Online Auctions and Generalized Secretary Problems

MOSHE BABAIOFF
Micosoft Research, Silicon Valley
and
NICOLE IMMORLICA
Centrum voor Wiskunde en Informatica (CWI) and Northwestern University
and
DAVID KEMPE
University of Southern California
and
ROBERT KLEINBERG
Cornell University

We present generalized secretary problems as a framework for online auctions. Elements, such as potential employees or customers, arrive one by one online. After observing the value derived from an element, but without knowing the values of future elements, the algorithm has to make an irrevocable decision whether to retain the element as part of a solution, or reject it. The way in which the secretary framework differs from traditional online algorithms is that the elements arrive in *uniformly random* order.

Many natural online auction scenarios can be cast as generalized secretary problems, by imposing natural restrictions on the feasible sets. For many such settings, we present surprisingly strong constant factor guarantees on the expected value of solutions obtained by online algorithms. The framework is also easily augmented to take into account time-discounted revenue and incentive compatibility. We give an overview of recent results and future research directions.

Categories and Subject Descriptors: J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Markov Processes*; F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Online Algorithms, Secretary Problems, Auctions, Knapsack, Matroids

1. INTRODUCTION

Suppose you decide to sell an item — say, your car — using the following procedure. People come to your garage one at a time, appraise your car, and make bids to buy it from you. Each person is allowed to submit only one bid, and after receiving an offer, you must make an immediate decision whether to accept or reject it. What is the best strategy for deciding which bids to accept or reject? This question has different answers under different modeling assumptions. Do we assume that the bids constitute a worst-case sequence, a random sequence, or something else? If the bids are a random sequence, does the seller have prior information about the bid values? Is there a correlation between the ordering of the bids and their values? How do we model the bidders' strategic behavior, if at all?

At one extreme of the modeling spectrum is the traditional approach in online algorithms [Borodin and El-Yaniv 1998] of assuming a *worst-case* input. This view leads to a very pessimistic bound: no online algorithm (randomized or deterministic) for selecting one bid from a sequence of n bids can achieve better than an $O(1/n)$ fraction of the optimum revenue[1]. At the other extreme is a sequence of bids independently and identically distributed (i.i.d.) according to a *known* distribution. In that case, the optimum strategy can be computed in polynomial time using dynamic programming.

Since the first extreme leads to unnecessarily pessimistic bounds, and the second assumes too much knowledge on the part of the seller, it is natural to investigate models in between the two extremes. Two natural candidate models suggest themselves. (1) The bids are i.i.d. from an *unknown* distribution. (2) An adversary gets to decide on the values of the bids, but not on the *order* in which they are presented to the seller. The former model is akin to approaches often taken in machine learning and AI. In fact, it is subsumed by the second model, since the sequence can be determined by first picking a multi-set of values from the (unknown) distribution, and then permuting them randomly.

Returning to your goal of selling your car, we can now state the problem formally as follows:

> Given a sequence of n numbers in random order[2], design an online algorithm for picking one element of the sequence, to maximize the expected value of the chosen element.

Stated this way, the problem is very similar to the famous *Secretary Problem*: designing an algorithm for picking one element of a randomly ordered sequence, to maximize the *probability of picking the maximum element* of the entire sequence. In fact, although the two problems have different objectives (optimizing the expected value of the chosen element versus optimizing the probability of choosing the maximum element), it is clear that a solution to the secretary problem immediately yields an algorithm for choosing one bid from a randomly-ordered bid sequence.

---

[1]If the n bids are a worst-case sequence, then for some $K \gg n$, the bid sequence could be $1, K, K^2, K^3, \ldots, K^m, 1, 1, \ldots, 1$, where the value of m is chosen to be the worst possible for the algorithm under consideration. In this case, no algorithm (even if it is randomized) can achieve more than an $O(1/n)$ fraction of the optimum revenue: any algorithm is simply too likely to stop before or after the maximum element is presented, since it has no way of knowing when the geometrically-increasing sequence of bids will end until it is too late.

[2]i.e., a sequence sampled from any distribution invariant under reordering the n elements of the sequence

The preceding example shows that secretary problems and, more generally, optimal stopping problems are closely related to online auctions, and can lead to surprisingly strong guarantees on the expected value selected. The rest of this article surveys a variety of recent results and open questions in this area, which arise naturally from thinking about generalizations and variations of the single-item auction application presented at the start of the article.

(1) Many auction scenarios involve the sale of more than one item. For instance, the seller could have k identical items to be sold in an online fashion to the bidders. In a further generalization, the bidders may require more than one of the items. This scenario naturally models auctions for ad impressions on a web site, or for human or machine resources. It gives the auction the flavor of an *Online Knapsack Problem.*

(2) The interactions between the bidders' desired item sets can also take on more complex forms. Natural examples include airline travelers seeking to purchase seats on one of multiple possible itineraries, or moviegoers seeking to purchase tickets for one of multiple movies or showings. As we describe in more detail below, these and several other scenarios are naturally cast as a *matroid structure* constraining the sets of bids which can be jointly accepted.

(3) In auction scenarios, earlier revenue or sales are often more valuable than later ones. Given the time-dependent nature of the problem, it is natural to study the generalization to time-discounted revenue, and the impact it has on the expected value an online auction can generate.

(4) It is well known that bidders might misrepresent their true valuation of an item (or other parameters) if it allows them to purchase the item more cheaply. Designing *truthful auctions*, in which self-interested bidders have no incentive to lie, is an important area of auction design. The question of incentive alignment is equally valid and important in the context of online auctions.

Beginning with the original secretary problem and a formal outline of our general framework, we explore each of the four generalizations discussed above. Most of the proofs will be omitted or sketched, and can be found in the cited literature.

2. THE SECRETARY PROBLEM

The problem of designing an online algorithm to optimize the probability of selecting the maximum element in a randomly-ordered sequence is traditionally known as the *Secretary Problem*, due to the following motivating application. Imagine that you manage a company, and you want to hire a secretary from a pool of n applicants. You are very keen on hiring only the best and brightest. Unfortunately, you cannot tell how good a secretary is until you interview him[3], and you must decide whether or not to make an offer at the time of the interview. As your decision is irrevocable, you will have to hire a secretary *before* observing the quality of applicants that have not yet been interviewed. What sort of guarantee can you give regarding the probability of hiring the best secretary?

For some intuition, let us consider the special case of just three applicants. You can easily guarantee that you hire the best secretary with probability $1/3$: simply hire a secretary at

---

[3]We use male pronouns throughout this paper for simplicity. No assumption on the genders of actual agents is intended.

random. Can you do better? In fact, you can. The idea is to interview the secretaries in a random order and use the quality of the first secretary to set the bar for the one you hire. That is, if the three applicants are Alice, Bob, and Charles, then

(1) Choose an applicant at random, say Alice, and interview her but do not hire her. Let her quality be v_A.
(2) Choose one of the remaining applicants, say Bob, at random, and observe his quality v_B. If Bob is better than Alice, i.e., $v_B \geq v_A$, hire Bob. Otherwise, hire the third applicant, Charles.

With probability $\frac{1}{3}$, you happen to interview the second-best applicant first, in which case you definitely hire the best applicant. Additionally, with probability $\frac{1}{6}$, you interview the worst applicant first and the best applicant second, in which case you also hire the best applicant. Thus, this strategy has probability $\frac{1}{2}$ of hiring the best applicant! Lindley [1961] and Dynkin [1963] proved that a generalization of this strategy to a setting with n applicants yields a probability approaching $1/e \approx 0.37$ of hiring the best secretary, and that this is the best possible guarantee.[4] The optimal strategy is appealingly simple: interview the first $m \sim n/e$ applicants without making any hiring decisions, then hire the next applicant whose quality exceeds the best of the first m. If you reach the end of the sequence without hiring anyone, then hire the last applicant no matter what.

Because the secretary algorithm selects the maximum element in a randomly-ordered sequence with probability at least $1/e$, the same algorithm can also be used in a single-item auction with randomly-ordered bids, to select a bid whose expected value is at least $1/e$ times the maximum bid. This factor $1/e$ is optimal for both the original secretary problem and the online single-item auction with randomly ordered bids, although its optimality in the latter case does not follow easily from its optimality in the former case.

3.   A UNIFIED FRAMEWORK

All of the problems we consider can be described in the following common framework that extends the "online single-item auction" interpretation of the secretary problem. A ground set $\mathcal{U}$ of so-called *elements* corresponds to the bidders, and a collection of subsets $\mathcal{I} \subseteq 2^{\mathcal{U}}$ (closed under containment) describes the sets of bidders whose bids can be simultaneously accepted. (These are also called *feasible sets*.) For example, in the case of a single-item auction, $\mathcal{I}$ is the collection of sets with at most one element. In the case of the ad impression auction, $\mathcal{I}$ is the collection of all sets S whose total requested impressions sum up to at most the total available impressions.

Each element $x \in \mathcal{U}$ has a non-negative *value* v_x. We wish to design online algorithms in which the structure of $\mathcal{U}$ and $\mathcal{I}$ is known at the outset, while the elements and their values are revealed one at a time in random order. As each element is presented, the algorithm must decide whether to select or reject it, subject to the following constraints:

(1) The decision to select or reject is irreversible.
(2) The decision must be made before the next element in the sequence is presented.
(3) The set of selected elements must belong to $\mathcal{I}$ at all times.

---

[4]As is the case with most simple and beautiful mathematical statements, this problem has been stated and solved multiple times throughout history. See Ferguson's survey [Ferguson 1989] for further details.

The algorithm's payoff is, in general, a function of the input sequence and the algorithm's choices. We will mostly be discussing the case in which the payoff obtained by the algorithm when selecting a set S is the sum of the values of its elements, $v(S) = \sum_{x \in S} v_x$. (Alternative definitions of payoff have been considered in the literature, including the time-discounted secretary problem discussed later in this article.) With $S^* \in \mathcal{I}$ denoting the set maximizing $v(S^*)$, we say that an algorithm is α-*competitive* if $\mathbf{E}\left[v(S)\right] \geq 1/\alpha \cdot v(S^*)$ for all possible valuation functions v. The expectation is taken over all orderings (permutations) and possible random choices of the algorithm. Together, these determine the set S.

We will refer to this general class of problems as *Generalized Secretary Problems*. There is an important difference between these problems and the original secretary problem. In the secretary problem itself, the objective (hiring the best applicant) is an *ordinal* criterion, i.e., it depends only on the relative quality of the applicants and not on any numerical values ascribed to them. In the extensions that we consider here, the objective is defined in terms of the numerical values v_x.

The best-known solutions to date for most of the extensions discussed above involve a simple algorithmic idea that is already foreshadowed in the secretary algorithm described above. The algorithms wait for some prescribed time t and observe the best-valued feasible subset $S \subseteq \{1, \ldots, t\}$. They then select additional elements if they "improve" (in some sense) upon the imaginary solution S. The intuitive interpretation is that the algorithms try to "get a feel for the market" before committing to any decisions. As such, they also bear similarity with pricing algorithms which use a subset of the bidders to determine prices for the others (see, e.g., [Goldberg et al. 2001]).

4.   THE K-CHOICE SECRETARY PROBLEM

When we consider selling k identical items instead of just one, we are led to the following extension of the secretary problem: design an online algorithm for picking k out of n nonnegative numbers presented in random order, to maximize their expected sum. In terms of the formal framework described above, this corresponds to the case in which $\mathcal{I}$ consists of all subsets of $\mathcal{U}$ with k or fewer elements.

When $k = 1$, we have seen that the optimal competitive ratio is e. How does the optimal competitive ratio vary with k, and what does the optimal algorithm look like? Perhaps the most natural algorithm to consider is the following generalization of Dynkin's algorithm:

(1) Observe the first $\lfloor n/e \rfloor$ elements.

(2) Remember the best k elements among these first $\lfloor n/e \rfloor$, and call this set T.[5]

(3) Whenever an element arrives whose value is greater than the minimum-value element in T, select this element and delete the minimum-value element from T.

Babaioff et al. [2007] show, via a complex counting argument, that this algorithm has competitive ratio no worse than e for all values of k, not just for $k = 1$. A slightly modified algorithm yields the same competitive ratio, with a much simpler proof. However, the constant e is far from optimal as $k \to \infty$. It was shown by Kleinberg [2005] that the optimal competitive ratio is bounded from above by $1 + C/\sqrt{k}$ and from below by $1 + c/\sqrt{k}$

---

[5]If $k > \lfloor n/e \rfloor$, then let T consist of the first $\lfloor n/e \rfloor$ elements observed, together with $k - \lfloor n/e \rfloor$ "dummy elements" of zero value.

for some pair of constants $c < C$. The upper bound is obtained using a recursive algorithm. It partitions the sequence into an initial segment and final segment of approximately equal length, recursively chooses at most $k/2$ elements from the initial segment, and sets a threshold value equal to the $(k/2)^{\text{th}}$ largest element of the initial segment. Subsequently, it chooses all elements of the final segment that meet this threshold until exhausting its k allotted choices.

5. THE KNAPSACK SECRETARY PROBLEM

Suppose you are running the official Super Bowl web site, which has a slot for a banner ad. You know that on Super Bowl day, the site will get some number of impressions (say, 50 million), and you would like to sell the ad slot for these impressions to advertisers. Each advertiser arrives with a request for some number of impressions and the price he is willing to pay. You must decide whether to accept the request, but of course, the total number of impressions promised to different advertisers cannot exceed 50 million. If all advertisers arrive simultaneously, the problem of determining which requests to accept (to maximize revenue) is an instance of the knapsack problem. If the advertisers arrive online in a random order instead, we will call this problem the *Knapsack Secretary Problem*. In terms of the formalism described above, each element $x \in \mathcal{U}$ has a weight $w_x > 0$ (in our Super Bowl advertising example, w_x is the fraction of the 50 million impressions requested by x), and $\mathcal{I}$ consists of the sets whose total weight does not exceed 1.

It is well known that the knapsack problem is NP-complete, and that it admits an FPTAS as well as a simple 2-approximation. The online knapsack problem, on the other hand, is inapproximable to within any non-trivial multiplicative factor. Several papers have investigated online and stochastic knapsack problems with varying assumptions [Kleywegt and Papastavrou 1998; Dean et al. 2004; Lueker 1995; Marchetti-Spaccamela and Vercellis 1995; Buchbinder and Naor 2005]. The Knapsack Secretary Problem is unique among these in that the set of cost/value pairs is *arbitrary*. Rather than making assumptions about the distributions of these pairs, we circumvent the inapproximability of the online knapsack problem by assuming that the elements arrive in a random order.

Under this assumption, we are able to adapt the 2-approximation algorithm to our setting to yield an algorithm with constant competitive ratio for the Knapsack Secretary Problem. As usual, the algorithm observes a constant fraction of requests without accepting any of them. It ranks the requests by their *value density* v_x/w_x and sets a threshold based on the value densities of the sampled requests. After the initial segment of the request sequence is observed, the algorithm accepts elements based on a criterion which primarily depends on whether their value density exceeds the threshold. However, the algorithm must be enhanced with some "tweaks" to accommodate requests whose weight is so large that they require a constant fraction of the knapsack's capacity. For full details we refer the reader to [Babaioff et al. 2007]. The algorithm presented there is $(10e)$-competitive, but the optimal competitive ratio for the knapsack secretary problem is still unknown.

6. THE MATROID SECRETARY PROBLEM

Every generalized secretary problem that we have described so far in this article admits a constant-competitive algorithm. Unfortunately, this is not always the case. Babaioff et al. [2007] exhibit a generalized secretary problem for which no algorithm has competitive ratio better than $\Omega(\log n / \log \log n)$. Besides the problems discussed earlier, what other

broad classes of generalized secretary problems admit algorithms with constant competitive ratio? One promising possibility is the *Matroid Secretary Problem* [Babaioff et al. 2007], in which the ground set $\mathcal{U}$ and the feasible subsets $\mathcal{I}$ constitute a matroid[6] structure. Many natural online auction scenarios fall into the framework of Matroid Secretary Problems.

(1) The k-choice secretary problem gives rise to the *uniform matroid*, in which the independent sets are exactly all sets of cardinality at most k.

(2) The movie ticket auction mentioned in Example 2 above can be formalized as follows. Given n customers with values v_x and m possible movies, as well as a bipartite graph indicating which movies each customer would be interested in. Under known seating capacities for each movie, find a subset of customers of maximum total value who can all see a movie of their choice simultaneously. The resulting matroid, in which all matchable sets of customers are independent, is called a *transversal matroid*.

(3) The airline travel setting mentioned in Example 2 above can be formalized as follows: each traveler, with value v_x, seeks a path from a source to a destination, and each edge has a known limited capacity. The goal is to route a set of passengers of maximum total value, without violating capacity constraints. So long as all sources are identical, the resulting structure is known to be a matroid, called a *gammoid*.

(4) A well-known class of matroids with less natural motivation in the auction context consists of *graphic matroids*, in which customers correspond to edges of a graph, and the goal is to select an acyclic set of maximum total value.

Matroids relate to secretary problems in a very natural way. For consider what happens if we relax Constraint 1 in the definition of the Generalized Secretary Problem, by stipulating that the decision to reject an element remains irreversible, but a decision to select an element may later be revoked. (Constraints 2 and 3 are still enforced. In particular, the algorithm is not allowed to accept an infeasible set and later reject some of its elements.) For the original secretary problem, there is now an obvious algorithm which always selects the best element: whenever a new element has a value exceeding all previous values, reject the current selection (if any) and select the new element. For any set system $(\mathcal{U}, \mathcal{I})$, one can naturally extend this greedy algorithm: whenever it is possible to improve the value of the currently selected set by picking a new element and possibly discarding a subset of the previous selections, do so. If there is choice of which subset to discard, choose the one with minimum total value. This greedy algorithm is guaranteed to select the maximum-value set $S \in \mathcal{I}$ *precisely* when $(\mathcal{U}, \mathcal{I})$ is a matroid.

This intuition motivates an appealing conjecture, namely that for some constant α (perhaps even $\alpha = e$), every Matroid Secretary Problem admits an α-competitive algorithm. This conjecture is tantamount to stating that the obligation to honor past commitments is not too costly, i.e., it only reduces the expected value of the chosen subset by a factor of α. The conjecture is currently open. The best positive result for general matroids to date is by Babaioff et al. [2007] who give an $O(\log k)$-competitive algorithm for Matroid Secretary Problems where k is the *rank* of the matroid, the maximum size of a feasible set.

---

[6]Matroids [Oxley 1992] are combinatorial constructs generalizing the notion of independence in vector spaces. A set system $(\mathcal{U}, \mathcal{I})$ with $\mathcal{I} \subseteq 2^{\mathcal{U}}$ is a matroid if $\mathcal{I}$ is nonempty and closed under containment, and it satisfies the *exchange axiom*: for all sets $A, B \in \mathcal{I}$ with $|B| > |A|$, there is an element $b \in B \setminus A$ such that $A \cup \{b\} \in \mathcal{I}$. The sets in $\mathcal{I}$ are called *independent sets*.

The matroid algorithm too consists of a sampling phase followed by a selection phase. In the sampling phase, the algorithm observes half of the elements and sets $\bar{v}$ to be the maximum value observed. After choosing a scale parameter j uniformly at random from $\{0, 1, 2, \ldots, \log_2 k\}$, the algorithm in the selection phase greedily selects all elements of value at least $\bar{v}/2^j$, so long as the addition does not violate feasibility.[7] The intuition for the competitive ratio is as follows. Let $v_1 \geq \ldots \geq v_k$ be the values of elements in the optimum solution. For each of these elements v_i, there is an $\Omega(1/\log k)$ probability that the selection threshold $\bar{v}/2^j$ lies just below v_i. In that case, the reordering assumption and matroid exchange property show that in expectation, at least $\Omega(i)$ elements of value at least v_i are selected by the algorithm. The bound now follows from linearity of expectation (with a separate argument required if only v_1 contributes significantly to the optimum).

While the conjecture of a constant-competitive online algorithm for all Matroid Secretary Problems remains open, it has been resolved positively for several special classes of matroids. Babaioff et al. [2007] give constant-competitive algorithms for several classes of matroids, including graphic matroids and transversal matroids of bounded degree (corresponding to the assumption that each moviegoer is only interested in a constant number of movies). A subsequent paper by Dimitrov and Plaxton [2008] gives a constant-competitive algorithm for general transversal matroids without the bounded-degree restriction. It is also known that the class of matroids admitting a constant-competitive secretary algorithm is closed under truncation, i.e., the operation of imposing a cardinality constraint on the sets in $\mathcal{I}$ [Babaioff et al. 2007].

7. TIME DISCOUNTING

Our discussion so far has treated the value of an element as independent of when it was selected. In many intertemporal choice scenarios, there are reasons why this assumption could be overly simplistic. For instance, in the auction setting (such as our initial example of selling a car), the seller could reinvest the revenue and obtain more interest from selling early. These considerations motivate discounting the value v_x by a time-dependent factor d_t at time t: the value of a set S is thus $\sum_{x \in S} v_x \cdot d_{\pi(x)}$, where $\pi(x)$ denotes the arrival time of x in the random ordering of bids. We assume here that the discount factors d_t are known beforehand to the algorithm designer; they are not revealed online.

Even the original secretary problem (i.e., choosing a single element from a sequence of n elements) becomes significantly more difficult with arbitrary discount factors. For instance, Babaioff et al. [2008] prove that no online algorithm for the resulting *Discounted Secretary Problem* can achieve a better competitive ratio than $\Omega(\frac{\log n}{\log \log n})$. In fact, even knowing the entire unordered set $\{v_x \mid x \in \mathcal{U}\}$ of values in advance, an online algorithm cannot be better than $\sqrt{2}$-competitive. On the positive side, [Babaioff et al. 2008] presents an $O(\log n)$-competitive online algorithm for the Discounted Secretary Problem, nearly matching the lower bound. They also prove that merely knowing the expected maximal benefit in advance (as is implied, for example, by knowing the unordered set of values) leads to dramatic improvements. The algorithm picking the first element whose benefit is at least half of the expected maximal benefit is a simple 4-competitive algorithm. Extensions of the Discounted Secretary Problem to various matroid domains are also discussed in [Babaioff et al. 2008]. Additional positive results have been obtained

---

[7]As described, the algorithm requires knowledge of the rank of the matroid, k. If the rank is unknown, the algorithm may estimate it in the sampling phase with just a constant-factor loss in the competitive ratio.

for the Discounted Secretary Problem with specific "well-behaved" discount functions like $d(t) = \beta^t$ [Rasmussen and Pliska 7576] or $d(t) = \sum_{t=1}^{n} \beta^t$ [Mahdian et al. 2008], as well as in a continuous-time model in which the values of the elements are independently and identically distributed and element arrivals are given by some renewal stochastic process [Gershkov and Moldovanu 2007].

8. INCENTIVE COMPATIBILITY

Our motivating examples have been drawn from various online auction scenarios. Yet we have so far ignored incentive issues: the problem that the bidders may not reveal their values or other parameters correctly if a misrepresentation stands to benefit them. Fortunately, all of the algorithms described so far are *value-monotonic*: each time a new element x is about to be observed, there is a predetermined threshold $u(x)$ (possibly infinite) such that x will be selected if and only if $v_x \geq u(x)$. Value-monotonicity directly implies that the algorithms are truthful, so long as bidders can only manipulate their declared value, and not — for example — their arrival time. To convert the algorithm into a truthful mechanism, one simply charges each bidder x a price of $u(x)$ if he is selected, and 0 otherwise.

The situation becomes notably more complex if the agents also have some control over their arrival times. For example, a person shopping for airline tickets may choose not to purchase them at the earliest possible time, instead preferring to wait until a later date in hopes of a price reduction. To model agents' ability to strategize about their arrival time, we assume that each agent has an arrival time $a(x)$ and departure time $d(x)$, such that the agents are randomly ordered by arrival time. The agent's *reported* arrival and departure times $\hat{a}(x), \hat{d}(x)$ may differ from the actual arrival and departure times. If agent x is selected during the time interval $[a(x), d(x)]$, he receives a value of v_x; otherwise, he receives no value for being selected. The agent's ability to manipulate his reported arrival and departure times is constrained by his true arrival and departure times; some common assumptions are *no early arrivals* ($\hat{a}(x) \geq a(x)$), *no late departures* ($\hat{d}(x) \leq d(x)$), or both. These assumptions are justified if the agent's presence can be directly verified, or (in the case of "no early arrivals") if we think of $a(x)$ as the time at which the agent first becomes aware of the existence of the auction or of his own desire to participate in it.

Some of the algorithms discussed earlier in this article can be transformed into truthful online mechanisms in the no-early-arrivals model, with little or no loss in the competitive ratio. Specifically, a variant of Dynkin's algorithm is truthful and e-competitive in the no-early-arrivals model [Hajiaghayi et al. 2004], and Kleinberg's $(1 + O(k^{-1/2}))$-competitive algorithm for the k-choice secretary problem [Kleinberg 2005] can be transformed into a truthful and $(1 + O(k^{-1/2}))$-competitive mechanism for the k-choice secretary problem in the no-early-arrivals model. The basic idea for both of these results is to modify the original algorithm so that whenever it sets a new threshold price, it scans through the list of agents who have not yet departed and selects those whose value exceeds the new threshold, even if they arrived much earlier. At the departure time of any agent who was selected as a winner of the auction, the agent is charged a price equal to the minimum threshold price attained during the agent's reported arrival-departure interval, even if this price is lower than the threshold price at the time the agent was selected as a winner.

9. CONCLUSIONS

The theory of online algorithms for resource allocation has been successful in dealing with many combinatorially complex optimization problems, but often suffers from overly pessimistic bounds based on pathological worst-case examples. Optimal stopping theory (including the study of secretary problems) avoids these pathologies by engaging in average-case analysis of online algorithms on random inputs, but until recently, the theory has primarily been limited to combinatorially simple problems such as picking any k elements in a sequence of n numbers. There is a clear opportunity to bring together ideas from these two areas and reap the benefits of both.

The resulting model of worst-case inputs presented in a random order delicately traces a boundary between the pessimistic bounds of online algorithms and the strong assumptions of i.i.d. sequences. Intuitively, it captures the fact that the adversary can choose the input, but not its precise presentation. In this respect, it is similar to the celebrated results on smoothed analysis (e.g., [Spielman and Teng 2004]), where worst-case inputs are perturbed by small amounts of random noise, and lead to significantly improved practical performance. Similarly, very simple online algorithms often achieve surprisingly strong bounds in the Secretary framework. Indeed, a similar phenomenon has been observed for random reordering of inputs to online facility location [Meyerson 2001].

In this article, we have sketched some initial successes of the Generalized Secretary research program and have illustrated their applications to online mechanism design. The field is replete with open questions, some inspired primarily by theoretical considerations (such as the matroid secretary conjecture presented earlier), others directly motivated by the application to auctions. The latter include multi-unit auctions, combinatorial auctions, and markets in which hard supply constraints are replaced by variable production costs.

REFERENCES

Babaioff, M., Dinitz, M., Gupta, A., Immorlica, N., and Talwar, K. 2008. Secretary problems: Weights and discounts.

Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. 2007. A knapsack secretary problem with applications. In *Proc. 10th Intl. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2007)*.

Babaioff, M., Immorlica, N., and Kleinberg, R. 2007. Matroids, secretary problems, and online mechanisms. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*.

Borodin, A. and El-Yaniv, R. 1998. *Online Computation and Competitive Analysis*. Cambridge University Press.

Buchbinder, N. and Naor, J. 2005. Online primal-dual algorithms for covering and packing problems. In *Proc. 13th European Symp. on Algorithms (ESA 2005)*.

Dean, B., Goemans, M., and Vondrák, J. 2004. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Proc. 45th IEEE Symp. on Foundations of Computer Science (FOCS 2004)*. 208–217.

Dimitrov, N. and Plaxton, C. G. 2008. Competitive weighted matching in transversal matroids. In *Proc. 35th Intl. Colloq. on Automata, Languages and Programming (ICALP 2008)*.

Dynkin, E. B. 1963. The optimum choice of the instant for stopping a markov process. *Sov. Math. Dokl. 4*.

Ferguson, T. S. 1989. Who solved the secretary problem? *Statist. Sci. 4, 3*, 282–289.

Gershkov, A. and Moldovanu, B. 2007. The dynamic assignment of heterogenous objects: A mechanism design approach. Tech. rep., University of Bonn.

Goldberg, A., Hartline, J., and Wright, A. 2001. Competitive auctions and digital goods. In *Proc. 12th ACM-SIAM Symp. on Discrete Algorithms (SODA 2001)*. 735–744.

Hajiaghayi, M. T., Kleinberg, R., and Parkes, D. 2004. Adaptive limited-supply online auctions. In *Proc. 5th ACM conference on Electronic commerce*. ACM Press, 71–80.

KLEINBERG, R. 2005. A multiple-choice secretary problem with applications to online auctions. In *Proc. 16th ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*. 630–631.

KLEYWEGT, A. J. AND PAPASTAVROU, J. D. 1998. The dynamic and stochastic knapsack problem. *Operations Research 46,* 1, 17–35.

LINDLEY, D. V. 1961. Dynamic programming and decision theory. *Applied Statistics 10*, 39–51.

LUEKER, G. 1995. Average-case analysis of off-line and on-line knapsack problems. In *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms (SODA 1995)*. 179–188.

MAHDIAN, M., MCAFFEE, P., AND PENNOCK, D. 2008. The secretary problem with durable employment. personal communication.

MARCHETTI-SPACCAMELA, A. AND VERCELLIS, C. 1995. Stochastic on-line knapsack problems. *Mathematical Programming 68*, 73–104.

MEYERSON, A. 2001. Online facility location. In *IEEE Symposium on Foundations of Computer Science (FOCS) 2001*. 426–431.

OXLEY, J. G. 1992. *Matroid Theory*. Oxford University Press.

RASMUSSEN, W. T. AND PLISKA, S. R. 1975/76. Choosing the maximum from a sequence with a discount function. *Appl. Math. Optim. 2,* 3, 279–289.

SPIELMAN, D. AND TENG, S.-H. 2004. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM 51*, 385–463.