2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Random Order Contention Resolution Schemes

Marek Adamczyk
University of Warsaw
m.adamczyk@mimuw.edu.pl

Michał Włodarczyk
University of Warsaw
m.wlodarczyk@mimuw.edu.pl

***Abstract*—Contention resolution schemes have proven to be an incredibly powerful concept which allows tackling a broad class of problems. The framework has been initially designed to handle submodular optimization under various types of constraints, that is, intersections of exchange systems (including matroids), knapsacks, and unsplittable flows on trees. Later on, it turned out that this framework perfectly extends to optimization under uncertainty, like stochastic probing and online selection problems, which further can be applied to mechanism design.**

We add to this line of work by showing how to create contention resolution schemes for intersection of matroids and knapsacks when we work in the random order setting. More precisely, we do know the whole universe of elements in advance, but they appear in an order given by a random permutation. Upon arrival we need to irrevocably decide whether to take an element or not. We bring a novel technique for analyzing procedures in the random order setting that is based on the martingale theory. This unified approach makes it easier to combine constraints, and we do not need to rely on the monotonicity of contention resolution schemes, as it was the case before.

Our paper fills the gaps, extends, and creates connections between many previous results and techniques. The main application of our framework is a $k + 4 + \varepsilon$ approximation ratio for the Bayesian multi-parameter unit-demand mechanism design under the constraint of k matroids intersection, which improves upon the previous bounds of $4k - 2$ and $e(k + 1)$. Other results include improved approximation ratios for stochastic k-set packing and submodular stochastic probing over arbitrary non-negative submodular objective function, whereas previous results required the objective to be monotone.

I. Introduction

Uncertainty in input data is a common feature of most practical problems and research in finding good solutions (both experimental and theoretical) for such problems has a long history. In recent years one technique in particular has proven to be very effective in tackling such problems, namely the Contention Resolution Schemes (CR schemes). They have been introduced by Chekuri et al. [18] in order to maximize submodular functions under various constraints. Submodular maximization arises in modeling various optimization problems that share the property of diminishing returns.

This framework has been initially designed for problems in deterministic setup, where all information is known at the beginning. However, its randomized approach has turned out to be perfect for handling problems where the uncertainty was part of the model, like stochastic probing and mechanism design [11].

This fact was elegantly leveraged by Feldman et al. [10], who adapted the framework of CR schemes to an online setting, and resolved a long-standing open question by Chawla et al. [7], by devising so called Oblivious Posted Price Mechanism for matroids. This implied a constant factor approximation for the Bayesian multi-parameter unit-demand mechanism design problem.

Inspired by this line of work we have asked ourselves a question:

> *What can contention resolution schemes do, if we shall consider them in the random order model?*

While trying to answer this question we drew from, extended, bridged some gaps between, and improved some of the results on CR schemes [18, 10], sequential posted price, multi-parameter mechanism design [7, 11, 12, 10], and stochastic probing [11, 1]. We describe these results precisely below.

A. Problems overview, known results, and our contributions

Contention resolution schemes: Let us start with an illustrative problem. Consider a matroid $\mathcal{M} = (E, \mathcal{I})$ and a fractional solution x from its polytope. Suppose we are given a weight vector $w : E \mapsto \mathbb{R}_+$, and we look for an algorithm that returns an independent set $S \in \mathcal{I}$ such that $\sum_{e \in S} w_e \geq c \cdot \sum_{e \in E} w_e x_e$ for some constant $c < 1$. In order to do so, the idea is to settle for a randomized algorithm and demand that every element is taken into S with probability at least $c \cdot x_e$. Such a property would immediately entail the desired guarantee.

How to design an algorithm returning S such that $\mathbb{P}[e \in S] \geq c \cdot x_e$? Chekuri et al. [18] presented a framework of contention resolution schemes (CR schemes) which addressed this problem, among other applications. The idea is to first draw a random set $R(x)$ such that $\mathbb{P}[e \in R(x)] = x_e$ for each $e \in E$ independently, and afterwards – since $R(x)$ is most likely not an independent set in $\mathcal{I}$ – to drop some elements from $R(x)$ to meet the feasibility constraint, that is, to resolve the contention between the elements.

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00080

Our contribution: Simply speaking, in this paper we show how to solve this problem when we work in a random order model, i.e., when elements of E appear according to a uniformly chosen random permutation, and upon arrival we need to make an irrevocable decision of whether to take an element or not. Since the algorithms of Feldman et al. [10] are order oblivious, we already know from their work that CR schemes can be obtained in such a random order model. However, the value of the paper lays in the way we obtain these results. Our arguments are specifically tailored for the random order setting, and they exhibit features other CR schemes do not possess.

One important feature is that – for a moment we assume that the reader is familiar with the previous works – we do not require the schemes to be monotone, unlike the known CR schemes. Monotonicity appeared to be an important feature in the previous works because it allowed to combine the schemes via the FKG inequality [2]. Due to the lack of monotonicity, we had to rebuild the whole framework accordingly. Still, we managed to obtain composable schemes for matroids, sparse column packings, and knapsacks, that work under both linear and non-negative submodular objectives. We believe that dropping the monotonicity assumption is an interesting fact on its own.

Our range of applicability matches the range of online contention resolution schemes of Feldman et al. [10], and due to a less restrictive model we work in we obtain better bounds. On the other hand, we were not able to match all the constraints and bounds of Chekuri et al. [18], since they work in an unrestricted setting. For optimization over a single matroid and a linear objective, Chekuri et al. obtained an approximation factor (the constant c) of $1 - \frac{1}{e}$, while we get factor $\frac{1}{2}$[1]. However, for an intersection of k matroids, starting with $k \geq 2$, we obtain a better bound of $\frac{1}{k+1}$, improving upon theirs $\frac{1}{e \cdot k + o(k)}$, even though we work in a more restrictive model[2]. A possible explanation for this is the fact that we combine the schemes in a new way, which does not require monotonicity.

Theorem I.1. *There exists a random-order CR scheme for intersection of k matroids with $c = \frac{1}{k+1}$.*

For submodular objective we also improve the bounds starting with $k \geq 2$.

Theorem I.2. *Maximization of a non-negative submodular function with respect to k matroid constraints admits a $(k+1+\varepsilon) \cdot e$ approximation algorithm in the random-order model.*

[1]Factor $1 - \frac{1}{e}$ is actually also feasible in the random order model as recently shown by Lee and Singla [14].

[2]Factor $e \cdot k$ has been asymptotically improved to $\frac{e}{2} \cdot k$ [8].

These results are not absolutely best when compared to more general techniques, since one can get ratio $(k-1)$ for linear objectives when $k \geq 2$ using iterative rounding [13], and $(k+1+o(1))$ for non-negative submodular functions via a combinatorial argument [15]. However, to the best of our knowledge, our result yields the best ratio in the random order model.

Mechanism Design: Consider the following mechanism design problem. There are n agents and a single seller providing a set of services. The agent i is interested in buying the i-th service and values its as v_i, which is drawn independently from a distribution D_i. Such a setting is called single-parameter. The valuation v_i is private, but the distribution D_i is known in advance. The seller can provide only a subset of services, that belongs to a system $\mathcal{I} \in 2^{[n]}$, which is specified by feasibility constraints. A mechanism accepts bids of agents, decides on subset of agents to serve, and sets individual prices for the service. A mechanism is called truthful if agents are motivated to bid their true valuations. Myerson's theory of virtual valuations yields truthful mechanisms that maximize the expected revenue of a seller [16], although they sometimes might be impractical [3]. On the other hand, practical mechanisms are often non-truthful [3]. The Sequential Posted Pricing Mechanism (SPM) introduced by Chawla et al. [7] gives a nice trade-off – it is truthful, simple to implement, and gives near-optimal revenue. An SPM offers each agent a 'take-it-or-leave-it' price for a service. After refusal the service shall not be provided, so it is easy to see that an SPM is indeed a truthful mechanism.

The paragraph above concerns only the single-parameter setup. In the Bayesian multi-parameter unit-demand mechanism design (BMUMD for short), we have n buyers and one seller. The seller offers a number of different services indexed by set $\mathcal{J}$. The set $\mathcal{J}$ is partitioned into groups $\mathcal{J}_i$, with the services in $\mathcal{J}_i$ being targeted by agent i. Each agent i is interested in getting any one of the services in $\mathcal{J}_i$, i.e., agents are unit-demand. Agent i has value v_j for service $j \in \mathcal{J}_i$. Value v_j is independent of all other values and is drawn from distribution D_j. Once again the seller faces a feasibility constraint specified by a set system $\mathcal{I} \subseteq 2^J$.

Unlike single-parameter setup, this problem is not solvable efficiently by the well-established Myerson's approach. The paper of Chawla et al. [7] launched a line of work in obtaining approximate results for the multi-parameter setup, by suggesting a possible avenue of a solution via the so-called Oblivious Posted Price mechanisms. One would have to first embed the multi-parameter problem into a single-parameter one, and later to ensure that the algorithm would work if the items are presented in an adversary order. Kleinberg and Weinberg [12] solved the BMUMD problem for matroid environments with approximation of $4k-2$ for intersection of k matroids (with 2-approximation for a single matroid), but they have not used the Oblivious Posted Price mechanisms. Feldman et al. [10] devised the first Oblivious Posted Price mechanisms and obtained an $ek+o(k)$ approximation for the intersection of k matroids.

Our contribution: We observe that the Oblivious Posted Price is an overly demanding notion, and we need to handle the oblivious order only when looking at the items of a given client, but there is no need to restrict the order of clients. In our algorithm we randomly shuffle clients, but cannot make assumption on the client's choice. This hybrid approach is what allows us to obtain improved bounds. For $k=2$ we match up to ε the 6-approximation of Kleinberg and Weinberg [12], but starting from $k \geq 3$ our ratios are better; for $k=3$ we get $7+\varepsilon$ improving over 9.48 of Feldman et al. [10].

Theorem I.3. *Bayesian multi-parameter unit-demand mechanism design over k matroid constraints admits a $(k+4+\varepsilon)$ approximation for any $\varepsilon > 0$.*

Non-negative submodular stochastic probing: We are given a universe E, where each element $e \in E$ is *active* with probability p_e independently. The only way to find out if an element is active is to *probe* it. We call a probe *successful* if an element turns out to be active. We execute an algorithm that probes the elements one-by-one. If an element is active, the algorithm is forced to add it to the current solution. In this way, the algorithm gradually constructs a solution consisting of active elements.

We consider the case in which we are given constraints on both the set of probed elements and the set of elements included in the solution. Formally, we are given two downward-closed independence systems: an *outer* system $(E, \mathcal{I}^{out})$ restricting the set of elements probed by the algorithm, and an *inner* system $\left(E, \mathcal{I}^{in}\right)$, restricting the set of elements taken by the algorithm. The goal is to maximize the expected value $\mathbb{E}\left[f\left(S\right)\right]$, where f is a given non-negative submodular function and S is the set of all successfully probed elements.

This problem has been stated by Gupta and Nagarajan [11] who gave an abstraction for couple of problems like stochastic matching and sequential-posted price mechanisms in a single-parameter setup. They obtained an $O(k_{in}+k_{out})$ approximation for linear objectives in an environment with k_{in} inner matroids and k_{out} outer matroids (together with results for more general constraints) using the CR-schemes of Chekuri et al. [18]. Later, Adamczyk et al. [1] showed how to obtain a $(k_{in}+k_{out})$-approximation for linear objectives and $\frac{e}{e-1} \cdot (k_{in}+k_{out}+1)$ for monotone submodular objectives.

Our contribution: We obtain the first results with respect to arbitrary non-negative submodular objective functions.

Theorem I.4. *Non-negative submodular stochastic probing with k_{in} inner matroid constraints and k_{out} outer matroid constraints admits a $(k_{in}+k_{out}+1+\varepsilon)\cdot e$ approximation for any $\varepsilon > 0$.*

Stochastic k-set packing: We are given n elements/columns, where each element $e \in [n]$ has a random profit $v_e \in \mathbb{R}_{+}$, and a random d-dimensional size $L_e \in \{0,1\}^d$. The sizes are independent for different elements, but v_e can be correlated with L_e, and the coordinates of L_e also might be correlated between each other. The values of v_e and L_e are revealed after e is probed, but their distributions are known in advance.

Additionally, for each element e we are given a set $Q_e \subseteq [d]$ of size at most k, such that the size vector L_e takes positive values only in these coordinates, i.e., $\{i \in \{1,...,d\} \mid L_e\left(i\right)=1\} \subseteq Q_e$ with probability 1. We are also given a capacity vector $b \in \mathbb{Z}_{+}^{d}$ into which elements must be packed, that is, the solution can consist of at most b_i elements with unit sizes in the i-th row. We say that the outcomes of L_e are monotone if for any possible realizations $x, y \in \{0,1\}^d$ of L_e, we have $x \leq y$ or $y \leq x$ coordinate-wise.

A strategy probes columns one by one, obeying the packing constraints, and the goal is to maximize the expected outcome of taken columns. The stochastic k-set packing problem was stated by Bansal et al. [4]. They have presented a $2k$-approximation algorithm for it, and a $(k+1)$-approximation algorithm with an assumption that the outcomes of size vectors L_e are monotone. Recently Brubach et al. [6] improved the approximation ratio to $k+o(k)$ in the general case.

Our contribution: We improve upon the recent bound of Brubach et al. [6]. Our algorithm also works in the case where we replace counting constraints on rows with arbitrary matroids.

Theorem I.5. *There exists a $(k+1)$ approximation algorithm for stochastic k-set packing over matroid row constraints.*

B. Our techniques

The main notion we use is a *controller mechanism*, which provides a handy abstraction, that allows us to combine various constraints without relying on the monotonicity of the schemes. For matroids it is implemented using a decomposition of a fractional solution into a convex combination of characteristic vectors of independent sets, and for knapsacks a controller is represented as a point from the unit interval. Additionally, knapsack constraints require a preprocessing procedure, that partitions the elements into big and small, which is inspired by [5, 10].

The controller mechanism of a constraint $\mathcal{I}$ randomly assigns each element $e \in E$ a controller C_e, which keeps track of its suitability to become a part of the solution when we iterate through the elements in a random order. More formally,

a) if S is the current solution and C_e has not been *blocked* yet, then $S \cup \{e\}$ must belong to $\mathcal{I}$,

b) for each element e the probability that 1) some element f has been chosen at step t, and 2) f has been assigned a controller C_f, that blocks C_e, is at most $\frac{\lambda}{n-t}$ (probability taken over all such f's and C_f's), for a constant λ depending on $\mathcal{I}$.

With these properties on hand, we can associate a submartingale with each element e and a fixed controller C_e. We define a stopping event of revealing the fate of e, i.e., we stop when we either take e into the solution or we block its controller. Before the stopping event for e occurs, we know that we still can either take it or block it. The bound on the probability of accepting the element comes then from the Doob's stopping theorem. This suffices to construct a random-order contention resolution scheme. Another martingale argument extends this reasoning to the submodular function maximization.

In the context of the stochastic probing problems, we are aware of only one usage of the martingale argument with the Doob's theorem, in the analysis of an iterative randomized rounding algorithm [1]. To the best of our knowledge, we present the first application of the martingale argument to analyze a random permutation, and we believe this technique can be handy and worth adding to a toolbox.

In order to handle Bayesian multi-parameter unit-demand mechanism design, we rely on the reduction to a single-parameter setup by Chawla et al. [7] via copies, and on the linear relaxation by Gupta and Nagarajan [11]. The last ingredient necessary to obtain the postulated approximation ratio for k matroids is a routine that processes a fractional solution for a single client menu, which later on allows us to give very tight upper and lower bounds on the probabilities of an item's acceptance and rejection. We present such a routine based on local search that reduces the discrepancy between these quantities in each step.

Arguments for stochastic probing and stochastic k-set packing exploit the same notion of the controller mechanism. However, in order to obtain an upper bound for a submodular objective case we need a stronger guarantee for the measured continuous greedy algorithm for optimizing the multilinear extension of a submodular function [9]. This bound is due to Justin Ward [19].

C. Organization of the paper

We start the technical part of the paper by showing a random-order CR scheme for a matroid in Section II. Section III contains the analysis of the CR scheme and introduces the language of our framework, that is, the controller mechanism and characteristic sequences. This allows us to present the extension to multiple matroids in a simple way, and later to explain how to deal with submodular functions.

In Section IV we present the more complicated algorithm for the Bayesian multi-parameter unit-demand mechanism design. This order of presentation allows us to explain both the framework and the main result within the extended abstract. The following Sections V and VI cover the submodular optimization and stochastic k-set packing. The chapter about stochastic probing can be found in the full version of the paper.

We deliberately avoid giving one procedure that captures all the results at once for the cleanest possible presentation of the paper. With each result comes an abstract formulation of the algorithm and the application in the matroid environment. Our framework also extends to knapsack constraints and we refer the reader to the full version of the paper for details.

II. Random-order contention resolution scheme for a matroid

We formulate our first goal as a motivation to present the simplest variant of the mechanism.

Theorem II.1. *There exists a random-order CR scheme for a matroid with $c = \frac{1}{2}$.*

Structural properties of matroids: To devise a CR scheme for a matroid constraint we shall need the following two properties. The proof of the lemma below about the existence of a convex decomposition can be found in [17].

Lemma II.2. *We can represent any $x \in \mathcal{P}(\mathcal{M})$ as $x = \sum_{i=1}^{m} \beta_i \cdot \mathbf{1}_{B_i}$, where $B_1, \ldots, B_m \in \mathcal{M}$ and $\beta_1, \ldots, \beta_m$ are non-negative weights such that $\sum_{i=1}^{m} \beta_i = 1$ and $m = |E|^{O(1)}$. We denote $\mathcal{S} = [m]$ and call $(\mathcal{S}, (B_i)_{i \in \mathcal{S}}, (\beta_i)_{i \in \mathcal{S}})$ a* support *of x in $\mathcal{P}(\mathcal{M})$.*

The following lemma is a slightly generalized basis exchange lemma, proof of which again can be found in [17].

Lemma II.3. *Let $A, B \in \mathcal{I}$ be two independent sets of matroid $\mathcal{M} = (E, \mathcal{I})$. We can find an exchange-mapping $\phi[A, B] : A \mapsto B \cup \{\perp\}$ such that:*

1) *$\phi[A, B](e) = e$ for every $e \in A \cap B$,*
2) *for each $f \in B$ there exists at most one $e \in A$ for which $\phi[A, B](e) = f$,*
3) *for $e \in A \setminus B$, if $\phi[A, B](e) = \perp$, then $B + e \in \mathcal{I}$, otherwise $B - \phi[A, B](e) + e \in \mathcal{I}$.*

Initialization: The procedure is shown in Algorithm 1. Given a vector $x \in \mathcal{P}(\mathcal{M})$, we begin with decomposing it into a *support* $x = \sum_{i\in\mathcal{S}} \beta_i \cdot \mathbf{1}_{B_i^0}$, where each set B_i^0 is independent in $\mathcal{M}$ (Lemma II.2), and exchange-mappings $\phi\left[B_i^0, B_j^0\right]$ between each pair of sets in the support (Lemma II.3). For each element $e \in E$ we choose as controller $j(e) \in \mathcal{S}$ a random set of $\mathcal{S}$ that includes e, where the probability of every such set B_i to become the controller is β_i/x_e. The set family given by the support is being modified after each step of the algorithm and we denote the sets in step t as $(B_j^t)_{j\in\mathcal{S}}$. The set $\mathcal{S}$ and scalars β_j remain the same. For sake of legibility we refer directly to set $B_{j(e)}^t$ as C_e^t and shorten it to C_e when it does not lead to a confusion.

Blocking events: We scan elements from E in a random order. If the element e chosen in step t happens to belong to $R(x)$ and its controller has not been blocked yet (to be explained shortly), we take it into the solution. Then we modify the set family $(B_j^t)_{j\in\mathcal{S}}$ by inserting e to each of them. This operation is performed according to the exchange-mappings. It may result in some other element f being removed from the set $C_f^t = B_{j(f)}^t$. When this happens, we say that (f, C_f) gets **blocked**.

Let us note that at the moment of doing so, in some circumstances, it could still be possible to take element f into the solution. However, we require a clean condition to know when an element is not considered any longer. This simplifies the analysis significantly. In the pseudocode shown below, we check for the blocking event of e in line 8.

Correctness: Let S^t stand for the solution constructed up to step t. We need to show that the output is indeed an independent set of the matroid. This follows from the two facts below.

Fact II.4. *For every t and $i \in \mathcal{S}$ it holds $S^t \subseteq B_i^t$.*

Proof: If we add an element e to S^t on line 9, then we add e to each B_i^t. ∎

Fact II.5. *For every t and $i \in \mathcal{S}$ the set B_i^t is independent in the matroid $\mathcal{M}$.*

Proof: All changes of the sets B_i^t are due do the exchange-mapping ϕ whose property (3) ensures that after each exchange sets B_i^t remain independent in $\mathcal{M}$. See Lemma II.3 for details. ∎

Approximation guarantee: In our setting we cannot assume we know the whole set $R(x)$ in advance, but rather we learn if $e \in R(x)$ after probing e in line 6. In the following arguments we fix an element e and condition all the probabilities on the fact that $e \in R(x)$, and on the controller C_e chosen in line 3. Since the choice of other controllers is irrelevant to e until an element f with a controller blocking C_e is revealed to exist in line 6, we can assume in the analysis that the assignment of C_f happens after the latest family of exchange-mappings has been established.

Algorithm 1 Random-order contention resolution scheme for a matroid

1: decompose x into its support in $\mathcal{M}$, that is, $x = \sum_{i\in\mathcal{S}} \beta_i \cdot \mathbf{1}_{B_i^0}$
2: find exchange-mappings $\phi\left[B_i^0, B_j^0\right]$ between all pairs $i, j \in \mathcal{S}$
3: for each element e choose a controller $j(e) \in \mathcal{S}$ such that $e \in B_j^0$ (each index j gets chosen with probability $\frac{\beta_j}{x_e}$); denote $B_{j(e)}^t$ by C_e^t
4: $S \leftarrow \emptyset$, $t \leftarrow 0$
5: **for** each element e in E in σ order **do**
6:   **if** $e \notin R(x)$ **then**
7:     continue
8:   **if** $e \in C_e^t$ **then**
9:     $S \leftarrow S \cup \{e\}$
10:     **for** each $i \in \mathcal{S} : e \notin B_i^t$ **do**
11:       **if** $\phi\left[C_e^t, B_i^t\right](e) = \perp$ **then** $B_i^{t+1} \leftarrow B_i^t + e$
12:       **if** $\phi\left[C_e^t, B_i^t\right](e) = f$ **then** $B_i^{t+1} \leftarrow B_i^t - f + e$
13:     **for** each $i \in \mathcal{S} : e \in B_i^t$ **do**
14:       $B_i^{t+1} \leftarrow B_i^t$
15:     find new exchange-mappings $\phi\left[B_i^{t+1}, B_j^{t+1}\right]$ between all pairs $i, j \in \mathcal{S}$
16:   $t \leftarrow t + 1$;
17: **return** S

The next two lemmas encapsulate the properties of the controller mechanism for a matroid. The main proof is postponed to Lemma III.5.

Lemma II.6. *Suppose that $\sum_{j\in\mathcal{S}} \beta_j \leq 1$ and $(B_j)_{j\in\mathcal{S}}$ is a family of independent sets from $\mathcal{M}$ with fixed exchange-mappings between each B_j and set $C \in \mathcal{M}$. Let us denote by $\Gamma(e, C) = \left\{(f, j) \;\middle|\; \phi_{[B_j, C]}(f) = e\right\}$ the set of all pairs (f, j) that makes e get removed from C. Then*

$$\sum_{f\in E} \sum_{j:(f,j)\in\Gamma(e,C)} \beta_j \leq 1.$$

Proof: For every set B_j there can be at most one element f such that $(f, j) \in \Gamma(e, C)$ because $\phi[B_j, C]$ cannot map two elements onto e (Lemma II.3). Therefore for a fixed $j \in \mathcal{S}$ we have $\sum_{f:(f,j)\in\Gamma_i(e,C)} \beta_j \leq \beta_j$. We change the summation order to obtain

$$\sum_{f\in E} \sum_{j:(f,j)\in\Gamma(e),C} \beta_j = \sum_{j\in\mathcal{S}} \sum_{f:(f,j)\in\Gamma(e,C)} \beta_j \leq \sum_{j\in\mathcal{S}} \beta_j \leq 1.$$

∎

Lemma II.7. *The probability of a blocking event for (e, C_e) in step t is at most $\frac{1}{n-t}$.*

Proof: We enumerate steps starting with 0. A blocking event occurs when we remove e from C_e^t. This happens if we choose $f \neq e$ in step t, that 1) turns out to belong to $R(x)$ in line 6, and 2) we choose a controller C_f such that $\phi_{C_f^t, C_e^t}(f) = e$ in line 3 (recall that in our analysis we can treat this event as happening after the existence f has been revealed). Let $\Gamma^t(e, C_e)$ be as in Lemma II.6 with respect to the set family $(B_j^t)_{j \in \mathcal{S}}$. Since there are $n - t$ elements to choose in step t, the probability that e gets removed from C_e^t is at most

$$\frac{1}{n-t} \sum_{f} \sum_{j:(f,j) \in \Gamma^t(e, C_e)} \mathbb{P}[f \in R(x)] \cdot \mathbb{P}[f \text{ chooses } j].$$

Element f belongs to $R(x)$ with probability $\mathbb{P}[f \in R(x)] = x_f$. Also, f is assigned controller j, i.e., $C_f^t = B_j^t$, with probability $\frac{\beta_j}{x_f}$. Therefore the above expression simplifies to

$$\frac{1}{n-t} \sum_{(f,j) \in \Gamma^t(e, C_e)} x_f \cdot \frac{\beta_j}{x_f} = \frac{1}{n-t} \sum_{(f,j) \in \Gamma^t(e, C_e)} \beta_j.$$

The claim now follows from Lemma II.6. ■

III. Controller mechanism

Before we are ready to finish the proof of Theorem II.1, we need to introduce our toolbox. In this section we abstract from the structure of the constraint and present the general framework for obtaining approximation ratios with the controller mechanism.

Algorithm 2 Abstract view of the random-order contention resolution scheme

1: assign each element $e \in E$ a controller C_e
2: $S \leftarrow \emptyset$
3: **foreach** element e in E in σ order **do**
4:     **if** $e \notin R(x)$ **then**
5:         continue
6:     **if** (e, C_e) has not been blocked **then**
7:         $S \leftarrow S \cup \{e\}$
8:         update controllers
9: **return** S

A. Characteristic sequences

In order to analyze the approximation guarantee we fix an element e and condition all the probabilities on the fact that $e \in R(x)$, and on the choice of controller C_e (using notation $\mathbb{P}[\text{event} \mid C_e]$). Since e is oblivious to the choice of other controllers until an element f with a controller blocking C_e is taken into the solution. We assume in the analysis that for $f \neq e$ the assignment of C_f happens after the last controller update and the disclosure of f.

Initially we know that e is *available* to take, i.e., there is still a possibility of accepting e via C_e^t for some t. As the process is being executed, at some point we get to know the fate of e: there comes a step in which we either 1) pick e in line 3, or 2) pick $f \neq e$ and choose a controller C_f which blocks (e, C_e) (in the matroid example: e gets removed from C_e).

Definition III.1 (Characteristic sequences). Consider an abstract routine, where in every turn each unseen element might be picked with equal probability and, if its controller has not been blocked, it gets accepted and might block other controllers. We shall associate three binary processes with (e, C_e):

- S_e^t : indicates whether e was taken into the solution before step t; initially $S_e^0 = 0$;
- Z_e^t : indicates whether the controller of e has been blocked before step t; initially $Z_e^0 = 0$;
- Y_e^t : we still **didn't get** to know the fate of e before step t; initially $Y_e^0 = 1$.

The sequences are bound with a following relationship

$$Y_e^t = 1 - S_e^t - Z_e^t.$$

We call the characteristic sequences λ-**bounded** if

$$\mathbb{E}\left[Z_e^{t+1} - Z_e^t \,\middle|\, \mathcal{F}^t\right] \leq \frac{\lambda \cdot Y_e^t}{n-t}.$$

Corollary III.2. *For the matroid constraint the characteristic sequences are 1-bounded.*

Proof: First let us note that if $Y_e^t = 0$, then we already got to know the fate of e before step t, and so the status of blocking e cannot change, i.e., $Z_e^t = Z_e^{t+1}$. If $Y_e^t = 1$, then the claim reduces to Lemma II.7. ■

Lemma III.3. *If the characteristic sequences of e are λ-bounded, then they satisfy*

$$\mathbb{E}\left[Z_e^{t+1} - Z_e^t \,\middle|\, \mathcal{F}^t\right] \leq \lambda \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t \,\middle|\, \mathcal{F}^t\right].$$

Proof: Consider step $t+1$ of the process. We claim the following relationship

$$\mathbb{E}\left[S_e^{t+1} - S_e^t \,\middle|\, \mathcal{F}^t\right] = \frac{Y_e^t}{n-t}.$$

We check this relation by a case-work. If $Y_{e,i}^t = 0$, then we already know the fate of e. In this case we either have $S_{e,i}^t = S_{e,i}^{t+1} = 0$ if e has been blocked, or we have $S_{e,i}^t = S_{e,i}^{t+1} = 1$, if we have taken e before step t. In both cases left-hand side and right-hand side are equal 0. Now if $Y_e^t = 1$, then we know that 1) we have not chosen e in line 3 before, and 2) (e, C_e) has not been blocked. Then we can pick e in step t with probability $\frac{1}{n-t}$, what means exactly that $S_e^t = 0$ but $S_e^{t+1} = 1$. The claim follows. ■

Lemma III.4. *Suppose characteristic sequences of e are λ-bounded. Then, the process $((1+\lambda) \cdot S_e^t + Y_e^t)_{t=0}^n$ is a submartingale.*

Proof: Recall that $Y_e^t = 1 - S_e^t - Z_e^t$. From Lemma III.3 we have

$$\begin{aligned} & \mathbb{E}\left[Y_e^t - Y_e^{t+1} \,\middle|\, \mathcal{F}^t\right] \\ = \; & \mathbb{E}\left[S_e^{t+1} + Z_e^{t+1} - S_e^t - Z_e^t \,\middle|\, \mathcal{F}^t\right] = \\ = \; & \mathbb{E}\left[S_e^{t+1} - S_e^t + Z_e^{t+1} - Z_e^t \,\middle|\, \mathcal{F}^t\right] \le \\ \le \; & (1+\lambda) \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t \,\middle|\, \mathcal{F}^t\right], \end{aligned}$$

$$\begin{aligned} & \mathbb{E}\left[\left((1+\lambda) \cdot S_e^{t+1} + Y_e^{t+1}\right) - \left((1+\lambda) \cdot S_e^t + Y_e^t\right) \,\middle|\, \mathcal{F}^t\right] \\ & = \mathbb{E}\left[(1+\lambda) \cdot \left(S_e^{t+1} - S_e^t\right) - \left(Y_e^t - Y_e^{t+1}\right) \,\middle|\, \mathcal{F}^t\right] \ge 0, \end{aligned}$$

which means that the process $((1+\lambda) \cdot S_e^t + Y_e^t)_{t=0}^n$ is indeed a submartingale. ■

Lemma III.5. *Suppose a random-order CR scheme yields a controller mechanism with λ-bounded characteristic sequences. Then the probability that e does not get blocked before it is picked is at least $\frac{1}{1+\lambda}$.*

Proof: Lemma III.4 guarantees that process $((1+\lambda) \cdot S_e^t + Y_e^t)_{t=0}^n$ is a submartingale. Let $\tau = \min\{t \mid Y_e^t = 0\}$ denote the first moment when we get to know what happens with e. Since τ is a bounded (always $\tau \le n$) stopping time, we can take advantage of the Doob's stopping theorem to get

$$\mathbb{E}\left[(1+\lambda) \cdot S_e^0 + Y_e^0\right] \le \mathbb{E}\left[(1+\lambda) \cdot S_e^\tau + Y_e^\tau\right].$$

Since $S_e^0 = 0 = Y_e^\tau$ and $Y_e^0 = 1$, we have

$$\begin{aligned} 1 = \mathbb{E}\left[(1+\lambda) \cdot S_e^0 + Y_e^0\right] \le \\ \mathbb{E}\left[(1+\lambda) \cdot S_e^\tau + Y_e^\tau\right] = (1+\lambda) \cdot \mathbb{E}\left[S_e^\tau\right], \end{aligned}$$

and so $\mathbb{E}[S_e^\tau] \ge \frac{1}{1+\lambda}$. Now one just has to note that $\mathbb{P}[e \text{ is available to take when picked} \mid C_e]$ is exactly equal to $\mathbb{E}[S_e^\tau]$ (conditioning on C_e comes from the fact that the derivation holds for this particular controller). Since this holds for any choice of the controller C_e, we get the same bound unconditionally. ■

Thus, the probability that element e will be taken into the solution under condition $e \in R(x)$ is at least $\frac{1}{1+\lambda}$. By combining Corollary III.2 and Lemma III.5 we finish the proof of Theorem II.1.

B. Combining constraints

Suppose now that we are given k constraints $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$. The combination of the mechanisms is simple. We assign each element k controllers independently with respect to each constraint. We scan elements in a random order and when an element gets accepted we independently update each controller mechanism. An element gets blocked if it is blocked in at least one constraint.

The correctness of the mechanism, i.e., the fact that we return a set that is independent in all constraints, is clear. We need to argue for the approximation ratio to be proper. Let us refer to the characteristic sequences of the i-th constraint as $({}^iS_e^t)$, $({}^iZ_e^t)$, $({}^iY_e^t)$. In order to construct the characteristic sequences describing the joint mechanism, observe that an element gets blocked if at least one of its controllers gets blocked, it gets accepted if it is accepted in all constraints, and we get to know its fate if it is revealed in at least one constraint. Recall that $Y_e^t = 1$ stands for fate of e **not being** revealed before step $t+1$. This can be summarized as

$$\begin{aligned} Z_e^t &= \max\left({}^1Z_e^t, {}^2Z_e^t, \ldots, {}^kZ_e^t\right), \\ S_e^t &= \min\left({}^1S_e^t, {}^2S_e^t, \ldots, {}^kS_e^t\right), \\ Y_e^t &= \min\left({}^1Y_e^t, {}^2Y_e^t, \ldots, {}^kY_e^t\right). \end{aligned}$$

We call these the *joint* characteristic sequences of e. The relationship between Z_e^t, S_e^t and Y_e^t becomes again $Y_e^t = 1 - S_e^t - Z_e^t$.

Lemma III.6. *Suppose the characteristic sequences for the i-th constraint are λ_i-bounded. Then the joint characteristic sequences are $(\sum_i \lambda_i)$-bounded.*

Proof: If $Y_e^t = 0$, then for some i we have ${}^iY_e^t = 0$, i.e., the fate of e has been revealed in the the i-th constraint. There are two cases: either ${}^iZ_e^t = 1$ or ${}^iS_e^t = 1$. In the first case we have $Z_e^t = Z_e^{t+1} = 1$. If ${}^iS_e^t = 1$, then the element e has been picked before step t and either it got accepted in all constraints or it had been blocked before in some other constraint. In both cases be have $Z_e^t = Z_e^{t+1}$.

If $Y_e^t = 1$, then it holds that ${}^iY_e^t = 1$ for all i. We estimate the probability of any event $\left(Z_e^{t+1} > Z_e^t\right)$ by the union bound, obtaining

$$\begin{aligned} & \mathbb{E}\left[Z_e^{t+1} - Z_e^t \,\middle|\, \mathcal{F}^t\right] \\ = \; & \mathbb{E}\left[\max_{i=1\ldots k}\left({}^iZ_e^{t+1}\right) - \max_{i=1\ldots k}\left({}^iZ_e^t\right) \,\middle|\, \mathcal{F}^t\right] \\ \le \; & \mathbb{E}\left[\max_{i=1\ldots k}\left({}^iZ_e^{t+1} - {}^iZ_e^t\right) \,\middle|\, \mathcal{F}^t\right] \\ \le \; & \mathbb{E}\left[\sum_{i=1\ldots k} {}^iZ_e^{t+1} - {}^iZ_e^t \,\middle|\, \mathcal{F}^t\right] \\ = \; & \sum_{i=1\ldots k} \mathbb{E}\left[{}^iZ_e^{t+1} - {}^iZ_e^t \,\middle|\, \mathcal{F}^t\right] \le \\ \le \; & \sum_{i=1\ldots k} \frac{\lambda_i \cdot {}^i Y_e^t}{n-t} = \frac{\sum_i \lambda_i \cdot Y_e^t}{n-t}. \end{aligned}$$

■

Theorem I.1. *There exists a random-order CR scheme for intersection of k matroids with $c = \frac{1}{k+1}$.*

Proof: The claims follows from Corollary III.2 and Lemmas III.5 and III.6. ■

IV. Multi-parameter mechanism design

Recall that each client $i \in \mathcal{I}$ is interested in purchasing one service from $\mathcal{J}_i$ and their valuation of an item $c \in \mathcal{J}_i$ is modeled by a random variable v_c, independent of other valuations, with a known distribution D_c. Following [11] we assume that the distribution D_c is always discrete and takes values over $\mathcal{B} = \{0, 1, \ldots, B\}$.

A. Bounding by auction with copies

Imagine a setting where for each item $c \in \mathcal{J}_i$ we create an independent copy-client c interested solely in this item. The new instance has the same constraint system as the original one plus additional partition matroid. We rely on the crucial lemma by Chawla et al. [7], saying that the optimal revenue in the new instance can be only greater because the competition increases.

This observation allows us to obtain an LP upper-bound for the true OPT. The linear program BMUMD-LP [11] models the auction with copy-clients, which is single-parameter. C denotes the set of copy-clients, which is equivalent to the set of items, and $\mathcal{P}$ is the polytope of the constraint system.

$$\begin{aligned} \max \quad & \textstyle\sum_{c \in C} \sum_p x_{c,p} \cdot p \cdot \mathbb{P}\left[v_c \geq p\right] \qquad (\text{BMUMD-LP}) \\ \text{s.t.} \quad & \left(\textstyle\sum_p x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right]\right)_{c \in C} \in \mathcal{P} \\ & \textstyle\sum_p x_{c,p} \leq 1 \qquad \forall c \in C \\ & \textstyle\sum_{c \in \mathcal{J}_i} \sum_p x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right] \leq 1 \quad \forall i \in \mathcal{I}. \end{aligned}$$

Lemma IV.1 ([7, 11])**.** *The optimal value of* BMUMD-LP *is an upper bound for the maximal revenue in the multi-parameter auction.*

B. Single client routine

The algorithm scans clients in random order, and presents a price menu to each client, from which the client picks one item which gives him the highest utility, or resigns from choosing if all utilities are negative. Such a procedure clearly yields a truthful mechanism. Let $x_{c,p}$ be the probability that we place item $c \in \mathcal{J}_i$ with price p in the menu of client i. The vector $\mathbf{x} = (x_{c,p})$ — which we shall call a *menu-vector* — describing randomized menu for client i must satisfy following constraints:

$$\begin{aligned} \textstyle\sum_p x_{c,p} \leq 1 \qquad & \forall c \in \mathcal{J}_i \\ \textstyle\sum_{c \in \mathcal{J}_i} \sum_p x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right] \leq 1. & \end{aligned}$$

Given menu-vector $\mathbf{x}$, we construct the menu as follows. Independently for each item c we choose price p with probability $x_{c,p}$ and discard the item with probability $1 - \sum_p x^t_{c,p}$. Then the client reveals her utilities for each item. We define $\mathbf{X}_{c,p}$ to be the event of the item c with price p being at the top of the menu. To ensure that it is well-defined we need to fix a mechanism to break the ties between items of equal utility to the client, e.g., lexicographically or by random choice. However we do not need to know the mechanism explicitly for the analysis sake.

The following lemma describes how to construct a menu-vector with almost tight guarantees on probabilities of item acceptance and rejection. The proof, based on $O(1/\varepsilon^2)$ rounds of a local search procedure, can be found in the full version of the paper.

Lemma IV.2. *Suppose we can compute values $\mathbb{P}\left[\mathbf{X}_{c,p}\right]$ in a polynomial time for a known menu-vector. Then for any $\varepsilon > 0$ there is a polynomial-time procedure, called by us* SingleClientSubroutine*, that, given menu-vector $\mathbf{x}$, finds another menu-vector $\mathbf{y}$, such that for each c, p:*

$$\frac{1}{4} x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right] \leq \mathbb{P}\left[\mathbf{Y}_{c,p}\right] \leq \left(\frac{1}{4} + \varepsilon\right) \cdot x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right].$$

C. The algorithm

With the subroutine to handle a single client, we are ready to prove the main result of this paper.

Theorem I.3. *Bayesian multi-parameter unit-demand mechanism design over k matroid constraints admits a $(k + 4 + \varepsilon)$ approximation for any $\varepsilon > 0$.*

Algorithm 3 Auction mechanism

```
assign each item c a controller C_c
for each client i ∈ I in random order do
    perform SingleClientSubroutine on the non-blocked items in J_i
    offer the chosen item c to client i
    if client i accepts c then
        update controllers
```

We begin with a relaxation to BMUMD-LP and obtain vector $(x_{c,p})_{c,p}$ that supplies the auction mechanism, that is based on the random-order contention resolution scheme. The abstract view of the auction mechanism is presented in Algorithm 3.

Matroid implementation: The controller mechanism for matroids is analogous to the one from Theorem II.1. We decompose vector $\left(\sum_p x^p_c \cdot \mathbb{P}\left[v_c \geq p\right]\right)_{c \in \mathcal{C}}$ into a support in matroid $\mathcal{M}$, that is $\sum_{i \in \mathcal{S}} \beta_i \cdot B^0_i$ (Lemma II.2). We find exchange-mappings $\phi\left[B^0_i, B^0_j\right]$ between each pair $i, j \in \mathcal{S}$ as in Lemma II.3. Then for each element c we choose $j(c) \in \mathcal{S}$ such that $c \in B_j$, with probability $\frac{\beta_j}{\sum_p x^p_c \cdot \mathbb{P}[v_c \geq p]}$, call it the controller of c, and denote $C^t_c = B^t_{j(c)}$.

When an item $c \in J_i$ gets accepted by client i, we update the controllers, as presented in Algorithm 4.

A pair (c, C_c) gets blocked when c is removed from C_c.

The correctness follows again from the invariant, that the set of served items is a subset of B_j^t, which is an independent set, for all $j \in \mathcal{S}$.

Algorithm 4 Controller mechanism update for a matroid, restated

```
for each i ∈ S : c ∉ B_i^t do
    if φ[C_c^t, B_i^t](c) = ⊥ then B_i^{t+1} ← B_i^t + c
    if φ[C_c^t, B_i^t](c) = d then B_i^{t+1} ← B_i^t − d + c
for each i ∈ S : c ∈ B_i^t do
    B_i^{t+1} ← B_i^t
find new exchange-mappings φ[B_i^{t+1}, B_j^{t+1}] between each pair i, j ∈ S
```

Approximation guarantee: We are interested in estimating the probability that a fixed item $c \in \mathcal{J}_i$ will be served to a client i at price p . In this paragraph we condition all the events on the critical set C_c and we argue that it will not get blocked until the turn of client i with high probability. We retrace the reasoning from Section III-A and assign each pair (c, C_c) the characteristic sequences (S_c^t, Z_c^t, Y_c^t). This time $S_c^t = 1$ carries semantics of c having ended up in the menu of client i before step $t+1$.

Lemma IV.3. *The characteristic sequences of the auction mechanism for a matroid are $\left(\frac{1}{4}+\varepsilon\right)$-bounded.*

Proof: We proceed as in Lemma II.7. We need to show that probability that (c, C_c) gets blocked in turn t in a single matroid is at most $\left(\frac{1}{4}+\varepsilon\right) \cdot \frac{1}{n-t}$. A blocking event happens when an item d of a different agent j gets chosen, that makes c removed from its controller set. The agent j is chosen with probability $\frac{1}{n-t}$. Then the agent has to pick item d from the menu. The properties of the single-agent subroutine (Lemma IV.2) guarantees that this happens with probability at most $\left(\frac{1}{4}+\varepsilon\right) \sum_p x_d^p \cdot \mathbb{P}\left[v_d \geq p\right]$. Then d must be assigned a controller $j \in \mathcal{S}$ which makes c removed from C_c – this occurs with probability $\frac{\beta_j}{\sum_p x_d^p \cdot \mathbb{P}[v_d \geq p]}$. The total probability of any of these events is

$$\frac{1}{n-t} \sum_d \left(\frac{1}{4}+\varepsilon\right) \sum_p x_d^p \cdot \mathbb{P}\left[v_d \geq p\right] \cdot \sum_{j:\phi_{[B_j^t, C_c^t]}(d)=c} \frac{\beta_j}{\sum_p x_d^p \cdot \mathbb{P}\left[v_d \geq p\right]} =$$

$$= \left(\frac{1}{4}+\varepsilon\right) \frac{1}{n-t} \sum_d \sum_{j:\phi_{[B_j^t, C_c^t]}(d)=c} \beta_j \leq$$

$$\leq \left(\frac{1}{4}+\varepsilon\right) \frac{1}{n-t},$$

where the last inequality follows from Lemma II.6. ∎

Proof of Theorem I.3: The joint mechanism for the intersection of k matroids is given by assigning each item k controllers and blocking the item if at least one of its controllers has been blocked. After an item is picked, it gets placed in the menu as long as it has not been blocked before. This leads to a construction analogous to the one from Section III-B.

In Lemma IV.3 we have analyzed a blocking event in a single turn and for a single matroid. We use Lemma III.6 to extend this result to k matroids. Then we apply Lemma III.5 to get the global probability of not being blocked in any turn and in any matroid.

$$\mathbb{P}\left[c \text{ is not blocked until the turn of client } i \mid C_c\right] \geq \frac{1}{k \cdot \left(\frac{1}{4}+\varepsilon\right)+1}$$

Lemma IV.2 guarantees that once c gets into the menu of client i, it will be served at price p with probability at least $\frac{1}{4} \cdot x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right]$. By multiplying these quantities and setting the final value of ε to $\frac{\varepsilon}{4k}$ we obtain

$$\mathbb{P}\left[c \text{ gets served client } i \text{ at price } p \mid C_c\right] \geq \geq \frac{\frac{1}{4} \cdot x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right]}{k \cdot \left(\frac{1}{4}+\frac{\varepsilon}{4k}\right)+1} = \frac{x_{c,p} \cdot \mathbb{P}\left[v_c \geq p\right]}{k+4+\varepsilon}.$$

Since this holds for every choice of C_c we get the same bound unconditionally. It means that the expected revenue of the mechanism is at least $\frac{1}{k+4+\varepsilon}$ times the optimal value of the linear program BMUMD-LP, which completes the proof. ∎

V. SUBMODULAR OPTIMIZATION

As another elegant application of our toolbox, as well as a building block for submodular stochastic probing, we show a framework for non-negative submodular function maximization. We are given the following optimization task over (possibly) a sequence of constraints $(\mathcal{I}_i)_{i=1}^k$.

$$\begin{array}{ll} \max & f(X) \\ \text{s.t.} & X \in \mathcal{I}_1 \cap \mathcal{I}_2 \cap \cdots \cap \mathcal{I}_k \end{array}$$

We first need to solve the multilinear relaxation for this problem.

$$\begin{array}{ll} \max & F(x) \\ \text{s.t.} & x \in \mathcal{P}(\mathcal{I}_1) \cap \mathcal{P}(\mathcal{I}_2) \cap \cdots \cap \mathcal{P}(\mathcal{I}_k) \end{array}$$

We can approximately solve such a optimization problem with the measured continuous greedy algorithm [9], that provides a vector x such that $F(x) \geq \left(\frac{1}{e}-\varepsilon\right) \cdot F(x_{OPT}) \geq \left(\frac{1}{e}-\varepsilon\right) \cdot f(X_{OPT})$. We are going to sample elements with respect to x and execute the random-order contention resolution scheme on the sampled set with a following postprocessing: we add an accepted element e to X only if $f(X \cup \{e\}) > f(X)$. We claim that this procedure returns a solution X such that $\mathbb{E}[f(X)] \geq \frac{1}{\lambda+1} F(x)$.

Lemma V.1. *Consider a sampling scheme with λ-bounded characteristic sequences, in which the chosen element e materializes with probability x_e. Suppose we are given a non-negative submodular function f and we accept the chosen element if it materializes and taking it increases the value of f. Such a procedure generates a random set X such that $\mathbb{E}[f(X)] \geq \frac{1}{\lambda+1} F(x)$.*

Proof: This time we need to track globally the solution that we create, and not just a particular element. Let X^t be the solution created up to step t and $S^t = \{S_e^t = 1 \mid e \in E\} \cap R(x)$. For all t it holds $X^t \subseteq S^t$. Also let Z^t be the set of all present elements that have been blocked up to step t, that is, $Z^t = \{Z_e^t = 1 \mid e \in E\} \cap R(x)$. We are going to show that the following sequence

$$\left((\lambda+1) \cdot f\left(X^t\right) - f\left(S^t \cup Z^t\right)\right)_{t=0}^{n}$$

is a submartingale. Let us consider the deltas

$$\begin{aligned} & \mathbb{E}\left[f\left(X^{t+1}\right) - f\left(X^t\right) \middle| \mathcal{F}^t\right] \\ = & \sum_{e \in E} x_e \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t \middle| \mathcal{F}^t\right] \cdot \\ & \max\left(f\left(X^t + e\right) - f\left(X^t\right), 0\right), \end{aligned}$$

and

$$\begin{aligned} & \mathbb{E}\left[f\left(S^{t+1} \cup Z^{t+1}\right) - f\left(S^t \cup Z^t\right) \middle| \mathcal{F}^t\right] \\ = & \sum_{e \in E} x_e \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t + Z_e^{t+1} - Z_e^t \middle| \mathcal{F}^t\right] \cdot \\ & \left(f\left(S^t \cup Z^t + e\right) - f\left(S^t \cup Z^t\right)\right). \end{aligned}$$

From Lemma III.3 we have $\mathbb{E}\left[Z_e^{t+1} - Z_e^t \middle| \mathcal{F}^t\right] \leq \lambda \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t \middle| \mathcal{F}^t\right]$, and from submodularity we know that $f\left(S^t \cup Z^t + e\right) - f\left(S^t \cup Z^t\right) \leq f\left(S^t + e\right) - f\left(S^t\right) \leq f\left(X^t + e\right) - f\left(X^t\right)$. Therefore

$$\begin{aligned} & \mathbb{E}\left[f\left(S^{t+1} \cup Z^{t+1}\right) - f\left(S^t \cup Z^t\right) \middle| \mathcal{F}^t\right] = \\ = & \sum_{e \in E} x_e \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t + Z_e^{t+1} - Z_e^t \middle| \mathcal{F}^t\right] \cdot \\ & \left(f\left(S^t \cup Z^t + e\right) - f\left(S^t \cup Z^t\right)\right) \\ \leq & \sum_{e \in E} x_e \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t + Z_e^{t+1} - Z_e^t \middle| \mathcal{F}^t\right] \cdot \\ & \max\left(f\left(X^t + e\right) - f\left(X^t\right), 0\right) \\ \leq & (\lambda+1) \cdot \sum_{e \in E} x_e \cdot \mathbb{E}\left[S_e^{t+1} - S_e^t \middle| \mathcal{F}^t\right] \cdot \\ & \max\left(f\left(X^t + e\right) - f\left(X^t\right), 0\right) \\ = & (\lambda+1) \cdot \mathbb{E}\left[f\left(X^{t+1}\right) - f\left(X^t\right) \middle| \mathcal{F}^t\right], \end{aligned}$$

and we conclude that the sequence $\left((\lambda+1) \cdot f\left(X^t\right) - f\left(S^t \cup Z^t\right)\right)_{t=0}^{n}$ is indeed a submartingale. Since $S^n \cup Z^n = R(x)$ and $f(\emptyset) \geq 0$, we have

$$\begin{aligned} 0 & \leq \lambda \cdot f(\emptyset) = (\lambda+1) \cdot f\left(X^0\right) - f\left(S^0 \cup Z^0\right) \leq \\ & \leq (\lambda+1) \cdot \mathbb{E}\left[f\left(X^n\right)\right] - \mathbb{E}\left[f\left(S^n \cup Z^n\right)\right] \\ & = (\lambda+1) \cdot \mathbb{E}\left[f\left(X^n\right)\right] - \mathbb{E}\left[f\left(R\left(x\right)\right)\right], \end{aligned}$$

and further, by the definition of the multilinear extension F,

$$(\lambda+1) \cdot \mathbb{E}\left[f\left(X^n\right)\right] \geq \mathbb{E}\left[f\left(R\left(x\right)\right)\right] = F\left(x\right).$$

■

Theorem I.2. *Maximization of a non-negative submodular function with respect to k matroid constraints admits a $(k+1+\varepsilon) \cdot e$ approximation algorithm in the random-order model.*

Proof: Corollary III.2 and Lemma III.6 implies that the characteristic sequences of the CR scheme for the intersection of k matroids are k-bounded. We find vector x, such that $F\left(x\right) \geq \left(\frac{1}{e} - \varepsilon\right) \cdot f\left(X_{OPT}\right)$, with the measured continuous greedy algorithm [9], sample elements accordingly to x, and apply Lemma V.1. In the end we adjust ε to appropriately depend on k. ■

VI. Stochastic k-set packing

In the basic stochastic k-set packing problem, we are given n elements/columns, where each item $e \in E = [n]$ has a profit $v_e \in \mathbb{R}_+$, and a random d-dimensional size $L_e \in \{0,1\}^d$. The sizes are independent for different items. Additionally, for each item e, there is a set Q_e of at most k coordinates such that each size vector L_e takes positive values only in these coordinates, i.e., $\{e \in E \mid L_e = 1\} \subseteq Q_e$ with probability 1. We are also given a capacity vector $b \in \mathbb{Z}_+^d$ into which items must be packed. We assume that v_e is a random variable that may be correlated with L_e. The coordinates of L_e also might be correlated between each other. After probing element e, its size L_e is revealed and the reward v_e is drawn.

Equivalently, one can consider d copies of each item: $e^1, e^2, \ldots, e^d$, so that if e is probed then its i-th copy materializes with probability p_e^i and $p_e^i = 0$ for $i \notin Q_e$. In this view the capacity vector b induces a constraint family U_{b_i} over the ground set E^i of i-th copies of each item. We can easily generalize this setting to consider arbitrary matroid constraint $\mathcal{M}_i$ over E^i. Let $R^i \subseteq E^i$ denote the random set of materialized i-th copies of elements.

The following linear program (used first in [4] in the case of uniform matroids) provides a relaxation for the problem.

$$\begin{aligned} \max \quad & \sum_{e\in E} \mathbb{E}\left[v_e\right]\cdot x_e & \text{(SETPACKING-LP)} \\ \text{s.t.} \quad & p^i \cdot x \in \mathcal{P}\left(\mathcal{M}_i\right) & \forall i \in [d] \\ & x_e \in [0,1] & \forall e \in [n]\,, \end{aligned}$$

where, as usual, x_e is interpreted as $\mathbb{P}$[optimal solution probes e]. We are going to present a probing strategy in which for every element e the probability of being probed is at least $\frac{x_e}{k+1}$. Having this property, a $(k+1)$-approximation guarantee will follow.

Theorem I.5. *There exists a $(k+1)$ approximation algorithm for stochastic k-set packing over matroid row constraints.*

Proof: We present the abstract view of the mechanism in Algorithm 5.

Algorithm 5 Controller mechanism for stochastic k-set packing

```
1: solve SETPACKING-LP; let (x_e) be the solution
2: for each element e do
3:     for each constraint i ∈ Q_e do
4:         assign e a controller C_e^i with respect to vector p^i · x
5: for each element e in random order do
6:     if (e^i, C_e^i) has been blocked for any i ∈ Q_e then
7:         continue
8:     take e into the solution with probability x_e
9:     for each constraint i ∈ Q_e do
10:        if e^i ∈ R^i then
11:            update controllers with respect to the i-th constraint
```

Matroid implementation: The controller mechanism is analogous to those in Sections II and IV. We decompose vector $p^i \cdot x$ into a support in matroid $\mathcal{M}_i$, that is $p^i \cdot x = \sum_{j\in\mathcal{S}_i} \beta_j \cdot B_j^0$ (Lemma II.2), and find exchange-mappings between each pair of sets (Lemma II.3). Then for each element e we choose $j(e,i) \in \mathcal{S}_i$ such that $e \in B_j$, with probability $\frac{\beta_j}{p_e^i \cdot x_e}$, call it the i-th controller of c, and denote[3] it by C_e^i.

When an element e gets accepted, we update the controllers in Q_e, as presented previously in Algorithm 4. A pair (e, C_e^i) gets blocked when e is removed from C_e^i.

Correctness: Let S^t denote the set of accepted elements up to step t. The controller mechanism ensures that if e has not been blocked, then $\left\{e^i \mid e \in S^t\right\} \cap R^i$ together with e belong to C_e^i, which is an independent set in $\mathcal{M}_i$ for all $i \in Q_e$. Therefore taking e into the solution would not break any constraint from Q_e and other constraints are oblivious to e.

Approximation guarantee: Consider the event that (e, C_e^i) gets blocked in step t in the i-th constraint. For this to happen, an element $f \neq e$ must be chosen with probability $\frac{1}{n-t}$, it must be taken into the solution with probability x_f and it must exist in R^i with probability p_f^i. A choice of particular controller $j(f,i)$ happens with probability $\frac{\beta_j}{p_e^i \cdot x_e}$. Let $\Gamma^{i,t}(e,C)$ denote the set of pairs (f,j) that would block (e,C) in step t in i-th constraint, as in Lemma II.6. Combining all of these, we get a bound on the probability of a blocking event

$$\begin{aligned} & \frac{1}{n-t}\sum_{f\neq e} \mathbb{P}\left[f \text{ gets accepted in line 8}\right]\cdot \mathbb{P}\left[f\in R^i\right]\cdot \\ & \qquad \sum_{j:(f,j)\in\Gamma^{i,t}(e,C_e^i)} \mathbb{P}\left[f \text{ chooses controller } j\right] \\ = \; & \frac{1}{n-t}\sum_{f\neq e} x_f \cdot p_f^i \sum_{j:(f,j)\in\Gamma^{i,t}(e,C_e^i)} \frac{\beta_j}{p_f^i \cdot x_f} \\ = \; & \frac{1}{n-t}\sum_{(f,j)\in\Gamma^{i,t}(e,C_e^i)} \beta_j \leq \frac{1}{n-t}, \end{aligned}$$

where the last inequality follows from Lemma II.6.

As in previous sections, we use characteristic sequences to keep track of status of e in the i-th constraint. From the derivation above we know that they are 1-bounded. Since e could be blocked only by constraints from Q_e, the joint characteristic sequences are given by a combination of at most k sequences, as described in Section III-B. Lemma III.6 says that these sequences are k-bounded and Lemma III.5 guarantees that e will reach line 8 with probability at least $\frac{1}{k+1}$. This finishes the proof. ■

[3] There is a notation conflict in the superscript of C_e^i as in previous sections we used that to refer to the set C_e in step t. This time we reserve it to denote the constraint index.

VII. ACKNOWLEDGMENTS

The second author has been supported by the National Science Center of Poland Grant UMO-2016/21/N/ST6/00968.

REFERENCES

[1] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. *Math. Oper. Res.*, 41(3):1022–1038, 2016.

[2] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* Wiley Publishing, 4th edition, 2016.

[3] Lawrence M. Ausubel and Paul Milgrom. The lovely but lonely vickrey auction. In *Combinatorial Auctions, chapter 1.* MIT Press, 2006.

[4] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When LP is the cure for your matching woes:

Improved bounds for stochastic matchings. *Algorithmica*, 63, 2012.

[5] Nikhil Bansal, Nitish Korula, Viswanath Nagarajan, and Aravind Srinivasan. On k-column sparse packing programs. In *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings*, pages 369–382, 2010.

[6] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Algorithms to approximate column-sparse packing problems. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 311–330, 2018.

[7] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC*, 2010.

[8] Moran Feldman. *Maximization Problems with Submodular Objective Functions*. PhD thesis, Technion, 2013.

[9] Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *FOCS*, 2011.

[10] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1014–1033, 2016.

[11] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization - 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings*, pages 205–216, 2013.

[12] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *STOC*, pages 123–136, 2012.

[13] Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[14] Euiwoong Lee and Sahil Singla. Optimal online contention resolution schemes via ex-ante prophet inequalities. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 57:1–57:14, 2018.

[15] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.*, 35(4):795–806, 2010.

[16] Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, February 1981.

[17] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. 2003.

[18] Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *STOC*, pages 783–792, 2011.

[19] Justin Ward. Private communication, 2015.