $\mathcal{A}_p$ for any value p.

LEMMA 9. *The algorithm $\mathcal{A}_p$ is position independent for each $0 \le p \le 1/n$ and selects the best candidate with probability $1-(1/(4pn)+pn/2)$.*

PROOF. We prove by induction that the algorithm $\mathcal{A}_p$ selects each position i with probability p. It is easy to verify that for $i=1$ this is true. For $i>1$, the probability the algorithm chooses the candidate at position i is

$$r \cdot \frac{1}{i} \cdot (1-(i-1)p) = \frac{i}{1/p+i-1}\frac{1}{i}(1-(i-1)p) = \frac{1-(i-1)p}{1/p+i-1} = p.$$

The probability the algorithm selects the best candidate is related to f_i. We have $f_i = ip$ for $1 \le i \le 1/(2p)$, and $f_i = 1-(i-1)p$ for $1/(2p) < i \le n$. Thus, we get

$$\frac{1}{n}\sum_{i=1}^{n} f_i = \frac{1}{n}\left(\sum_{i=1}^{1/2p} ip + \sum_{i=1/2p+1}^{n}(1-(i-1)p)\right) = 1-\left(\frac{1}{4pn}+\frac{pn}{2.}\right)$$

Optimizing the linear programs (P1), (P2), and (P3) exactly, we get the following theorem. The optimality of the algorithms can also be shown by exhibiting an optimal dual solution.

THEOREM 3. *The family of algorithms $\mathcal{A}_p$ achieves the following*:
1. *Algorithm $\mathcal{A}_{1/\sqrt{2}n}$ is position independent with efficiency of* $1-1/\sqrt{2} \approx 0.29$.
2. *Algorithm $\mathcal{A}_{1/2n}$ is position independent and regret free with efficiency* $1/4$.
3. *Algorithm $\mathcal{A}_{1/n}$ is position independent and must hire with efficiency* $1/4$.

Moreover, all these algorithm are optimal for selecting the best candidate along with the additional properties.

5. Further discussion. Characterizing the set of algorithms in secretary type problems as a linear polytope possesses many advantages as we demonstrate by obtaining optimal algorithms for many variants of the secretary problem. In contrast to methods of factor revealing LPs in which linear programs are used to analyze a single algorithm, here we characterize *all* algorithms by a linear program. However, we do not have a systematic approach of obtaining a linear program for any variant of the secretary problem at hand, and it is unclear which variants can benefit from this approach.

One interesting direction for future research is trying to capture settings of a more combinatorial nature. One such example is the problem studied in Babaioff et al. [3] and Chakraborty and Lachish [10] in which elements of a matroid arrive one by one. This problem seems extremely appealing since matroid constraints are exactly captured by a linear program. Very recently, our techniques were shown to be useful in this regard (Gharan and Vondrák [15]) and other variants (Kumar et al. [22]).

Acknowledgments. The first author is supported by ISF [Grant 954/11] and BSF [Grant 2010426].

Appendix. Omitted proofs

Proof of Observation 1

PROOF. We start with the first part. Let $\mathcal{A}$ be a monotone algorithm for the (K, K)-secretary problem that is c-competitive. It directly implies an algorithm for the cardinal version that ignores the values and only looks at the relative rankings of the values. Consider the performance of this algorithm. Since it selects at most K elements, it always outputs a feasible solution. Let $v_1 \ge v_2, \ldots, \ge v_n$ denote the valuations of the elements in a nonincreasing order. Thus, $OPT = \sum_{i=1}^{K} v_i$. Let $\alpha_1, \ldots, \alpha_n$ denote the probability that $\mathcal{A}$ selects each of the elements. Let $E[\mathcal{A}]$ denote the expected value of the solution returned by the algorithm.

Since the algorithm is c-competitive for (K, K)-secretary problem we have that

$$\frac{1}{K}\sum_{i=1}^{K}\alpha_i \ge c.$$

Also by the monotonicity property of the algorithm, we get that

$$\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_k.$$

We conclude that

$$E[\mathcal{A}] = \sum_{i=1}^{n} v_i \cdot \alpha_i \geq \sum_{i=1}^{K} v_i \cdot \alpha_i$$
$$\geq \left(\sum_{i=1}^{K} v_i\right)\left(\frac{1}{K}\sum_{i=1}^{K} \alpha_i\right)$$
$$\geq c \cdot \sum_{i=1}^{K} v_i = c \cdot OPT, \tag{.14}$$

where inequality (.14) follows from the Chebyshev's sum inequality.

For the second part of the claim, let $\mathcal{A}$ be an oblivious algorithm for the cardinal secretary problem. Since $\mathcal{A}$ is an oblivious, for any set of values the probability it selects the ith highest element is always the same. Moreover, for any set of values the input of the algorithm is a random permutation. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ denote these probabilities. Now, $\mathcal{A}$ implies an algorithm for the (K, K)-secretary problem where we assign arbitrary values as items arrive such that values respect the given linear ordering. The performance of the algorithm for the (K, K)-secretary problem is exactly

$$\frac{\sum_{i=1}^{K} \alpha_i}{K}.$$

To lower bound this value, we consider the execution of the cardinal algorithm with the following values for the items. For $1 \leq i \leq K$, $v_i = 1 - \epsilon i$ and for $K+1 \leq i \leq n$, $v_i = \epsilon \cdot (n+1-i)$. That is, for small ϵ, the first K elements are approximately 1 and the last $n-K$ elements are approximately zero. On this input the expected performance of the algorithm is

$$E[\mathcal{A}] = \sum_{i=1}^{n} v_i \cdot \alpha_i = \sum_{i=1}^{K} \alpha_i - O(n^2\epsilon) \geq c \cdot K - O(n^2\epsilon),$$

where the inequalities follow since v_i are approximately 1 for $1 \leq i \leq K$ and at most $n\epsilon$ for $i > K$. Picking ϵ to be arbitrary small, we get that $(1/K)\sum_{i=1}^{K} \alpha_i \geq c$ as required.

References

[1] Awerbuch B, Azar Y, Meyerson A (2003) Reducing truth-telling online mechanisms to online optimization. *Proc. ACM Sympos. Theory Comput.* (ACM, New York), 503–510.

[2] Babaioff M, Immorlica N, Kempe D, Kleinberg R (2008) Online auctions and generalized secretary problems. *SIGecom Exchange* 7:1–11.

[3] Babaioff M, Immorlica N, Kleinberg R (2007a) Matroids, secretary problems, and online mechanisms. Bansal N, Pruhs K, Stein C, eds. *Proc. Eighteenth Annual ACM-SIAM Sympos. Discrete Algorithms SODA '07* (SIAM, Philadelphia), 434–443.

[4] Babaioff M, Immorlica N, Kempe D, Kleinberg R (2007b) A knapsack secretary problem with applications. *Proc. 10th Internat. Workshop on Approximation and the 11th Internat. Workshop on Randomization, Combin. Optim. Algorithms and Techniques APPROX '07/RANDOM '07*, 16–28.

[5] Babaioff M, Dinitz M, Gupta A, Immorlica N, Talwar K (2009) Secretary problems: Weights and discounts. *SODA '09: Proc. Nineteenth Annual ACM-SIAM Sympos. Discrete Algorithms*, 1245–1254.

[6] Borodin A, El-Yaniv R (1998) *Online Computation and Competitive Analysis* (Cambridge University Press, Cambridge, UK).

[7] Buchbinder N, Jain K, Naor J(S) (2007) Online primal-dual algorithms for maximizing ad-auctions revenue. *Proc. 15th Annual Eur. Sympos.*, 253–264.

[8] Buchbinder N, Jain K, Singh M (2010a) Incentives in online auctions via linear programming. Saberi A, ed. *Proc. 6th Internat. Conf. Internet Network Econom.* (Springer-Verlag, Berlin), 106–117.

[9] Buchbinder N, Jain K, Singh M (2010b) Secretary problems via linear programming. *Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, Vol. 6080 (Springer-Verlag, Berlin), 163–176.

[10] Chakraborty S, Lachish O (2012) Improved competitive ratio for the matroid secretary problem. *Proc. 23rd Ann. ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 1702–1712.

[11] Dynkin EB (1963) The optimum choice of the instant for stopping a Markov process. *Sov. Math. Dokl.* 4:627–629.

[12] Ferguson TS (1989) Who solved the secretary problem? *Statist. Sci.* 4:282–289.

[13] Freeman PR (1983) The secretary problem and its extensions: A review. *Internat. Statist. Rev.* 51:189–206.

[14] Gardner M (1960) Mathematical games. *Scientific Amer.* (February):150–153.

[15] Gharan SO, Vondrák J (2011) On variants of the matroid secretary problem. Demetrescu C, Halldórsson M, eds. *Proc. 19th Eur. Conf. Algorithms* (Springer-Verlag, Berlin, Heidelberg), 335–346.

[16] Goemans M, Kleinberg J (1998) An improved approximation ratio for the minimum latency problem. *Math. Programming* 82:111–124.

[17] Hajiaghayi MT, Kleinberg R, Parkes DC (2004) Adaptive limited-supply online auctions. Breese J, Feigenbaum J, Seltzer M, eds. *Proc. 5th ACM Conf. Electronic Commerce* EC '04 (ACM, New York), 71–80.

[18] Immorlica N, Kalai AT, Lucier B, Moitra A, Postlewaite A, Tennenholtz M (2011) Dueling algorithms. Fortnow L, Vadhan S, eds. *Proc. ACM Sympos. Theory Comput.* (ACM, New York), 215–224.

[19] Jain K, Mahdian M, Markakis E, Saberi A, Vazirani VV (2003) Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *J. ACM* 50(6):795–824.

[20] Kleinberg R (2005) A multiple-choice secretary algorithm with applications to online auctions. *Proc. Sixteenth Annual ACM-SIAM Sympos. Discrete Algorithms SODA '05* (SIAM, Philadelphia), 630–631.

[21] Korula N, Pál M (2009) Algorithms for secretary problems on graphs and hypergraphs. *Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 5556 (Springer-Verlag, Berlin), 508–520.

[22] Kumar R, Lattanzi S, Vassilvitskii S, Vattani A (2011) Hiring a secretary from a poset. Shoham Y, Chen Y, Roughgarden T, eds. *ACM Conf. Electronic Commerce* (ACM, New York), 39–48.

[23] Lavi R, Nisan N (2004) Competitive analysis of incentive compatible on-line auctions. *Theor. Comput. Sci.* 310(1–3):159–180.

[24] Lindley DV (1961) Dynamic programming and decision theory. *J. Royal Statist. Soc. Ser. C (Appl. Statist.)* 10(1):39–51.

[25] Mehta A, Saberi A, Vazirani U, Vazirani V (2007) Adwords and generalized online matching. *J. ACM* 54(5):Article 22.

[26] Samuels SM (1991) Secretary problems. *Handbook of Sequential Anal.* 118:381–405.