






Laminar Matroid Secretary: Greedy Strikes Back

Zhiyi Huang   

Department of Computer Science, The University of Hong Kong, Hong Kong

Zahra Parsaeian 

University of Freiburg, Germany

Zixuan Zhu  

Department of Computer Science, The University of Hong Kong, Hong Kong

Abstract

We show that a simple greedy algorithm is 4.75-competitive for the Laminar Matroid Secretary Problem, improving the $3\sqrt{3} \approx 5.196$ -competitive algorithm based on the forbidden sets technique (Soto, Turkieltaub, and Verdugo, 2018).

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Matroid Secretary, Greedy Algorithm, Laminar Matroid

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.73

Supplementary Material

Software (Source code): <https://github.com/ZixuannnZhu/Laminar-Matroid-Secretary>
archived at `swh:1:dir:954b919a6c837d42d75462980b05ab1529af703d`

Funding This work is supported by General Research Fund from Research Grants Council of Hong Kong (Grand No. 17211723).

1 Introduction

Decision-making under uncertainty has always been a central research topic in Computer Science and Operations Research. A classical problem, and one of the oldest in this area, is the Secretary Problem [4, 6, 11]. Here a decision-maker wants to hire the best secretary out of n applicants. The applicants are interviewed one at a time by a random order. After each applicant's interview, the decision-maker learns how he/she ranks compared to the previous applicants, and must then immediately decide whether to hire him/her. The goal is to maximize the probability of hiring the best secretary. This problem has been extensively studied in the literature and several optimal e -competitive algorithms have been proposed. A well-known elegant solution by Dynkin [4] goes as follows: reject the first $\frac{n}{e}$ applicants, and then hire the first subsequent applicant that is better than all previous applicants.

Babaioff, Immorlica, and Kleinberg [2] formulated a generalization called the Matroid Secretary Problem, which considers hiring $r \geq 1$ secretaries subject to a matroid constraint. From now on, we will refer to the secretaries as elements. We are given a matroid $\mathcal{M} = (E, \mathcal{I})$, where E is a finite set of elements and $\mathcal{I} \subseteq 2^E$ is a collection of *independent sets* of elements from E . We will defer the definition of general matroids to the next section, and refer interested readers to the textbook by Oxley [13] for an in-depth exposition of matroid theory. For now, readers may think of \mathcal{I} as the collection of all subsets with size at most r , known as the r -uniform matroid, as a running example. Each element $e \in E$ has a positive weight w_e . Elements' weights are revealed to the algorithm one at a time by a random order. After observing an element's weight, the algorithm must immediately decide whether to select the element. The goal is to select an independent set that maximizes the sum of the weights of the elements in it. An algorithm is c -competitive if the expected total weight of its selected elements is at least a $\frac{1}{c}$ fraction of the total weight of the optimal solution in hindsight.



© Zhiyi Huang, Zahra Parsaeian, and Zixuan Zhu;
licensed under Creative Commons License CC-BY 4.0
32nd Annual European Symposium on Algorithms (ESA 2024).

Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 73; pp. 73:1–73:8
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Babai et al. [2] conjectured the existence of a constant competitive algorithm for the Matroid Secretary Problem, which has been a major open question in the literature for more than a decade. The best competitive ratio so far is $\mathcal{O}(\log \log r)$, due to Lachish [10] and Feldman, Svensson, and Zenklusen [5].

Although the general Matroid Secretary Problem remains elusive, much progress has been made toward designing constant competitive algorithms for special matroids. This paper focuses on the laminar matroids. A *laminar family* \mathcal{F} over elements E is a collection of subsets of elements such that any pair of distinct subsets A and B are either related by containment or disjoint, i.e., either $A \subset B$, or $B \subset A$, or $A \cap B = \emptyset$. A *laminar matroid* puts a capacity $c(B)$ on each subset $B \in \mathcal{F}$; a subset of elements is independent if it has at most $c(B)$ elements in each $B \in \mathcal{F}$. We call it the Laminar Matroid Secretary Problem.

Im and Wang [7] gave the first constant competitive algorithm for laminar matroids, with a competitive ratio of $\frac{16000}{3}$. Jaillet, Soto, and Zenklusen [8] proposed an improved $3\sqrt{3}e \approx 14.12$ -competitive algorithm by reducing it to the problem with partition matroids and running an e -competitive algorithm on the resulting problem. Ma, Tang, and Wang [12] showed that directly running a greedy algorithm is 9.6-competitive. Finally, Soto, Turkieltaub, and Verdugo [15] introduced the forbidden sets technique and applied it to various matroids. For laminar matroids, they combined it with the reduction to partition matroids by Jaillet et al. [8] to yield the best existing $3\sqrt{3} \approx 5.196$ -competitive ratio.

This paper considers a simple greedy algorithm that is different from the previous greedy algorithm by Ma et al. [12]. We show that our algorithm is 4.75-competitive for the Laminar Matroid Secretary Problem, improving upon the best existing result by Soto et al. [15].

2 Preliminaries

We will assume that the elements' weights are distinct, which simplifies the arguments by avoiding tie-breaking cases. This is without loss of generality as we can randomly perturb the weights with negligible Gaussian noise.

For notational simplicity, we write $S + e$ for $S \cup \{e\}$ and $S - e$ for $S \setminus \{e\}$.

► **Definition 1.** A matroid $\mathcal{M} = (E, \mathcal{I})$ consists of a set of elements E and collection of subsets of elements \mathcal{I} called the independent sets, with the following properties:

- **Non-empty Property:** $\emptyset \in \mathcal{I}$;
- **Hereditary Property:** For any $S \subset T \subseteq E$, if $T \in \mathcal{I}$ then $S \in \mathcal{I}$;
- **Augmentation Property:** For any $S, T \in \mathcal{I}$ such that $|S| > |T|$, there exists an element $e \in S \setminus T$ such that $T + e \in \mathcal{I}$.

Another useful property of matroids, even considered to be the fourth axiom by some, is the following **Exchange Property**.

► **Lemma 2** (c.f., Schrijver [14], Corollary 39.12a). For any $S, T \in \mathcal{I}$ such that $|S| = |T|$, for any $e \in S \setminus T$, there is an element $e' \in T \setminus S$ such that $S - e + e', T - e' + e \in \mathcal{I}$.

Let OPT denote the independent set that maximizes the sum of its elements' weights, which is unique when all the elements have distinct weights. We will use a well-known property of matroids, whose proof we include for self-containedness.

► **Lemma 3.** For any matroid $\mathcal{M} = (E, \mathcal{I})$, and any subset of elements $S \subseteq E$, the elements in $OPT \cap S$ also belong to the optimal solution of the sub-matroid restricted to the elements in S , i.e., $\mathcal{M}_S = (S, \mathcal{I} \cap 2^S)$.

Proof. Let $OPT(S)$ denote the optimal solution of sub-matroid \mathcal{M}_S . Suppose to the contrary that there exists an element $e \in OPT \cap S$ such that $e \notin OPT(S)$.

We first show that there is an element $e' \in OPT(S)$ such that:

$$OPT - e + e', OPT(S) - e' + e \in \mathcal{I}.$$

If $|OPT(S)| = |OPT|$, it follows by the exchange property.

Next suppose that $|OPT(S)| < |OPT|$. By repeatedly applying the augmentation property, we can find elements $e_1, e_2, \dots, e_k \in OPT \setminus OPT(S)$, where $k = |OPT| - |OPT(S)|$, such that $OPT(S) + e_1 + \dots + e_k \in \mathcal{I}$. Further, the optimality of $OPT(S)$ and the fact that $e \notin OPT(S)$ indicate $e_1, \dots, e_k \neq e$. We next apply the exchange property to $OPT(S) + e_1 + \dots + e_k$ and OPT to conclude that there is an element $e' \in (OPT(S) + e_1 + \dots + e_k) \setminus OPT = OPT(S) \setminus OPT$ such that $OPT - e + e' \in \mathcal{I}$ and $OPT(S) + e_1 + \dots + e_k - e' + e \in \mathcal{I}$. Finally, the latter implies $OPT(S) - e' + e \in \mathcal{I}$ by the hereditary property.

Given such an element e' , we can now derive a contradiction from the optimality of OPT and $OPT(S)$: the optimality of OPT implies $w_e > w_{e'}$, while the optimality of $OPT(S)$ indicates $w_{e'} > w_e$ (recall that $e \notin S$). ◀

3 Greedy Algorithm

This section introduces our greedy algorithm. Instead of letting the elements arrive one by one in n discrete time steps, we consider an equivalent continuous time model, in which each element's arrival time is drawn independently and uniformly between 0 and 1.¹ The continuous time model implies independence of different elements' arrival times, which leads to a simpler analysis.

We further define the following notations:

- Let $E(t)$ denote the set of elements that arrive from time 0 to t (inclusive).
- Let $OPT(t)$ be the optimal solution with respect to elements in $E(t)$.
- Let $OPT = OPT(1)$ denote the optimal solution with respect to all elements.

For some time threshold t_0 to be determined, our greedy algorithm rejects all elements that arrive before t_0 , like the classical algorithm by Dynkin [4]. For an element that arrives at time $t_0 < t \leq 1$, the algorithm selects it under two conditions: it belongs to the offline optimal solution with respect to the arrived elements in $E(t)$; and adding it to the selected elements gives an independent set.

Algorithm 1 Greedy.

input : matroid $\mathcal{M} = (E, \mathcal{I})$ and a threshold time t_0
output : set of selected elements ALG , initially set to be the empty set \emptyset .
for each element e that arrives at time $0 \leq t \leq 1$ **do**
 | **let** $ALG \leftarrow ALG \cup e$ **if** (1) $t > t_0$, (2) $e \in OPT(t)$, and (3) $ALG \cup e \in \mathcal{I}$.
end

Algorithm 1 is computationally efficient. In particular, $OPT(t)$ at any time t can be computed in polynomial time by processing the elements in $E(t)$ by descending order of weights, and selecting each element whenever independence is still satisfied.

¹ To simulate the continuous time model in the discrete-time model, we can first draw n arrival times $t_1 < t_2 < \dots < t_n$ between 0 and 1, and let t_i be the arrival time of the i -th element.

We next compare Algorithm 1 with the 9.6-competitive greedy algorithm by Ma et al. [12]. The only difference is the second condition for selecting an element e . Their algorithm checks if e is in the optimal solution with respect to $E(t_0) \cup \{e\}$, i.e., the elements that arrive before the threshold time t_0 and element e . By contrast, our algorithm examines $OPT(t)$, i.e., the optimal solution with respect to all arrived elements. By Lemma 3, elements in the optimal solution OPT would pass both criteria regardless of their arrival time. Ours is stricter, and hence, is less likely to select elements outside OPT .

This natural idea of checking local optimality was applied in Babaioff et al's [1] e -competitive *virtual algorithm* for uniform matroids, which maintains a reference set at each time t to record the current top- k elements and selects an element if it belongs to the reference set on arrival. Same idea also appeared in the e -competitive online algorithm for weighted bipartite matching given by Kesselheim et al [9].

4 Competitive Analysis

► **Theorem 4.** *Algorithm 1 (with $t_0 = 0.7$) is 4.75 probability-competitive for the Laminar Matroid Secretary Problem.*

Proof. Consider any element $e^* \in OPT$. It suffices to prove that Algorithm 1 selects e^* with probability at least $\frac{1}{4.75}$.

Since e^* would satisfy the second condition regardless of its arrival time, i.e., $e^* \in OPT(t)$ (Lemma 3), we consider the case when it arrives at time $t_0 < t^* \leq 1$, and lower bound the probability that all subsets containing e^* in the laminar family, denoted as $B_1 \subset B_2 \subset \dots \subset B_k$, still have unused capacities. We can assume without loss of generality that $c(B_i) < c(B_{i+1})$. Otherwise, we can remove B_i and its capacity and still have the same laminar matroid.

We first bound the failure probability for each subset B_i separately. That is, we consider the probability that the algorithm selects $c(B_i)$ elements from time t_0 to t^* . To avoid the complex dependence across different subsets, we will analyze a relaxed event. Consider the sub-laminar matroid restricted to elements in B_i , denoted as \mathcal{M}_{B_i} . We say that an element $e \in B_i$ arriving at $0 \leq t \leq 1$ *qualifies* (with respect to B_i) if it belongs to the optimal solution with respect to sub-matroid $\mathcal{M}_{B_i \cap E(t)}$, denoted as $OPT_i(t)$. By Lemma 3 with matroid $\mathcal{M}_{E(t)}$ and subset $B_i \cap E(t)$, any element $e \in B_i$ selected by the algorithm, in particular, any $e \in OPT(t) \cap B_i$, belongs to $OPT_i(t)$ and thus qualifies. Hence, it suffices to upper bound the probability that at least $c(B_i)$ elements in B_i qualify from time t_0 to t .

To do so, we next establish two lemmas that characterize the arrival times of qualified elements. For ease of presentation, we will assume without loss of generality that $|OPT_i(t)| = c(B_i)$ at all time, by adding infinitely many dummy elements with negligible weights.

► **Lemma 5.** *Fix any time $0 < t \leq 1$ and subset B_i in the laminar family. Let t_{-1} be the arrival time of the last qualified element of B_i from time 0 to t (exclusive). Then, $\ln \frac{t}{t_{-1}}$ follows the exponential distribution with rate parameter $c(B_i)$, i.e., $\Pr \left[\ln \frac{t}{t_{-1}} \leq x \right] = 1 - e^{-c(B_i)x}$.*

Proof of Lemma 5. We first rewrite the cumulative distribution function:

$$\Pr \left[\ln \frac{t}{t_{-1}} \leq x \right] = 1 - \Pr \left[\ln \frac{t}{t_{-1}} > x \right] = 1 - \Pr \left[t_{-1} < e^{-x}t \right] .$$

It remains to show that:

$$\Pr \left[t_{-1} < e^{-x}t \right] = e^{-c(B_i)x} .$$

Let $E(t^-)$ be the elements that arrive from time 0 to t (exclusive). Accordingly, let $OPT_i(t^-)$ be the optimal solution with respect to sub-matroid $\mathcal{M}_{E(t^-) \cap B_i}$. We will prove that t_{-1} is the arrival time of the last element in $OPT_i(t^-)$. In other words, the last element in $OPT_i(t^-)$ is the last qualified element.

On the one hand, by Lemma 3 with matroid $\mathcal{M}_{E(t^-) \cap B_i}$, any element in $OPT_i(t^-)$ must qualify. On the other hand, consider any element e' that arrives at some time t' after all elements in $OPT_i(t^-)$ arrive. This means $OPT_i(t^-) \subseteq E(t')$. Further by Lemma 3 with matroid $\mathcal{M}_{E(t^-) \cap B_i}$, we get that $OPT_i(t^-) = OPT_i(t^-) \cap E(t') \subseteq OPT_i(t')$. This implies $OPT_i(t^-) = OPT_i(t')$ and thus $e' \notin OPT_i(t')$. That is, e' cannot qualify.

The lemma now follows because the probability that all $c(B_i)$ elements in $OPT_i(t^-)$ arrive before $e^{-x}t$ equals $e^{-c(B_i)x}$, where these elements' arrival times distribute independently and uniformly between time 0 and t (exclusive). ◀

► **Lemma 6.** *Fix any time t and subset B_i in the laminar family. Let t_{-k} be the arrival time of the k -th last qualified element of B_i from time 0 and t (exclusive). Then, $\ln \frac{t}{t_{-c(B_i)}}$ follows the Gamma distribution with shape and rate parameters equal to $c(B_i)$.*

To be self-contained, we include below the cumulative function of Gamma distribution with shape parameter α and rate parameter β :

$$F(x; \alpha, \beta) = \int_0^x \frac{y^{\alpha-1} e^{-\beta y} \beta^\alpha}{(\alpha-1)!} dy .$$

We refer interested readers to Chapter 8.4 of Blitzstein and Hwang [3] for further information about Gamma distributions.

Proof of Lemma 6. Observe that

$$\ln \frac{t}{t_{-c(B_i)}} = \ln \frac{t}{t_{-1}} + \ln \frac{t_{-1}}{t_{-2}} + \cdots + \ln \frac{t_{-c(B_i)+1}}{t_{-c(B_i)}} .$$

By Lemma 5, $\ln \frac{t}{t_{-1}}$ follows the exponential distribution with rate parameter $c(B_i)$. Further note that conditioned on t_{-1} and the subset of elements that arrive before t_{-1} , these elements independently and uniformly arrive from 0 to t_{-1} (exclusive). Hence, by Lemma 5 with $t = t_{-1}$, we get that $\ln \frac{t_{-1}}{t_{-2}}$ follows the exponential distribution with rate parameter $c(B_i)$, independent to $\ln \frac{t}{t_{-1}}$. Repeating this argument shows that $\ln \frac{t}{t_{-c(B_i)}}$ is the sum of $c(B_i)$ independent exponential random variables with rate parameter $c(B_i)$, and thus follows the Gamma distribution with shape and rate parameters equal to $c(B_i)$.² ◀

We are now ready to upper bound the probability that at least $c(B_i)$ elements in B_i qualify from time t_0 to t^* , because B_i has at least $c(B_i)$ qualified elements arrive from time t_0 to t^* if and only if $t_{-c(B_i)} > t_0$. As a corollary of Lemma 6, the probability of both events equals $F(\ln \frac{t^*}{t_0}; c(B_i), c(B_i))$. Further by union bound, and by that $c(B_1) < c(B_2) < \cdots < c(B_k)$, with probability at least $1 - \sum_{i=1}^{\infty} F(\ln \frac{t^*}{t_0}, i, i)$ we have that less than $c(B_i)$ qualified elements arrive from time t_0 to t for any B_i . Hence, Algorithm 1 selects element e^* with probability at least:

$$\int_{t_0}^1 1 - \sum_{i=1}^{\infty} F\left(\ln \frac{t^*}{t_0}, i, i\right) dt^* .$$

² The fact that the sum of exponential distributions with the same parameter forms a Gamma distribution can be found, e.g., in Theorem 8.4.3 of Blitzstein and Hwang [3].

The theorem follows as one can verify numerically that it is at least $\frac{1}{4.75}$ when $t_0 = 0.7$.

For completeness, we next present the details of the numerical verification. We divide the infinite sum of cumulative distribution functions into two parts:

$$\sum_{i=1}^{3000} F\left(\ln \frac{t^*}{t_0}, i, i\right) \quad \text{and} \quad \sum_{i=3001}^{\infty} F\left(\ln \frac{t^*}{t_0}, i, i\right).$$

The first part is finite and hence can be calculated numerically.³ We next upper bound the second part. Recall that the cumulative distribution function of Gamma distribution is:

$$F(x, i, i) = \int_0^x \frac{y^{i-1} e^{-iy} i^i}{(i-1)!} dy = \int_0^x \frac{i^i}{i!} e^{-iy} dy^i.$$

By $e^{-iy} \leq 1$, this integral is bounded from above by $\frac{x^i i^i}{i!}$, which is further upper bound by $(xe)^i$ because $i! \geq i^i e^{-i}$ by Stirling's approximation. Hence:

$$\sum_{i=3001}^{\infty} F\left(\ln \frac{t^*}{t_0}, i, i\right) < \sum_{i=3001}^{\infty} \left(e \ln \frac{t^*}{t_0}\right)^i,$$

which is a geometric sequence. The convergence condition requires that $e \ln \frac{t^*}{t_0} < 1$, where $t^* \in [0, 1]$. This restricts our choice of t_0 to be greater than $e^{-1/e} \approx 0.692$. On the other hand, delaying the threshold time to a very late stage will lead to significant loss by missing all the optimal elements arriving before t_0 . Numerical calculations suggest choosing $t_0 = 0.7$ to utilize the success probability,³ with which we have:

$$\sum_{i=3001}^{\infty} \left(e \ln \frac{t^*}{t_0}\right)^i < \sum_{i=3001}^{\infty} \left(e \ln \frac{1}{0.7}\right)^i < 10^{-38}.$$

Hence, the probability that Algorithm 1 selects element e^* is at least:

$$\int_{0.7}^1 1 - \sum_{i=1}^{\infty} F\left(\ln \frac{t^*}{0.7}, i, i\right) dt^* > \int_{0.7}^1 1 - \sum_{i=1}^{3000} F\left(\ln \frac{t^*}{0.7}, i, i\right) - 10^{-38} dt^* > \frac{1}{4.75}. \quad \blacktriangleleft$$

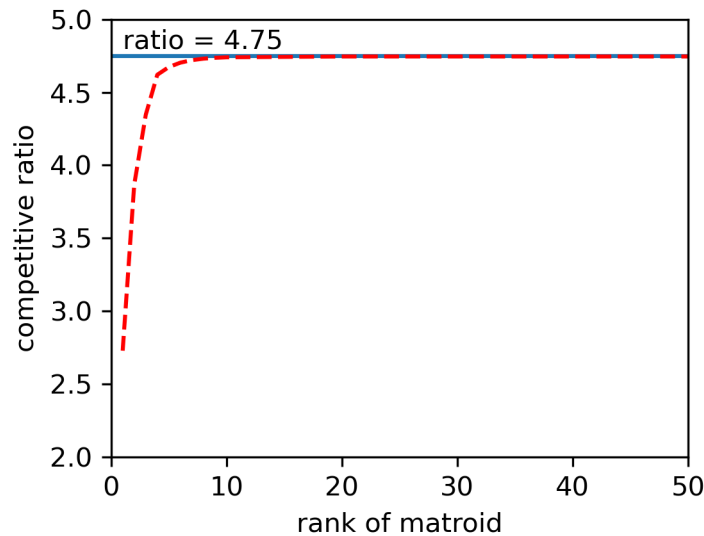
5 Conclusion

We conclude the article with a few remarks. First, our algorithm is 4.75 probability competitive, i.e., any element from the optimal solution is selected with probability at least $\frac{1}{4.75}$. This is slightly stronger than the standard notion of competitive analysis, a.k.a., utility competitiveness in the context of Matroid Secretary Problem.

Further, our algorithm works in the ordinal model posed by Soto et al. [15], where the decision-maker does not have numerical information about elements' weights, but only knows their relative ranks.

Finally, some readers may notice that our analysis has underestimated the algorithm's probability of selecting an element from the optimal solution, because the union bound sums over all possible capacities from 1 to infinity, instead of, say, to the rank of the matroid. Nevertheless, the ratio does not change significantly even if we only sum to 10. The competitive ratio given by our method for different ranks, with t_0 optimized for each rank, is shown in the following figure.³ When rank = 1, Algorithm 1 is e probability competitive by taking $t_0 = \frac{1}{e}$, matching the results of Dynkin [4] for the original Secretary Problem.

³ The numerical calculations are conducted in Python. Interested readers may find the source codes through: <https://github.com/ZixuannnZhu/Laminar-Matroid-Secretary>.



References

- 1 Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Matroid secretary problems. *J. ACM*, 65(6):35:1–35:26, 2018. doi:10.1145/3212512.
- 2 Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 434–443, 2007. URL: <http://dl.acm.org/citation.cfm?id=1283383.1283429>.
- 3 Joseph K Blitzstein and Jessica Hwang. *Introduction to probability*. Crc Press, 2019.
- 4 Evgenii B Dynkin. Optimal choice of the stopping moment of a markov process. In *Doklady Akademii Nauk*, volume 150, pages 238–240. Russian Academy of Sciences, 1963.
- 5 Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $O(\log \log(\text{rank}))$ -competitive algorithm for the matroid secretary problem. *Math. Oper. Res.*, 43(2):638–650, 2018. doi:10.1287/MOOR.2017.0876.
- 6 Thomas S. Ferguson. Who solved the secretary problem. *Statistical Science*, 4:282–289, 1989.
- 7 Sungjin Im and Yajun Wang. Secretary problems: Laminar matroid and interval scheduling. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1265–1274, 2011. doi:10.1137/1.9781611973082.96.
- 8 Patrick Jaillet, José A. Soto, and Rico Zenklusen. Advances on matroid secretary problems: Free order model and laminar case. In *Integer Programming and Combinatorial Optimization - 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, pages 254–265, 2013. doi:10.1007/978-3-642-36694-9_22.
- 9 Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 589–600, 2013. doi:10.1007/978-3-642-40450-4_50.
- 10 Oded Lachish. $O(\log \log \text{rank})$ competitive ratio for the matroid secretary problem. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 326–335, 2014. doi:10.1109/FOCS.2014.42.
- 11 Dennis V. Lindley. Dynamic programming and decision theory. *Journal of The Royal Statistical Society Series C-applied Statistics*, 10:39–51, 1961.

73:8 Laminar Matroid Secretary: Greedy Strikes Back

- 12 Tengyu Ma, Bo Tang, and Yajun Wang. The simulated greedy algorithm for several submodular matroid secretary problems. *Theory Comput. Syst.*, 58(4):681–706, 2016. doi:10.1007/S00224-015-9642-4.
- 13 James G Oxley. *Matroid Theory*. Oxford University Press, 2006.
- 14 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.
- 15 José A. Soto, Abner Turkieltaub, and Victor Verdugo. Strong algorithms for the ordinal matroid secretary problem. *Math. Oper. Res.*, 46(2):642–673, 2021. doi:10.1287/MOOR.2020.1083.