# CSC2421 Topics in Algorithms: Online and Other Myopic Algorithms
# Fall 2019

Allan Borodin

November 13, 2019

# Announcements

- Today, in the second half of the lecture, I will discuss online algorithms with stochastic inputs.
- In the first half, Kayman will present his reading project on online algorithm with advice
- Next week, Caroline will present her reading project on papers relating to chapter 6, namely non monotone submodular maximization and max-sat.
- No class on Wednesday, November 27
- Final class on Wed, December 4. Greg will present his project on dynamic algorithms.

# Beyond worst case

So far in the course, our discussions have assumed an adversarial input model. For each of the problems and computational models we have considered, once we have formulated both a precise definition of an input item, and a set of all possible input items, an adversary is free to choose a worst case sequence of input items. There is growing interest within theoretical computer science to take a less adversarial approach so that the theory becomes a more accurate reflection of what we experience "in practice". There are essentially two ways to go *beyond worst case analysis*:

- Restrict the set of possible input items to a set of items that better reflect real applications. We have seen this approach, for example, when we considered restricted classes of graphs in Chapter 5 of the text. However, this is still done within an adversarial input model.
- Assume that the input items are coming from some known or unknown input distribution. This is, of course, a (or even the most) common practice in both experimental and theoretical aspects of science. Note: We considered stochastic inputs when using Yao's Principle for proving negative results for randomized algorithms.

# The secretary problem and the ROM model

We will begin a discussion of some results in what is commonly called *stochastic optimization* as well studied within the field of operations research. This discussion will be continued in later chapters of the text, especially when we consider problems directly related to online advertising.

We start our discussion of stochastic input models with the random order model (ROM) which was arguably first formally articulated with regard to the famous *secretary problem*. The secretary problem has an interesting history as discussed in the Historical Notes for this chapter. Although often stated in terms of choosing the best secretary, the basic problem obviously applies to choosing the winning candidate for any position. The secretary problem is also the starting point for the study of *optimal stopping rules* and for more general variants of the secretary problem (e.g., selecting a set of candidates subject to some constraint).

# The secretary problem

The problem is essentially the online bipartite matching problem when there are $n$ online nodes and one offline node. The $i^{th}$ online node (i.e., the $i^{th}$ candidate) arrives with a value $v_i$ and the goal is to choose the candidate with the highest value.

If the values and order of arrivals is chosen adversarially, then it is nxiot hard to see that no deterministic or randomized algorithm can have a constant probability of choosing the winning candidate (having the highest value). When the values are arbitrary this is equivalent to saying that there is no constant competitive ratio.

Note that this problem is equivalent to what we called the time-series problem in chapter 2. But there we analyzed the problem when say we are given lower $L$ and upper $U$ bounds on the possible values.

# The random order model

To get beyond the impossibility result for arbitrary values, we change the input model and consider the *random order model (ROM)*. In this model, an adversary selects a *set* $S = \{v_1, \ldots, v_n\}$ of $n$ values, and then the input sequence is $\{v_{\pi(1)}, \ldots, v_{\pi(n)}\}$ for a permutation $\pi : [1, n] \to [1, n]$ chosen uniformly at random. The goal now is to maximize the probability (wrt the the random choice of $\pi$) of choosing the best candidate (equivalently, maximizing the expected competitive ratio).

For stochastic inputs (such as the ROM model) when the distribution is essentially unknown, a standard idea is to sample some initial sequence of input items to "learn" information about the distribution. The secretary algorithm (next slide) is sampling an initial sequence to gain information about the maximum value in the input sequence.

# The secretary algorithm

---

**Algorithm 33** The SECRETARY ALGORITHM

---

```
procedure SECRETARY
    v_best ← v_1
    i ← 2
    r ← n/e
    while i ≤ r do            ▷ Find the best candidate amongst the first r = n/e candidates
        if v_i > v_best then
            v_best = i
    i ← i + 1
    while i ≤ n do            ▷ Output next candidate (if any) better than initial best
        if v_i > v_best then
            Halt and return i
```

---

**Theorem**

*As $n \to \infty$, the secretary algorithm selects the best candidate with probability $\frac{1}{e}$.*

# Proof of the secretary theorem

It is interesting to ask why an initial "sampling" at $r = c \cdot n$ inputs with $0 < c < 1$ will result in a constant probability of success.

Consider the following simple (but limited) combinatorial argument for a modified secretary algorithm where we use $r = n/2$ rather than $n/e$ for the initial sampling. A *sufficient condition* for this modified algorithm to output the best candidate is that the second best candidate must occur in the first half and the best candidate must occur in the second half of the input sequence. The probability for this to happen is $\frac{n/2}{n} \cdot \frac{n/2}{n-1} \approx \frac{1}{4}$.

But, of course, this is not a *necesary condition*. It could, for example, be that the third best candidate is in the first half of the input sequence and the best and second best are both in the second half of the input sequence with the best candidate preceding the second best. Trying to enumerate all the cases or just estimating probabilities for any given initial sampling of $r = c \cdot n$ inputs becomes a combinatorial nightmare.

# Proof of the secretary theorem continued

We will now see why choosing $r = n/e$ is the right place to stop sampling. We are interested in an asymptotic result as $n \to \infty$. We can assume that all candidate values are distinct. In the ROM model, for all $i$, the maximum value will occur as the $i^{th}$ input with probability $\frac{1}{n}$. Since we are interested in an asymptotic constant result as $t \to \infty$, we can ignore the possibility that the maximum value occurs as the 1st item. The following is then a *necessary and sufficient condition* for the algorithm to choose the correct candidate: :

- The best applicant $[1, t]$ is the same as the best applicant in $[1, r]$ for $t \geq r$.
- The best applicant occurs in position $t + 1$.

These conditions are met with probability $\frac{r}{t} \cdot \frac{1}{n}$.
Let $E_r$ be the event that the algorithm outputs the best candidate for a given $r$. It follows then that the probability of outputting the best candidate for a given $r$ is $\Pr(E_r) = \frac{r}{n} \sum_{t=r}^{n-1} \frac{1}{t}$.

# Continuous analysis to avoid the combinatorial nightmare

Now here is where we use the continuous analysis. Letting $x = \lim_{n\to\infty} \frac{r}{n}$ and $y = \lim_{n\to\infty} \frac{t}{n}$, we have;

$$Prob(E_r) = \lim_{n\to\infty} \frac{r}{n} \sum_{t=r}^{n-1} \frac{n}{t} \frac{1}{n} = x \int_x^1 \frac{1}{y} dy = x \ln x$$

To obtain the optimal choice of $r$, we differentiate and set $Prob'(E_r) = -\ln x - 1 = 0$ so that $x = \frac{1}{e}$ and therefore $Prob(E_{(\frac{n}{e})}) = \frac{1}{e}$. This is of course only an optimal choice of $r$ in the limit but for sufficiently large $n$, it is a very good approximation.

We have left open the question as to whether or not there is a different type of algorithm (yielding a better probability of success) other than one than chooses a "stopping rule" $r$ as in the Secretary algorithm.

# Optimality of the secretary algorithm

Consider any algorithm for the Secretary problem. The algorithm must at some point output its choice for the best candidate. That is after seeing (but not acting on) some $r$ candidates, it eventually outputs candidate $r+1$. Cleary $v_j > v_i$ for any $i \leq r$ otherwise the probability of success is zero. Combined with the analysis of the secretary algorithm, we can conclude that the secretary algorithm is asymptotically optimal. That is, we have an oprtimal *stopping rule*.

**Corollary**

*The scretary algorithm with $r = n/e$ provides the asymptiotically best approximation for the Secretary prooblem.*

# Extensions of the basic secretary problem

We again note that the secretary problem is a special case of edge weighted online bipartite matching when there is only one offline vertex. It is natural to consider the edge weighted bipartite matching problem in the ROM model. We know that we cannot do better than the $\frac{1}{e}$ approximation ratio, so the question is whether or not we can achieve this ratio. The following algorithm achieves the optimal ROM bound.

**Algorithm 34** The EDGE-WEIGHTED BIPARTITE MATCHING ALGORITHM

```
procedure WEIGHTED MATCHING
    ▷ V is the set of offline vertices
    ▷ Online vertices u_1, ..., u_n arrive according to the ROM model
    U' ← {u_1, ..., u_⌊n/e⌋}                 ▷ U' is the current set of online vertices
    M ← ∅                                    ▷ M will be the constructed matching
    ℓ ← ⌈n/e⌉
    while ℓ ≤ n do
        U' ← U' ∪ {u_ℓ}
        M^(ℓ) ← optimal matching on edge weighted graph with online vertices U' and offline
vertices V
        if (ℓ, r) ∈ M^(ℓ) and r not yet matched then
            M ← M ∪ {(ℓ, r)}
        ℓ ← ℓ + 1
```

# Statement and proof of the ROM edge weighted bipartite matching algorithm

**Theorem**

*The Algorithm (due to Kesselheim et al [2013]) has expected approximation ratio $\frac{1}{e}$ in the ROM model. More spefically*

$$\mathbb{E}[w(M)] \geq (\frac{1}{e} - \frac{1}{n}) \cdot OPT$$

The proof relies on the following lemma for estimating the expected contribution of each online node $u_\ell$ for $\ell \geq \lceil n/e \rceil$.

**Lemma**

*Let $A_\ell$ denote the contribution (i.e. the weight added to the solution) of online vertex $u_\ell \in U$ for $\lceil n/e \rceil \leq \ell \leq n$. Then $\mathbb{E}[A_\ell] \geq \frac{\lfloor n/e \rfloor}{\ell - 1} \cdot \frac{OPT}{n}$.*

# Proof of the lemma for the edge weighted matching algorithm

It is helpful to view the random order of online vertices so that $u_\ell$ is chosen uniformly at random from $U'$. Then, conditioned on $r$ being unmatched thus far,
(1) the expected weight $\mathbb{E}[(u_\ell, r)]$ of the edge $(u_\ell, r)$ is $\frac{w(M^{(\ell)})}{\ell}$
where $w(M^\ell)$ is the weight of the optimal matching on the current set of vertices.
Furthermore, $U'$ is a uniformly at random set of size $\ell$ chosen from $U$ so that
(2) $\mathbb{E}[w(M^{(\ell)}] \geq \frac{\ell}{n} OPT$. It follows that
$\mathbb{E}[w(u_\ell, r)] \geq \frac{OPT}{n}$
The expectation of the above inequality is in terms of the random choice of $U'$ and the choice of $\ell$ as the last arrival in $U'$.

# Proof of the lemma for the edge weighted matching algorithm continued

We now need to consider the randomness in the preceding $\ell - 1$ arrivals to determine the probability that the intended match $r$ for $u_\ell$ was not already matched. Using the same view that the last element in any initial input sequence is being chosen randomly from the initial set of inputs and independent of the order of the previous elements, the probability that $r$ is not chosen in the $k^{th}$ iteration (for $k = \ell - 1, \ell - 2, \ldots 1$) is $\frac{k-1}{k}$.

Hence $Prob(r$ is unmatched) when $u_\ell$ arrives is equal to
$\prod_{k=\lceil n/e \rceil}^{\ell-1} \frac{k-1}{k} = \frac{\lceil n/e \rceil - 1}{\ell - 1}$
s Summarizing, the expected contribution of the $\ell^t h$ online vertex is
$\mathbb{E}[A_\ell] = \mathbb{E}[(u_\ell, r) | r$ is not yet matched$] \cdot Prob(r$ is not yet matched).
Namely $\mathbb{E}[A_\ell] \geq \frac{\lfloor n/e \rfloor}{\ell - 1} \cdot \frac{OPT}{n}$ as claimed.

# Completing the proof of the ROM matching algorithm

The bound on the competitive ratio is obtained by summing up the individual contributions $\mathbb{E}[A_\ell]$. That is,

$$\mathbb{E}[w(M)] = \mathbb{E}[\sum_{\ell=1}^{n} A(\ell) \geq \sum_{\lceil n/e \rceil}^{n} \frac{\lfloor n/e \rfloor}{\ell - 1} \cdot \frac{OPT}{n} = \frac{\lfloor n/e \rfloor}{n} \sum_{\lfloor n/e \rfloor}^{n-1} \frac{OPT}{\ell}$$

This can be simplified to yield the desired bound on $\mathbb{E}[w(M)]$ by observing that
$\frac{\lfloor n/e \rfloor}{n} \geq (\frac{1}{e} - \frac{1}{n})$ and $\sum_{\lfloor n/e \rfloor}^{n-1} \frac{1}{\ell} \geq \ln(\frac{n}{\lfloor n/e \rfloor}) \geq 1$

# Other extensions of the secretary problem

The matching algorithm is selecting a set of winners; that is, a subset of the online nodes. Of course, the nodes chosen are subject to a condition, namely the selected set *can be matched* to the offline side. This condition is the defintion of a *traversal matroid*. Note that this is different than saying we are selecting a matching as in a set of edges, which is the intersection of two matroids.

We want to discuss the general problem of selecting (within the ROM model) a set of elements subject to a matroid condition. Which means we have to define the concept of a matroid and the resulting generalized seccrtary problem. There are many equivalent defintions but we will use what is probably the most often stated defintion.

# Matroids

Let $U$ be a set of elements and $\mathcal{I}$ be a collection of subsets of $U$. $(U, \mathcal{I})$ is a matroid if the following hold:

- (Hereditary property) If $I \in \mathcal{I}$ and $I' \subset I$, then $I' \in \mathcal{I}$.
- (Exchange property) If $I', I \in \mathcal{I}$ and $|I'| < |I|$, then $\exists u \in I \setminus I'$ such that $I' \cup \{u\} \in \mathcal{I}$.

An *hereditary set system* $(U, \mathcal{I})$ is any set system satisfying the hereditary property so that a matroid is an hereditary set system that also satisfies the exchange property.

The sets $I \in \mathcal{I}$ are referred to as the *independent sets*. As we noted, there are alternative equivalent definitions. In particular, an alternative to the exchange property is that every maximal independent set has the same size, and this maximum size is call the *rank* of the matroid.

# Matroid secretary problems

Let $(U, \mathcal{I})$ be a matroid and $w : U \to \mathbb{R}^{\geq 0}$. The matroid secretary problem is to choose an independent set $I \subseteq U$ in the matroid so as to maximize $\sum_{u \in I} w_u$.

As an immediate consequence of the ROM bipartitie matching result, we obtain a constant competitive ratio $\frac{1}{e}$ for three secretary problems, namely choosing a set of candidates so as to maximize the sum of the element weights for the following matroid constraints.

- A uniform matroid; that is, where the independent sets $I$ have cardinality at most $k$ for some fixed $k$. Here it is immediate to see that the rank of such a matroid is the cardinality constraint $k$.
- A partition matroid; that is, there is a partition $(U_1, \ldots, U_m)$ (for some $m$) of the universe $U$ and for each $U_j$ there is a capacity $k_j$; the independent sets $I$ are those satisfying $|u \in U_j \cap I| \leq k_j$. Clearly, every uniform matroid is a partition matroid.
- A transversal matroid; Every partition matroid is a transversal matroid.

# The general matroid secretary problem

Of course, since these particular matroid constraint problems are special cases of the bipartite matching problem it may be possible to obtain better constant approximation ratios.

Indeed this is the case for the uniform matroid with cardinality constraint $k$ for which there is a $\frac{1}{(1-\Omega(\sqrt{1/k})}$ approximation. Hence for uniform matroids the approximation ratio limits to 1 as $k$ increases. (Here I am still using a fraction to indicate the competitive ratio to be consistent with the original secretary problem.

It is an open problem whether or not there is a constant approximation for all matroid constraints. Currently, the best known approximation guarantee for an arbitrary matroid constraint is $\frac{1}{\Omega(\log \log k)}$ for matroids of rank $k$.

# The Ranking Algorithm in the ROM Model

While the Secretary problem and algorithm seem to be the first explicit use of the ROM model, it has become an important input model following the introduction of the Ranking algorithm for the unweighted bipartite maximimum matching (BMM) problem. The Ranking algorithm can be viewed as a randomized algorithm in the adversarial input model or alternatively as a deterministic algorithm in the ROM input model. This duality in perspectives of the Ranking algorithm can be seen as follows.

Consider an arbitary (i.e. adversarialy created) priority order of the offline nodes. Let Fixed Rank refer to the deterministic matching of each online node to the highest (if any) priority available offline node. Fixed Rank has as asymptotically optimal approximation ratio of $\frac{1}{2}$. Considering Ranking as a deterministic ROM algorithm, we think of the offline vertices being ordered arbitarily while nature creates a random order for the online nodes. Fixed Rank then becomes a deterministic $1 - \frac{1}{e}$ approximatiom in the ROM input model. The analysis is exactly the same.

# The Ranking Algorithm in the ROM Model continued

The same duality in perspective holds for the vertex weighted case. That is, we can think of the offline vertices being unweighted and ordered arbitrarily and the online vertices ariving in random order, each online vertex having a weight.

While Ranking is the optimal randomized algorithm in the adversarial online model, it is an open problem if the dual interpretation as the Fixed Ranking ROM algorithm is optimal amongst all deterministic ROM algorithms for BMM. However, using randomization for both the offline and online vertices does lead to an improved approximation ratio.

**Theorem**

*Consider the Ranking algorithm as a randomized algorithm (i.e., the offline nodes are randomly ordered) in the ROM input model for the unweighetd maximum bipartite matching problem. The expected approximation ratio* $\rho(\textit{Ranking})$ *satisfies the following bounds:* $.696 \leq \rho(\textit{Ranking}) \leq .727$

# Stochastic input models

Although most of complexity theory as well as the analysis of algorithms (as taught in undergraduate CS courses) is based on worst case complexity, a more traditional approach was to analyze algorithms assumming that inputs are derived from some distribution. Of course, trying to define the appropriate distribution is very application specific. The reality of many current applications is that large amounts of data is making it more realistic to be able to understand how data is distributed.

It is interesting that an initial argument for competitive analysis (i.e. a worst case perspective) was that stochastic analysis (especially if in terms of some naive distribution) did not model real applications. With the emergence of large amounts of data it now becomes more plausible that the data can be utilized to better understand the possible distribution that underlies a given application. Of course, it is a much more general approach if we can just assume that there is an input distribution without assuming a specific distribution.

# What is the measure of performance when the inputs are being generated by a distribution

In offline or online computation, we need an analogue for the approximation ratio (respectively, the competitive ratio). Since we are now discussing a maximization problem, lets stay with the fractional way to express the ratio.

Let $\mathcal{I}$ be a set (or sequence) of inputs and let $\mathcal{D}$ be a distribution, then the accepted definition for the (distributional) competitive ratio wrt $\mathcal{D}$ is $liminf_{\mathbb{E}[OPT(\mathcal{I})]\to\infty} \frac{\mathbb{E}[ALG(\mathcal{I})}{\mathbb{E}[OPT(\mathcal{I})]}$
Here the expectation is with respect to the distribution. For an online algorithm, we would have a sequence of distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ where the $i^{th}$ input $x_i$ is drawn from $\mathcal{D}_i$.

We can refer to this ratio as the *distributional competitive ratio* noting that the distributions are chosen by an adversary. In the context of stochastic inputs it is implied that the competitive ratio means the distributional competitive ratio.

# Online computation and the i.i.d. model

With regard to online computation, there has been a substantial amount of interest in the i.i.d. *independent identically distributed* input model. In this model, each input item is independentally drawn in sequence from some underlying distribution. The distribution $\mathcal{D}$ may be known to the algorithm (the *known i.i.d. model*) or unknown (*the unknown i.i.d. model*). In either the known or unknown i.i.d. model, it is usually assumed that independent samples from $\mathcal{D}$ can be obtained. The generality in this model is that $\mathcal{D}$ can be an arbitrary distribution.

Of course, there is major assumption of independence whereas in many applications input items are correlated. In particular, in paging, *locality of reference* induces an obvious correlation between consecutive input items in the input sequence. It might still be argued that the i.i.d. model is applicable in some applications and in any case can provide useful and additional insight into the "real" performance of an algorithm, arguably better than what can be obtained from worst case performance.
Of course, a worst case guarantee is an absolute guarantee that protects against rare but still possible and potentially problematic instances.

# ROM results imply i.i.d. results

The following result might be termed an observation; however, it is a very basic and useful observation that we will label as a Theorem.

**Theorem**

*Consider any problem $\mathcal{P}$ for which the ROM model and i.i.d. models are applicable. Suppose there is an algorithm $\mathcal{A}$ such that $\mathcal{A}$ obtains expected competitive ratio $c$ in the ROM model. Then $\mathcal{A}$ achieves expected competitive ratio at least $c$ in the unknown i.i.d. model. This of course, implies that $\mathcal{A}$ obtains at least that ratio in the known i.i.d. model.*

The proof is remarkably simple given how useful is this fact. Consider the algorithm on problem instances (i.e. multi sets) consisting on $n$ input items. Partition the input instances into classes, each of size $n!$ such that the class is made up of the $n!$ ways to permute some set of input items. Each input sequence in a class occurs with the same probability. Thus each class becomes an instance of the random order model and hence algorithm $\mathcal{A}$ has competitive ratio at least $c$ on each class. We can then take the expectation over the different clsses to obtain the desired result.

# Bipartite matching in the i.i.d. model

For the unweighted bipartite maximum matching (BMM) we just observed that Ranking ca be viewed as a deterministic algorithm in the ROM model that achieves expected competitive ratio $1 - \frac{1}{e}$. It then follows immediately that Fixed Rank achieves the same $1 - \frac{1}{e}$ ratio in the unknown (and hence also known) i.i.d. model for the BMM problem. It is currently unknown if there is a better deterministic approximation for the BMM problem in the ROM model. However, there are significantly better approximations for both the unweighted (and even edge weighted) bipartite matching problems in the known i.i.d. model.

For the edge weighted case, the i.i.d. approximations are therefore much better than what can be achieved in the ROM model since in the ROM model we know that the ratio $\frac{1}{e}$ is asymptotical the best we can do.

For the unweighted (or offline vertex weighted) case the known results are better than what is known for the ROM model so far, we do not know what is the best possible ROM approximation nor do we know what is the best i.i.d. approximation.

# The known i.i.d. model for the BMM problem

In the known i.i.d. model, an adversary first chooses a *type graph* $G = (L, R, E)$ and a distribution $p : L \to [0, 1]$ on the LHS nodes. In this case, the nodes in $L$ are also referred to as types. The type graph together with the distribution is given to the algorithm in advance.

In the known i.i.d. model, an actual input instance $\hat{G} = (\hat{L}, R, \hat{E})$ is a random variable and is generated from $G$ as follows. The right hand side $R$ is the same in $G$ and $\hat{G}$, but the left-hand-side of $\hat{G}$ consists of $m$ i.i.d. samples from $p$. Thus, say a given node $\hat{\ell} \in \hat{L}$ has type $\ell \in L$, then the neighbors of $\hat{\ell}$ in $\hat{G}$ are the same as the neighbors of $\ell$ in $G$. The graph $\hat{G}$ is presented to the algorithm in the vertex arrival model (the order of vertices is the same as the order in which they were generated).

Note that a particular type $\ell$ can be absent altogether or can be repeated a number of times in $\hat{G}$. We refer to $\hat{G}$ as the *instance graph*.

# Known i.i.d. distributions with integral types

A known i.i.d. problem is said to have *integral types* if the expected number of times a particular type occurs is integral. We will denote the number of times type $\ell$ occurs in an instance by the random variable $Z_\ell$. Then the condition of integral types is that $\mathbb{E}[Z_\ell] = p(\ell) \cdot m \in \mathbb{Z}$.

While the parameters $|L|, |R|$, and $m$ can all be different, the most common setting is $m = |L|$. This assumption together with integral types implies that without loss of generality one can take $p$ to be the uniform distribution on $L$ (by duplicating types as necessary). An additional common assumption is that $|L| = |R|$. In that case we talk about a single parameter $n = |L| = |R| = m$.

The first algorithm to beat the $1 - 1/e$ barrier in the known i.i.d. model is due to Feldman et al. [2009]. Their algorithm achieved a competitive ratio of .73. We rushed the explanation of Feldman et al so we will start the next lecture with this first i.i.d. result for the BMM problem.