Combinatorial Secretary Problems with Ordinal Information[*][†]

Martin Hoefer[1] and Bojana Kodric[2]

1   **Institute of Computer Science, Goethe-University Frankfurt am Main, Frankfurt, Germany**
      `mhoefer@cs.uni-frankfurt.de`
2   **MPI for Informatics and Saarland University, Saarland Informatics Campus, Saarbrücken, Germany**
      `bojana@mpi-inf.mpg.de`

―――― **Abstract** ――――

The secretary problem is a classic model for online decision making. Recently, combinatorial extensions such as matroid or matching secretary problems have become an important tool to study algorithmic problems in dynamic markets. Here the decision maker must know the numerical value of each arriving element, which can be a demanding informational assumption. In this paper, we initiate the study of combinatorial secretary problems with *ordinal information*, in which the decision maker only needs to be aware of a preference order consistent with the values of arrived elements. The goal is to design online algorithms with small competitive ratios.

For a variety of combinatorial problems, such as bipartite matching, general packing LPs, and independent set with bounded local independence number, we design new algorithms that obtain constant competitive ratios.

For the matroid secretary problem, we observe that many existing algorithms for special matroid structures maintain their competitive ratios even in the ordinal model. In these cases, the restriction to ordinal information does not represent any additional obstacle. Moreover, we show that ordinal variants of the submodular matroid secretary problems can be solved using algorithms for the linear versions by extending [18]. In contrast, we provide a lower bound of $\Omega(\sqrt{n}/(\log n))$ for algorithms that are oblivious to the matroid structure, where n is the total number of elements. This contrasts an upper bound of $O(\log n)$ in the cardinal model, and it shows that the technique of thresholding is not sufficient for good algorithms in the ordinal model.

1   Introduction

The secretary problem is a classic approach to model online decision making under uncertain input. The interpretation is that a firm needs to hire a secretary. There are n candidates and they arrive sequentially in random order for an interview. Following an interview, the firm learns the value of the candidate, and it has to make an immediate decision about hiring him

―――――――――――――

[*] A full version is available at `http://arxiv.org/abs/1702.01290`.

before seeing the next candidate(s). If the candidate is hired, the process is over. Otherwise, a rejected candidate cannot be hired at a later point in time. The optimal algorithm is a simple greedy rule that rejects all candidates in an initial learning phase. In the following acceptance phase, it hires the first candidate that is the best among all the ones seen so far. It manages to hire the best candidate with optimal probability $1/e$. Notably, it only needs to know if a candidate is the best seen so far, but no exact numerical values.

Since its introduction [15], the secretary problem has attracted a huge amount of research interest. Recently, a variety of combinatorial extensions have been studied in the literature [7] capturing a variety of fundamental online allocation problems in networks and markets, such as network design [28], resource allocation [25], medium access in networks [22], or competitive admission processes [12]. Prominently, in the matroid secretary problem [8], the elements of a weighted matroid arrive in uniform random order (e.g., weighted edges of an undirected graph G). The goal is to select a max-weight independent set of the matroid (e.g., a max-weight forest of G). The popular *matroid secretary conjecture* claims that for all matroids, there exists an algorithm with a constant *competitive ratio*, i.e., the expected total weight of the solution computed by the algorithm is at least a constant fraction of the total weight of the optimum solution. Despite much progress on special cases, the conjecture remains open. Beyond matroids, online algorithms for a variety of combinatorial secretary problems with downward-closed structure have recently been studied (e.g., matching [28, 25], independent set [22], linear packing problems [26] or submodular versions [18]).

The best known algorithms for matroid or matching secretary problems rely heavily on knowing the exact weight structure of elements. They either compute max-weight solutions to guide the admission process or rely on advanced bucketing techniques to group elements based on their weight. For a decision maker, in many applications it can be quite difficult to determine an exact cardinal preference for each of the incoming candidates. In contrast, in the original problem, the optimal algorithm only needs *ordinal information* about the candidates. This property provides a much more robust guarantee, since the numerical values can be arbitrary, as long as they are consistent with the preference order.

In this paper, we study algorithms for combinatorial secretary problems that rely only on ordinal information. We assume that there is an unknown value for each element, but our algorithms only have access to the *total order* of the elements arrived so far, which is consistent with their values. We term this the *ordinal model*; as opposed to the *cardinal model*, in which the algorithm learns the exact values. We show bounds on the competitive ratio, i.e., we compare the quality of the computed solutions to the optima in terms of the exact underlying but unknown numerical values. Consequently, competitive ratios for our algorithms are robust guarantees against uncertainty in the input. Our approach follows a recent line of research by studying the potential of algorithms with ordinal information to approximate optima based on numerical values [4, 3, 1, 10].

1.1 Our Contribution

We first point out that many algorithms proposed in the literature continue to work in the ordinal model. In particular, a wide variety of algorithms for variants of the matroid secretary problem with constant competitive ratios continue to obtain their guarantees in the ordinal model (see Table 1 for an overview). This shows that many results in the literature are much stronger, since the algorithms require significantly less information. Notably, the algorithm of [9] extends to the ordinal model and gives a ratio of $O(\log^2 r)$ for general matroids, where r is the rank of the matroid. In contrast, the improved algorithms with ratios of $O(\log r)$ and $O(\log \log r)$ [8, 30, 16] are not applicable in the ordinal model.

For several combinatorial secretary problems we obtain new algorithms for the ordinal model. For online bipartite matching we give an algorithm that is $2e$-competitive. More generally, it obtains a ratio of $3e$ when the value of the matching is a submodular set function of the chosen edges. We also extend this result to online packing LPs with at most d non-zero entries per variable. Here we obtain an $O(d^{(B+1)/B})$-competitive algorithm, where B is a tightness parameter of the constraints. Another extension is matching in general graphs, for which we give a 8.78-competitive algorithm.

We give an $O(\alpha_1^2)$-competitive algorithm for the online weighted independent set problem in graphs, where α_1 is the local independence number of the graph. For example, for the prominent case of unit-disk graphs, $\alpha_1 = 5$ and we obtain a constant-competitive algorithm.

For matroids, we extend a result of [18] to the ordinal model: The reduction from submodular to linear matroid secretary can be done with ordinal information for marginal weights of the elements. More specifically, we show that whenever there is an algorithm that solves the matroid secretary problem in the ordinal model on some matroid class and has a competitive ratio of α, there is also an algorithm for the submodular matroid secretary problem in the ordinal model on the same matroid class with a competitive ratio of $O(\alpha^2)$. The ratio can be shown to be better if the linear algorithm satisfies some further properties.

Lastly, we consider the importance of knowing the weights, ordering, and structure of the domain. For algorithms that have complete ordinal information but cannot learn the specific matroid structure, we show a lower bound of $\Omega(\sqrt{n}/(\log n))$, even for partition matroids, where n is the number of elements in the ground set. This bound contrasts the $O(\log^2 r)$-competitive algorithm and indicates that learning the matroid structure is crucial in the ordinal model. Moreover, it contrasts the cardinal model, where thresholding algorithms yield $O(\log r)$-competitive algorithms without learning the matroid structure.

For structural reasons, we present our results in a slightly different order. We first discuss the matroid results in Section 3. Then we proceed with matching and packing in Section 4 and independent set in Section 5. Due to spatial constraints, all missing proofs are deferred to the full version [23].

2 Preliminaries and Related Work

In the typical problem we study, there is a set E of elements arriving sequentially in random order. The algorithm knows $n = |E|$ in advance. It must accept or reject an element before seeing the next element(s). There is a set $\mathcal{S} \subseteq 2^E$ of *feasible solutions*. $\mathcal{S}$ is downward-closed, i.e., if $S \in \mathcal{S}$, then $S' \in \mathcal{S}$ for every $S' \subseteq S$. The goal is to accept a feasible solution that maximizes an objective function f. In the *linear version*, each element has a *value* or *weight* w_e, and $f(S) = \sum_{e \in S} w_e$. In the *submodular version*, f is submodular and $f(\emptyset) = 0$.

In the linear ordinal model, the algorithm only sees a strict total order over the elements seen so far that is consistent with their weights (ties are broken arbitrarily). For the submodular version, we interpret the value of an element as its marginal contribution to a set of elements. In this case, our algorithm has access to an *ordinal oracle* $\mathcal{O}(S)$. For every subset S of arrived elements, $\mathcal{O}(S)$ returns a total order of arrived elements consistent with their marginal values $f(e|S) = f(S \cup \{e\}) - f(S)$. We chose this particular model because it elegantly aligns with the online setting. Both stronger and weaker definitions are possible and these are interesting avenues for future work.

Given this information, we strive to design algorithms with small competitive ratio $f(S^*)/\mathbb{E}[f(S_{alg})]$. Here S^* is an optimal feasible solution and S_{alg} the solution returned by the algorithm. Note that S_{alg} is a random variable due to random-order arrival and possible internal randomization of the algorithm.

■ **Table 1** Existing algorithms for matroid secretary problems that provide the same guarantee in the ordinal model.

Matroid	general	k-uniform	graphic	cographic	transversal	laminar	regular
Ratio	$O(\log^2 r)$	$1 + O(\sqrt{1/k})$, e	$2e$	$3e$	16	$3\sqrt{3}e$	$9e$
Reference	[9]	[15, 27, 6]	[28]	[31]	[13]	[24]	[14]

In the matroid secretary problem, the pair $\mathcal{M} = (E, \mathcal{S})$ is a matroid. We summarize in Table 1 some of the existing results for classes of the (linear) problem that transfer to the ordinal model. The algorithms for all restricted matroid classes other than the graphic matroid assume a-priori complete knowledge of the matroid – only weights are revealed online. The algorithms do not use cardinal information, their decisions are based only on ordinal information. As such, they translate directly to the ordinal model. Notably, the algorithm from [9] solves even the general submodular matroid secretary problem in the ordinal model.

2.1   Related Work

Our work is partly inspired by [4, 5], who study ordinal approximation algorithms for classical optimization problems. They design constant-factor approximation algorithms for matching and clustering problems with ordinal information and extend the results to truthful mechanisms. Our approach here differs due to online arrival. Anshelevich et al. [3] examine the quality of randomized social choice mechanisms when agents have metric preferences but only ordinal information is available to the mechanism. Previously, [1, 10] studied ordinal measures of efficiency in matchings, for instance the average rank of an agent's partner.

The literature on the secretary problem is too broad to survey here. We only discuss directly related work on online algorithms for combinatorial variants. In[29, 21], the authors study hiring a single secretary when only a partial ordering of the candidates is available. For multiple-choice secretary, where we can select any k candidates, there are algorithms with ratios that are constant and asymptotically decreasing in k [27, 6]. More generally, the matroid secretary problem has attracted a large amount of research interest [8, 11, 30, 16], and the best-known algorithm in the cardinal model has ratio $O(\log \log r)$. For results on specific matroid classes, see the overview in Table 1. Extensions to the submodular version are treated in [9, 18].

Another prominent domain is online bipartite matching, in which one side of the graph is known in advance and the other arrives online in random order, each vertex revealing all incident weighted edges when it arrives [28]. In this case, there is an optimal algorithm with ratio e [25]. Moreover, our paper is related to Göbel et al. [22] who study secretary versions of maximum independent set in graphs with bounded inductive independence number ρ. They derive an $O(\rho^2)$-competitive algorithm for unweighted and an $O(\rho^2 \log n)$-competitive algorithm for weighted independent set.

In addition, algorithms have been proposed for further variants of the secretary problem, e.g., the temp secretary problem (candidates hired for a fixed duration) [19], parallel secretary (candidates interviewed in parallel) [17], or local secretary (several firms and limited feedback) [12]. For these variants, some existing algorithms (e.g., for the temp secretary problem in [19]) directly extend to the ordinal model. In general, however, the restriction to ordinal information poses an interesting challenge for future work in these domains.

Algorithm 1: Greedy [18]

> **Input** : ground set E
> **Output** : independent set I

1 Let $I \leftarrow \emptyset$ and $E' \leftarrow E$.
2 while $E' \neq \emptyset$ do
3 $\quad$ Let $u \leftarrow \max_{u'} f(u'|I)$ and $E' \leftarrow E' \setminus \{u\}$;
4 $\quad$ if $(I \cup \{u\}$ independent in $\mathcal{M}) \wedge (f(u|I) \geq 0)$ then add u to I;

Algorithm 2: Online(p) algorithm [18]

> **Input** : $n = |E|$, size of the ground set
> **Output** : independent set $Q \cap N$

1 Choose X from the binomial distribution $B(n, 1/2)$.
2 Reject the first X elements of the input. Let L be the set of these elements.
3 Let M be the output of Greedy on the set L.
4 Let $N \leftarrow \emptyset$.
5 for *each element $u \in E \setminus L$* do
6 $\quad$ Let $w(u) \leftarrow 0$.
7 $\quad$ if *u accepted by Greedy applied to $M \cup \{u\}$* then
8 $\quad\quad$ With probability p do the following:
9 $\quad\quad$ Add u to N.
10 $\quad\quad$ Let $M_u \subseteq M$ be the solution of Greedy immediately before it adds u to it.
11 $\quad\quad$ $w(u) \leftarrow f(u|M_u)$.
12 $\quad$ Pass u to Linear with weight $w(u)$.
13 **return** $Q \cap N$, where Q is the output of Linear.

---

3 Matroids

3.1 Submodular Matroids

We start our analysis by showing that – in addition to algorithms for special cases mentioned above – a powerful technique for submodular matroid secretary problems [18] can be adjusted to work even in the ordinal model. More formally, in this section we show that there is a reduction from submodular matroid secretary problems with ordinal information (SMSPO) to linear matroid secretary problems with ordinal information (MSPO). The reduction uses Greedy (Algorithm 1) as a subroutine and interprets the marginal value when added to the greedy solution as the value of an element. These values are then forwarded to whichever algorithm (termed Linear) that solves the linear version of the problem. In the ordinal model, we are unable to see the exact marginal values. Nevertheless, we manage to construct a suitable ordering for the forwarded elements. Consequently, we can apply algorithm Linear as a subroutine to obtain a good solution for the ordinal submodular problem.

Let $\mathcal{M} = (E, \mathcal{S})$ be the matroid, f the submodular function, and E the ground set of elements. The marginal contribution of element u to set M is denoted by $f(u|M) = f(M \cup \{u\}) - f(M)$. Since f can be non-monotone, Greedy in the cardinal model also checks if the marginal value of the currently best element is positive. While we cannot explicitly make this check in the ordinal model, note that $f(u|M) \geq 0 \iff f(M \cup \{u\}) \geq f(M) = f(M \cup \{u'\})$

for every $u' \in M$. Since the ordinal oracle includes the elements of M in the ordering of marginal values, there is a way to check positivity even in the ordinal model. Therefore, our results also apply to non-monotone functions f.

A potential problem with Algorithm 2 is that we must compare marginal contributions of different elements w.r.t. different sets. We can resolve this issue by following the steps of the Greedy subroutine that tries to add new elements to the greedy solution computed on the sample. We use this information to construct a correct ordering over the marginal contributions of elements that we forward to Linear.

▶ **Lemma 1.** *Let us denote by s_u the step of Greedy in which the element u is accepted when applied to $M + u$. Then $s_{u_1} < s_{u_2}$ implies $f(u_1|M_{u_1}) \geq f(u_2|M_{u_2})$.*

Proof. First, note that $M_{u_1} \subset M_{u_2}$ when $s_1 < s_2$. We denote by m_{u_1} the element of M that would be taken in step s_{u_1} if u_1 would not be available. Then we know that $f(u_1|M_{u_1}) \geq f(m_{u_1}|M_{u_1})$. Furthermore, since $s_1 < s_2$, $f(m_{u_1}|M_{u_1}) \geq f(u_2|M_{u_1})$. Lastly, by using submodularity, we know that $f(u_2|M_{u_1}) \geq f(u_2|M_{u_2})$. ◀

When $s_{u_1} = s_{u_2}$, then $M_{u_1} = M_{u_2}$ so the oracle provides the order of marginal values. Otherwise, the lemma yields the ordinal information. Thus, we can construct an ordering for the elements that are forwarded to Linear that is consistent with their marginal values in the cardinal model. Hence, the reduction can be applied in the ordinal model, since the algorithm executes exactly the same as in the cardinal model, and all results from [18] continue to hold. We mention only the main theorem. It implies constant ratios for all problems in Table 1 in the submodular version.

▶ **Theorem 2.** *Given an arbitrary algorithm Linear for MSPO that is α-competitive on a matroid class, there is an algorithm for SMSPO with competitive ratio is at most $24\alpha(3\alpha+1) = O(\alpha^2)$ on the same matroid class. For SMSPO with monotone f, it can be improved to $8\alpha(\alpha + 1)$.*

3.2   A Lower Bound

Another powerful technique in the cardinal model is thresholding, where we first sample a constant fraction of the elements to learn their weights. Based on the largest weight observed, we pick a threshold and accept subsequent elements greedily if they exceed the threshold. This approach generalizes the classic algorithm [15] and provides logarithmic ratios for many combinatorial domains [8, 28, 22, 12]. Intuitively, these algorithms *learn the weights but not the structure.*

We show that this technique does not easily generalize to the ordinal model. The algorithms with small ratios in the ordinal model rely heavily on the matroid structure. Indeed, in the ordinal model we show a polynomial lower bound for algorithms in the matroid secretary problem that learn the ordering but not the structure. Formally, we slightly simplify the setting as follows. The algorithm receives the global ordering of all elements in advance. It determines (possibly at random) a threshold position in the ordering. Then elements arrive and are accepted greedily if ranked above the threshold. Note that the algorithm does not use sampling, since in this case the only meaningful purpose of sampling is learning the structure. We call this a *structure-oblivious* algorithm.

▶ **Theorem 3.** *Every structure-oblivious randomized algorithm has a competitive ratio of at least $\Omega(\sqrt{n}/(\log n))$.*
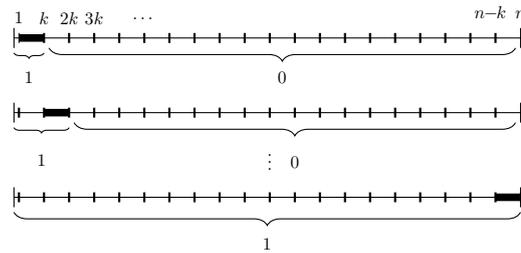
Figure 1 Values for the family of instances described in the proof of Theorem 11, where the position of the "valuable" ones is denoted by the thick segment.
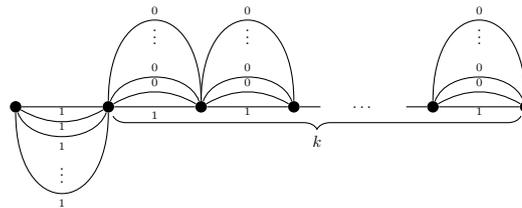
Figure 2 One instance from the family described in the proof of Theorem 11.

Proof. In the proof, we restrict our attention to instances with weights in $\{0, 1\}$ (for a formal justification, see the full version [23]). We give a distribution of such instances on which every deterministic algorithm has a competitive ratio of $\Omega(\sqrt{n}/(\log n))$. Using Yao's principle, this shows the claimed result for randomized algorithms.

All instances in the distribution are based on a graphic matroid (in fact, a partition matroid) of the following form. There is a simple path of $1 + k$ segments. The edges in each segment have weight of 0 or 1. We call the edges with value 1 in the last k segments the "valuable edges". The total number of edges is the same in each instance and equals $n + 1$. All edges in the first segment have value 1 and there is exactly one edge of value 1 in all other segments (that being the aforementioned valuable edges). In the first instance there are in total $k + 1$ edges of value 1 (meaning that there is only one edge in the first segment). In each of the following instances this number is increased by k (in the i-th instance there are $(i - 1) \cdot k + 1$ edges in the first segment) such that the last instance has only edges with value 1 (there are $n - k + 1$ edges in the first segment). The zero edges are always equally distributed on the last k path segments. The valuable edges are lower in the ordering than any non-valuable edge with value 1 (see Figure 1). Each of the instances appears with equal probability of $\frac{k}{n}$ (see Figure 2 for one example instance).

A deterministic algorithm picks a threshold at position i. The expected value of the solution is

$$\mathbb{E}[w(S_{alg})] \leq 1 + \frac{k}{n} \sum_{\ell=1}^{\frac{i}{k}} \frac{k}{\ell} \leq 1 + \frac{k^2}{n} \log \frac{i}{k} \leq \frac{k^2}{n} \log \frac{n}{k} + 1 \ ,$$

where log denotes the natural logarithm and the expression results from observing that the algorithm cannot obtain more than a value of 1 if its threshold i falls above the valuable 1's. Otherwise it gets an additional fraction of k, depending on how close the threshold is positioned to the valuable 1's. For instance, if the threshold is set between 1 and k positions

below the valuable 1's, the algorithm will in expectation select edges of total value of at least $1 + k/2$. This follows from the random arrival order of the edges and the fact that the ratio of valuable to non-valuable edges that the algorithm is ready to accept is at least $1 : 2$. Furthermore, we see that for this distribution of instances the optimal way to set a deterministic threshold is at the lowest position. Using $k = \sqrt{n}$, a lower bound on the competitive ratio is

$$\frac{k}{\frac{k^2}{n}\log\frac{n}{k} + 1} = \frac{n}{k\log\frac{n}{k} + \frac{n}{k}} = \Omega\left(\frac{\sqrt{n}}{\log n}\right) \quad . \qquad\qquad \blacktriangleleft$$

4   Matching and Packing

4.1   Bipartite Matching

In this section, we study online bipartite matching. The vertices on the right side of the graph (denoted by R) are static and given in advance. The vertices on the left side (denoted by L) arrive sequentially in a random order. Every edge $e = (r, l) \in R \times L$ has a non-negative weight $w(e) \geq 0$. In the cardinal model, each vertex of L reveals upon arrival the weights of all incident edges. In the ordinal model, we are given a total and strict order of all edges that have arrived so far, consistent with their weights[1]. Before seeing the next vertex of L, the algorithm has to decide to which vertex $r \in R$ (if any) it wants to match the current vertex l. A match that is formed cannot be revoked. The goal is to maximize the total weight of the matching.

The algorithm for the cardinal model in [25] achieves an optimal competitive ratio of e. However, this algorithm heavily exploits cardinal information by repeatedly computing max-weight matchings for the edges seen so far. For the ordinal model, our Algorithm 3 below obtains a competitive ratio of $2e$. While similar in spirit, the main difference is that we rely on a greedy matching algorithm, which is based solely on ordinal information. It deteriorates the ratio only by a factor of 2.

Here we assume to have access to ordinal preferences over all the edges in the graph. Note that the same approach works if the vertices provide correlated (ordinal) preference lists consistent with the edge weights, for every vertex from R and every arrived vertex from L. In this case, the greedy algorithm can still be implemented by iteratively matching and removing a pair that mutually prefers each other the most, and it provides an approximation guarantee of 2 for the max-weight matching (see, e.g., [2]). In contrast, if we receive only preference lists for vertices on one side, there are simple examples that establish super-constant lower bounds on the competitive ratio[2].

▶ **Lemma 4.** *Let the random variable A_v denote the contribution of the vertex $v \in L$ to the output, i.e. weight assigned to v in M. Let $w(M^*)$ denote the value of the maximum-weight matching in G. For $l \in \{\lceil\frac{n}{e}\rceil, \dots, n\}$,*

$$\mathbb{E}\big[A_l\big] \geq \frac{\lceil\frac{n}{e}\rceil}{l-1} \cdot \frac{w(M^*)}{2n} \quad .$$

---

[1] Ties are broken arbitrarily, but consistently over the arrival process.
[2] Consider a bipartite graph with two nodes on each side (named A,B and 1,2). If we only know that both A and B prefer 1 to 2, the ratio becomes at least 2 even in the offline case. Similar examples imply that the (offline) ratio must grow in the size of the graph.

Algorithm 3: Bipartite Matching

Input   : vertex set R and cardinality $n = |L|$
Output: matching M

1 Let L' be the first $\lfloor \frac{n}{e} \rfloor$ vertices of L, and $M \leftarrow \emptyset$;
2 **for** *each $\ell \in L \setminus L'$* **do**
3 $\quad$ $L' \leftarrow L' \cup \{\ell\}$;
4 $\quad$ $M^{(\ell)} \leftarrow$ greedy matching on $G[L' \cup R]$;
5 $\quad$ Let $e^{(\ell)} \leftarrow (\ell, r)$ be the edge assigned to ℓ in $M^{(\ell)}$;
6 $\quad$ **if** $M \cup \{e^{(\ell)}\}$ is a matching **then** add $e^{(\ell)}$ to M;

Proof. We first show that $e^{(\ell)}$ has a significant expected weight. Then we bound the probability of adding $e^{(\ell)}$ to M.

In step ℓ, $|L'| = \ell$ and the algorithm computes a greedy matching $M^{(\ell)}$ on $G[L' \cup R]$. The current vertex ℓ can be seen as selected uniformly at random from L', and L' can be seen as selected uniformly at random from L. Therefore, $\mathbb{E}[w(M^{(\ell)})] \geq \frac{\ell}{n} \cdot \frac{w(M^*)}{2}$ and $\mathbb{E}[w(e^{(\ell)})] \geq \frac{w(M^*)}{2n}$. Here we use that a greedy matching approximates the optimum by at most a factor of 2 [2].

Edge $e^{(\ell)}$ can be added to M if r has not been matched already. The vertex r can be matched only when it is in $M^{(k)}$ where $\lceil n/e \rceil \leq k \leq l - 1$. The probability of r being matched in step k is at most $\frac{1}{k}$ and the order of the vertices in steps $1, \ldots, k-1$ is irrelevant for this event.

$$\Pr[r \text{ unmatched in step } \ell] = \Pr\left[\bigwedge_{k=\lceil n/e \rceil}^{\ell-1} r \notin e^{(k)} \right] \geq \prod_{k=\lceil n/e \rceil}^{\ell-1} \frac{k-1}{k} = \frac{\lceil \frac{n}{e} \rceil - 1}{\ell - 1}$$

We now know that $\Pr[M \cup e^{(\ell)} \text{ is a matching}] \geq \frac{\lfloor n/e \rfloor}{\ell - 1}$. Using this and $\mathbb{E}[w(e^{(\ell)})] \geq \frac{w(M^*)}{2n}$, the lemma follows.   ◂

▶ **Theorem 5.** *Algorithm 3 for bipartite matching is $2e$-competitive.*

Proof. The weight of matching M can be obtained by summing over random variables A_ℓ.

$$\mathbb{E}[w(M)] = \mathbb{E}\left[\sum_{\ell=1}^{n} A_\ell \right] \geq \sum_{\ell=\lceil n/e \rceil}^{n} \frac{\lfloor n/e \rfloor}{\ell - 1} \cdot \frac{w(M^*)}{2n} = \frac{\lfloor n/e \rfloor}{2n} \sum_{\ell=\lfloor n/e \rfloor}^{n-1} \frac{1}{\ell} \cdot w(M^*)$$

Since $\frac{\lfloor n/e \rfloor}{n} \geq \frac{1}{e} - \frac{1}{n}$ and $\sum_{\ell=\lfloor n/e \rfloor}^{n-1} \frac{1}{l} \geq \ln \frac{n}{\lfloor n/e \rfloor} \geq 1$, it follows that

$$\mathbb{E}[w(M)] \geq \left(\frac{1}{e} - \frac{1}{n} \right) \cdot \frac{w(M^*)}{2} \ . \qquad\qquad\qquad ◂$$

We can use the same algorithm and the same analysis in the submodular version, where greedy gives a 3-approximation [20]. It builds the matching by greedily adding an edge that maximizes the marginal improvement of f, which is the information delivered by the ordinal oracle. Hence, our algorithm is $3e$-competitive for the submodular version.

Algorithm 4: Packing LP

Input : capacities $\mathbf{b}$, total number of requests n, probability $p = \frac{e(2d)^{1/B}}{1+e(2d)^{1/B}}$
Output : assignment vector $\mathbf{y}$

1 Let L' be the first $p \cdot n$ requests, and $\mathbf{y} \leftarrow \mathbf{0}$;
2 **for** *each* $j \notin L'$ **do**
3 $L' \leftarrow L' \cup \{j\}$;
4 $\mathbf{x}^{(L')} \leftarrow$ greedy assignment on the LP for L';
5 $\mathbf{y}_j \leftarrow \mathbf{x}_j^{(L')}$;
6   **if** $\neg(\mathbf{A}(\mathbf{y}) \leq \mathbf{b})$ **then** $\mathbf{y}_j \leftarrow \mathbf{0}$;

4.2 Packing

Our results for bipartite matching can be extended to online packing LPs of the form $\max \mathbf{c}^\tau \mathbf{x}$ s.t. $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$, which model problems with m resources and n online requests coming in random order. Each resource $i \in [m]$ has a capacity b_i that is known in advance, together with the number of requests. Every online request comes with a set of options, where each option has its profit and resource consumption. Once a request arrives, the coefficients of its variables are revealed and the assignment to the variables has to be determined.

Formally, request $j \in [n]$ corresponds to variables $x_{j,1}, \ldots, x_{j,K}$ that represent K options. Each option $k \in [K]$ contributes with profit $c_{j,k} \geq 0$ and has resource consumption $a_{i,j,k} \geq 0$ for resource i. Overall, at most one option can be selected, i.e., there is a constraint $\sum_{k \in [K]} x_{j,k} \leq 1, \forall j \in [n]$. The objective is to maximize total profit while respecting the resource capacities. The offline problem is captured by the following linear program:

$$\max \sum_{j \in [n]} \sum_{k \in [K]} c_{j,k} x_{j,k} \quad \text{s.t.} \quad \sum_{j \in [n]} \sum_{k \in [K]} a_{i,j,k} x_{j,k} \leq b_i \qquad i \in [m]$$

$$\sum_{k \in [K]} x_{j,k} \leq 1 \qquad j \in [n]$$

As a parameter, we denote by d the maximum number of non-zero entries in any column of the constraint matrix $\mathbf{A}$, for which by definition $d \leq m$. We compare the solution to the fractional optimum, which we denote by $\mathbf{x}^*$. The competitive ratio will be expressed in terms of d and the capacity ratio $B = \min_{i \in [m]} \left\lfloor \frac{b_i}{\max_{j \in [n], k \in [K]} a_{i,j,k}} \right\rfloor$.

Kesselheim et al. [25] propose an algorithm that heavily exploits cardinal information – it repeatedly solves an LP-relaxation and uses the solution as a probability distribution over the options. Instead, our Algorithm 4 for the ordinal model is based on greedy assignments in terms of profits $c_{j,k}$. More specifically, the greedy assignment considers variables $x_{j,k}$ in non-increasing order of $c_{j,k}$. It sets a variable to 1 if this does not violate the capacity constraints, and to 0 otherwise.

▶ **Theorem 6.** *Algorithm 4 for online packing LPs is $O(d^{(B+1)/B})$-competitive.*

4.3 Matching in General Graphs

Here we study the case when vertices of a general undirected graph arrive in random order. In the beginning, we only know the number n of vertices. Each edge in the graph has a non-negative weight $w(e) \geq 0$. Each vertex reveals the incident edges to previously arrived

---

Algorithm 5: General Matching

Input : vertex set V and cardinality $n = |V|$
Output: matching M

1 Let R be the first $\lfloor \frac{n}{2} \rfloor$ vertices of V;
2 Let L' be the further $\lfloor \frac{n}{2e} \rfloor$ vertices of V, and $M \leftarrow \emptyset$;
3 for *each $\ell \in V \setminus L'$* **do**
4 $\quad$ $L' \leftarrow L' \cup \{\ell\}$;
5 $\quad$ $M^{(\ell)} \leftarrow$ greedy matching on $G[L' \cup R]$;
6 $\quad$ Let $e^{(\ell)} \leftarrow (\ell, r)$ be the edge assigned to ℓ in $M^{(\ell)}$;
7 $\quad$ **if** $M \cup \{e^{(\ell)}\}$ is a matching **then** add $e^{(\ell)}$ to M;

---

vertices and their weights (cardinal model), or we receive a total order over all edges among arrived vertices that is consistent with the weights (ordinal model). An edge can be added to the matching only in the round in which it is revealed. The goal is to construct a matching with maximum weight.

We can tackle this problem by prolonging the sampling phase and dividing the vertices into "left" and "right" vertices. Algorithm 5 first samples $n/2$ vertices. These are assigned to be the set R, corresponding to the static side of the graph in bipartite matching. The remaining vertices are assigned to be the set L. The algorithm then proceeds by sampling a fraction of the vertices of L, forming a set L'. The remaining steps are exactly the same as in Algorithm 3.

▶ **Theorem 7.** *Algorithm 5 for matching in general graphs is $12e/(e+1)$-competitive, where $12e/(e+1) < 8.78$.*

5 Independent Set and Local Independence

In this section, we study maximum independent set in graphs with bounded local independence number. The set of elements are the vertices V of an underlying undirected graph G. Each vertex has a weight $w_v \geq 0$. We denote by $N(v)$ the set of direct neighbors of vertex v. Vertices arrive sequentially in random order and reveal their position in the order of weights of vertices seen so far. The goal is to construct an independent set of G with maximum weight. The exact structure of G is unknown, but we know that G has a bounded local independence number α_1.

▶ **Definition 8.** An undirected graph G has local independence number α_1 if for each node v, the cardinality of every independent set in the neighborhood $N(v)$ is at most α_1.

We propose Algorithm 6, which is inspired by the Sample-and-Price algorithm for matching in [28]. Note that Göbel et al. [22] construct a more general approach for graphs with bounded *inductive* independence number ρ. However, they only obtain a ratio of $O(\rho^2 \log n)$ for the weighted version, where a competitive ratio of $\Omega(\log n / \log^2 \log n)$ cannot be avoided, even in instances with constant ρ. These algorithms rely on ρ-approximation algorithms for the offline problem that crucially exploit cardinal information.

Similar to [28], we reformulate Algorithm 6 into an equivalent approach (Algorithm 7) for the sake of analysis. Given the same arrival order, the same vertices are in the sample. Algorithm 7 drops all vertices from S that have neighbors in S while Algorithm 6 keeps one of them. Hence, $\mathbb{E}[w(S_{Alg_6})] \geq \mathbb{E}[w(S_{Sim})]$. In what follows, we analyze the performance

---

Algorithm 6: Independent Set in Graphs with Bounded Local Independence Number

---

Input : $n = |G|$, $p = \sqrt{\alpha_1/(\alpha_1 + 1)}$
Output : independent set of vertices S

1 Set $k \leftarrow \text{Binom}(n, p)$, $S \leftarrow \emptyset$;
2 Reject first k vertices of G, denote this set by G';
3 Build a maximal independent set of vertices from G' greedily, denote this set by M_1;
4 **for** *each $v \in G \setminus G'$* **do**
5 $\quad w^* \leftarrow \max\{w \mid \mathcal{N}(v) \cap M_1\}$;
6 $\quad$ **if** $(v > w^*) \wedge (S \cup \{v\} \text{ independent set})$ **then** add v to S;

Algorithm 7: Simulate

Input : $n = |G|$, $p = \sqrt{\alpha_1/(\alpha_1 + 1)}$
Output : independent set of vertices S

1 Sort all vertices in G in non-increasing order of value;
2 Initialize $M_1, M_2 \leftarrow \emptyset$;
3 **for** *each $v \in G$ in sorted order* **do**
4 $\quad$ **if** $M_1 \cup \{v\}$ *independent set* **then**
5 $\quad\quad$ flip a coin with probability p of heads;
6 $\quad\quad$ **if** heads **then** $M_1 \leftarrow M_1 \cup \{v\}$; **else** $M_2 \leftarrow M_2 \cup \{v\}$;

7 $S \leftarrow M_2$;
8 **for** *each $w \in S$* **do**
9 $\quad$ **if** w has neighbors in S **then** remove w and all his neighbors from S;

---

of Simulate. The first lemma follows directly from the definition of the local independence number.

▶ **Lemma 9.** $\mathbb{E}[w(M_1)] \geq p \cdot \frac{w(S^*)}{\alpha_1}$, where $\alpha_1 \geq 1$ is the local independence number of G.

▶ **Lemma 10.** $\mathbb{E}\big[|\mathcal{N}(v) \cap M_2| \mid v \in M_2\big] \leq \frac{\alpha_1(1-p)}{p}$.

Proof. Let us denote by X_u^1 and X_u^2 the indicator variables for the events $u \in M_1$ and $u \in M_2$ respectively. Then,

$$\mathbb{E}\big[|\mathcal{N}(v) \cap M_2| \mid v \in M_2\big] = \mathbb{E}\left[\sum_{u \in \mathcal{N}(v)} X_u^2 \mid v \in M_2\right] = \sum_{u \in \mathcal{N}(v)} \mathbb{E}\big[X_u^2 \mid v \in M_2\big]$$

$$= \frac{1-p}{p} \sum_{u \in \mathcal{N}(v)} \mathbb{E}\big[X_u^1 \mid v \in M_2\big] \leq \frac{1-p}{p} \cdot \alpha_1 \ . \qquad \blacktriangleleft$$

▶ **Theorem 11.** *Algorithm 7 for weighted independent set is $O(\alpha_1^2)$-competitive, where α_1 is the local independence number of the graph.*

Proof. By using Markov's inequality and Lemma 10,

$$\Pr[|\mathcal{N}(v) \cap M_2| \geq 1 \mid v \in M_2] \quad \leq \quad \alpha_1 \cdot (1-p)/p$$

$$\text{and} \quad \Pr[|\mathcal{N}(v) \cap M_2| < 1 \mid v \in M_2] \quad > \quad 1 - (\alpha_1(1-p)/p) \ .$$

Thus, we can conclude that

$$\mathbb{E}[w(S)] \geq \left(1 - \alpha_1 \cdot \frac{1-p}{p}\right) \cdot \mathbb{E}[w(M_2)] \geq \left(1 - \alpha_1 \cdot \frac{1-p}{p}\right) \cdot \frac{1-p}{\alpha_1} \cdot w(S^*) \ .$$

The ratio is optimized for $p = \sqrt{\frac{\alpha_1}{\alpha_1 + 1}}$, which proves the theorem. ◀

As a prominent example, $\alpha_1 = 5$ in the popular class of unit-disk graphs. In such graphs, our algorithm yields a constant competitive ratio for online independent set in the ordinal model.

References

1 David Abraham, Robert Irving, Telikepalli Kavitha, and Kurt Mehlhorn. Popular matchings. *SIAM J. Comput.*, 37(4):1030–1045, 2007.

2 Elliot Anshelevich and Martin Hoefer. Contribution games in networks. *Algorithmica*, 63(1–2):51–90, 2012.

3 Elliot Anshelevich and John Postl. Randomized social choice functions under metric preferences. In *Proc. 25th Intl. Joint Conf. Artif. Intell. (IJCAI)*, pages 46–59, 2016.

4 Elliot Anshelevich and Shreyas Sekar. Blind, greedy, and random: Algorithms for matching and clustering using only ordinal information. In *Proc. 13th Conf. Artificial Intelligence (AAAI)*, pages 390–396, 2016.

5 Elliot Anshelevich and Shreyas Sekar. Truthful mechanisms for matching and clustering in an ordinal world. In *Proc. 12th Conf. Web and Internet Economics (WINE)*, pages 265–278, 2016.

6 Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. In *Proc. 10th Workshop Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 16–28, 2007.

7 Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exchanges*, 7(2), 2008.

8 Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proc. 18th Symp. Discrete Algorithms (SODA)*, pages 434–443, 2007.

9 MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Trans. Algorithms*, 9(4):32, 2013.

10 Deeparnab Chakrabarty and Chaitanya Swamy. Welfare maximization and truthfulness in mechanism design with ordinal preferences. In *Proc. 5th Symp. Innovations in Theoret. Computer Science (ITCS)*, pages 105–120, 2014.

11 Sourav Chakraborty and Oded Lachish. Improved competitive ratio for the matroid secretary problem. In *Proc. 23rd Symp. Discrete Algorithms (SODA)*, pages 1702–1712, 2012.

12 Ning Chen, Martin Hoefer, Marvin Künnemann, Chengyu Lin, and Peihan Miao. Secretary markets with local information. In *Proc. 42nd Intl. Coll. Automata, Languages and Programming (ICALP)*, volume 2, pages 552–563, 2015.

13 Nedialko Dimitrov and Greg Plaxton. Competitive weighted matching in transversal matroids. *Algorithmica*, 62(1-2):333–348, 2012.

14 Michael Dinitz and Guy Kortsarz. Matroid secretary for regular and decomposable matroids. *SIAM J. Comput.*, 43(5):1807–1830, 2014.

15 Eugene Dynkin. The optimum choice of the instant for stopping a Markov process. In *Sov. Math. Dokl*, volume 4, pages 627–629, 1963.

16 Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $O(\log \log(\text{rank}))$-competitive algorithm for the matroid secretary problem. In *Proc. 26th Symp. Discrete Algorithms (SODA)*, pages 1189–1201, 2015.

17   Moran Feldman and Moshe Tennenholtz. Interviewing secretaries in parallel. In *Proc. 13th Conf. Electronic Commerce (EC)*, pages 550–567, 2012.

18   Moran Feldman and Rico Zenklusen. The submodular secretary problem goes linear. In *Proc. 56th Symp. Foundations of Computer Science (FOCS)*, pages 486–505, 2015.

19   Amos Fiat, Ilia Gorelik, Haim Kaplan, and Slava Novgorodov. The temp secretary problem. In *Proc. 23rd European Symp. Algorithms (ESA)*, pages 631–642, 2015.

20   Marshall Fisher, George Nemhauser, and Laurence Wolsey. An analysis of approximations for maximizing submodular set functions-II. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

21   Byrn Garrod and Robert Morris. The secretary problem on an unknown poset. *Random Struct. Algorithms*, 43(4), 2013.

22   Oliver Göbel, Martin Hoefer, Thomas Kesselheim, Thomas Schleiden, and Berthold Vöcking. Online independent set beyond the worst-case: Secretaries, prophets and periods. In *Proc. 41st Intl. Coll. Automata, Languages and Programming (ICALP)*, volume 2, pages 508–519, 2014.

23   Martin Hoefer and Bojana Kodric. Combinatorial secretary problems with ordinal information. *CoRR*, abs/1702.01290, 2017. URL: `http://arxiv.org/abs/1702.01290`.

24   Patrick Jaillet, José Soto, and Rico Zenklusen. Advances on matroid secretary problems: Free order model and laminar case. In *Proc. 16th Intl. Conf. Integer Programming and Combinatorial Optimization (IPCO)*, pages 254–265, 2013.

25   Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Proc. 21st European Symp. Algorithms (ESA)*, pages 589–600, 2013.

26   Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal beats dual on online packing LPs in the random-order model. In *Proc. 46th Symp. Theory of Computing (STOC)*, pages 303–312, 2014.

27   Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proc. 16th Symp. Discrete Algorithms (SODA)*, pages 630–631, 2005.

28   Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *Proc. 36th Intl. Coll. Automata, Languages and Programming (ICALP)*, pages 508–520, 2009.

29   Ravi Kumar, Silvio Lattanzi, Sergei Vassilvitskii, and Andrea Vattani. Hiring a secretary from a poset. In *Proc. 12th Conf. Economics and Computation (EC)*, 2011.

30   Oded Lachish. O(log log rank) competitive ratio for the matroid secretary problem. In *Proc. 55th Symp. Foundations of Computer Science (FOCS)*, pages 326–335, 2014.

31   José Soto. Matroid secretary problem in the random-assignment model. *SIAM J. Comput.*, 42(1):178–211, 2013.