SIAM J. COMPUT.
Vol. 42, No. 1, pp. 178–211
© 2013 Society for Industrial and Applied Mathematics

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

MATROID SECRETARY PROBLEM IN THE RANDOM-ASSIGNMENT MODEL*

JOSÉ A. SOTO†

Abstract. The *matroid secretary problem* admits several variants according to the order in which the matroid's elements are presented and how the elements are assigned weights. As the main result of this article, we devise the first constant competitive algorithm for the model in which both the order and the weight assignment are selected uniformly at random, achieving a competitive ratio of approximately 5.7187. This result is based on the nontrivial fact that every matroid can be approximately decomposed into uniformly dense minors. Based on a preliminary version of this work, Oveis Gharan and Vondrák [*Proceedings of the* 19*th Annual European Symposium on Algorithms, ESA*, 2011, pp. 335–346] devised a $40e/(e-1)$-competitive algorithm for the stronger random-assignment adversarial-order model. In this article we present an alternative algorithm achieving a competitive ratio of $16e/(e-1)$. As additional results, we obtain new algorithms for the standard model of the matroid secretary problem: the adversarial-assignment random-order model. We present an $O(\log r)$-competitive algorithm for general matroids which, unlike previous ones, uses only *comparisons* among seen elements. We also present constant competitive algorithms for various matroid classes, such as column-sparse representable matroids and low-density matroids. The latter class includes, as a special case, the duals of graphic matroids.

Key words. secretary problems, matroids, principal partition, online selection

AMS subject classifications. 68W27, 05B35, 68W40

DOI. 10.1137/110852061

1. Introduction.

In the simplest form of the *secretary problem*, an employer wants to select the best secretary among n applicants arriving in random order. Once a secretary is interviewed, the employer must immediately decide whether to accept or reject the applicant. Say that a secretary is a *record* if he or she is the best applicant seen so far. If one is only interested in selecting the *best secretary*, then the only strategies that make sense are those that outputs records. Lindley [20] and Dynkin [11] have shown that the strategy that rejects the first $\lfloor n/e \rfloor$ candidates and then selects the first arriving record (if any) has a probability of at least $1/e$ of selecting the best secretary. They have also shown that no algorithm can beat this constant.

An important generalization of this problem is known as the *multiple choice secretary problem* (see [17]). The objective in this problem is to select a group of at most k secretaries from a pool of n applicants having a combined value as large as possible.

Babaioff et al. [4] introduce the *generalized secretary problem* as a natural class of extensions of the above setting in which the set returned by the algorithm must obey certain combinatorial restrictions. In this problem, we are given a finite set E with hidden nonnegative weights and a collection of *feasible sets* $\mathcal{I} \subseteq 2^E$ closed under inclusion. The collection $\mathcal{I}$, known as the *domain* of the problem, describes the sets of elements that can be simultaneously accepted. The elements of E are presented to an online algorithm in random order. When an element is revealed, the algorithm learns

*Received by the editors October 18, 2011; accepted for publication (in revised form) October 4, 2012; published electronically January 29, 2013. An extended abstract of this article appeared in the Proceedings of SODA 2011. This work was supported, at different stages, by NSF contract CCF-0829878, ONR grant N00014-05-1-0148, and Nucleo Milenio Información y Coordinación en Redes ICM/FIC P10-024F.
http://www.siam.org/journals/sicomp/42-1/85206.html

†Departamento de Ingeniería Matemática & Centro de Modelamiento Matemático (UMI 2807 CNRS), Universidad de Chile, Santiago, Chile (jsoto@dim.uchile.cl).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

its weight and decides whether or not to accept it while keeping the set of accepted elements feasible at every step. This decision must be made before the next element is revealed. The objective is to output a feasible set of maximum total weight. The condition that the elements are presented in random order is not mandatory: we can also consider variants in which the order is selected adversarially.

We remark that other lines of generalizations of the multiple choice secretary problem with different objective functions have been considered before. These include, among others, minimizing the sum of the relative ranks of the selected elements (studied by Ajtai, Megiddo, and Waarts [1]), the weighted and time discounted secretary problems of Babaioff et al. [2], the J-choice K-best secretary problem studied by Buchbinder, Jain, and Singh [7], and the submodular secretary problem of Bateni, Hajiaghayi, and Zadimoghaddam [6].

The generalized secretary problem is of interest due to its connection to online auctions. In both the original and multiple choice secretary problems, we can regard the algorithm as an auctioneer having one or many identical items, and the secretaries as agents arriving at random times, each having a different valuation for the item. The goal of the algorithm is to assign the items to the agents as they arrive while maximizing the total social welfare. In more complex situations, the auctioneer may have access to a collection of goods that it wishes to assign to agents, subject to some restrictions. In many cases, these restrictions can be modeled by matroid constraints. For that reason, the *matroid secretary problem*, in which the feasible sets are the independent sets of a matroid, is of special interest (see, e.g., [4]).

The difficulty of the matroid secretary problem changes depending on the information available beforehand about the matroid, the weights, and the order in which the elements are presented. In this work, we mostly restrict our attention to those settings in which the matroid is fully known to the algorithm beforehand. Following Oveis Gharan and Vondrák [26] we classify models of the matroid secretary problems as follows:

1. *Assignment of weights:* An adversary selects a list $W = \{w_1, \dots, w_n\}$ of weights hidden to the algorithm which are then assigned to the elements of E.

(i) *Adversarial-assignment*: The list W is assigned to E via an adversarial bijection.

(ii) *Random-assignment*: The list W is assigned to E via a uniform random permutation. In other words, every element of e receives a unique, randomly chosen, weight from W, disallowing repetitions.

2. *Order of the elements:*

(i) *Adversarial-order*: An adversary selects the ordering of the elements. This order is unknown to the algorithm.

(ii) *Random-order*: The elements are presented in a uniform random order.

There are four models arising from combining the variants above. We assume that the random choices are performed after the adversarial choices. For example, in the adversarial-assignment random-order model, the adversary selecting the weights does not know the order in which the elements are going to be presented. If randomness is involved for both parameters, we assume the choices to be independent. The main variant studied in the literature is the adversarial-assignment random-order model. We call this the *standard* model of the matroid secretary problem.

It is worth noting that for both the classical and the multiple choice secretary problems, the three models involving randomness coincide; therefore, these three models of the matroid secretary problem are direct extensions of the classical and multiple

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

choice secretary problems, where the considered matroid is uniform.

A natural scenario to consider is when each element receives a weight independent and identically distributed (i.i.d.) from a (known or unknown) fixed distribution at the moment it is presented to the algorithm. This scenario can be seen as a particular case of random-assignment, in which the list W consists of weights selected i.i.d. from that distribution beforehand. We denote as *full information setting* (resp., *partial information setting*) the model obtained in the way just described when the distribution is known to the algorithm beforehand (resp., if it is not known).

The difficulty of the matroid secretary problem also changes depending on whether the algorithm learns the actual weight of the elements or just the relative order of the weights seen so far. See the surveys of Freeman [14] and Ferguson [13] for variations of the classical secretary problem according to this parameter.

There is a long line of work on the standard model of the matroid secretary problem. Constant competitive algorithms are known for partition matroids (corresponding to the classical [20, 11] and multiple choice secretary problems [17, 3]), transversal matroids [5, 10, 18], graphical matroids [5, 18, 2], and laminar matroids [16]. It is also known [5] that if a matroid admits a constant competitive algorithm in the standard model, then so do its restrictions and truncations. The best known competitive ratio for general matroids is $O(\sqrt{\log r})$, where r is the rank function of the matroid. This recent result by Chakraborty and Lachish [9] improves on the previously best $O(\log r)$-competitive algorithm due to Babaioff, Immorlica, and Kleinberg [5].

Nonmatroidal domains have also been considered in the literature. Babaioff et al. [3] show a $10e$-competitive algorithm for knapsack domains even for the case where both the weights and lengths are revealed online. Korula and Pál [18] give constant competitive algorithms for certain intersections of partition matroids in the standard model, namely, for matchings in hypergraphs whose edges have constant size.

Babaioff, Immorlica, and Kleinberg [5] have shown a particular domain for which no algorithm has a competitive ratio smaller than $o(\log n/\log\log n)$ even in a random-order model with full information setting. However, matroid domains have the following property: If we are allowed to reject elements which have been previously accepted, while keeping at every moment an independent set, then it is possible to output the optimum independent set no matter in which order the elements are presented. This intuition motivated Babaioff et al. (see [5, 4]) to conjecture that the matroid secretary problem admits a constant competitive algorithm, provided that at least one of the order or the assignment is selected at random.

1.1. Main results. In this paper, we partially solve the above conjecture, devising a constant-competitive algorithm (cf. Algorithm 4 in section 4) for the random-assignment random-order model.

On a very high level our algorithm is based on a simple divide and conquer approach: replace the matroid by a collection of matroids of a simpler class, apply a constant-competitive algorithm in each one, and then return the union of the answers. The simpler matroids we use are known as uniformly dense matroids.

Uniformly dense matroids are those for which the *density* of a set, i.e., the ratio of its cardinality to its rank, is maximized on the entire ground set. The simplest examples are precisely the uniform matroids. We show that uniformly dense matroids and uniform matroids of the same rank behave similarly, in the sense that the distribution of the rank of a random set is similar for both matroids. We use this fact to devise a constant-competitive auxiliary algorithm (cf. Algorithm 2 in section 3) for uniformly dense matroids.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

In order to extend the auxiliary algorithm to general matroids we exploit some notions coming from the *theory of principal partition* of a matroid, particularly its *principal sequence*. Roughly speaking, the principal sequence of a matroid $\mathcal{M}$ is a decomposition of its ground set into a sequence of parts, each of which is the ground set of a uniformly dense minor of $\mathcal{M}$. Furthermore, if we select one independent set in each of these minors, their union is guaranteed to be independent in $\mathcal{M}$. By employing separately the auxiliary algorithm on each of these minors, we can return an independent set of $\mathcal{M}$, while only increasing an extra factor of $e/(e-1)$ on the competitive ratio. By comparing the weight of our solution to the optimum of certain randomly defined matroids, we obtain a tighter analysis for the competitive ratio.

It is worth noting that a slightly weaker $2e^2/(e-1) \approx 8.6$-competitive algorithm for this task has already been presented by the author in a conference version of this paper [29]. Both presented algorithms are different but share the same divide and conquer approach explained above. We remark that they are also the first algorithms achieving a constant competitive ratio even for the weaker partial and full information settings (under random-order) of the matroid secretary problem.

As first noticed by Oveis Gharan and Vondrák [26] after the publication of a first draft of this article, it is possible to apply the above methods to the stronger random-assignment adversarial-order model. They devise a 40-competitive algorithm for uniformly dense matroids in this model. By using our techniques, they modify it to obtain a $40/(1-1/e)$-competitive algorithm for general matroids. In this article, we also present an alternative algorithm for that model achieving a competitive ratio of $16/(1-1/e)$.

Our study of random-assignment models is organized as follows. In section 2, we describe formally the problem and the divide and conquer approach. In section 3, we focus on uniformly dense matroids and give algorithms for both random-assignment models. In section 4, we present algorithms working on general matroids. To analyze them, we use some technical results that are described later in sections 5 and 6.

For the reader's convenience, we include below a list of our main algorithmic results, in the order in which we prove them.

THEOREM 1.1. *There is a* 4.92078*-competitive algorithm (cf. Algorithm* 2 *in subsection* 3.1*) for the random-assignment random-order model on uniformly dense matroids.*

THEOREM 1.2. *There is a* $16/(1-1/e)$*-competitive algorithm (cf. Algorithm* 3 *in subsection* 3.2*) for the random-assignment adversarial-order model on uniformly dense matroids.*

THEOREM 1.3. *There is a* 5.7187*-competitive algorithm (cf. Algorithm* 4 *in subsection* 4.1*) for the random-assignment random-order model on general matroids.*

THEOREM 1.4. *There is a* $16/(1-1/e)$*-competitive algorithm (cf. Algorithm* 5 *in subsection* 4.2*) for the random-assignment adversarial-order model on general matroids.*

1.2. Additional results. Babaioff et al.'s conjecture is still open for the standard adversarial-assignment random-order model. In section 7, we present simple algorithms for various matroid classes working on this model. We show a ke-competitive algorithm for the case in which the matroid is representable by a matrix having at most k nonzero entries per column. This result generalizes the $2e$-competitive algorithm for graphic matroids of Korula and Pál [18]. We also give algorithms for general matroids having competitive ratios proportional to the density of the matroid. Using this, we obtain a $3e$-competitive algorithm for cographic matroids, and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

a k-competitive algorithm for matroids where each element is in a cocircuit of size at most k.

For general matroids, we give a new $O(\log r)$-competitive algorithm. Unlike the $O(\log r)$-competitive algorithm of Babaioff, Immorlica, and Kleinberg [5] and the $O(\sqrt{\log r})$-competitive algorithm of Chakraborty and Lachish [9], ours does not use the numerical value of the weights. It only needs the ability to make comparisons among seen elements. This is a desirable property since the features revealed by the elements may be of qualitative type (for example, the *qualifications* of a person applying for a job), but the actual value or profit may be an unknown increasing function of the features revealed. In fact, all the algorithms presented in this article have the mentioned desirable property.

2. Preliminaries. We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$, $\Pr(\mathcal{E})$ to denote the probability of a given event $\mathcal{E}$, and $\mathbb{E}[X]$ to denote the expected value of a random variable X. We use subindices on $\Pr(\cdot)$ and $\mathbb{E}[\cdot]$ to be specific about the probability space over which the probability or expectation is taken. We assume familiarity with basic concepts in matroid theory. For an introduction, we refer the reader to Oxley's book [27].

Consider a matroid $\mathcal{M} = (E, \mathcal{I})$ with ground set $E = \{e_1, \ldots, e_n\}$. An adversary selects a set W of n nonnegative weights $w_1 \geq \cdots \geq w_n \geq 0$, which are assigned to the elements of the matroid using an *ordering* of E,

$$\sigma(1), \sigma(2), \sigma(3), \ldots, \sigma(n),$$

defined by a bijective map $\sigma\colon [n] \to E$; i.e., the weight assignment is given by

$$w(\sigma(i)) = w_i, \quad i \in [n].$$

The elements are then presented to an online algorithm in the order

$$\pi(1), \pi(2), \pi(3), \ldots, \pi(n),$$

defined by a certain bijective function $\pi\colon [n] \to E$. When an element is presented, the algorithm must decide whether to add it to the current solution set, denoted as ALG, under the condition that this set is independent (ALG $\in \mathcal{I}$) at all times. The set ALG is returned after all elements have been presented. The objective is to output a set whose *payoff*, defined as $w(\text{ALG}) = \sum_{e \in \text{ALG}} w(e)$, is as high as possible.

Depending on how the assignment σ and the ordering π are selected, we recover the four models discussed in the introduction. Each of σ and π can be selected in an adversarial way or uniformly at random. We further assume that when the ith element of the stream, $\pi(i)$, is presented, *the algorithm only learns the relative order of the current weight with respect to the previously seen ones*; i.e., it can compare $w(\pi(j))$ with $w(\pi(k))$ for all $j, k \leq i$, but it cannot use the numerical weight values. Without loss of generality, we assume that there are no ties in W, because otherwise we can break them using a random bijection $\tau\colon E \to [n]$ (independent of W, σ, and π): if there is a tie between two seen elements, we consider heavier the one having larger τ-value.

To analyze the performance of the algorithm, we use its competitive ratio. This quantity is usually defined as the ratio between the maximum possible payoff and the algorithm's payoff. Since there is randomness involved, expected values are used. However, since the payoff can be zero, the previous ratio is not always well defined. For that reason, we use the following definition: We say that a randomized algorithm

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

returning a set ALG is *α-competitive* for some constant $\alpha \geq 1$ if, for any adversarial selection that the model allows,

$$\mathbb{E}[\alpha w(\text{ALG}) - w^*] \geq 0. \tag{2.1}$$

In the above expression, the expectation is taken over both the random choices given by the model and the random choices performed by the algorithm; and w^* is the maximum payoff of a feasible set under the current realization of the weights. The *competitive ratio* of the algorithm is the minimum value of α for which this algorithm is α-competitive.

Provided that $\mathbb{E}[w^*] \neq 0$, we can rewrite (2.1) in the more familiar form

$$\frac{\mathbb{E}[w(\text{ALG})]}{\mathbb{E}[w^*]} \leq \frac{1}{\alpha}. \tag{2.2}$$

An interesting aspect of random-assignment models is that the quantity w^* is a random-variable itself. So, instead of studying the ratio of the expected outcomes $\frac{\mathbb{E}[w(\text{ALG})]}{\mathbb{E}[w^*]}$, one could also study the expectation of the ratio, i.e., $\mathbb{E}\big[\frac{w(\text{ALG})}{w^*}\big]$. The latter notion is, in the author's opinion, not as robust as the first one. For instance, in most cases $\mathbb{E}\big[\frac{w(\text{ALG})}{w^*}\big]^{-1} \neq \mathbb{E}\big[\frac{w^*}{w(\text{ALG})}\big]$, so people who define competitive ratios for online maximization problems as quantities smaller than one and those who define them as quantities bigger than one would have trouble translating results from one setting to the other. In this article we restrict ourselves to the ratio-of-expectation notion. However, analyzing random-assignment models, or any model in which w^* is a random variable, using an expectation-of-ratio notion of competitiveness seems like a good problem to work out in the future.

An important observation to keep in mind for the rest of this article is that even though the maximum payoff w^* depends on both W and σ, the *set* achieving this maximum payoff depends only on σ. Indeed, let $\text{OPT}_{\mathcal{M}}(\sigma)$ be the *lexicographic first base* of M under ordering σ. In other words, $\text{OPT}_{\mathcal{M}}(\sigma)$ is the set obtained by applying the *greedy procedure* that includes an element if it can be added to the previously included ones while preserving independence in $\mathcal{M}$, on the sequence $\sigma(1), \sigma(2), \ldots, \sigma(n)$. Standard matroid arguments imply that $\text{OPT}_{\mathcal{M}}(\sigma)$ is a maximum independent set with respect to any weight function v for which $v(\sigma(1)) \geq \cdots \geq v(\sigma(n)) \geq 0$. In particular, this is true for the weight function w defined before and $\mathbb{E}[w^*] = \mathbb{E}_\sigma[w(\text{OPT}_{\mathcal{M}}(\sigma))]$.

2.1. Divide and conquer. Given two or more matroids $\{\mathcal{M}_i = (E_i, \mathcal{I}_i)\}_i$ with disjoint ground sets, their *direct sum* $\bigoplus_i \mathcal{M}_i$ is the matroid with ground set $\bigcup_i E_i$ such that a set I is independent if for all i the set $E_i \cap I$ is independent in $\mathcal{M}_i$.

Consider this *divide and conquer* approach for any model of the matroid secretary problem: Given $\mathcal{M} = (E, \mathcal{I})$, find a (possibly random) family $\mathcal{F} = \{\mathcal{M}_i = (E_i, \mathcal{I}_i)\}_i$ of matroids with disjoint ground sets satisfying the next three properties.

(i) Every independent set of $\bigoplus_i \mathcal{M}_i$ is independent in $\mathcal{M}$.

(ii) For each matroid $\mathcal{M}_i$, we have access to an α-competitive algorithm $\mathcal{A}_i$, with $\alpha \geq 1$ for the same model of the matroid secretary problem.

(iii) There is a constant $\beta \geq 1$ such that the (expected) maximum weight of an independent set in $\bigoplus_i \mathcal{M}_i$ is at least $1/\beta$ times the (expected) maximum weight of an independent set in $\mathcal{M}$.

PROPOSITION 1. *Let $\mathcal{F} = \{\mathcal{M}_i = (E_i, \mathcal{I}_i)\}_i$ be a family of disjoint matroids satisfying the above properties; then $\mathcal{M}$ admits an $\alpha\beta$-competitive algorithm.*

Proof. Consider the procedure that runs algorithm $\mathcal{A}_i$ on each matroid $\mathcal{M}_i$ in parallel and returns the union of their answers. This procedure is $\alpha\beta$-competitive. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

When the family $\mathcal{F}$ contains only uniform matroids, this approach corresponds exactly to the β-partition property defined by Babaioff et al. in [2]. They show that the standard matroid secretary problem can be *reduced*, for certain classes of matroids, to the case of partition matroids, i.e., to disjoint sums of uniform matroids. In particular, they use this approach to unify the results of [5].

In the following sections we use a slight variation of the divide and conquer approach explained above to devise constant-competitive algorithms for random-assignment models of the matroid secretary problem.

3. Uniformly dense matroids. Define the *density* $\gamma(\mathcal{M})$ of a loopless matroid[1] $\mathcal{M} = (E, \mathcal{I})$ with rank function $\mathrm{rk}\colon 2^E \to \mathbb{Z}_+$ as the maximum over all nonempty sets $|X|$ of the quantity $|X|/\mathrm{rk}(X)$. The matroid $\mathcal{M}$ is *uniformly dense* if $\gamma(\mathcal{M})$ is attained by the entire ground set, that is, if $\frac{|X|}{\mathrm{rk}(X)} \leq \frac{|E|}{\mathrm{rk}(E)}$ for every nonempty $X \subseteq E$. Examples of uniformly dense matroids include uniform matroids, the graphic matroid of a complete graph, and all projective geometries $PG(r-1, \mathbb{F})$. The following property of uniformly dense matroids is important for our analysis.

LEMMA 3.1. *Let $(x_1, \ldots, x_j)$ be a sequence of different elements of a uniformly dense matroid of total rank r, chosen uniformly at random. The probability that element x_j is selected by the greedy procedure on that sequence is at least $1-(j-1)/r$.*

Proof. An element is selected by the greedy procedure only if it is outside the span[2] of the previous elements. Let $A_j = \{x_1, \ldots, x_j\}$ denote the set of the first j elements of the sequence, and let n be the number of elements of the matroid. Then

$$\Pr[x_j \text{ is selected}] = \mathbb{E}\left[\frac{n - |\operatorname{span}(A_{j-1})|}{n-(j-1)}\right] \geq \frac{n - \mathbb{E}[\mathrm{rk}(A_{j-1})]n/r}{n-(j-1)} \geq 1 - \frac{j-1}{r},$$

where the first equality follows because element x_j is chosen uniformly at random, the next inequality holds because the matroid is uniformly dense, and the last one holds because the rank of a set is always at most its cardinality. ∎

For a simple application of the previous lemma, consider a uniform random set X of j elements. The rank of $X = \{x_1, \ldots, x_j\}$ equals the cardinality of the set returned by the greedy procedure on any ordering of its elements. By Lemma 3.1,

$$\mathbb{E}[\mathrm{rk}(X)] \geq \sum_{i=1}^{j} \max\left\{\left(1 - \frac{i-1}{r}\right), 0\right\} = \sum_{i=1}^{\min\{j,r\}} \left(1 - \frac{i-1}{r}\right). \tag{3.1}$$

Note that if $j \leq r$, the right-hand side is $j - \frac{j(j-1)}{2r} \geq j/2$, and if $j \geq r$, the right-hand side is $r - \frac{r(r-1)}{2r} \geq r/2$. In any case, the expected rank of X is at least $\min\{j, r\}/2$. This shows that the rank of a random set in a uniformly dense matroid is close to what it would be were the matroid uniform. The following lemma tightens this bound.

LEMMA 3.2. *Let X be a set of a fixed cardinality j whose elements are chosen uniformly at random from a uniformly dense matroid. Then*

$$\mathbb{E}[\mathrm{rk}(X)] \geq r\left(1 - \left(1 - \frac{1}{r}\right)^j\right) \geq r\left(1 - e^{-j/r}\right). \tag{3.2}$$

[1] A *circuit* is a minimal nonindependent set of a matroid. A *loop* is an element e such that $\{e\}$ is a circuit. A loopless matroid is a matroid having all singletons independent.

[2] The *span* of a set $A \subseteq E$ is defined as the set of elements that, when added to A, do not increase its rank, i.e., $\operatorname{span}(A) = \{x \in E \colon \mathrm{rk}(A \cup \{x\}) = \mathrm{rk}(A)\}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

In particular, $\mathbb{E}[\mathrm{rk}(X)] \geq \min\{j, r\}\left(1 - \frac{1}{e}\right)$.

Proof. Let $(x_1, \ldots, x_n)$ be a random ordering of the elements of E, and let X_j be the set $\{x_1, \ldots, x_j\}$. As j increases, the rank of X_j increases by one unit every time x_j is outside the span of the previous elements. Then, for all $1 \leq j \leq n$,

$$\mathbb{E}[\mathrm{rk}(X_j)] - \mathbb{E}[\mathrm{rk}(X_{j-1})] = \Pr(x_j \notin \mathrm{span}(X_{j-1})) = \mathbb{E}\left[\frac{n - |\mathrm{span}(X_{j-1})|}{n - |X_{j-1}|}\right]$$
$$\geq \mathbb{E}\left[\frac{n - \mathrm{rk}(X_{j-1})n/r}{n-(j-1)}\right] \geq 1 - \frac{\mathbb{E}[\mathrm{rk}(X_{j-1})]}{r}.$$

Therefore, the sequence $Z_j = \mathbb{E}[\mathrm{rk}(X_j)]$, for $j = 0, \ldots, n$, satisfies

$$Z_0 = 0; \quad \text{and} \quad Z_j \geq 1 + Z_{j-1}\left(1 - \frac{1}{r}\right) \text{ for } j \geq 1.$$

Solving the previous recurrence yields

$$Z_j \geq \sum_{i=0}^{j-1}\left(1 - \frac{1}{r}\right)^i = r\left(1 - \left(1 - \frac{1}{r}\right)^j\right) \geq r(1 - e^{-j/r}). \tag{3.3}$$

The function $(1 - e^{-x})$ is increasing; thus if $j \geq r$, $Z_j \geq r\left(1 - e^{-j/r}\right) \geq r\left(1 - e^{-1}\right)$. In addition, the function $(1 - e^{-x})/x$ is decreasing; thus if $j \leq r$, $Z_j \geq j\frac{\left(1-e^{-j/r}\right)}{j/r} \geq j(1 - e^{-1})$. ∎

3.1. Algorithm for the random-assignment random-order model. Lindley [20] and Dynkin [11] have given a very simple e-competitive algorithm for the classical secretary problem: Observe and reject the first $\lfloor n/e \rfloor$ candidates and then select the first arriving record (if any). In other words, we select the first element whose value is higher than all the previous ones.

Our constant-competitive algorithm for uniformly dense matroids is based on the same idea. Before explaining the details, it is useful to first modify the Lindley–Dynkin algorithm by choosing the number of elements to "observe and reject" as a binomial random variable with expectation np. This modification is very standard and has proven to be useful for the matroid secretary problem (see, e.g., Kleinberg's algorithm [17] for the multiple choice secretary problem). The modified procedure is depicted as Algorithm 1 below.

Algorithm 1. For the classical secretary problem over a set of n applicants.

1: Choose m from the binomial distribution $\mathrm{Bin}(n, p)$.
2: Observe and reject the first m elements—call this set the *sample*.
3: Accept the first record (if any) of the remaining elements.

The next lemma gives bounds for the probability that any specific element of the stream appears in the output of Algorithm 1. This will be useful later.

Lemma 3.3. *Let w_i be the ith top weight of the stream. Algorithm* 1 *returns the empty set with probability p, and it returns the singleton $\{w_i\}$ with probability at least*

$$p\int_p^1 \frac{(1-t)^{i-1}}{t}\,dt.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

In particular, it returns $\{w_1\}$ with probability at least $(-p \ln p)$. Therefore, by setting $p = 1/e$, we get an e-competitive algorithm for the classical secretary problem.

Proof. Consider the following offline simulation. Let $w_1 \geq w_2 \geq \cdots \geq w_n \geq 0$ be the adversarial weights in nonincreasing order. Each weight w_i selects an arrival time t_i in the open interval $(0, 1)$ uniformly and independently at random. The simulation processes the weights in order of arrival, rejecting those arriving before time p (the set of those weights is called the *sample*). When a weight w arriving on or after time p is processed, the simulation checks if w is a record of the sequence seen so far. In that case, it returns $\{w\}$ and halts; otherwise, it continues. If the simulation runs out of elements, it returns the empty set. As the cardinality of the sampled set has binomial distribution with parameters n and p, the set returned by this simulation has the same distribution as the one returned by Algorithm 1. Hence, in what follows we analyze the simulation.

Let w_i be the ith top weight of the stream. The weight w_i is selected by the simulation if and only if the following events hold simultaneously:

($\mathcal{E}1$) The arrival time t_i of w_i is at least p.
($\mathcal{E}2$) The weight w_i is a record.
($\mathcal{E}2$) There is no record in the time interval $[p, t_i)$.

Event $\mathcal{E}2$ holds if and only if all $(i-1)$ weights higher than w_i arrive after time t_i. This happens with probability $(1 - t_i)^{i-1}$. To analyze event $\mathcal{E}3$, consider the set $A(t_i) = \{w_j : t_j < t_i\}$ of weights seen before the arrival of w_i. If this set is empty, then event $\mathcal{E}3$ trivially holds. If $A(t_i)$ is not empty, let w_A be its top weight and t_A be the arrival time of w_A. No matter in what order the weights in $A(t_i)$ arrive, w_A is always the last record seen before t_i. Hence, event $\mathcal{E}3$ holds if and only if $t_A < p$. Furthermore, conditioned on the set $A(t_i)$ being nonempty, the variable t_A is a uniform random variable in $(0, t_i)$. Thence, for every $t \in [p, 1]$ we have

$$\begin{aligned}\Pr(\mathcal{E}3 \mid t_i = t, \mathcal{E}2) &= \Pr(A(t_i) = \emptyset \mid t_i = t, \mathcal{E}2) + \Pr(A(t_i) \neq \emptyset, t_A < p \mid t_i = t, \mathcal{E}2)\\ &= \Pr(A(t_i) = \emptyset \mid t_i = t, \mathcal{E}2) + \Pr(A(t_i) \neq \emptyset \mid t_i = t, \mathcal{E}2)\, p/t\\ &\geq\ p/t.\end{aligned}$$

Since event $\mathcal{E}2$ holds with probability $(1-t_i)^{i-1}$, we conclude that the probability of selecting w_i in the simulation is

$$\begin{aligned}\Pr(\mathcal{E}1, \mathcal{E}2, \mathcal{E}3) &= \int_p^1 \Pr(\mathcal{E}2, \mathcal{E}3 \mid t_i = t)\, ds = \int_p^1 \Pr(\mathcal{E}3 \mid t_i = t, \mathcal{E}2)(1-t)^{i-1}\, dt\\ &\geq \int_p^1 \frac{p(1-t)^{i-1}}{t}\, dt.\end{aligned}$$

In particular, the probability that w_1 is selected is $\int_p^1 \frac{p}{t}\, dt = -p \ln p$. This is maximized by setting $p = 1/e$. For that value, the probability above also equals $1/e$.

To finish the proof, note that the only way for the simulation to return an empty set is for w_1 to arrive before time p. This happens with probability p. ∎

To obtain an algorithm for the random-assignment random-order model that works on any uniformly dense matroid of n elements and total rank r, we propose the following approach: Perform a random partition of the sequence of presented elements into r consecutive groups, each one having roughly n/r elements. For each group, apply Algorithm 1 to find its top-valued element and then try to include it to the output set whenever this preserves independence. The detailed procedure is depicted as Algorithm 2.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

ALGORITHM 2. For uniformly dense matroids of n elements and rank r in the random-assignment random-order model.

ALG $\leftarrow \emptyset$.
Select n values $v_1, \ldots, v_n$ uniformly at random from $\{1, \ldots, r\}$ and let N_i be the number of times value i was selected.
Let $E_1 \subseteq E$ be the sequence of the first N_1 arriving elements, $E_2 \subseteq E$ the sequence of the next N_2 elements, and so on.
For each i in $[r]$, run Algorithm 1 with parameter p (not necessarily $p = 1/e$) and $n = N_i$ on the sequence E_i. Mark the elements that are selected.
Whenever an element x is marked, check if ALG $\cup \{x\}$ is independent. If so, add x to ALG.
Return the final set ALG.

Before analyzing Algorithm 2, it is convenient to define an auxiliary construction that will be useful for the rest of the article.

For any matroid $\mathcal{M} = (E, \mathcal{I})$ of total rank r, the *associated random partition matroid* $\mathcal{R}(\mathcal{M}) = (E, \mathcal{I}')$ is obtained as follows. Partition the set E into r classes, where each element selects its own class independently and uniformly at random. A set of elements in E is independent in $\mathcal{R}(\mathcal{M})$ if it contains at most one element of each class. The following lemma states that the weight of the optimum base of $\mathcal{R}(\mathcal{M})$ is at least a constant fraction of the weight of the optimum in $\mathcal{M}$.

LEMMA 3.4. *Let $w_1 \geq w_2 \geq \cdots \geq w_n \geq 0$ be the set of weights determined by the adversary. Then*

$$\mathbb{E}_{\sigma,\mathcal{R}(\mathcal{M})}[w(\mathrm{OPT}_{\mathcal{R}(\mathcal{M})}(\sigma))] \geq \left(1 - \left(1 - \frac{1}{r}\right)^r\right) \sum_{i=1}^{r} w_i \geq (1 - 1/e)\mathbb{E}_\sigma[w(\mathrm{OPT}_{\mathcal{M}}(\sigma))]. \tag{3.4}$$

To prove this lemma, we need Chebyshev's sum inequality (see, e.g., [21]), which states that if $a_1 \geq a_2 \geq \cdots \geq a_r$ and $b_1 \geq b_2 \geq \cdots \geq b_r$, then $\sum_{i=1}^r a_i b_i \geq \frac{1}{r}\left(\sum_{i=1}^r a_i\right)\left(\sum_{i=1}^r b_i\right)$.

Proof of Lemma 3.4. For $i \in [r]$, element $\sigma(i) \in E$ receiving weight w_i is in $\mathrm{OPT}_{\mathcal{R}(\mathcal{M})}(\sigma)$ if and only if elements $\sigma(1), \ldots, \sigma(i-1)$ are assigned to a different class of $\mathcal{R}(\mathcal{M})$ than the one of $\sigma(i)$. Then $\Pr(\sigma(i) \in \mathrm{OPT}_{\mathcal{R}(\mathcal{M})}(\sigma)) = (1 - 1/r)^{i-1}$. Note that both $((1-1/r)^{i-1})_{i=1,\ldots,r}$ and $(w_i)_{i=1,\ldots,r}$ are nonincreasing sequences. Using Chebyshev's sum inequality, we have

$$\begin{aligned}\mathbb{E}_{\sigma,\mathcal{R}(\mathcal{M})}[w(\mathrm{OPT}_{\mathcal{R}(\mathcal{M})}(\sigma))] &\geq \sum_{i=1}^{r} w_i \left(1 - \frac{1}{r}\right)^{i-1} \geq \left(\frac{1}{r}\sum_{i=1}^{r}\left(1 - \frac{1}{r}\right)^{i-1}\right)\left(\sum_{i=1}^{r} w_i\right) \\ &= \left(1 - \left(1 - \frac{1}{r}\right)^r\right)\sum_{i=1}^{r} w_i.\end{aligned}$$

We conclude by using that $\sum_{i=1}^r w_i$ is a trivial upper bound on $w(\mathrm{OPT}_{\mathcal{M}}(\sigma))$. ∎

In what follows, we give two different analyses for Algorithm 2. In the first one, we compare the weight of the outcome with the optimum of the random partition matroid $\mathcal{R}(\mathcal{M})$. This approach does not lead to the best competitive ratio. However, as we will see later, this analysis is more useful for dealing with general matroids. In the second analysis, we directly compare the weight of the outcome to the sum of the top r weights of the matroid, obtaining a much better competitive ratio for uniformly

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

dense matroids. Observe, in particular, that the second analysis of Theorem 3.5 below implies Theorem 1.1, which is the first of our main results.

THEOREM 3.5. *Let* ALG *be the set returned by Algorithm* 2 *when applied to a uniformly dense matroid* $\mathcal{M}$ *of total rank* r.

(i) *For* $p \in (0,1)$, *we have*

$$\mathbb{E}_{\sigma,\pi}[w(\mathrm{ALG}(\sigma))] \geq \gamma_1(p) \cdot \mathbb{E}_{\sigma,\mathcal{R}(\mathcal{M})}[w(\mathrm{OPT}_{\mathcal{R}(\mathcal{M})}(\sigma))],$$
$$\text{where } \gamma_1(p) = \frac{1}{(1-p)}(1-e^{-(1-p)})(-p\ln p).$$

This is optimized by setting $p = p_1 \approx 0.433509$. *By Lemma* 3.4, *this algorithm is* $((1-1/e)\gamma_1(p_1))^{-1} \approx 5.7187$*-competitive.*

(ii) *For* $p \in (0,1)$, *we have*

$$\mathbb{E}_{\sigma,\pi}[w(\mathrm{ALG}(\sigma))] \geq \gamma_2(p) \cdot \sum_{i=1}^{r} w_i,$$
$$\text{where } \gamma_2(p) = \frac{p(1-e^{-(1-p)})}{1-p}\int_p^1 \frac{1-e^{-t}}{t^2}\,dt.$$

This is optimized by setting $p = p_2 \approx 0.384374$. *Thus, the competitive ratio of this algorithm is at most* $1/(\gamma_2(p_2)) \approx 4.92078$.

Proof. We analyze Algorithm 2 using a two-step offline simulation.

Step 1. Every *weight* w_i of the adversarial list selects a color $c(w_i)$ in $[r]$ uniformly at random. Let W_j be the collection of weights selecting color j. For every color j, present the weights of W_j in uniform random order to Algorithm 1 with parameter p. This algorithm returns either a singleton weight or an empty set. Let T be the collection of returned weights. Note that T contains at most one weight of each color.

Step 2. Randomly assign to each weight in T a different element of the ground set to obtain a set $X(T) \subseteq E$. Finally, apply the greedy procedure on the elements $X(T)$ in *uniform random order*. Let ALG $\subseteq E$ be the obtained set and $W(\mathrm{ALG}) \subseteq W$ be the corresponding set of weights.

It is straightforward to check that the distributions of ALG and $W(\mathrm{ALG})$ remain unchanged if we apply the greedy procedure on $X(T)$ in random order or if we apply it in increasing ordering of their colors (this follows from the random-assignment assumption). We use a random order in step 2 because it makes our analysis simpler. From here, we deduce that the set of elements ALG and the set of weights $W(\mathrm{ALG})$ returned by the two-step simulation have the same distributions as the corresponding ones returned by Algorithm 2.

Consider a fixed weight $w \in W$. In what follows we estimate the probability that w is included in $W(\mathrm{ALG})$, conditioned on the fact that w is selected in the first step, i.e., $w \in T$. For that, let t be a number between 1 and r. Because of the way it was constructed, $W(\mathrm{ALG})$ is a uniform random subset of T of size $\mathrm{rk}(X(T))$. Hence,

$$\Pr[w \in W(\mathrm{ALG}) \mid w \in T,\ |T| = t] = \frac{\mathbb{E}[\mathrm{rk}(X(T))]}{t} \geq \frac{r}{t}\left(1-\left(1-\frac{1}{r}\right)^t\right), \tag{3.5}$$

where the last inequality comes from (3.2) of Lemma 3.2.

In order to remove the conditioning on $|T| = t$, we compute the expected value of the right-hand side of (3.5), conditioned on the event that $w \in T$. The random variable $|T|$ equals one unit (since $w \in T$) plus the number of colors different from

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$c(w)$, for which Algorithm 1 returns a nonempty set in step 1 of the simulation. For each color, Algorithm 1 returns a nonempty set with probability at most[3] $(1-p)$ (see Lemma 3.3). Therefore, the variable $|T|$ is stochastically dominated by one plus the sum of $r-1$ Bernoulli random variables (one for each color different from $c(w)$) with parameter $(1-p)$. As the right-hand side of (3.5) is decreasing in $|T|$ we can effectively replace $|T|$ by the quantity $1+k$, where k is a random variable chosen from the binomial distribution $\mathrm{Bin}(r-1, 1-p)$, to obtain that

$$
\begin{aligned}
\Pr(w \in W(\mathrm{ALG}) \mid w \in T) &\geq \mathbb{E}_{k\sim \mathrm{Bin}(r-1,1-p)}\left[\frac{r}{1+k}\left(1-\left(1-\frac{1}{r}\right)^{1+k}\right)\right] \\
&= \sum_{k=0}^{r-1}\left[\frac{r}{1+k}\left(1-\left(1-\frac{1}{r}\right)^{1+k}\right)\right]\binom{r-1}{k}(1-p)^k p^{r-1-k} \\
&= \frac{1}{1-p}\sum_{k=0}^{r-1}\binom{r}{k+1}\left[(1-p)^{k+1}p^{r-(k+1)} - \left(1-\frac{1}{r}\right)^{k+1}(1-p)^{k+1}p^{r-(k+1)}\right]. \qquad (3.6)
\end{aligned}
$$

Using the binomial theorem, the right-hand side of (3.6) is equal to

$$
\frac{[1-p^r]-[((1-\frac{1}{r})(1-p)+p)^r - p^r]}{1-p} = \frac{1-\left(1-\frac{1-p}{r}\right)^r}{1-p} \geq \frac{1-e^{-(1-p)}}{1-p}. \qquad (3.7)
$$

We have thus obtained that $\Pr(w \in W(\mathrm{ALG}) \mid w \in T) \geq \frac{1-e^{-(1-p)}}{1-p}$. Let us use this bound to conclude the analysis of (i). Recall that Algorithm 1 returns the best weight of each class with probability $-p\ln p$ (see Lemma 3.3). Hence, for each color class $j \in [r]$, the expected weight of $T\cap W_j$ is at least $-p\ln p$ fraction of the maximum weight w_j^* in W_j (set $w_j^* = 0$ for the pathological case where W_j is empty). Therefore,

$$
\mathbb{E}[\Sigma W(\mathrm{ALG})] \geq \sum_{j=1}^{r}\frac{1-e^{-(1-p)}}{1-p}\mathbb{E}[\Sigma(T\cap W_j)] \geq \gamma_1(p)\sum_{j=1}^{r}\mathbb{E}[w_j^*],
$$

where $\gamma_1(p) = \frac{1-e^{-(1-p)}}{1-p}(-p\ln p)$. This concludes the proof of part (i).

The analysis for part (ii) is a refinement of the previous one. For this one, we directly compute the probability that the simulation selects each one of the top r weights in the adversarial list. Let w_i, for $i \leq r$, be the ith top weight. Let $\mathcal{E}_{ij}$ be the event that w_i is the jth top weight of its own color class. Note that $\mathcal{E}_{ij}$ holds if and only if exactly $j-1$ weights in $\{w_1,\ldots,w_{i-1}\}$ are in the same color class as w_i. Since each weight selects a color uniformly at random in $[r]$, we have

$$
\Pr(\mathcal{E}_{ij}) = \begin{cases} \binom{i-1}{j-1}\left(\frac{1}{r}\right)^{j-1}\left(1-\frac{1}{r}\right)^{i-j} & \text{if } j \leq i, \\ 0 & \text{otherwise.}\end{cases} \qquad (3.8)
$$

Using (3.8), Lemma 3.3, and the binomial theorem, we obtain that the probability

[3]The reason this is not exactly $(1-p)$ is the pathological case where the color class is empty. In this case the algorithm will always return an empty set.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

that w_i is added to the set T is

$$\begin{aligned}
\Pr(w_i \in T) &= \sum_{j=1}^{i} \Pr(w_i \in T \mid \mathcal{E}_{ij}) \Pr(\mathcal{E}_{ij}) \\
&= \sum_{j=1}^{i} p \int_p^1 \frac{(1-t)^{j-1}}{t}\, dt \binom{i-1}{j-1} \left(\frac{1}{r}\right)^{j-1} \left(1-\frac{1}{r}\right)^{i-j} \\
&= \int_p^1 \frac{p}{t} \sum_{j=1}^{i} \binom{i-1}{j-1} \left(\frac{1-t}{r}\right)^{j-1} \left(1-\frac{1}{r}\right)^{i-j} dt \\
&= \int_p^1 \frac{p}{t} \left(\frac{1-t}{r} + 1 - \frac{1}{r}\right)^{i-1} dt = p \int_p^1 \frac{(1-t/r)^{i-1}}{t}\, dt. \qquad (3.9)
\end{aligned}$$

The equation above implies that the sequence $(\Pr(w_i \in T))_{i=1,\ldots,r}$ is nonincreasing in i. Using (3.6), (3.7), and Chebyshev's sum inequality, we conclude that

$$\begin{aligned}
\mathbb{E}[\Sigma W(\mathrm{ALG})] &\geq \frac{1-e^{-(1-p)}}{1-p} \sum_{i=1}^{r} w_i \Pr(w_i \in T) \\
&\geq \frac{1-e^{-(1-p)}}{1-p} \left(\sum_{i=1}^{r} w_i\right) p \int_p^1 \frac{1}{r} \sum_{i=1}^{r} \frac{(1-t/r)^{i-1}}{t}\, dt \\
&= \frac{1-e^{-(1-p)}}{1-p} \left(\sum_{i=1}^{r} w_i\right) p \int_p^1 \frac{1-(1-t/r)^r}{t^2}\, dt \\
&\geq \frac{1-e^{-(1-p)}}{1-p} \left(\sum_{i=1}^{r} w_i\right) p \int_p^1 \frac{1-e^{-t}}{t^2}\, dt. \qquad \square
\end{aligned}$$

The second part of the above analysis for Algorithm 2 gives the tightest competitive ratio (≈ 4.92078). It is worth noting that this ratio is better than that of the algorithm for uniformly dense matroids presented in the conference version of this paper [29]. We also remark that even though the first part of the analysis is not as tight, it is much more useful (see section 4) to obtain a tighter analysis of the algorithm that we propose for general matroids.

3.2. Algorithm for the random-assignment adversarial-order model. After the first publication [29] of some of the results of this article, Oveis Gharan and Vondrák [26] have devised a 40-competitive algorithm for uniformly dense matroids on the random-assignment adversarial-order model. Using the techniques present in this article (see subsection 4.2), they obtain a $40e/(e-1)$-competitive algorithm for general matroids.

In what follows we devise an alternative algorithm for uniformly dense matroids of rank r in this model that is similar to Algorithm 2. We start by creating a random partition of the matroid's ground set into r groups, or *color classes*, of roughly the same size. Unlike Algorithm 2, the partition used is independent of the order in which the elements are presented. Afterwards, we use a modification of Algorithm 1 to find and mark the heaviest element of each color class. We do this in a coupled way: the sample of a color class (that is, the set of unconditionally rejected elements) consists of all those elements seen in a certain first fraction of the stream. Finally, we try to add each marked elements to the output set. But unlike Algorithm 2, we only do this step

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

with certain probability (otherwise, we discard the element). The complete procedure is depicted as Algorithm 3 below. This algorithm depends on the two parameters p and q in $(0, 1)$.

ALGORITHM 3. For uniformly dense matroids of n elements and rank r in the random-assignment adversarial-order model.

1: ALG $\leftarrow \emptyset$.
2: Assign to each element of the matroid a color $j \in [r]$ uniformly at random.
3: Choose m from the binomial distribution $\mathrm{Bin}(n, p)$.
4: Observe and reject the first m elements of the stream, denoting this set as the sample.
5: **for** each element x arriving on or after time p **do**
6:     **if** the color j of x has not been tagged as *completed* yet and x is the heaviest element seen so far with color j, **then**
7:         Tag color j as completed.
8:         With probability q ignore x and continue.
9:         Otherwise, check if $\mathrm{ALG} \cup \{x\}$ is independent. If so, add x to ALG.
10:     **end if**
11: **end for**
12: Return the set ALG.

The next offline simulation algorithm will be useful in analyzing Algorithm 3. Given a set of weights W, a sorted list E of matroid elements, and two values $p, q \in (0, 1)$, do the following steps.

Step 1. Select, for every weight $w \in W$, an arrival time $t(w)$ in the open interval $(0, 1)$ uniformly at random. Assign to the weight with the kth smallest arrival time, the corresponding kth element $e(w)$ of the adversarially sorted list.

Step 2. Select, for very weight $w \in W$, a color $c(w) \in [r]$.

Step 3. For each color $j \in [r]$ do the following: among all the weights of color j arriving on or after time p, *mark* the first one (if any) that is larger than all the weights of color j arriving before time p.

Step 4. Independently toss a coin for each *color* with probability of heads equal to q. *Unmark* the weights whose coin came up heads.

Step 5. Apply the greedy procedure on the elements that are still marked in increasing arrival order. Return the answer of this procedure.

To analyze Algorithm 3, it is enough to study the output of the simulation.

PROPOSITION 2. *The sets of elements and weights that the above simulation return have the same distribution as the corresponding ones returned by Algorithm* 3 *in the random-assignment adversarial-order model.*

Proof. The first step of the simulation is just a reinterpretation of the model: every matroid element of the adversarial list is assigned to a weight selected uniformly at random from W. The second step indirectly assigns a uniform random color in $[r]$ to each matroid element, mimicking line 2 of Algorithm 3. In the third step, the simulation skips all the elements arriving before time p. Since the number of those elements has binomial distribution with parameters p and n, we conclude that the set of skipped element behaves exactly as the sample defined in line 4 of Algorithm 3.

Observe that at most one weight of each color is marked in the third step of the simulation and that each marked weight is the first record of its color class arriving in the time interval $[p, 1)$. A similar situation occurs in Algorithm 3: If K denotes

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

the set of elements passing the test of line 6, then K contains at most one element of each color and each element in K is the first nonsampled record of the subsequence defined by its color. This means that K has the same distribution as the set L of matroid elements assigned to those weights that were marked at the end of the third step of the simulation.

To conclude the proof of the proposition, note that the output of Algorithm 3 is obtained by applying the greedy procedure on the subset of K obtained by dropping each element with probability q, in their order of appearance. This is exactly what the fourth and fifth steps of the simulation do on set L. ∎

The partition matroid on E whose independent sets are those containing at most one element of each color (in the simulation) behaves exactly as the random partition matroid $\mathcal{R}(\mathcal{M})$ associated to $\mathcal{M}$ that we defined in the previous section, so we consider it as such. We will show that the output of the simulation contains a significant fraction of the optimum independent set of $\mathcal{R}(\mathcal{M})$. This, together with Proposition 2 and Lemma 3.4, will be enough to prove that Algorithm 3 is constant-competitive.

More precisely, say that a color $j \in [r]$ is *successful* if either the heaviest weight of color j appears in the output of the simulation or no weight in the stream has color j.

PROPOSITION 3. *Every color is successful with probability at least* $p(1-p)q(1-q)$.

Before proving this proposition, let's use it to prove the main theorem of this section.

THEOREM 3.6. *Let* ALG *be the set returned by Algorithm* 3 *when applied to a uniformly dense matroid* $\mathcal{M}$ *of total rank* r *in the random-assignment adversarial-order model. Then*

$$\mathbb{E}_\sigma[w(\mathrm{ALG}(\sigma))] \geq pq(1-p)(1-q) \cdot \mathbb{E}_{\mathcal{R}(\mathcal{M}),\sigma}[w(\mathrm{OPT}_{\mathcal{R}(\mathcal{M})}(\sigma))].$$

By Lemma 3.4*, Algorithm* 3 *is* $16/(1-1/e) \approx 25.31$*-competitive for* $p = q = 1/2$.

Proof. Propositions 2 and 3 imply that the total weight returned by Algorithm 3 is at least $p(1-p)q(1-q)$ times the expected weight of the optimum of the random partition matroid $\mathcal{R}(\mathcal{M})$. ∎

We remark here that Theorem 3.6 implies the second of our main results, Theorem 1.2. The only part left is to prove Proposition 3.

Proof of Proposition 3. We prove the following stronger claim instead. Fix an arbitrary coloring $c^*\colon W \to [r]$. Then, conditioned on the event that c^* is the coloring selected in the second step of the simulation, every color is successful with probability at least $p(1-p)q(1-q)$.

Without loss of generality, relabel the colors so that the nonempty color classes are $1, \ldots, s$. We assume that $2 \leq s \leq r$, as for the pathological case in which $s = 1$ we can use the proof of Lemma 3.3 to show that the only nonempty color class is successful with probability at least $(-p \ln p)(1-q) \geq p(1-p)q(1-q)$.

It is enough to show that the claim holds for the first color class. Let v_1 and v_2 be the two top weights of color class 1. If this class has only one element, let v_2 be an arbitrary different weight of the list (to simplify our later discussion, assume that v_2 is not the top weight of its own color class). By definition, color 1 is successful if weight v_1 appears in the output of the algorithm. In what follows, we estimate the probability that this event occurs.

Let A be the (random) set of weights arriving before time p and $B = W \setminus A$ the weights arriving on or after time p. Furthermore, let $Y \subseteq W$ be the set of weights marked in the third step and $X \subseteq Y$ the set of weights that are still marked at the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

end of Step 4. For every set V of weights, let $e(V)$ be the associated collection of elements (i.e., $e(V) = \{e(v) \colon v \in V\}$).

Condition on the event that $v_1 \in B$ and $v_2 \in A$. Under this condition, weight v_1 will definitely be marked in the third step of the simulation (i.e., $v_1 \in Y$). In order for v_1 to be part of the output of the simulation, it is enough that this weight is not unmarked in the fourth step (i.e., $v_1 \in X$) and that the element $e_1 = e(v_1)$ assigned to weight v_1 is outside the span of the set of elements assigned to other weights in X (i.e., $e_1 \notin \operatorname{span}(e(X \setminus \{v_1\}))$). Then

$$\begin{aligned}
&\Pr(v_1 \text{ is selected by the algorithm} \mid v_1 \in B, v_2 \in A)\\
&\qquad \geq \Pr(v_1 \in X \text{ and } e_1 \notin \operatorname{span}(e(X \setminus \{v_1\})) \mid v_1 \in B, v_2 \in A)\\
&\qquad = (1-q)\Pr(e_1 \notin \operatorname{span}(e(X \setminus \{v_1\})) \mid v_1 \in B, v_2 \in A). \qquad (3.10)
\end{aligned}$$

If we fix the sets B, $Y' = Y \setminus v_1$, and $X' = X \setminus v_1$, then the element e_1 is a uniformly random element of $e(B \setminus Y')$. Therefore, using that $X' \subseteq Y' \subseteq B$, the probability that e_1 is not spanned by $e(X')$ is

$$\frac{|e(B \setminus Y') \setminus \operatorname{span}(e(X'))|}{|e(B \setminus Y')|} = \frac{\left|e(B) \setminus \Big(\operatorname{span}(e(X')) \cup (e(Y') \setminus e(X'))\Big)\right|}{|B| - |Y'|}. \qquad (3.11)$$

As the matroid is uniformly dense, $|\operatorname{span}(e(X'))| \leq \frac{n}{r}\operatorname{rk}(e(X')) \leq \frac{n}{r}|X'|$. Therefore, using that $X' \subseteq Y'$, we conclude that (3.11) is at least

$$\frac{|B| - \left(\frac{n}{r}|X'| + |Y'| - |X'|\right)}{|B| - |Y'|} = 1 - \left(\frac{n}{r} - 1\right)\frac{|X'|}{|B| - |Y'|}. \qquad (3.12)$$

It follows that the probability that v_1 is selected by the algorithm is at least

$$p(1-p)(1-q)\left(1 - \frac{n-r}{r}\mathbb{E}\left[\frac{|X'|}{|B| - |Y'|}\,\middle|\, v_1 \in B, v_2 \in A\right]\right). \qquad (3.13)$$

We need to upper bound the expectation in the expression above. For each $k \in \{1, \ldots, s\}$, let Y_k be the indicator variable for the event that the top weight of color k arrives after time p (note that under the condition that $v_1 \in B$, Y_1 is deterministic and equal to 1). Define also $n - s$ other indicator random variables, $(Y_{s+1}, \ldots, Y_n)$, one for each weight that is not the top weight of its color. Each of these indicates the event that its associated weight arrives after time p. Without loss of generality associate Y_n to weight v_2 (note that under the condition that $v_2 \in A$, Y_n is deterministic and equal to 0). Finally, for each $k \in \{1, \ldots, s\}$, let X_k be the indicator random variable for the event that color k is not unmarked. We will use the independent indicator variables just defined to express $|B|$, $|X'|$, and $|Y'|$.

Observe that for every nonempty color class $k \in \{2, \ldots, s\}$, a weight of color k is marked in the second step of the simulation if and only if $Y_k = 1$. From here it is easy to see that $|Y'| = \sum_{k=2}^{s} Y_k$, $|X'| = \sum_{k=2}^{s} X_k Y_k$, and $|B| = \sum_{k=1}^{n} Y_k$. Therefore,

$$\mathbb{E}\left[\frac{|X'|}{|B| - |Y'|}\,\middle|\, v_1 \in B, v_2 \in A\right] = \sum_{k=2}^{s} \mathbb{E}\left[\frac{X_k Y_k}{1 + \sum_{j=s+1}^{n-1} Y_j}\right]. \qquad (3.14)$$

By symmetry, all $(s-1)$ terms in the above summation have the same expectation. Noting that X_k, Y_k, and the denominator are mutually independent, we conclude that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

(3.14) is equal to

$$(s-1)(1-q)(1-p)\mathbb{E}\left[\frac{1}{1+\sum_{j=s+1}^{n-1} Y_j}\right]. \tag{3.15}$$

Since each Y_j is a Bernoulli random variable of parameter $1-p$, we can compute the expected value in (3.15) as follows:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{1+\sum_{j=s+1}^{n-1} Y_j}\right] &= \sum_{i=0}^{n-s-1} \frac{1}{1+i}\binom{n-s-1}{i}(1-p)^i p^{n-s-1-i} \\
&= \frac{1}{(1-p)(n-s)} \sum_{i=0}^{n-s-1} \binom{n-s}{i+1}(1-p)^{i+1}p^{n-s-(i+1)} \\
&= \frac{1-p^{n-s}}{(1-p)(n-s)} \le \frac{1}{(1-p)(n-s)}. \qquad (3.16)
\end{aligned}$$

By putting all together and using that $s \le r \le n$, we have

$$\begin{aligned}
\Pr(\text{Color 1 is successful}) &\ge p(1-p)(1-q)\left(1-\frac{n-r}{r}\cdot\frac{(s-1)(1-q)}{n-s}\right) \\
&\ge p(1-p)(1-q)\left(1-\frac{n-r}{r}\cdot\frac{(r-1)(1-q)}{n-r}\right) \\
&\ge p(1-p)(1-q)q. \qquad (3.17)
\end{aligned}$$

This concludes the proof. □

The attentive reader may observe that the above analysis is not very tight. There are many ways in which a weight can be marked: If we let $v_1, \dots, v_i$ be the top i weights of the first color class, and we are in the situation where v_i arrives before time p, v_1 arrives after time p, and all $v_2, \dots, v_{i-1}$ arrive after v_1, then the weight v_1 is marked. In the proof above we only considered the case $i = 2$. A more careful analysis would improve our bound on the competitive ratio. For instance, let us consider the case $i = 3$ in the above scenario. By conditioning on the event that the time $t(v_3)$ is smaller than p, and on the time $t \ge p$ of arrival of v_2, we can show that

$$\begin{aligned}
&\Pr(v_1 \text{ is selected by the algorithm} \,|\, t(v_3) < p, t = t(v_2)) \\
&\ge (t-p)(1-q)\mathbb{E}\left[\frac{|B(t)| - \left(\frac{n}{r}-1\right)|X'(t)| - |Y'(t)|}{|B(t)|}\,\middle|\, t(v_3) < p, t = t(v_2), v_1 \in B(t)\right] \\
&= (t-p)(1-q)\left(1-\left(\frac{n}{r}(1-q)+q\right)\mathbb{E}\left[\frac{|Y'(t)|}{|B(t)|}\,\middle|\, t(v_3) < p, t = t(v_2), v_1 \in B(t)\right]\right),
\end{aligned}$$

where $B(t)$, $Y'(t)$, and $X'(t)$ are, respectively, the sets of weights arriving in the time interval (p,t), the set of weights marked in (p,t) of color different from 1, and the set of weights in $Y'(t)$ that are not unmarked. Using arguments similar to those in the proof of Proposition 3, we can estimate the probability above and finally obtain that

$$\begin{aligned}
&\Pr(\text{Color 1 is successful due to case } i = 3) \\
&\quad\ge p(1-q)\int_p^1 (t-p)\left(1-\left(\frac{n}{r}(1-q)+q\right)\frac{r-1}{t(n-2)}\right)dt.
\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

This expression is not easily simplified; however, if the matroid has high density, namely, if $n \geq 2r$, then $\left(\frac{n}{r}(1-q)+q\right)\frac{r-1}{n-2} \leq (1-q)+q/2 = 1-q/2$. Therefore, when the matroid has density at least 2, the probability that a color is successful (due to the disjoint cases $i=2$ and $i=3$) is at least

$$p(1-p)q(1-q) + p(1-q)\int_p^1 \frac{(t-p)(t-(1-q/2))}{t}\,dt.$$

The expression above is optimized when we select $p \approx 0.51399$ and $q \approx 0.523138$, giving a lower bound of approximately 0.0652158. In particular, by selecting these values of p and q, we have that Algorithm 3 is $(0.0652158(1-1/e))^{-1} \approx 24.2575681$-competitive for uniformly dense matroids of density at least 2.

It is possible to further improve the bound for the competitive ratio of Algorithm 3 by considering cases where $i \geq 4$; unfortunately, in order to simplify the resulting expressions we require higher matroid density and more cumbersome analysis. We will not pursue that goal in this article; however, we do remark that it is possible to obtain an algorithm having competitive ratio strictly better than $16/(1-1/e)$ for uniformly dense matroids of arbitrary density. Indeed, if the density of the matroid is high, we use Algorithm 3; if not, we use Algorithm 8 for low-density matroids that we describe in subsection 7.3 (that algorithm works for any model of the matroid secretary problem, including random-assignment adversarial-order).

4. Algorithms for random-assignment models in general matroids. In order to devise constant competitive algorithms for general matroids, we use some technical result involving matroids constructed using the theory of principal partition. In what follows, we state the results we need, and how to use them. The discussion of these results is deferred to section 5.

Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and L its set of loops. By Theorem 5.1 (see section 5), there exists a particular collection of uniformly dense matroids $(\mathcal{M}_i = (E_i, \mathcal{I}_i))_{i=1}^k$, called *principal minors of* $\mathcal{M} \setminus L$, such that the family $\{L, E_1, \ldots, E_k\}$ partitions the ground set E.

Define the matroid $\mathcal{M}'$ and the random matroid $\mathcal{Q}'$ as

$$\mathcal{M}' = \mathcal{U}(L,0) \oplus \bigoplus_{i=1}^k \mathcal{M}_i, \qquad \mathcal{Q}' = \mathcal{U}(L,0) \oplus \bigoplus_{i=1}^k \mathcal{R}(\mathcal{M}_i),$$

where $\mathcal{U}(X,r)$ denotes the uniform matroid on a set X with rank r and $\mathcal{R}(\mathcal{N})$ denotes the random partition matroid associated to $\mathcal{N}$. In section 5 we prove the following technical lemmas.

Lemma 4.1. *Any independent set of $\mathcal{M}'$ is independent in $\mathcal{M}$.*

Lemma 4.2. *For every adversarial list of weights $w_1 \geq w_2 \geq \cdots \geq w_n \geq 0$,*

$$\mathbb{E}_{\sigma,\mathcal{Q}'}[w(\mathrm{OPT}_{\mathcal{Q}'}(\sigma))] \geq \left(1-\frac{1}{e}\right)\mathbb{E}_\sigma[w(\mathrm{OPT}_{\mathcal{M}}(\sigma))].$$

In subsections 4.1 and 4.2 we use the above lemmas together with the divide and conquer idea of subsection 2.1 to devise constant competitive algorithms for general matroids.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

4.1. Random-assignment random-order model. Consider Algorithm 4 depicted below.

ALGORITHM 4. For general matroids in the random-assignment random-order model.

1: Compute the principal minors $(\mathcal{M}_i)_{i=1}^k$ of the matroid obtained by removing the loops of $\mathcal{M}$.
2: Run Algorithm 2 (with parameter p) for uniformly dense matroids in parallel on each $\mathcal{M}_i$ and return the union of the answers.

As the set returned is independent in the matroid $\mathcal{M}' = \mathcal{U}(L,0) \oplus \bigoplus_{i\in[k]} \mathcal{M}_i$, it is also independent in $\mathcal{M}$ (see Lemma 4.1). Therefore, the algorithm is correct. To estimate its competitive ratio, we use part (i) of the analysis of Algorithm 2 for uniformly dense matroids. Theorem 3.5 states that, when applied on a uniformly dense matroid $\mathcal{N}$, Algorithm 2 with parameter p returns a set of expected weight at least $\gamma_1(p)$ times the expected weight of the optimum in the random partition matroid $\mathcal{R}(\mathcal{N})$. The random partition matroid $\mathcal{Q}'$ associated to $\mathcal{M}$ contains a summand $\mathcal{R}(\mathcal{M}_i)$ for every uniformly dense matroid $\mathcal{M}_i$. Since every summand is treated independently in Algorithm 4, we conclude that, in expectation, this algorithm recovers $\gamma_1(p)$-fraction of the optimum weight of the random partition matroid $\mathcal{Q}'$.

LEMMA 4.3. *Let* ALG *be the set returned by Algorithm* 4. *Then*

$$\mathbb{E}_{\sigma,\pi}[w(\mathrm{ALG})] \geq \gamma_1(p)\,\mathbb{E}_{\sigma,\mathcal{Q}'}[w(\mathrm{OPT}_{\mathcal{Q}'}(\sigma))]. \tag{4.1}$$

Proof. The only ingredient left is to argue that Algorithm 2 is effectively applied over an instance of the random-assignment random-order model. Observe that the random bijection $\sigma\colon [n] \to W$ used to assign the weights in W to the elements of the matroid can be viewed as the composition of a random partition of $[n]$ and W into blocks of sizes $(|L|, |E_1|, |E_2|, \dots, |E_k|)$, and a collection of random bijections between the corresponding blocks. Conditioned on the random partition, each block receives a hidden list of weights which are assigned uniformly at random to the elements of the block. To complete the proof we only need to observe that the elements within each block are presented to Algorithm 2 in a uniform random order. ◻

Now we are ready to give a bound on the competitive ratio of Algorithm 4. The next theorem implies, in particular, our third main result, Theorem 1.3.

THEOREM 4.4. *For $p = p_1 \approx 0.433509$, Algorithm 4 is $1/(\gamma_1(p_1)(1-1/e)) \approx 5.7187$-competitive for the random-assignment random-order model of the matroid secretary problem.*

Proof. The proof follows directly from Lemmas 4.3 and 4.2. ◻

4.2. Random-assignment adversarial-order model. Oveis Gharan and Vondrák [26] noticed that by combining their 40-competitive algorithm for uniformly dense matroids on the random-assignment adversarial-order model with our techniques, one can get a $40/(1-1/e)$-competitive algorithm for general matroids. We improve this result by using Algorithm 3 instead. Consider the procedure depicted as Algorithm 5 below, whose correctness follows from Lemma 4.1.

Theorem 3.6 states that for every uniformly dense matroid $\mathcal{N}$ of total rank r, Algorithm 3 returns a set of expected weight at least $1/16$ times the expected weight of the optimum in the random partition matroid $\mathcal{R}(\mathcal{N})$. In particular, since each matroid $\mathcal{M}_i$ is uniformly dense, and since within each $\mathcal{M}_i$ there is a random assignment of weights (see the proof of Lemma 4.3), Algorithm 5 recovers in expectation $1/16$ times

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ALGORITHM 5. For general matroids in the random-assignment adversarial-order model.

1: Compute the principal minors $(\mathcal{M}_i)_{i=1}^k$ of the matroid obtained by removing the loops of $\mathcal{M}$.
2: Run Algorithm 3 for uniformly dense matroids in parallel on each $\mathcal{M}_i$ and return the union of the answers.

the optimum weight of the partition matroid $\mathcal{Q}'$. Therefore, we have the following theorem.

THEOREM 4.5. *Let* ALG *be the set returned by Algorithm* 5 *when applied on a uniformly dense matroid. Then*

$$\mathbb{E}_\sigma[w(\mathrm{ALG})] \geq \frac{1}{16}\mathbb{E}_{\sigma,\mathcal{Q}'}[w(\mathrm{OPT}_{\mathcal{Q}'}(\sigma))]. \tag{4.2}$$

By Lemma 4.2*, Algorithm* 5 *is* $16/(1-1/e) \approx 25.31$*-competitive for the random-assignment adversarial-order model of the matroid secretary problem.*

The previous theorem implies our fourth main result, Theorem 1.4.

5. Uniformly dense minors of a matroid. In this section we revisit some useful concepts coming from the theory of principal partitions of discrete systems. For a thorough introduction to this area, we refer the reader to a monograph by Narayanan [24] and a survey by Fujishige [15].

The main result we need from this theory (which follows, e.g., from [15, Theorem 3.11] or [8]) states that every loopless matroid $\mathcal{M}$ contains a nice collection of uniformly dense minors.[4]

THEOREM 5.1 (principal sequence of a loopless matroid). *Let* $\mathcal{M} = (E, \mathcal{I})$ *be a loopless matroid. Then there are a unique sequence of sets* $\emptyset = F_0 \subsetneq F_1 \subsetneq F_2 \subsetneq \cdots \subsetneq F_k$ *and a unique sequence of nonnegative real values* $\lambda_1 > \lambda_2 > \cdots > \lambda_k \geq 1$ *satisfying that the matroid* $\mathcal{M}_i = (\mathcal{M}/F_{i-1})|_{(F_i \setminus F_{i-1})}$ *is uniformly dense with density* $\lambda_i = \frac{|F_i \setminus F_{i-1}|}{\mathrm{rk}_\mathcal{M}(F_i) - \mathrm{rk}_\mathcal{M}(F_{i-1})}$ *for every* $1 \leq i \leq k$*. Moreover, if for every* i*,* I_i *is an independent set of* $\mathcal{M}_i$*, then the set* $\bigcup_{i=1}^k I_i$ *is independent in* $\mathcal{M}$*.*

The sequence $\emptyset = F_0 \subset F_1 \subset \cdots \subset F_k = E$ is called the *principal sequence* of the matroid $\mathcal{M}$; $\lambda_1 > \cdots > \lambda_k \geq 1$ is the associated sequence of *critical values*, and $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are known as the *principal minors* of $\mathcal{M}$. Polynomial time algorithms to compute the principal sequence of a given matroid can also be found in the literature (see, e.g., [25] or [24, Chapters 10 and 11]). These sequences have been extensively studied in the past under different names (see, e.g., [23, 30, 25]), but unfortunately, as with many other concepts coming from a theory developed mainly in Japan, they are still not well known in the western world. To keep our discussion self-contained, in section 6 we include a proof of Theorem 5.1 using only elementary matroid arguments.

The rest of this section is devoted to the study the properties of certain direct sums of matroids which can be constructed using Theorem 5.1. In particular, we prove Lemmas 4.1 and 4.2 that we used in section 4. We start by giving a formal definition of the matroids $\mathcal{M}'$ and $\mathcal{Q}'$ involved in both lemmas.

Consider a general (not necessarily loopless) matroid $\mathcal{M} = (E, \mathcal{I})$ and let L be its set of loops. Let $(F_i)_{i=0}^k$, $(\lambda_i)_{i=1}^k$, and $(\mathcal{M}_i)_{i=1}^k$ be the principal sequence, critical values, and principal minors of the loopless matroid $\mathcal{M} \setminus L$. Also, for every $i \in [k]$,

[4] A *minor* of $\mathcal{M}$ is a matroid obtained by deleting or contracting elements from $\mathcal{M}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

let $E_i = F_i \setminus F_{i-1}$ and r_i denote the ground set and the total rank of matroid $\mathcal{M}_i$. Recall also that the *density* $\gamma(\mathcal{M}_i)$ of the matroid $\mathcal{M}_i$ is equal to $\lambda_i = |E_i|/r_i$.

The family $\{L, E_1, \dots, E_k\}$ is a partition of the ground set E. Define the matroid $\mathcal{M}'$ with ground set E and independent set family $\mathcal{I}(\mathcal{M}') = \left\{ \bigcup_{i=1}^k I_i \colon I_i \in \mathcal{I}(\mathcal{M}_i) \right\}$. In other words, if $\mathcal{U}(X, r)$ is the uniform matroid on set X having rank r, then

$$\mathcal{M}' = \mathcal{U}(L, 0) \oplus \bigoplus_{i=1}^{k} \mathcal{M}_i.$$

Recall the definition of the random partition matroid $\mathcal{R}(\mathcal{M}_i)$ associated to $\mathcal{M}_i$. In $\mathcal{R}(\mathcal{M}_i)$, each element of E_i receives a color in $[r_i]$ uniformly at random. Let B_{ij} be the set of elements in E_i that are assigned color j. The independent sets of $\mathcal{R}(\mathcal{M}_i)$ are those subsets of E_i having at most one element in each part B_{ij}. Consider the random matroid $\mathcal{Q}'$ obtained by replacing each summand $\mathcal{M}_i$ of $\mathcal{M}'$ by the matroid $\mathcal{R}(\mathcal{M}_i)$. That is,

$$\mathcal{Q}' = \mathcal{U}(L, 0) \oplus \bigoplus_{i=1}^{k} \mathcal{R}(\mathcal{M}_i) = \mathcal{U}(L, 0) \oplus \bigoplus_{i=1}^{k} \bigoplus_{j=1}^{r_i} \mathcal{U}(B_{ij}, 1).$$

Now we show some properties of $\mathcal{M}'$ and $\mathcal{Q}'$. We start by proving Lemma 4.1 of the previous section, i.e., that every independent set of $\mathcal{M}'$ is independent in $\mathcal{M}$.

Proof of Lemma 4.1. This follows directly from the definition of each $\mathcal{M}_i = (\mathcal{M}/F_{i-1})\big|_{E_i}$ and from Theorem 5.1. ∎

The following theorem is the main result of this section. It states that random independent sets in $\mathcal{Q}'$ are likely to have large rank in the original matroid $\mathcal{M}$.

THEOREM 5.2. *Let X_ℓ be a uniform random subset of ℓ elements of E, where $1 \le \ell \le n$. Then*

$$\mathbb{E}_{X_\ell, \mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell)] \ge \left(1 - \frac{1}{e}\right) \mathbb{E}_{X_\ell}[\mathrm{rk}_{\mathcal{M}}(X_\ell)]. \tag{5.1}$$

In order to prove Theorem 5.2, we need two technical lemmas.

LEMMA 5.3. *For every $1 \le \ell \le n$ and every $1 \le i \le k$,*

$$\mathbb{E}_{X_\ell, \mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)] \ge r_i\big(1 - \exp(-\lambda_i \ell / n)\big).$$

Proof. Given that the partition $\{L, E_1, \dots, E_k\}$ of E is fixed beforehand, the value on the left-hand side depends only on X_ℓ and on the subpartition $\{B_{ij}\}_{j \in [r_i]}$ of E_i. As these objects are chosen independently at random, we can assume they are constructed as follows: We first select $X_\ell \subseteq E$ uniformly at random. Then every element $e \in E_i$ chooses a color $j \in \{1, \dots, r_i\}$ uniformly at random and is assigned to B_{ij}. Let $X_{\ell,j}$ be the set of elements in X_ℓ having color j; therefore,

$$\mathbb{E}_{X_\ell, \mathcal{Q}'}\left[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)\right] = \sum_{j=1}^{r_i} \Pr(X_{\ell,j} \cap E_i \neq \emptyset) = \sum_{j=1}^{r_i} \left(1 - \Pr(X_{\ell,j} \cap E_i = \emptyset)\right). \tag{5.2}$$

Focus on the jth term of the sum above and condition on the size t of $X_{\ell,j}$. Under this assumption, $X_{\ell,j}$ is a uniform random subset of E of size t. From here,

$$\Pr\left(X_{\ell,j} \cap E_i = \emptyset \;\middle|\; |X_{\ell,j}| = t\right) = \frac{\binom{n - |E_i|}{t}}{\binom{n}{t}} = \prod_{\ell=0}^{t-1} \left(1 - \frac{|E_i|}{n - \ell}\right) \le \left(1 - \frac{|E_i|}{n}\right)^t.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

By removing the conditioning and using that t is a binomial random variable with parameters ℓ and $1/r_i$,

$$\begin{aligned}\Pr(X_{\ell,j} \cap E_i = \emptyset) &\leq \sum_{t=0}^{\ell}\left(1-\frac{|E_i|}{n}\right)^t \cdot \binom{\ell}{t}\left(\frac{1}{r_i}\right)^t\left(1-\frac{1}{r_i}\right)^{\ell-t}\\ &= \left(\frac{1}{r_i}\left(1-\frac{|E_i|}{n}\right)+\left(1-\frac{1}{r_i}\right)\right)^{\ell}\\ &= \left(1-\frac{|E_i|}{nr_i}\right)^{\ell}.\end{aligned}$$

Replacing this in (5.2), and using that $\lambda_i = |E_i|/r_i$, we have

$$\mathbb{E}_{X_\ell,\mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)] \geq r_i\left(1-\left(1-\frac{\lambda_i}{n}\right)^{\ell}\right) \geq r_i\big(1-\exp(-\lambda_i\ell/n)\big). \qquad \square$$

Consider a uniform random set X_ℓ of size ℓ in E. The rank of X_ℓ in $\mathcal{Q}'|_{E_i} = \mathcal{R}(\mathcal{M}_i)$ is simply the number of subparts in $\{B_{i1},\dots,B_{ir_i}\}$ this set intersects. If E_i has high density (say $\lambda_i \geq n/\ell$), then we expect E_i to contain $|E_i|(\ell/n) \geq r_i$ elements of X_ℓ. As they are roughly equally distributed among the subparts of E_i, we expect the rank of $X_\ell \cap E_i$ to be close to r_i. On the other hand, if the set E_i has low density, then we expect it to contain less than r_i elements of X_ℓ, and so the rank of $X_\ell \cap E_i$ should be close to its expected cardinality. The following lemma formalizes this intuition.

LEMMA 5.4. *For every $1 \leq \ell \leq n$ and every $1 \leq i \leq k$,*

$$\begin{aligned}\mathbb{E}_{X_\ell,\mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)] &\geq \left(1-\frac{1}{e}\right)\min\{\mathbb{E}_{X_\ell}[|X_\ell \cap E_i|], r_i\}\\ &= \begin{cases}(1-1/e)\ r_i & \text{if } \lambda_i \geq n/\ell,\\ (1-1/e)\,|E_i|\,\ell/n & \text{if } \lambda_i \leq n/\ell.\end{cases}\end{aligned}$$

Proof. First note that $\mathbb{E}_{X_\ell}[|X_\ell \cap E_i|] = |E_i|(\ell/n)$. This quantity is larger than or equal to r_i if and only if $\lambda_i \geq n/\ell$. Suppose that this is the case. Using Lemma 5.3 and that the function $(1-e^{-x})$ is increasing, we obtain

$$\begin{aligned}\mathbb{E}_{X_\ell,\mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)] &\geq \big(1-\exp(-\lambda_i\ell/n)\big)r_i\\ &\geq \left(1-\frac{1}{e}\right)r_i = \left(1-\frac{1}{e}\right)\min\{\mathbb{E}_{X_\ell}[|X_\ell \cap E_i|], r_i\}.\end{aligned}$$

Suppose now that $\lambda_i \leq n/\ell$. Since the function $(1-e^{-x})/x$ is decreasing, we obtain

$$\begin{aligned}\mathbb{E}_{X_\ell,\mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)] &\geq \frac{\big(1-\exp(-\lambda_i\ell/n)\big)}{\lambda_i\ell/n}r_i\lambda_i\ell/n \geq \left(1-\frac{1}{e}\right)|E_i|\,\ell/n\\ &= \left(1-\frac{1}{e}\right)\min\{\mathbb{E}_{X_\ell}[|X_\ell \cap E_i|], r_i\}. \qquad \square\end{aligned}$$

Now we are ready to prove Theorem 5.2.

Proof of Theorem 5.2. Since the densities $(\lambda_i)_{i=1}^k$ form a decreasing sequence, there is an index i^* such that $\lambda_i \geq n/\ell$ if and only if $1 \leq i \leq i^*$. The set $\bigcup_{i=1}^{i^*} E_i$ is equal to the set F_{i^*} in the principal sequence of the matroid $\mathcal{M} \setminus L$. Let $F = L \cup F_{i^*}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Every set in the principal sequence has the same rank in both $\mathcal{M}$ and $\mathcal{Q}'$. Using this fact, properties of the rank function, and Lemma 5.4, we have

$$\begin{aligned}
\mathbb{E}_{X_\ell}[\mathrm{rk}_{\mathcal{M}}(X_\ell)] &\le \mathbb{E}_{X_\ell}[\mathrm{rk}_{\mathcal{M}}(X_\ell \cap F) + \mathrm{rk}_{\mathcal{M}}(X_\ell \cap (E \setminus F))] \\
&\le \mathrm{rk}_{\mathcal{M}}(F_{i^*}) + \mathbb{E}_{X_\ell}[|X_\ell \cap (E \setminus F)|] \\
&= \sum_{i=1}^{i^*} r_i + \sum_{i=i^*+1}^{k} |E_i|(\ell/n). \\
&\le \frac{\sum_{i=1}^{k} \mathbb{E}_{X_\ell,\mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell \cap E_i)]}{(1-1/e)} = \frac{\mathbb{E}_{X_\ell,\mathcal{Q}'}[\mathrm{rk}_{\mathcal{Q}'}(X_\ell)]}{(1-1/e)}. \qquad \square
\end{aligned}$$

Next, we prove a lemma that translates the result of Theorem 5.2 to a more useful setting for the matroid secretary problem. Let E be a set of size n. Consider two arbitrary distributions over matroids on E, and let $\mathcal{M}_1$ and $\mathcal{M}_2$ be chosen independently according to those distributions.

Lemma 5.5. *Let $\alpha \ge 0$ be a nonnegative constant. The following are equivalent:*

(i) *For all $1 \le \ell \le n$,*

$$\mathbb{E}_{X_\ell,\mathcal{M}_1}[\mathrm{rk}_{\mathcal{M}_1}(X_\ell)] \ge \alpha\, \mathbb{E}_{X_\ell,\mathcal{M}_2}[\mathrm{rk}_{\mathcal{M}_2}(X_\ell)], \tag{5.3}$$

where X_ℓ is a cardinality ℓ subset of E selected uniformly at random, independently from any random choice defining the matroids.

(ii) *For every adversarial list of weights $w_1 \ge w_2 \ge \cdots \ge w_n \ge 0$,*

$$\mathbb{E}_{\sigma,\mathcal{M}_1}[w(\mathrm{OPT}_{\mathcal{M}_1}(\sigma))] \ge \alpha \mathbb{E}_{\sigma,\mathcal{M}_2}[w(\mathrm{OPT}_{\mathcal{M}_2}(\sigma))], \tag{5.4}$$

where $\sigma\colon [n] \to E$ is a bijective map selected uniformly at random, independently from any random choice defining the matroids.

Proof. We start by rewriting $\mathbb{E}_{\sigma,\mathcal{M}}[w(\mathrm{OPT}_{\mathcal{M}}(\sigma))]$ in a more useful way. Let $X_\ell^\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(\ell)\}$ be the (random) set of elements in E receiving the top ℓ weights of the adversarial list of weights. Note that

$$\begin{aligned}
\Pr(\sigma(\ell) \in \mathrm{OPT}_{\mathcal{M}}(\sigma)) &= \Pr(\mathrm{rk}_{\mathcal{M}}(X_\ell^\sigma) - \mathrm{rk}_{\mathcal{M}}(X_{\ell-1}^\sigma) = 1) \\
&= \mathbb{E}_{\sigma,\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_\ell^\sigma)] - \mathbb{E}_{\sigma,\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_{\ell-1}^\sigma)] \\
&= \mathbb{E}_{X_\ell,\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_\ell)] - \mathbb{E}_{X_{\ell-1},\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_{\ell-1})],
\end{aligned}$$

where for the last line we used that X_ℓ^σ is a uniform random set of ℓ elements. Then

$$\begin{aligned}
\mathbb{E}_{\sigma,\mathcal{M}}[w(\mathrm{OPT}_{\mathcal{M}}(\sigma))] &= \sum_{\ell=1}^{n} w_\ell \Pr(\sigma(\ell) \in \mathrm{OPT}_{\mathcal{M}}(\sigma)) \\
&= \sum_{\ell=1}^{n} w_\ell \left(\mathbb{E}_{X_\ell,\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_\ell)] - \mathbb{E}_{X_{\ell-1},\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_{\ell-1})]\right) \\
&= w_n\, \mathbb{E}_{X_n,\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_n)] + \sum_{\ell=1}^{n-1} (w_\ell - w_{\ell+1})\, \mathbb{E}_{X_\ell,\mathcal{M}}[\mathrm{rk}_{\mathcal{M}}(X_\ell)]. \qquad (5.5)
\end{aligned}$$

Assume that condition (i) holds; then each term for $\mathcal{M}_1$ in the above sum is at least α times the corresponding term for $\mathcal{M}_2$, implying that condition (ii) holds. On the other hand, if condition (i) does not hold, then there is an index ℓ for which

$$\mathbb{E}_{X_\ell,\mathcal{M}_1}[\mathrm{rk}_{\mathcal{M}_1}(X_\ell)] < \alpha\, \mathbb{E}_{X_\ell,\mathcal{M}_2}[\mathrm{rk}_{\mathcal{M}_2}(X_\ell)].$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Consider the sequence of weights given by $w_1 = w_2 = \cdots = w_\ell = 1$ and $w_{\ell+1} = \cdots = w_n = 0$. For this sequence,

$$\begin{aligned}\mathbb{E}_{\sigma,\mathcal{M}_1}[w(\mathrm{OPT}_{\mathcal{M}_1}(\sigma))] &= \mathbb{E}_{X_\ell,\mathcal{M}_1}[\mathrm{rk}_{\mathcal{M}_1}(X_\ell)]\\ &< \alpha\,\mathbb{E}_{X_\ell,\mathcal{M}_2}[\mathrm{rk}_{\mathcal{M}_2}(X_\ell)] = \alpha\,\mathbb{E}_{\sigma,\mathcal{M}_2}[w(\mathrm{OPT}_{\mathcal{M}_2}(\sigma))]. \qquad \square\end{aligned}$$

We can use the results in this section to prove Lemma 4.2.

Proof of Lemma 4.2. The inequality follows directly from Theorem 5.2 and Lemma 5.5. $\square$

6. Principal sequence revisited. In this section we offer a self-contained, constructive proof of Theorem 5.1.

Consider a loopless matroid $\mathcal{M} = (E, \mathcal{I})$ that is not uniformly dense, and let E_1 be a *maximum cardinality set* achieving the density of $\mathcal{M}$. That is,

$$\gamma(\mathcal{M}) = \max_{\emptyset \neq X \subseteq E} \frac{|X|}{\mathrm{rk}(X)} = \frac{|E_1|}{\mathrm{rk}(E_1)}, \tag{6.1}$$

and $|E_1| \geq |X|$ for any set X achieving the same density.

We claim that the matroid $\mathcal{M}_1 = \mathcal{M}|_{E_1}$, obtained by restricting $\mathcal{M}$ to the set E_1, is uniformly dense with density $\lambda_1 = \gamma(\mathcal{M})$. Indeed, since the rank function of $\mathcal{M}_1$ is equal to that of $\mathcal{M}$, every subset of E_1 has the same density in both $\mathcal{M}$ and $\mathcal{M}_1$, making E_1 the densest set in $\mathcal{M}_1$, with density $|E_1|/\mathrm{rk}(E_1) = \gamma(\mathcal{M}) = \lambda_1$.

Consider now the matroid $\mathcal{M}_1' = \mathcal{M}/E_1$ obtained by contracting E_1 in $\mathcal{M}$. We can show that this matroid is loopless and has density strictly smaller than $\mathcal{M}_1$. Indeed, recall that the rank function of the contracted matroid (see, e.g., [27]) is

$$\mathrm{rk}_{\mathcal{M}_1'}(X) = \mathrm{rk}_{\mathcal{M}}(E_1 \cup X) - \mathrm{rk}_{\mathcal{M}}(E_1) \text{ for all } X \subseteq E \setminus E_1.$$

Hence, if $x \in E \setminus E_1$ is a loop of $\mathcal{M}_1'$, then $\mathrm{rk}_{\mathcal{M}}(E_1 \cup \{x\}) = \mathrm{rk}_{\mathcal{M}}(E_1)$, and so

$$\frac{|E_1 \cup \{x\}|}{\mathrm{rk}_{\mathcal{M}}(E_1 \cup \{x\})} = \frac{|E_1| + 1}{\mathrm{rk}_{\mathcal{M}}(E_1)} > \frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)},$$

contradicting the definition of E_1. Therefore, $\mathcal{M}_1'$ is loopless. By maximality of E_1, every set X with $\emptyset \neq X \subseteq E \setminus E_1$ satisfies

$$\frac{|E_1 \cup X|}{\mathrm{rk}_{\mathcal{M}}(E_1 \cup X)} < \frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)}.$$

Hence,

$$\frac{|X|}{\mathrm{rk}_{\mathcal{M}_1'}(X)} < \frac{\frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)}\left(\mathrm{rk}_{\mathcal{M}}(E_1 \cup X) - \mathrm{rk}_{\mathcal{M}}(E_1)\right)}{\mathrm{rk}_{\mathcal{M}}(E_1 \cup X) - \mathrm{rk}_{\mathcal{M}}(E_1)} = \frac{|E_1|}{\mathrm{rk}_{\mathcal{M}}(E_1)},$$

implying that $\gamma(\mathcal{M}_1') < \gamma(\mathcal{M})$. Thus, we have the following lemma.

LEMMA 6.1. *Let $\mathcal{M} = (E, \mathcal{I})$ be a loopless matroid on E_1 that is not uniformly dense, and E_1 the* unique *maximum cardinality set with $\gamma(\mathcal{M}) = |E_1|/\mathrm{rk}_{\mathcal{M}}(E_1)$. Then the matroid $\mathcal{M}_1 = \mathcal{M}|_{E_1}$ is uniformly dense with density $\lambda_1 = \gamma(\mathcal{M})$ and the matroid $\mathcal{M}_1' = \mathcal{M}/E_1$ is loopless with density strictly smaller than $\gamma(\mathcal{M})$.*

Proof. The only missing step to prove is that E_1 is unique. Indeed, suppose that there are distinct sets E_1 and E_1' of the same cardinality achieving the density λ_1 of $\mathcal{M}$, in particular, $\mathrm{rk}_{\mathcal{M}}(E_1) = \mathrm{rk}_{\mathcal{M}}(E_1')$. By submodularity of the rank function,

$$\frac{|E_1 \cup E_1'|}{\mathrm{rk}_{\mathcal{M}}(E_1 \cup E_1')} \geq \frac{|E_1| + |E_1'| - |E_1 \cap E_1'|}{\mathrm{rk}_{\mathcal{M}}(E_1) + \mathrm{rk}_{\mathcal{M}}(E_1') - \mathrm{rk}_{\mathcal{M}}(E_1 \cap E_1')}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Using that $\lambda_1 = |E_1| / \mathrm{rk}_{\mathcal{M}}(E_1) = |E_1'| / \mathrm{rk}_{\mathcal{M}}(E_1') \geq |E_1 \cap E_1'| / \mathrm{rk}_{\mathcal{M}}(E_1 \cap E_1')$, we have

$$\frac{|E_1 \cup E_1'|}{\mathrm{rk}_{\mathcal{M}}(E_1 \cup E_1')} \geq \frac{\lambda_1 \left(\mathrm{rk}_{\mathcal{M}}(E_1) + \mathrm{rk}_{\mathcal{M}}(E_1') - \mathrm{rk}_{\mathcal{M}}(E_1 \cap E_1')\right)}{\mathrm{rk}_{\mathcal{M}}(E_1) + \mathrm{rk}_{\mathcal{M}}(E_1') - \mathrm{rk}_{\mathcal{M}}(E_1 \cap E_1')} = \lambda_1.$$

Thus, $E_1 \cup E_1'$ is a set strictly larger than E_1 with the same density, contradicting E_1's choice. ∎

If the loopless matroid $\mathcal{M}_1'$ is not uniformly dense, we can use the above lemma in this matroid to find a second uniformly dense matroid $\mathcal{M}_2 = \mathcal{M}_1'|_{E_2}$ with density

$$\lambda_2 = \gamma(\mathcal{M}_1') = \frac{|E_2|}{\mathrm{rk}_{\mathcal{M}_1'}(E_2)} = \frac{|E_2|}{\mathrm{rk}_{\mathcal{M}}(E_1 \cup E_2) - \mathrm{rk}_{\mathcal{M}}(E_1)} < \lambda_1$$

and a loopless matroid $\mathcal{M}_2' = \mathcal{M}_1'/E_2$ of strictly smaller density. Here, E_2 is the maximum cardinality set achieving $\mathcal{M}_1'$'s density. By repeating this process we obtain a sequence of sets $(E_1, \ldots, E_k)$ partitioning E and a sequence of values $\lambda_1 > \lambda_2 > \cdots > \lambda_k \geq 0$. Furthermore, for every $0 \leq i \leq k$, define $F_i = \bigcup_{j=1}^{i} E_i$. The sequences $(F_i)_{i=0}^{k}$ and $(\lambda_i)_{i=1}^{k}$ are exactly the ingredients we need to prove Theorem 5.1.

Proof of Theorem 5.1. We know from properties of matroids that contractions and deletions (restrictions) commute (see [27]). Therefore, the matroids $\mathcal{M}_i = (\mathcal{M}/F_{i-1})|_{(F_i \setminus F_{i-1})}$, for $1 \leq i \leq k$, coincide with the uniformly dense matroids constructed iteratively using Lemma 6.1. The density condition in the statement of the theorem also holds since

$$\gamma(\mathcal{M}_i) = \frac{|E_i|}{\mathrm{rk}_{\mathcal{M}_i}(E_i)} = \frac{|E_i|}{\mathrm{rk}_{\mathcal{M}}(E_i \cup F_{i-1}) - \mathrm{rk}_{\mathcal{M}}(F_{i-1})}.$$

To finish the proof, let I_i be an independent set of matroid $\mathcal{M}_i$ for every $i \in [k]$. We prove by induction that $\bigcup_{i=1}^{j} I_i$ is independent in $\mathcal{M}|_{F_j}$. The claim trivially holds for $j = 1$. For $2 \leq j \leq k$, the fact that I_j is independent in $\mathcal{M}_j = (\mathcal{M}/F_{j-1})|_{E_j} = \left(\mathcal{M}|_{F_j}\right)/F_{j-1}$ implies that $I_j \cup J$ is independent in $\mathcal{M}|_{F_j}$ for any J independent in $\mathcal{M}|_{F_{j-1}}$. By setting $J = \bigcup_{i=1}^{j-1} I_i$ we conclude the proof. ∎

For an example illustrating the principal sequence of a graphic matroid, we refer the reader to [24, section 11.4.3].

7. Results for the adversarial-assignment random-order model. Constant competitive algorithms for the adversarial-assignment random-order model of the matroid secretary problem remain elusive.

Unfortunately, uniformly dense matroids are as hard as general matroids in this model: Any algorithm $\mathcal{A}$ for uniformly dense matroids can be modified into an algorithm for general matroids having the same competitive ratio as $\mathcal{A}$. This follows from a result of Lai and Lai [19] stating that every matroid $\mathcal{M}$ is a restriction of a uniformly dense matroid $\mathcal{M}'$. The algorithm for $\mathcal{M}$ would virtually complete the matroid $\mathcal{M}'$ by adding a dummy set of zero weight elements and then run algorithm $\mathcal{A}$ on $\mathcal{M}'$, simulating the augmented input in such a way that the dummy elements arrive uniformly at random similarly to the real ones.

In 2007, Babaioff et al. proposed an $O(\log r)$-competitive algorithm for the standard model. This procedure has many features, including that it does not need to know the matroid beforehand; it only needs to know the number of elements and to have access to an oracle for testing independence only on subsets of elements it has already seen. Nevertheless, this algorithm makes use of the actual values of the

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

weights being revealed. We start this section by presenting a new algorithm having the same features but using only the relative order of the weights and not their numerical values. Later, we show new constant-competitive algorithms in this model for certain matroid classes.

Remark. During the realization of this work, Chakraborty and Lachish [9] proposed an improved $O(\sqrt{\log r})$-competitive algorithm for this problem. In its current form, their algorithm relies on the numerical values of the observed weights. It is not clear whether their techniques can be adapted to the setting in which the only available information is the relative order of the observed weights.

7.1. General $O(\log r)$-competitive algorithm. The following algorithm returns an independent set of the matroid $\mathcal{M}$: With probability $1/2$, run the classical secretary algorithm (say, Algorithm 1) on the set of nonloops of the matroid. This returns the heaviest nonloop of the stream with probability at least $1/e$. Otherwise, observe and reject the first m elements of the stream, where m is chosen from the binomial distribution $\mathrm{Bin}(n, 1/2)$ (as usual, denote this set of elements as the *sample*) and compute the optimum base $A = \{a_1, \dots, a_k\}$ (with $w(a_1) > \cdots > w(a_k)$) of the sampled elements. Afterwards, select uniformly at random a number $\ell \in \{1, 3, 9, \dots, 3^t\}$ with $t = \lfloor \log_3 r \rfloor$, run the greedy procedure on the set of nonsampled elements having weight at least that of a_ℓ as they arrive, and return its answer (if $\ell > k$, run the greedy procedure over the entire set of nonsampled elements).

It is possible to implement this algorithm without even knowing the matroid beforehand: it is enough to know the number of elements n and have access to an oracle to test independence on subsets of already seen elements. For that we need to make two changes to the algorithm above.

First we require making a slight modification to the algorithm for the classical secretary problem we use (Algorithm 1) so that it considers only the nonloops of the stream *without knowing a priori the number of nonloops.* The modification samples the first N elements of the stream, where N is distributed as $\mathrm{Bin}(n, 1/e)$, and then returns the first nonloop having weight larger than any sampled nonloop (if any). Observe that if n' is the number of nonloops of the matroid, then the number of nonloops sampled has distribution $\mathrm{Bin}(n', 1/e)$. This observation implies this algorithm does exactly what Algorithm 1 would do if it knew the number of nonloops beforehand.

The second change deals with the number $t = \lfloor \log_3 r \rfloor$ in the algorithm above. As we do not know the rank of the matroid a priori, we cannot use this value. Instead, we use the rank k of the sampled set (which we can compute) to estimate it: We select $t \in \{\lfloor \log_3 k \rfloor, \lfloor \log_3 k \rfloor + 1\}$ uniformly at random and use this value in the previous algorithm. The full description of this algorithm is depicted as Algorithm 6.

To analyze this algorithm, note that for every number ℓ the algorithm can choose (provided that $\ell \le k$), the sample contains an independent set of size ℓ containing only elements of weight at least the one of a_ℓ (namely the set $\{a_1, \dots, a_\ell\}$ itself). Since the sampled set behaves similarly to the nonsampled one, we expect the same to happen outside the sample. In particular, the greedy procedure should recover a weight of roughly $\ell w(a_\ell)$. By taking the expectation over the choices of ℓ it is not hard to check that the expected weight returned by the algorithm is at least $\Omega(\mathbb{E}[w(A)/\log_3(r)]) = \mathbb{E}[w(\mathrm{OPT})]\,\Omega(1/\log_3(r))$. We give the formal result below.

THEOREM 7.1. *Algorithm* 6 *is* $O(\log r)$*-competitive for any matroid of rank* r*.*

Proof. Assume first that the rank r of the matroid is known. Let $\mathrm{OPT} = \{x_1, \dots, x_r\}$, with $w(x_1) > \cdots > w(x_r)$, be the maximum independent set of the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 6. For general matroids.

Flip a fair coin. On heads, run the modified version of Algorithm 1 that returns the largest nonloop with probability at least $1/e$.
Otherwise, set ALG $\leftarrow \emptyset$.
Choose m from the binomial distribution $\mathrm{Bin}(n, 1/2)$.
Observe the first m elements and denote this set as the sample.
Compute the optimum base A for the sample. Let $a_1, \dots, a_k$ be the elements of A in decreasing order of weight.
If the total rank r of the matroid is known, set $t = \lfloor \log_3 r \rfloor$; otherwise, select $t \in \{\lfloor \log_3 k \rfloor, \lfloor \log_3 k \rfloor + 1\}$ uniformly at random.
Select ℓ uniformly at random from the set $\{1, 3, 9, \dots, 3^t\}$.
for each element x arriving after the first m elements **do**
    If ALG $\cup \{x\}$ is independent and $w(x) \geq w(a_\ell)$ (where $w(a_\ell) = 0$ if $\ell > k$), add x to ALG.
end for
Return the set ALG.

matroid, let T be the set of sampled elements, and let $T' = E \setminus T$ be the set of nonsampled ones.

Let $A = \{a_1, \dots, a_k\}$ be the optimum set in T, with $w(a_1) > \cdots > w(a_k)$ (independent of whether the algorithm computes A or not). An element in T is included into A if and only if it is not spanned by the elements in T that are heavier than it. In particular, if an element x_i of the optimum is in T, then as x_i is outside the span of all elements in E heavier than it, x_i must appear in A.

Every element of the matroid is sampled independently with probability 1/2, including the elements of the optimum. Therefore, by the previous paragraph,

$$\mathbb{E}[w(A)] \geq \frac{w(\mathrm{OPT})}{2}. \tag{7.1}$$

To simplify our analysis, in the following we assume that for $i > k$, a_i is a dummy element with $w(a_i) = 0$. Given the number ℓ chosen by the algorithm (if the algorithm reaches that state), the weight of the returned set will be at least $w(a_\ell)$ times the number of elements the greedy procedure selects; therefore, $\mathbb{E}[w(\mathrm{ALG})]$ is at least

$$\frac{w(x_1)}{2e} + \frac{1}{2(1 + \lfloor \log_3 r \rfloor)} \sum_{j=0}^{\lfloor \log_3 r \rfloor} \mathbb{E}\left[w(a_\ell) \cdot |\mathrm{ALG}| \;\middle|\; \ell = 3^j \text{ was selected} \right]. \tag{7.2}$$

Let $H(a_\ell)$ be the collection of nonsampled elements that are at least as heavy as a_ℓ. If the algorithm chooses the number ℓ, it will then execute the greedy procedure on $H(a_\ell)$ and return a set of cardinality equal to the rank of $H(a_\ell)$. Note that for every ℓ, $w(x_\ell) \geq w(a_\ell)$; therefore, the rank of $H(a_\ell)$ is at least the number of nonsampled elements in $\{x_1, \dots, x_\ell\}$.

By a Chernoff bound (see, e.g., [22]), $\Pr\Big(|\{x_1, \dots, x_\ell\} \cap T'| \leq \ell/4 \Big) \leq \exp(-\ell/8)$. In particular, if $\ell \geq 9$,

$$\mathbb{E}[w(a_\ell) \cdot |\mathrm{ALG}| \mid \ell\,] \geq \mathbb{E}[w(a_\ell)]\,(1 - \exp(-\ell/8))\,\ell/4 \geq \mathbb{E}[w(a_\ell)]\,\ell/6.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Replacing this in (7.2) and dropping the values of j for which $\ell < 9$, we get

$$\begin{aligned}\mathbb{E}[w(\mathrm{ALG})] &\geq \frac{w(x_1)}{2e} + \frac{1}{12(1+\lfloor \log_3 r\rfloor)} \sum_{j=2}^{\lfloor \log_3 r\rfloor} \mathbb{E}[w(a_{3^j})]\,3^j \\ &\geq \mathbb{E}\left[\frac{w(\{a_1,\dots,a_8\})}{16e} + \frac{1}{24(1+\lfloor \log_3 r\rfloor)} \sum_{j=2}^{\lfloor \log_3 r\rfloor} w(\{a_{3^j},\dots,a_{3^{j+1}-1}\})\right] \\ &\geq \frac{\mathbb{E}[w(A)]}{16e(1+\lfloor \log_3 r\rfloor)}.\end{aligned}$$

Using inequality (7.1), we obtain

$$\mathbb{E}[w(\mathrm{ALG})] \geq \frac{w(\mathrm{OPT})}{32e(1+\lfloor \log_3 r\rfloor)},$$

which implies the algorithm is $O(\log r)$-competitive.

Suppose now that the rank r is unknown. If r is small, say $r \leq 12$, then with probability $1/(2e)$ the algorithm will run the standard secretary algorithm and return element x_1. This element has weight at least $1/12$ fraction of the optimum; therefore the algorithm is $24e$-competitive for this case.

For the case where $r > 12$ we use a different analysis. The random variable k denoting the rank of the sampled set could be strictly smaller than r. However, the probability that $k \leq r/3$ is small. Indeed, for that event to happen we require that at most $1/3$ of the elements of OPT are in the sample. By Chernoff bound, this happens with probability

$$\Pr\Big(|\{x_1,\dots,x_r\} \cap T| \leq r/3\Big) \leq \exp(-r/18) \leq \exp(-13/18) \leq 1/2.$$

Noting that $r/3 \leq k \leq r$ implies that $\lfloor \log_3 r\rfloor \in \{\lfloor \log_3 k\rfloor, \lfloor \log_3 k\rfloor + 1\}$, we deduce that with probability at least $1/4$ our algorithm guesses $t = \lfloor \log_3 r\rfloor$ right; therefore, the competitive ratio of this algorithm is at most 4 times worse than the one that knows the rank beforehand. $\square$

7.2. Column-sparse linear matroids. Let A be an $m \times n$ matrix over a field $\mathbb{F}$ whose columns are indexed by a set V. The *linear matroid* represented by A is the matroid $\mathcal{M} = (V, \mathcal{I})$ whose independent sets are those for which the corresponding columns are linearly independent as vectors. Formally, if $\{A_v : v \in V\}$ is the multiset[5] of columns of A, then a set $I \subseteq V$ is independent in $\mathcal{M}$ if and only if the family $\{A_v : v \in I\}$ of vectors is linearly independent in $\mathbb{F}^m$ (in particular, this family cannot contain repeated vectors). A matroid is *linear* if there exists a matrix A that represents it. Examples of such matroids include partition, laminar, and graphic matroids.

Consider the following algorithm, depicted as Algorithm 7, for a linear matroid $\mathcal{M} = (V, \mathcal{I})$ represented by A: Randomly permute the rows of A to obtain a new matrix A'. Define for every $i \in [m]$ the sets $C_i = \{v \in V : v_i \neq 0\}$ and $B_i = C_i \setminus \bigcup_{j<i} C_j$, where $v_i = A'_{i,v}$ denotes the ith coordinate of the column indexed by v in the permuted matrix A'. Next, run any e-competitive secretary algorithm for the partition matroid that accepts at most one element from each B_i, and return its answer.

[5]Note that A may contain several repeated columns.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ALGORITHM 7. For a matroid $\mathcal{M} = (V, \mathcal{I})$ represented by a matrix A.

1: Permute the rows of A at random to obtain a matrix A'. Index the rows of A' as $1, 2, \ldots, m$.
2: Let $C_i = \{v \in V : v_i \neq 0\}$ and $B_i = C_i \setminus \bigcup_{j<i} C_j$, where v_i is the ith coordinate of the column associated to v in matrix A'.
3: Let $\mathcal{P}$ be the partition matroid on V whose independent sets contain at most one element from each B_i.
4: Run any e-competitive secretary algorithm for $\mathcal{P}$ on the stream of elements and return its answer.

THEOREM 7.2. *Algorithm* 7 *returns an independent set of* $\mathcal{M}$. *Furthermore, if every column of the matrix* A *representing* $\mathcal{M}$ *contains at most* k *nonzero entries, then this algorithm is* ke*-competitive (assuming a random-order model).*

Proof. We first show that the returned set is independent. If this were not the case, there would be a circuit $S \subseteq V$ in the output. Let i be the smallest index such that $B_i \cap S \neq \emptyset$, and let v be the unique element in $B_i \cap S$. Since S is a circuit, the vector A'_v is a linear combination of vectors in $\{A'_w : w \in S \setminus \{v\}\}$. This is a contradiction since, by definition of i and B_i, we have $S \cap C_i = \{v\}$, meaning that v is the only element of S for which $v_i = A'_{i,v} \neq 0$.

To show that the algorithm is ke-competitive, construct the bipartite graph G with parts the rows and columns of A, where there is an edge (i, v) from row $i \in [m]$ to column $v \in V$ if the entry $A_{i,v}$ is nonzero. Assign to every edge of G incident to column v a weight equal to the weight of the matroid element v.

Consider the following simulation algorithm: Select a random permutation τ of $[m]$, and let G^τ be the subgraph of G that contains only those edges going from a column vertex v to its lowest neighbor according to τ. In other words, $(i, v) \in E(G^\tau)$ if and only if, for all $j \in [m], (j, v) \in E(G) \Rightarrow \tau(i) < \tau(j)$. Finally, run any e-competitive secretary algorithm for the partition matroid on the edge set of G^τ that accepts for each row vertex at most one edge incident to it. It is easy to see that this simulation returns a matching in G with the same weight as the set of elements Algorithm 7 returns. Thence, we can analyze the output of the simulation.

If $X \subseteq V$ is independent in $\mathcal{M}$, then the row-rank of the submatrix of A induced by X equals its cardinality. In particular, the number of row vertices that X dominates in G is at least $|X|$. Using Hall's theorem we conclude that for every independent set of columns, there is a matching covering the associated vertices. From here we conclude that the weight of the maximum weight matching M^* of G is at least that of the optimum independent set of $\mathcal{M}$. On the other hand, the weight of M^* is at most that of the edge set $\{(i, v^*(i)) : i \in [m]\}$, where $v^*(i) = \arg\max\{w(v) : v \in V, (i, v) \in G\}$ is the maximum weight neighbor of i in G. Since each edge $(i, v^*(i))$ is in $E(G^\tau)$ with probability $1/k$ and that, given this event, the simulation selects $(i, v^*(i))$ with probability $1/e$, we conclude that for the set ALG returned by the algorithm,

$$w(\mathrm{ALG}) \geq \frac{1}{ke} \sum_{i=1}^{n} w(v^*(i)) \geq \frac{1}{ke} w(M^*) \geq \frac{1}{ke} w(\mathrm{OPT}). \qquad \square$$

By applying Algorithm 7 to graphic matroids, which are representable by matrices having only two 1's per column, we recover the $2e$-competitive algorithm of Korula and Pál [18].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

7.3. Low-density matroids. The *matroid polytope* $\mathrm{P}_{\mathcal{M}} \subseteq \mathbb{R}^E$ of a matroid $\mathcal{M} = (E, \mathcal{I})$ is the convex hull of the indicator vectors of its independent sets. This polytope can be characterized (see, e.g., [28, Chapter 40]) as

$$\mathrm{P}_{\mathcal{M}} = \{y \in \mathbb{R}^E \colon y \geq 0 \text{ and } y(U) \leq \mathrm{rk}(U) \text{ for all } U \subseteq E\}.$$

Let $\mathcal{M} = (E, \mathcal{I})$ be a loopless matroid of density $\gamma(\mathcal{M}) = \max_{\emptyset \neq U \subseteq E} \frac{|U|}{\mathrm{rk}(U)}$, and let $\tau^{\mathcal{M}} \in \mathbb{R}^E$ be the vector having all its coordinates equal to $1/\gamma(\mathcal{M})$. We have the following property.

Lemma 7.3. *For every loopless matroid $\mathcal{M}$, the vector $\tau^{\mathcal{M}}$ is in the matroid polytope $\mathrm{P}_{\mathcal{M}}$.*

Proof. For every $U \subseteq E$, $\tau^{\mathcal{M}}(U) = \sum_{u \in U} \tau^{\mathcal{M}}(u) = \frac{|U|}{\gamma(\mathcal{M})} \leq \mathrm{rk}(U)$. ◻

The previous lemma implies that $\tau^{\mathcal{M}}$ admits a decomposition as a convex combination of independent sets of $\mathcal{M}$:

$$\tau^{\mathcal{M}} = \sum_{I \in \mathcal{I}} \lambda_I \chi_I, \text{ with } \sum_{I \in \mathcal{I}} \lambda_I = 1.$$

This decomposition can be found in polynomial time given access to an independence oracle of $\mathcal{M}$ (see [28, Chapter 40]). Consider the following algorithm (depicted as Algorithm 8) for a loopless matroid $\mathcal{M} = (E, \mathcal{I})$.

Algorithm 8. For loopless matroid $\mathcal{M} = (E, \mathcal{I})$.

1: Compute the decomposition $\tau^{\mathcal{M}} = \sum_{I \in \mathcal{I}} \lambda_I \chi_I$.
2: Select and return a set $I \in \mathcal{I}$ according to the probability distribution $(\lambda_I)_{I \in \mathcal{I}}$.

Lemma 7.4. *For every loopless matroid $\mathcal{M}$, Algorithm 8 is $\gamma(\mathcal{M})$-competitive in* any *model of the matroid secretary problem (even adversarial-assignment adversarial-order).*

Proof. Every element $u \in E$ is in the output with probability

$$\sum_{I \in \mathcal{I} \colon u \in I} \lambda_I = \tau^{\mathcal{M}}(u) = 1/\gamma(\mathcal{M}).$$

Therefore, the expected weight returned is at least $1/\gamma(\mathcal{M})$ times the collective total weight of all the elements in the matroid. ◻

If a loopless matroid $\mathcal{M}$ contains parallel elements,[6] we can potentially get a better competitive ratio by using its *simple version* $\mathcal{M}'$. This matroid $\mathcal{M}' = (E', \mathcal{I}')$ is obtained by deleting all but one element in each parallel class of $\mathcal{M} = (E, \mathcal{I})$. In particular, for every element u in $\mathcal{M}$ there is a unique element in $\mathcal{M}'$ representing u's parallel class.

For every independent set $I' \in \mathcal{M}'$, let $\mathcal{Q}(I')$ be the partition matroid in E induced by I'. In other words, the independent sets of $\mathcal{Q}(I')$ are those subsets of E containing at most one element from each parallel class represented in I'. In particular, every independent set in $\mathcal{Q}(I')$ is independent in the original matroid $\mathcal{M}$. Consider Algorithm 9 depicted below.

[6] Two elements u and v are parallel in $\mathcal{M}$ if $\{u, v\}$ is a minimal dependent set. Being parallel is an equivalence relation (considering that every element is parallel to itself). The parallel classes of $\mathcal{M}$ are the equivalence classes of this relation. A matroid is called simple if it has no loops and no pair of parallel elements.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 9. For loopless matroid $\mathcal{M} = (E, \mathcal{I})$.

1: Construct the simple matroid $\mathcal{M}' = (E', \mathcal{I}')$ and the decomposition $\tau^{\mathcal{M}'} = \sum_{I' \in \mathcal{I}'} \lambda_{I'} \chi_{I'}$.
2: Select a set $I' \in \mathcal{I}'$ according to the probability distribution $(\lambda'_I)_{I' \in \mathcal{I}'}$.
3: Run any e-competitive algorithm for partition matroids on the matroid $\mathcal{Q}(I')$ and return its answer.

Lemma 7.5. *For every loopless matroid $\mathcal{M}$, Algorithm 9 is $e\gamma(\mathcal{M}')$-competitive in the adversarial-assignment random-order model of the matroid secretary problem.*

Proof. The returned set is independent in the original matroid $\mathcal{M}$; hence the algorithm is correct. Note that every element u of the optimum base of $\mathcal{M}$ is the heaviest of its own parallel class. Provided that the parallel class of u is represented in the set I' selected in line 2, Algorithm 9 returns u with probability at least $1/e$. We conclude the proof by noting that every parallel class (i.e., every element of E') is selected with probability $\gamma(\mathcal{M}')$. ∎

We can modify the previous algorithms to work on matroids having loops: Simply run them on the matroid obtained by removing the loops. The competitive ratios of the described algorithms are linear in the density of the matroid $\mathcal{M}$ (or in the density of its simple version $\mathcal{M}'$). In particular for matroids of constant density, these algorithms are constant-competitive.

In the next subsections we describe two interesting classes of matroids having this property: cographic matroids and small cocircuit matroids.

7.4. Cographic matroids. The cographic matroid $\mathcal{M}^*(G)$ of a graph G is the dual of its graphic matroid $\mathcal{M}(G)$. The independent sets in $\mathcal{M}^*(G)$ are those edge sets whose removal does not increase the number of connected components of G. The bases in $\mathcal{M}^*(G)$ are the complements of the maximum forests of G. The circuits (minimal dependent sets) of $\mathcal{M}^*(G)$ are exactly the minimal edge-cuts of the connected components of G. This means that the loops of $\mathcal{M}^*(G)$ are the bridges of G.

A well-known result of graph theory states that when G is 3-edge-connected we can find three spanning trees T_1, T_2, and T_3, such that the union of their complements covers $E(G)$ (this follows from, e.g., Edmonds' matroid partitioning theorem [12]). In particular, we have the following.

Lemma 7.6. *If every connected component of $G = (V, E)$ is 3-edge-connected, then $\gamma(\mathcal{M}^*(G)) \leq 3$.*

Proof. The above result implies that there are 3 forests F_1, F_2, and F_3 whose complements cover all the edges of G. Let $B_i = E \setminus F_i$. For every set of edges $X \subseteq E$,

$$|X| \leq \sum_{i=1}^{3} |X \cap B_i| = \sum_{i=1}^{3} \mathrm{rk}_{\mathcal{M}^*(G)}(X \cap B_i) \leq 3\,\mathrm{rk}_{\mathcal{M}^*(G)}(X),$$

where the middle equality follows since $X \cap B_i \subseteq B_i$ is independent in $\mathcal{M}^*(G)$. ∎

This result implies that Algorithm 8 is 3-competitive for cographic matroids of graphs having only 3-edge-connected components. An alternative algorithm in the same spirit is depicted as Algorithm 10.

Lemma 7.7. *Algorithm 10 is 3-competitive in any model of the matroid secretary problem for cographic matroids of graphs whose components are 3-edge-connected.*

Proof. This holds as every edge is selected with probability at least 1/3. ∎

We cannot extend the previous results to arbitrary graphs: the cographic matroid of a bridgeless graph G can have arbitrarily high density. To see this, consider the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

---

ALGORITHM 10. For the cographic matroid of a graph $G = (V, E)$ having only 3-edge-connected components.

1: Find three forests F_1, F_2, and F_3 whose complements cover E.
2: Select $i \in \{1, 2, 3\}$ uniformly at random and return the set $B_i = E \setminus F_i$.

---

cycle C_n on n vertices. For this graph $|E(C_n)|/\mathrm{rk}_{\mathcal{M}^*}(C_n) = n$; nevertheless we can still show the following result.

LEMMA 7.8. *For any bridgeless graph G, the simple version of its cographic matroid has density at most* 3.

Proof. Consider the collection $\{C_1, \ldots, C_k\}$ of 2-edge-connected components[7] of the bridgeless graph G, and the graph H obtained by taking the disjoint union of the graphs C_i (using copies of the vertices that are in two or more of the C_i's). The graph H has the same graphic and cographic matroids as G, so we use H instead.

Let $P_1, \ldots, P_\ell$ be the cographic parallel classes of $\mathcal{M}^*(H)$. Recall that to obtain the simple version $\mathcal{M}'$ of $\mathcal{M}$ we need to delete (in the matroid sense) all but one element of each P_j. Since deleting an element of $\mathcal{M}$ corresponds to contracting the same element in its dual, which is a graphic matroid, we conclude that $\mathcal{M}'$ is the cographic matroid of the graph H' obtained by contracting in H (in the graph sense) all but one edge of each P_j.

Since $\mathcal{M}'$ has no pair of parallel elements, the components of H' have no edge-cut of size 2. Therefore, all the components of H' are 3-edge-connected. Using Lemma 7.6, we conclude that $\mathcal{M}'$ has density at most 3. ∎

Let $\mathcal{M}$ be the cographic matroid of an arbitrary graph G. The result above implies that the procedure that removes the bridges of G and then applies Algorithm 9 to the resulting graph is $3e$-competitive for $\mathcal{M}$. Alternatively, we can also use the procedure depicted as Algorithm 11.

ALGORITHM 11. For the cographic matroid of a graph $G = (V, E)$.

1: Remove the bridges of G.
2: Construct the associated graph H' described in the proof of Lemma 7.8.
3: Find three forests F_1, F_2, and F_3 whose complements cover H'.
4: Define the partition matroids $\mathcal{Q}_i = \mathcal{Q}(E(H') \setminus F_i)$ having $E(G)$ as ground set (see subsection 7.3).
5: Select $i \in \{1, 2, 3\}$ uniformly and run any e-competitive algorithm for partition matroids on $\mathcal{Q}_i$, returning its answer.

LEMMA 7.9. *Algorithm* 11 *is* $3e$*-competitive for general cographic matroids in the adversarial-assignment random-order model.*

Proof. The proof is analogous to that of Lemma 7.5. ∎

7.5. Matroids with small cocircuits. For each element u of a loopless matroid $\mathcal{M} = (E, \mathcal{I})$, let $c^*(u)$ be the size of the smallest cocircuit (i.e., circuits of the dual matroid) containing u, and let

$$c^*(\mathcal{M}) = \max_{u \in E} c^*(u). \tag{7.3}$$

[7]A 2-edge-connected component is a maximal 2-edge-connected subgraph. The 2-edge-connected components of a bridgeless graph provide a partition of the edges of the graph.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Consider the algorithm that greedily constructs an independent set of $\mathcal{M}$ selecting elements as they appear without even looking at their weights.

THEOREM 7.10. *The algorithm described above is $c^*(\mathcal{M})$-competitive (in random-order models).*

Proof. Fix $u \in E$ and let C^* be a cocircuit of minimum size containing it. If u appears before all the other elements of C^* in the random order, then it has to be selected by the algorithm. Otherwise, there would be a circuit C that intersects C^* only in element u, which is a contradiction (see, e.g., [27, Proposition 2.1.11]). We conclude that u is selected with probability at least $1/c^*(u) \geq 1/c^*(\mathcal{M})$. ∎

LEMMA 7.11. *For every loopless matroid $\mathcal{M} = (E, \mathcal{I})$, $\gamma(\mathcal{M}) \leq c^*(\mathcal{M})$.*

Proof. Let n be the size of E. An element u is selected by the algorithm above if and only if u is in the lexicographic first base $\mathrm{OPT}(\pi)$ of the ordering $\pi\colon [n] \to E$ in which the elements are presented. Consider the vector $\rho \in \mathbb{R}^E$ having each coordinate equal to $1/c^*(\mathcal{M})$. Using the proof of Theorem 7.10, we conclude that

$$\rho(u) \leq \Pr_\pi(u \text{ is selected by the algorithm}) = \frac{1}{n!}\sum_\pi \chi_{\mathrm{OPT}(\pi)}(u).$$

In particular, for every set $U \subseteq E$ we have

$$\frac{|U|}{c^*(\mathcal{M})} = \rho(U) \leq \frac{1}{n!}\sum_\pi |U \cap \mathrm{OPT}(\pi)| \leq \frac{1}{n!}\sum_\pi \mathrm{rk}(U) = \mathrm{rk}(U),$$

where we used the fact that $U \cap \mathrm{OPT}(\pi)$ is an independent subset of U. ∎

The last lemma shows that the algorithm presented in this section is no better than Algorithm 8 for low-density matroids in terms of its competitive ratio. Nevertheless, the algorithm above is simpler and does not require knowledge of the matroid beforehand.

Acknowledgment. The author would like to thank the anonymous reviewers for their valuable comments and suggestions, which were helpful in improving this article.

REFERENCES

[1] M. AJTAI, N. MEGIDDO, AND O. WAARTS, *Improved algorithms and analysis for secretary problems and generalizations*, in Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science, FOCS, Washington, DC, 1995, pp. 473–482.

[2] M. BABAIOFF, M. DINITZ, A. GUPTA, N. IMMORLICA, AND K. TALWAR, *Secretary problems: Weights and discounts*, in Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, Philadelphia, 2009, pp. 1245–1254.

[3] M. BABAIOFF, N. IMMORLICA, D. KEMPE, AND R. KLEINBERG, *A knapsack secretary problem with applications*, in Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX, 2007, pp. 16–28.

[4] M. BABAIOFF, N. IMMORLICA, D. KEMPE, AND R. KLEINBERG, *Online auctions and generalized secretary problems*, SIGecom Exchanges, 7 (2008), pp. 1–11.

[5] M. BABAIOFF, N. IMMORLICA, AND R. KLEINBERG, *Matroids, secretary problems, and online mechanisms*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, Philadelphia, 2007, pp. 434–443.

[6] M. BATENI, M. HAJIAGHAYI, AND M. ZADIMOGHADDAM, *Submodular secretary problem and extensions*, in Proceedings of the 13th International Workshop on Approximation and the 14th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX, 2010, pp. 39–52.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/22/14 to 200.89.68.74. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

[7] N. Buchbinder, K. Jain, and M. Singh, *Secretary problems via linear programming*, in Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, IPCO, 2010, pp. 163–176.

[8] P. A. Catlin, J. W. Grossman, A. M. Hobbs, and H.-J. Lai, *Fractional arboricity, strength, and principal partitions in graphs and matroids*, Discrete Appl. Math., 40 (1992), pp. 285–302.

[9] S. Chakraborty and O. Lachish, *Improved competitive ratio for the matroid secretary problem*, in Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Philadelphia, 2012, pp. 1702–1712.

[10] N. B. Dimitrov and C. G. Plaxton, *Competitive weighted matching in transversal matroids*, in Proceedings of the 35th International Colloquium on Automata, Languages and Programming, ICALP, Part I, Springer, Berlin, Heidelberg, 2008, pp. 397–408.

[11] E. B. Dynkin, *The optimum choice of the instant for stopping a Markov process*, Soviet Math. Dokl., 4 (1963), pp. 627–629.

[12] J. Edmonds, *Minimum partition of a matroid into independent subsets*, J. Res. Nat. Bur. Standards Sect. B, 69 (1965), pp. 67–72.

[13] T. S. Ferguson, *Who solved the secretary problem?*, Statist. Sci., 4 (1989), pp. 282–289.

[14] P. R. Freeman, *The secretary problem and its extensions: A review*, Internat. Statist. Rev., 51 (1983), pp. 189–206.

[15] S. Fujishige, *Theory of principal partitions revisited*, in Research Trends in Combinatorial Optimization, Springer, Berlin, 2009, pp. 127–162.

[16] S. Im and Y. Wang, *Secretary problems: Laminar matroid and interval scheduling*, in Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, Philadelphia, 2011, pp. 1265–1274.

[17] R. Kleinberg, *A multiple-choice secretary algorithm with applications to online auctions*, in Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, Philadelphia, 2005, pp. 630–631.

[18] N. Korula and M. Pál, *Algorithms for secretary problems on graphs and hypergraphs*, in Proceedings of the 36th International Colloquium on Automata, Languages and Programming, ICALP, Springer, Berlin, Heidelberg, 2009, pp. 508–520.

[19] H.-J. Lai and H. Lai, *Every matroid is a submatroid of a uniformly dense matroid*, Discrete Appl. Math., 63 (1995), pp. 151–160.

[20] D. V. Lindley, *Dynamic programming and decision theory*, Appl. Statist., 10 (1961), pp. 39–51.

[21] R. B. Manfrino, J. A. G. Ortega, and R. V. Delgado, *Inequalities: A Mathematical Olympiad Approach*, Birkhäuser, Basel, 2009.

[22] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, New York, 1995.

[23] H. Narayanan, *Theory of Matroids and Network Analysis*, Ph.D. thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, 1974.

[24] H. Narayanan, *Submodular Function and Electrical Networks*, Ann. Discrete Math. 54, North–Holland, Amsterdam, The Netherlands, 1997.

[25] H. Narayanan and M. N. Vartak, *An elementary approach to the principal partition of a matroid*, Trans. Inst. Electronics Commun. Engineers Japan Sect. E, 64 (1981), pp. 227–234.

[26] S. Oveis Gharan and J. Vondrák, *On variants of the matroid secretary problem*, in Proceedings of the 19th Annual European Symposium on Algorithms, ESA, 2011, pp. 335–346.

[27] J. G. Oxley, *Matroid Theory*, Oxford University Press, New York, 2006.

[28] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, Berlin, 2003.

[29] J. A. Soto, *Matroid secretary problem in the random assignment model*, in Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, Philadelphia, 2011, pp. 1275–1284.

[30] N. Tomizawa, *Strongly irreducible matroids and principal partition of a matroid into strongly irreducible minors*, Electron. Commun. Japan, 59 (1976), pp. 1–10.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.