Online Contention Resolution Schemes for Matchings and Matroids

by

Iliad Ramezani

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

September 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Online Contention Resolution Schemes for Matchings and Matroids

submitted by **Iliad Ramezani** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Bruce Shepherd, Professor, Department of Computer Science, UBC
Co-supervisor

Hu Fu, Professor, Department of Computer Science, UBC
Co-supervisor

Abstract

We investigate the problem of designing (preferably optimal) online contention resolution schemes (OCRSs). They are a powerful framework for optimization under various combinatorial constraints such as matroids, matchings, knapsacks and their intersections. OCRSs immediately imply prophet inequalities in Bayesian online selection problem. They also have other important applications such as stochastic probing and oblivious posted price mechanisms.

We present several novel OCRSs which improve the current state-of-the-art algorithms. We design an optimal 0.5-selectable OCRS over matroids with rank 2, and another optimal 0.5-selectable OCRS over transversal matroids. Previously the best result applicable to these types of matroids was a 0.25-selectable OCRS. Furthermore, we design a $1/(k+1)$-selectable OCRS over matchings in k-partite hypergraphs.

Lay Summary

A widely used technique in online optimization problems under various constraints is online contention resolutions schemes. They were initially designed as a generic technique for rounding a fractional solution. However, they can be used to tackle a broad class of problems. We consider the problem of designing optimal online contention resolution schemes over combinatorial structures such as matchings in graphs. These results have many important implications in areas such as Bayesian online selection problem and other problems in mechanism design.

Preface

The entire work presented in this thesis is original, unpublished, independent research conducted by the author, Iliad Ramezani, under supervision of Dr. Hu Fu and Dr. Bruce Shepherd. The OCRS for matroids with rank 2 was developed in joint work with Abner Turkieltaub. The OCRS for matchings in $k-$partite hypergraphs was designed and analysed by the author of this thesis.

Table of Contents

Acknowledgments

I would like to express a deep sense of gratitude to my two supervisors, Hu Fu and Bruce Shepherd. They have provided invaluable guidance throughout the research process. I am extremely grateful for the lessons I have learned from them. I would like to thank Abner Turkieltaub, for his ideas and collaboration.

To my closest friend, Arezou Fatemi, thank you for always supporting me.

Lastly, I would like to thank my parents for their love and sacrifices, I wouldn't be able to achieve anything without them.

Chapter 1

Introduction

Combinatorial optimization problems have been a major subject of study in theoretical computer science. More recently, a technique has been frequently and successfully used to tackle such problems, namely *contention resolution schemes* (CRS). They were first introduced by Chekuri et al. [7] as a tool to maximize non-negative submodular set functions under various combinatroial constraints such as matroids and knapsacks. Later on, an adaptation of CRSs to online settings was introduced by Feldman et. al. [10], and is called Online Contention Resolution Schemes (OCRS). They have applications in many problems such as prophet inequalities and oblivious posted pricing mechanisms.

1.1 Preliminaries

We will define some of the concepts used throughout the thesis.
Let N be a finite set.

Definition 1.1.1. *A family of subsets $S \subseteq 2^N$ is called downward closed, if $A \in S$ and $B \subseteq A$, then $B \in S$, for any $A, B \subseteq N$.*

Let $\mathscr{F}$ be a downward-closed family of subsets of N. We sometimes refer to sets in $\mathscr{F}$ as *feasible* or *independent* subsets.

Definition 1.1.2. *$P_{\mathscr{F}} \subseteq [0,1]^N$, the polytope corresponding to $\mathscr{F}$, is the convex hull of all characteristic vectors of feasible subsets.*

When the polytope is clear from the context we may refer to it as P.

1.1.1 Matroids

Definition 1.1.3. *A matroid M is a pair $(N, \mathscr{F})$ where N is the ground set and $\mathscr{F} \subseteq N$ is the family of independent sets that has the following properties:*

1. *Downward closure: $\mathscr{F}$ is downward closed.*
2. *Exchange property: If A and B are two independent sets such that $|A| > |B|$, then there exist an element $e \in A \setminus B$ such that $B + e$ is also independent.*

In order to define what transversal matroids are, we first need to define set systems and partial transversals.

A *set system* is a collection A of subsets of a finite set S:

$$A = (A_j : j \in J)$$

A *partial transversal* of A is a set $T \subseteq S$ such that there is an injective function $\phi : T \to J$ where:

$$t \in A_{\phi(t)} \ \ \forall t \in T$$

The set of all partial transversals of a set system A forms the independent sets of a matroid M [8].

Definition 1.1.4. *Let A be a set system on a set S. Let $\mathscr{F}$ be the set of all partial transversals of A. The matroid $M(A) = (S, \mathscr{F})$ is called a transversal matroid.*

We can also represent transversal matroids via a bipartite graph $G = (S, A, E)$ where the edges in the graph indicate the membership of the elements of S in the sets of the set system A.

1.2 Maximizing submodular functions via contention resolution schemes

Before introducing CRSs, we take a brief digression to discuss the problem they were originally designed to solve. Consider the problem of maximizing a non-negative submodular set function. Let N be the ground set of elements, $\mathscr{F}$ be the feasibility family (constraint), and its polytope P. Moreover, we're given a non-negative submodular function $f : 2^N \to \mathrm{R}_+$ which we want to maximize. We first extend f to a continuous function over P. An appropriate extension which used in [7] is the multilinear extension [4]:

$$F(x) = \sum_{S \subseteq N} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$$

Note that the multilinear extension agrees with f on integral points. Let $R(x)$ be a random set that contains each element $i \in N$ with probability x_i, independent from other elements. Also note that by the definition of the multilinear extension, the value $F(x)$ is equivalent to the expected value of $f(R(x))$.

The approach consists of 2 parts:

- We must find a point x^* such that $F(x^*)$ is approximately maximum over integral points of P, i.e., $F(x^*) \geq c_1 \max_{x \in P \cap \{0,1\}^N} F(x)$ for some parameter $c_1 > 0$
- We should design an algorithm that effectively rounds any fractional solution $x \in P$ to a feasible solution $S \in \mathscr{F}$ such that the objective value is approximately preserved, i.e., $F(1_S) \geq c_2 F(x)$ for some parameter $c_2 > 0$.

Combining these two ideas results in a set that approximately maximizes f over $\mathscr{F}$. The approximation factor of this algorithm would be $c_1 c_2$, the product of approximation factors of the algorithm in each of the 2 parts. Similar approaches have been used before in [3, 4, 6, 14, 16]. We will discuss each of these two parts in further details in the following sections.

1.2.1 Maximizing the multilinear extension

If the submodular function f is assumed to be monotone, then there is a $(1 - 1/e)$-approximation for the problem $\max\{F(x) : x \in P\}$ where F is the multilinear extension of f and P is any solvable polytope in [5, 17]. Furthermore, from [18] we know that if f is non-monotone, then there is no constant factor approximation for the matroid base polytope which is not down-monotone. If f is a non-negative submodular function and P is a solvable down-monotone polytope, [7] showed that a $(0.25 - o(1))$-approximation algorithm exists. They also designed a 0.309 algorithm for the problem $\max\{F(x) : x \in P \cap \{0,1\}^N\}$. This algorithm can also be used in this approach since we only need to compare the fractional solutions to the optimal integral solution.

1.2.2 Contention resolution schemes

Contention resolutions schemes provide a powerful framework for rounding fractional solutions. Let $x \in P$. The random set $R(x)$ is an intuitive way to round x, however $R(x)$ is not necessarily feasible. CRSs produce a set $S \subseteq R(x)$ by removing some elements from $R(x)$. They aim to produce this set in a way that each element $i \in N$ belongs to S with probability at least cx_i for some parameter c. Such CRSs are called $c-$balanced.

Definition 1.2.1. *Let $c \in [0,1]$. A $c-$balanced CRS for P is a procedure that for every $x \in P$, returns a random set $S(x) \subseteq R(x)$ and satisfies the following:*

1. *$S(x) \in \mathscr{F}$ with probability 1 $\quad \forall x \in P$*
2. *$Pr[e \in S(x) \mid e \in R(x)] \geq c \;\; \forall e \in N \, \forall x \in P$*

1.3 Online contention resolution schemes

An adaptation of CRSs to online settings was introduced by Feldman et al. [10]. Similar to before, consider a finite set N of ground elements, downward-closed feasible family $\mathscr{F} \subseteq 2^N$ and its polytope P. We are also given a point x inside the polytope.

Definition 1.3.1. *Suppose the elements of N arrive one by one. Upon arrival of element $e \in N$, it is revealed to us whether e is active or not. The element e is active with probability x_e independent from the other elements. If the element is active, we must make a decision to either select it or not. This decision must be made immediately (before looking at the next elements), and it cannot be changed in the future. An OCRS is an algorithm outputs a selected feasible subset $S \in \mathscr{F}$ of active elements.*

By this definition, an OCRS can be as trivial as returning an empty set. We need to define a property of OCRSs similar to CRSs which would specify how effective they are. Let $R(x)$ be the random set of active elements.

Definition 1.3.2. *An OCRS is called $c-$selectable if for any element $e \in N$, given that e is active, the probability that the algorithm selects e is at least c. i.e., $P[e \in S \mid e \in R(x)] \geq c$.*

In the online setting we assume the arrival order of the elements is controlled by an adversary. When desigining OCRSs, another important aspect to keep in mind is the level of power of the adversary. The different types of adversaries include:

1. *offline adversary*: This adversary fixes the arrival order of the elements at the beginning (before the elements start arriving). In this case we assume that we are aware of the arrival order of the elements.

2. *online adversary*: This adversary chooses which element will be revealed next online. This type of adversary has exactly the same information as the OCRS does. For instance, the online adversary can make the decision based on the elements accepted by the OCRS so far. Clearly an online adversary is more powerful than an offline adversary.

3. *almighty adversary*: This is the most powerful adversary. He has access to all information, such as the random set $R(x)$ and results of the random coins used by the OCRS. Therefore, this adversary can produce the worst possible order based on all the information.

4. *random adversary (random order)*: In this case, the arrival order of the elements is chosen uniformly at random from all possible arrival orders.

In all the OCRSs presented in this paper, we assume we are working under the online adversary assumption, unless stated otherwise.

1.4 Prophet inequalities

Bayesian online selection problem: Suppose there is a set of elements N, and each element $e \in N$ is associated with an independent random variable $\mathscr{V}_e$ from a known distribution $\mathscr{D}_e$. We also have a downward closed family of feasible subsets $\mathscr{F} \subseteq 2^N$. The realization of the random variables are revealed to us online by an adversary. Upon each revelation, we have to irrevocably decide whether we want to take that element or not, while always maintaining a feasible set of taken elements A. Let v_e be the realization of the random variable $\mathscr{V}_e$. Our goal is to maximize the sum of the value of the selected elements:

$$\sum_{e \in A} v_e$$

Suppose someone can foresee the realization of all the random variables, and therefore can select the optimal feasible set. This person is commonly referred to as the *prophet* in the literature. The expected value that the prophet can achieve is:

$$\mathrm{E}[max\{\sum_{e \in I} v_e \mid I \in \mathscr{F}\}]$$

Definition 1.4.1. *A c-competitive prophet inequality is an algorithm for the Bayesian online selection problem that selects a set $A \in \mathscr{F}$ of elements satisfying:*

$$\mathrm{E}[\sum_{e \in A} v_e] \geq c \cdot \mathrm{E}[max\{\sum_{e \in I} v_e \mid I \in \mathscr{F}\}]$$

We will also make use of a stronger benchmark than the prophet's expected gain called *ex-ante* prophet inequality.

Definition 1.4.2. *A c-approximation ex-ante prophet inequality is an algorithm for the Bayesian online selection problem that selects a set $A \in \mathscr{F}$ of elements satisfying:*

$$\mathrm{E}[\sum_{e \in A} v_e] \geq c \cdot max_{x \in P_{\mathscr{F}}} \mathrm{E}[V_i \mid V_i \textit{ takes value in its top } x_i \textit{ quantile }]$$

1.5 Thesis Organization

We have organized this thesis as follows:

First, in Chapter 2, we review the related work and literature. We also discuss different results under different assumptions and types of adversaries for various restriction families.

In Chapter 3, we present a $\frac{1}{2}-$selectable OCRS for rank 2 matroids. This OCRS is optimal and works against the online adversary.

Then in Chapter 4, we present a $\frac{1}{2}-$selectable OCRS for transversal matroids. This OCRS is also optimal.

Then we move on to describe our last OCRS in Chapter 5. It is an $\frac{1}{k+1}$-selectable OCRS over matchings in $k-$partite hypergraphs.

In Chapter 6, we summarize our work and suggest possible directions for future works.

Chapter 2

Related Work

In their work, Feldman et al. [10] established a reduction from prophet inequalities to OCRSs. They showed that a c-selectable OCRS implies a c-competitive prophet inequality. It is easy to see that there exists no c-selectable OCRS (or c-competitive prophet inequality) over matroids for any $c > 1/2$. . The example involves a matroid with rank 1 that has 2 elements. Later, Lee and Singla [15] gave a reduction for the other direction. They showed a so-called c-approximation ex-ante prophet inequality implies a c-selectable OCRS. By Kleinberg and Weinberg [13], we have a $1/2$-competitive prophet inequality. However, their proof does not immediately extend to a $1/2$-approximation ex-ante prophet inequality. By building up on the algorithm in [13], the first $1/2$-approximation ex-ante prophet inequality was designed in [15]. This together with the reduction, implies that a $1/2$-selectable OCRS over matroids exists. However, this OCRS is given by an LP solved by the Ellipsoid algorithm, and an explicit description of such an OCRS is not known. The best known OCRS over general matroids is a $1/4$-selectable OCRS by [10]. Alaei [2] in his earlier work designed an optimal $1/2$-selectable OCRS over matroids with rank 1.

Another well studied type of OCRSs is for the case of random arrival order, also referred to as random-order contention resolution schemes (RCRS). Adamczyk and Wlodarczyk [1] designed a $\frac{1}{k+1}$-selectable RCRS for the intersection of k matroids. Their analysis made use of the martingale theory. In their work, Lee and Singla [15] also showed that a $c-$approximation ex-ante prophet inequality for random arrival order Bayesian online selection problem over a downward closed family $\mathscr{F}$ implies a $c-$ selectable RCRS over $\mathscr{F}$. By extendending the work of Ehsani et. al. they designed a $(1-\frac{1}{e})$-approximation ex-ante prophet inequality for random arrival order for matroids, thus implying a $(1-\frac{1}{e})$-selectable RCRS for matroids. This is also optimal due to an example given in [7]. We drew inspirations from the algorithm in [1] for designing our OCRS for transversal matroids.

When the feasibility family is the set of all matching (elements are edges of a graph and a feasible set is a subset of edges where no two of them share a common node), Gravin and Wang [12] showed $1/3$-approximation prophet inequality. This does not immediately imply a $1/3$-selectable OCRS over matching, since one needs to show that the algorithm is a $\frac{1}{3}$-approximation ex-ante prophet inequality. Also their approach is based specifically on bipartite graphs, and cannot be extended to $k-$partite hypergraphs. Later, Fu et al. [11] designed RCRSs for matchings in general graphs in the model where arriving elements are vertices.

It is important to note that the $3-$selectable OCRS for matchings presented in this thesis is also presented by Ezra et al. [9]. Our work was done independently in the spring of 2020.

Chapter 3

Optimal OCRS for Matroids with Rank 2

Suppose we have a matroid $M = (N, \mathscr{F})$ such that $rank(M) = 2$. Let P be the matroid polytope. An input consists of a point $x \in P$. Since the matroid polytope is defined as the convex hull of the characteristic vectors of $\mathscr{F}$, we can represent x as a weighted sum of characteristic vectors of some independent sets of M. For now we assume these independent sets are bases. Let $B = \{b_1, b_2, \ldots, b_m\}$ be a set of bases of M and $w : B \to \mathrm{R}_+$ a weight function such that $x = \sum_{b \in B} w(b) 1_b$ and $\sum_{b \in B} w(b) \leq 1$. Let $e_{2i}, e_{2i+1} \in N$ represent the elements in base b_i.

We also assume all elements e_i are distinct, later we will show how to relax these assumptions.

Given these assumptions, for each element $e \in N$, we can let $\bar{e}$ be the element that together with e forms one of the bases in B.

3.1 Constructing blocking sets

A key idea in our approach, is to restrict the elements that can be taken together. For each element e, we partition all the elements into two sets $Q(e)$ and $P(e)$. We prohibit taking e alongside any element in $P(e)$. Furthermore, for each base b, we want to include exactly one of its elements in $P(e)$ and the other one in $Q(e)$.

The construction of the sets $Q(e)$ can be seen in Algorithm 1. Some of the important properties of these sets are:

1. $|Q(e) \cap b| = 1 \quad \forall e \in N \, \forall b \in B$
2. $\{e, e'\} \in \mathscr{F} \quad \forall e \in N \, \forall e' \in Q(e)$
3. $Q(e) \cap Q(\bar{e}) = \emptyset \quad \forall e \in N$
4. $e \in Q(a) \quad \forall e \in N \, \forall a \in Q(e)$

Let $P(e) = N \setminus Q(e)$. By the first property, exactly one element of each base is included in $Q(e)$, and therefore the other element of the base belongs to $P(e)$. By the second property, e together with any element in $Q(e)$ forms an independent set. The way we will use these sets, is that whenever an element e arrives, we only consider selecting it if we have not selected any element in $P(e)$. The other properties of $Q(e)$ will be useful in the proof of our algorithm.

Algorithm 1 Constructing $Q(e)$

```
for i ∈ [m] do
    Q(e_{2i−1}) ← e_{2i}
    Q(e_{2i}) ← e_{2i−1}
    for j ∈ [m] \ [i] do
        if {e_{2i−1}, e_{2j}} ∈ F ∧ {e_{2i}, e_{2j−1}} ∈ F then
            Q(e_{2i−1}) ← e_{2j}
            Q(e_{2i}) ← e_{2j−1}
            Q(e_{2j−1}) ← e_{2i}
            Q(e_{2j}) ← e_{2i−1}
        else
            Q(e_{2i−1}) ← e_{2j−1}
            Q(e_{2i}) ← e_{2j}
            Q(e_{2j−1}) ← e_{2i−1}
            Q(e_{2j}) ← e_{2i}
        end if
    end for
end for
```

To construct $Q(e)$, we have used two nested loops. When the outer loop is on base b_i, we have already added exactly one element from each of the bases $b_k, k < i$

to $Q(e_{2i-1})$. Also the other element of base b_k has been added to $Q(e_{2i})$. We then add e_{2i} to $Q(e_{2i-1})$. This is a valid choice since $\{e_{2i}, e_{2i-1}\}$ is a base and therefore an independent set. Then the inner loop goes through all the bases $b_j, j > i$, and we add exactly one element $e \in b_j$ to $Q(e_{2i-1})$. The first, third and the fourth properties together then uniquely determine which element from base b_j must be added to $Q(e_{2i})$, and which elements from the base B_i must be added to $Q(e_{2j-1})$ and $Q(e_{2j})$. We now only need to check if the second property is satisfied. If the the condition in the if statement is true, then the second property is clearly satisfied. Otherwise, we are in the else statement, and the second property is satisfied by the following lemma.

Lemma 3.1.1. *For any two independent sets $b_1 = \{a,b\}$ and $b_2 = \{c,d\}$ of some matroid M, if at least one of the sets $\{a,d\}$ and $\{b,c\}$ is dependent, then both sets $\{a,c\}$ and $\{b,d\}$ are independent.*

Proof. First, note that by downward closure all the singletons are also independent sets. By symmetry, assume $\{a,d\}$ is dependent. Since $\{a,b\}$ is an independent set, by considering $\{a,b\}$ and $\{d\}$, the exchange lemma indicates that at least one of the sets $\{a,d\}$ or $\{b,d\}$ is independent. Since $\{a,d\}$ is a dependent set, $\{b,d\}$ must be an independent set. Similarly, by considering $\{c,d\}$ and $\{a\}$, the exchange lemma indicates that $\{a,c\}$ is an independent set. □

3.2 The OCRS

We present our OCRS in Algorithm 2.

The algorithm consists of two phases:

In the first phase, we simulate the elements. We suppose elements e_1 and e_2 from b_1 arrive first, then elements e_3 and e_4 arrive from b_2, and so on. Whenever an element e is supposed to have arrived, we simulate a random event to decide if it is active with probability x_e. If previously we have selected an element that belongs to $P(e)$, or if we have already selected two elements, we discard e. Otherwise, we select it with some probability that can be seen in the algorithm. We denote the set of selected elements in the algorithm by Alg. After deciding to either select e or not, we add e to the set of arrived elements A.

Algorithm 2 OCRS for matroids with rank 2

Run Algorithm 1 to construct the family of sets $Q(e)$
$P(e) \leftarrow N \setminus Q(e)$, $\forall e \in N$
$Alg \leftarrow \emptyset$
$A \leftarrow \emptyset$
for $i \in [1,\ldots,2m]$ **do**
  Flip a coin to decide whether e_i is *supposedly* active w.p. x_i
  if $Alg \subseteq Q(e_i) \wedge |Alg| < 2 \wedge e_i$ is supposedly active **then**
    $Alg \leftarrow e_i$ w.p. $(2 - \sum_{e \in A \cap P(e_i)} x_e)^{-1}$
  end if
  $A \leftarrow A + e_i$
end for
while an element e_i arrives **do**
  $A \leftarrow A - e_i$
  $Alg \leftarrow A - e_i$
  if $Alg \subseteq Q(e_i) \wedge |Alg| < 2 \wedge e_i$ is revealed to be active **then**
    $Alg \leftarrow e_i$ w.p. $(2 - \sum_{e \in A \cap P(e_i)} w_e)^{-1}$
  end if
  $A \leftarrow A + e_i$
end while

In the second phase, we begin to actually look at the real elements. When element e arrives, we remove it from the set of already arrived elements, and from the set of selected elements (if it is there). We then do the same selection process in the first phase as if e was the last element to arrive. An important difference is that we no longer decide whether e is active ourselves, since this information is given to us for each element.

Theorem 3.2.1. *Algorithm 2 is a* $1/2$*-selectable OCRS over matroids with rank 2.*

Correctness Clearly the algorithm selects at most 2 elements. If $|Alg| < 2$, then $Alg \in \mathscr{F}$ follows immediately. Now suppose $Alg = \{e, y\}$, and moreover, e was the last element that was selected. Since e was selected, then $y \in Q(e)$, and by the second property of $Q(e)$ we have $Alg \in \mathscr{F}$.

Let S_e be the event that $e \in Alg$. Also, let V_e be the event that when considering

e, it can be selected, more precisely $Alg \subseteq Q(e) \wedge |Alg| < 2$ holds at the time.

Lemma 3.2.2. *When the algorithm has considered to select the element e, we have:*

1. $P(S_y) = \frac{x_y}{2} \;\; \forall y \in A$
2. $P(S_y \cap S_z) = \frac{x_y x_z}{4} \;\; \forall y, z \in A, y \in Q(z)$
3. $P(S_y \cap S_z) = 0 \;\; \forall y, z \in A, y \in P(z)$

Proof. We prove this by using induction on the size of A.

Base Case. When $A = \emptyset$, the statement is trivial.

Induction Step. Suppose the statement holds for the set A. We will prove it holds for $A + e$.

To verify (1), we need to show $P(S_e) = \frac{x_e}{2}$, since equality holds for other elements by the induction hypothesis. First let's calculate $P(V_e)$. The event $\bar{V}_e$ can be partitioned into two disjoint events V_1 and V_2 where V_1 is the event that at least one element from $P(e)$ was selected, and V_2 is the event that two elements from $Q(e)$ were selected. So we have:

$$P(V_2) = \sum_{y,z \in A \cap Q(e)} P(S_y \cap S_z)$$

To calculate $P(V_1)$, we use the inclusion exclusion principle. However, since $|Alg| \leq 2$ we have:

$$P(V_1) = \sum_{y \in A \cap P(e)} P(S_y) - \sum_{y,z \in A \cap P(e)} P(S_y \cap S_z)$$

By combining these equations we get:

$$P(V_e) = 1 - \sum_{y \in A \cap P(e)} P(S_y) + \sum_{y,z \in A \cap P(e)} P(S_y \cap S_z) - \sum_{y,z \in A \cap Q(e)} P(S_y \cap S_z)$$

$$= 1 - \sum_{y \in A \cap P(e)} \frac{x_y}{2} + \sum_{y,z \in A \cap P(e), y \in Q(z)} \frac{x_y x_z}{4} - \sum_{y,z \in A \cap Q(e), y \in Q(z)} \frac{x_y x_z}{4}$$

Where in the second line we used the induction hypothesis and the fourth property of $Q(e)$.

Since elements from bases arrive one by one in the first phase, all the bases (possibly except for the base of the current element) have either both of their elements in A or neither of them are in A. Therefore, by the first and third property of $Q(e)$ if $y \in A \cap P(e)$ then $\bar{y} \in A \cap Q(e)$. Similarly if $y \in (A \cap Q(e)) - \bar{e}$ then $\bar{a} \in A \cap Q(e)$. Therefore, we can rewrite the last summation as the following:

$$P(V_e) = 1 - \sum_{y \in A \cap P(e)} \frac{x_y}{2} + \sum_{y,z \in A \cap P(e), y \in Q(z)} \frac{x_y x_z}{4} - \sum_{y,z \in A \cap P(e), y \in Q(z)} \frac{x_{\bar{y}} x_{\bar{z}}}{4}$$

$$= 1 - \sum_{y \in A \cap P(e)} \frac{x_y}{2}$$

Since the element e is selected only if the event V_e occurs (by definition):

$$P(S_e) = P(S_e | V_e) P(V_e)$$

By the algorithm, $P(S_e|V_e) = w_e(2 - \sum_{y \in A \cap P(e)} x_y)^{-1}$ where x_e is the probability of the event that e is active. Therefore:

$$P(S_e) = \frac{x_e}{2 - \sum_{y \in A \cap P(e)} x_y} (1 - \sum_{y \in A \cap P(e)} \frac{x_y}{2}) = \frac{x_e}{2}$$

To verify (2), for $y \in A \cap Q(e)$ we have:

$$P(S_e \cap S_y) = P(S_e | S_y) P(S_y)$$

$$= \frac{x_e}{2 - \sum_{z \in A \cap P(e)} w_z} P(V_e | S_y) \frac{x_y}{2}$$

$$= \frac{x_e x_y}{4} \frac{1 - \sum_{z \in A \cap Q(y)} P(S_z | S_y)}{1 - \sum_{z \in A \cap P(e)} \frac{x_z}{2}}$$

$$= \frac{x_e x_y}{4} \frac{1 - \sum_{z \in A \cap Q(y)} \frac{x_z}{2}}{1 - \sum_{z \in A \cap P(e)} \frac{x_z}{2}}$$

$$= \frac{x_e x_y}{4}$$

Lastly,(3) is true because the algorithm never selects an item z if it has already selected an item y such that $y \in P(z)$.

□

3.3 Relaxing the assumptions

In this section we show how to remove the assumption that the decomposition only uses bases and also the assumption that all the elements in bases are unique.

To relax the first assumption, suppose element e appears as an independent set with weight w in the decomposition. We create a new element e_r such that $x_{e_r} = w$. Add $\{e_r\}$ to the independent sets. Also, for any other element e, add $\{e, e_r\}$ to the independent sets. We claim that the new family of independent sets is still a matroid. The family is clearly still downward-closed. To check the exchange property, consider two independent sets A and B such that $|A| > |B|$. Since the rank of the matroid is still 2, we have $|B| < 2$. If neither A nor B contain e_r, the property holds since the original family was a matroid. Now suppose $e_r \in B$. Since $|B| < 2$, e_r must be the only element in B and any element in $A \setminus B$ can be added to B. If A contains e_r, then e_r can be added to B (by the construction of the new independent sets). Therefore the new family is still a matroid. Now replace the set $\{e\}$ with $\{e, e_r\}$ in the decomposition with same weight w. If we do this for all single item sets in the decomposition, we would get a new point x' that could be written as a convex combination of bases. Note that the newly created elements are not actual elements, and at the end if they are included in the solution we can safely remove them.

To relax the second assumption, suppose element e is an element that appears in more than one bases. Let $B_e \subseteq B$ be the subset of bases that e belongs to. For any base $b \in B_e$, we create a new element e_b and replace base b with $b - e + e_b$. We should also adjust the independent subsets. All subsets of consisting of two of the newly created elements are dependent. All subsets of the form $\{e_b, y\}$ where e_b is a new element is independent if and only if $\{e, y\}$ was independent. It is easy to see that the new independent sets still form a matroid.
In the Algorithm 2, when the element e arrives and is revealed to be active, we throw a $|B_e|$ sided dice where for each base $b \in B_e$, we have a corresponding side in dice that has probability $\frac{w(b)}{x_e}$ of appearing on top. Note that $\sum_{b \in B_e} w(b) = x_e$. Suppose base $b \in B_e$ is the top side if the thrown dice. Then we say element e_b is

active and all other elements $e_{b'}$ where $b' \in B_e - b$ are not active.
We do the process described above for all elements $e \in N$ that appear in multiple bases in B. Since the sets consisting of two copies of one these elements such as e are dependent, by Algorithm 1 for any $b, b' \in B_e$ we would have $e_{b'} \in P(e_b)$. Therefore, the copies of the same element will not interact with each other, and our proof would still correct.

This section together with Lemma 3.2.2 proves Theorem 3.2.1.

Chapter 4

Optimal OCRS for Transversal Matroids

Let $G = ((U,V),E)$ be a bipartite graph that corresponds to a transversal matroid. Let $\mathscr{F} \subseteq 2^U$ be the set of independent sets of the matroid. Without loss of generality (see Proposition 4.0.1), we may assume the rank of the matroid is $|V|$.

Proposition 4.0.1. *Let M be a maximum matching in G, with L_1 and R_1 being the sets of matched nodes in U and V, respectively. If M' is another maximum matching in G, with L_2 and R_2 being the sets of matched nodes in U and V, respectively, then there is a matching with L_2 and R_1 being the sets of matched nodes in U and V, respectively.*

Proof. The symmetric difference between M and M' consists of cycles and paths. Every node in $L_1 \cap L_2$ has degree two in $M \triangle M'$. Any path starting from a node in $R_1 \setminus R_2$ must end in a node in $R_2 \setminus R_1$, by passing only nodes in $(L_1 \cap L_2) \cup (R_1 \cap R_2)$. By taking these alternating paths from M' we obtain a new matching whose matched nodes are L_2 and R_1. $\square$

For every $B \in \mathscr{F}$, fix a matching which matches all nodes in B. This matching defines an injection $\phi_B : B \to V$. You can see the OCRS for trasnversal matroids in Algorithm 3.

Algorithm 3 OCRS for transversal matroids

Given a point x in the polytope of the transversal matroid, decompose it into $x = \sum_{B \in \mathscr{F}} \alpha_B 1_B$.
Let $b(u,v)$ denote the set of matchings in which u is mapped to v: $\{B \in \mathscr{F} \mid \phi_B(u) = v\}$.
For each vertex $v \in V$, run a 0.5-selectable single-item OCRS $\mathscr{A}_v$ on the set $\{b(u,v)\}_u$. The arrival order for algorithm $\mathscr{A}_v$ is $b(u_1,v), \ldots, b(u_n,v)$. $b(u,v)$ will be active with probability $\sum_{B \in b(u,v)} \alpha_B$.
When node u arrives and is shown active, draw $B \in \mathscr{F}$ randomly, with probability α_B / x_u. Tell algorithm $\mathscr{A}_{\phi_B(u)}$ that $b(u, \phi_B(u))$ is active.
Select u if and only if $\mathscr{A}_{\phi_B(u)}$ selects $b(u, \phi_B(u))$.

Correctness. First note that the input to each single-item OCRS is a valid single-item OCRS, since $\sum_u \sum_{B \in b(u,v)} \alpha_B = \sum_{B \in \mathscr{F}} \alpha_B \leq 1$.
The set of selected elements must be independent, because each $\mathscr{A}_v$ selects at most one node, which must have an edge going to v. Therefore the set of nodes that are selected in the end can all be simultaneously matched to nodes in V.

Theorem 4.0.2. *Algorithm 3 is a* $1/2$*-selectable OCRS over transversal matroids.*

Proof. Each $b(u,v)$, when declared to be active, is accepted with probability exactly, $1/2$ by the guarantee of the single-item OCRS; the probability with which u calls algorithm $\mathscr{A}_v$ is $[u] \cdot \frac{\sum_{B \in b(u,v)} \alpha_B}{[u]} = \sum_{B \in b(u,v)} \alpha_B$, the probability with which $b(u,v)$ is declared active to $\mathscr{A}_v$. Therefore, conditioning on u being active, it is selected with probability $\frac{1}{2}$. □

Chapter 5

OCRS for Matchings

In this chapter we will present a $1/3$-selectable OCRS for matchings in bipartite graphs, and then generalize it to matchings in $k-$uniform hypergraphs. These algorithms work against the offline adversary.

5.1 Bipartite graphs

Let $G = (V, E)$ be a bipartite graph. The ground set for this problem would be the set of edges E. Let $\mathscr{F}$ be the set of all matchings of G. Also, let $P_{\mathscr{F}}$ be the matching polytope, which is defined as:

$$\sum_{v \in \delta(u)} x_v \leq 1 \quad \forall u \in V$$

$$x_e \geq 0 \quad \forall e \in E$$

Let S_e be the event that the edge e is selected by our OCRS. If some other edge t is adjacent to e and is selected by the OCRS, we say t *blocks* e. We also say an edge e is *blocked* if there exists some edge t that blocks it. Let A_e be the event that e is not blocked. You can see the description of the OCRS in Algorithm 4.

We first discuss how $P(A_e)$ can be calculated. Suppose edge $e = (u, v)$ has arrived. Let $B(u)$ be the set of all the edges adjacent to u that have arrived before e. Define $B(v)$ similarly. Note that e is blocked, if and only if any of the edges in

Algorithm 4 OCRS for matchings in bipartite graphs

while an edge e arrives **do**
  Calculate $P(A_e)$
  If the edge e is revealed to be active and it is not blocked, take it with probability $\frac{1}{3P(A_e)}$
end while

$B(u) \cup B(v)$ has already been selected by the OCRS. Equivalently:

$$\overline{A_e} = \bigcup_{t \in B(u) \cup B(v)} S_t$$

.

Theorem 5.1.1. *Algorithm 4 is a* $1/3$*-selectable OCRS over matchings in bipartite graphs.*

Proof. Since we only select edges when they are active, we equivalently need to show $P(S_e) = \frac{x_e}{3}$. We also need to show $\frac{1}{3P(A_e)}$ is a valid property. We prove these by using induction on elements.

Base Case. Suppose e is the first edge to arrive. Clearly e is never blocked and thus $P(A_e) = 1$. Therefore, when e is active, we take it with probability $\frac{1}{3}$ and $P(S_e) = \frac{x_e}{3}$

Induction Step. Suppose the statement holds for the all the edges that have previously arrived, and the edge $e = (u, v)$ arrives next. Then:

$$P(\overline{A_e}) = P(\bigcup_{t \in B(u) \cup B(v)} S_t) \leq \sum_{t \in B(u) \cup B(v)} P(S_t)$$

$$= \frac{1}{3} \sum_{t \in B(u)} x_t + \frac{1}{3} \sum_{t \in B(v)} x_t \leq \frac{1}{3} \sum_{t \in \delta(u)} x_t + \frac{1}{3} \sum_{t \in \delta(v)} x_t$$

In the second equality we used the induction hypothesis. Since $x \in P_{\mathscr{F}}$, we have $\sum_{t \in \delta(u)} x_t \leq 1$ and $\sum_{t \in \delta(v)} x_t \leq 1$. Therefore:

$$P(\overline{A_e}) \leq \frac{2}{3} \implies P(A_e) \geq \frac{1}{3}$$

Thus $\frac{1}{3P(A_e)}$ is a valid probability. Since we take e with probability $\frac{1}{3P(A_e)}$ if it is active and not blocked we have:

$$P(S_e) = x_e P(A_e) \frac{1}{3P(A_e)} = \frac{x_e}{3}$$

Which completes the proof. □

5.2 $K-$partite hypergraphs

Recall that a $k-$partite hypergraph is a hypergraph $G = (V, E)$ with node set $V = \cup_{i\in[k]} V_i$, and each hyperedge $e \in E$ contains precisely k vertices, $(v_1, v_2, \ldots, v_k)$ such that $v_i \in V_i \ \forall i$. The ground set is the set of hyperedges E. Let $\mathscr{F}$ be the set of all matchings in G, and $P_{\mathscr{F}}$ be the matching polytope, similar to before. Much of the algorithm and proof closely resembles the algorithm and proof for the special case of bipartite graphs. Let the events S_e and A_e be defined in exactly the same way as before. The OCRS can be seen in Algorithm 5.

Algorithm 5 OCRS for matchings in $k-$partite hypergraphs

while a hyperedge e arrives **do**
  Calculate $P(A_e)$
  If the hyperedge e is revealed to be active and it is not blocked, take it with probability $\frac{1}{(k+1)P(A_e)}$
end while

Theorem 5.2.1. *Algorithm 5 is a* $1/(k+1)$*-selectable OCRS over matchings in* $k-$*partite hypergraphs.*

Proof. We prove this by using induction.
Base Case. Suppose e is the first hyperedge to arrive. Clearly e is never blocked. Thus $P(S_e) = \frac{x_e}{k+1}$
Induction Step. Suppose the statement holds for the all the hyperedges that have previously arrived, and the hyperedge $e = (v_1, v_2, \ldots, v_k)$ arrives next. Then:

$$P(\overline{A_e}) = P(\bigcup_{t\in\cup_i B(v_i)} S_t) \leq \sum_{t\in\cup_i B(v_i)} P(S_t)$$

$$= \frac{1}{k+1} \sum_{t \in \cup_i B(v_i)} x_t \leq \frac{1}{k+1} \sum_{i \in [k]} \sum_{t \in \delta(v_i)} x_t$$

In the second equality we used the induction hypothesis. Since $x \in P_{\mathscr{F}}$, we have $\sum\limits_{t \in \delta(v_i)} x_t \leq 1$. Therefore:

$$P(\overline{A_e}) \leq \frac{k}{k+1} \implies P(A_e) \geq \frac{1}{k+1}$$

Thus $\frac{1}{(k+1)P(A_e)}$ is a valid probability and we have:

$$P(S_e) = x_e P(A_e) \frac{1}{(k+1)P(A_e)} = \frac{x_e}{k+1}$$

Which completes the proof.

□

Chapter 6

Conclusion and future work

6.1 Conclusion

In this thesis, we begin by introducing the concept of online contention resolution schemes (OCRS). After that we describe the current state of the art results for various feasibility families, including a 0.25-selectable OCRS for matroids against the almighty adversary. We also discuss a purely existential proof for an optimal 0.5-selectable OCRS over matroids against the offline adversary. We then move on to presenting our optimal 0.5-selectable OCRS over matroids with rank 2. Although we prove the correctness of the algorithm against an online adversary, it is worth noting that our OCRS can work against an even stronger adversary. Then we propose an optimal 0.25 selectable OCRS over transversal matroids. Finally, we present a $\frac{1}{k+1}$-selectable OCRS over matchings in a $k-$partite hypergraph.

6.2 Future directions

An interesting direction for future work would be to design an optimal 0.5 selectable OCRS over any matroid. We already know that such an OCRS exists, but there is no explicit description for such an algorithm yet.

Another direction is to design an OCRS for matroids that is better than 0.25-selectable against the almighty adversary, or to find an upper-bound lower than the trivial 0.5 that we are already aware of.

Bibliography

[1] M. Adamczyk and M. Wlodarczyk. Random order contention resolution schemes. *CoRR*, abs/1804.02584, 2018. URL http://arxiv.org/abs/1804.02584. → pages 8, 9

[2] S. Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 512–521. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.90. → pages 8

[3] N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. Solving packing integer programs via randomized rounding with alterations. *Theory of Computing*, 8(24):533–565, 2012. ISSN 1557-2862. doi:10.4086/toc.2012.v008a024. → pages 3

[4] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Proceedings of the Twelfth Conference on Integer Programming and Combinatorial Optimization (IPCO) 2007*, 2007. → pages 3

[5] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40 (6):1740–1766, 2011. doi:10.1137/080733991. → pages 3

[6] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 575–584. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.60. → pages 3

[7] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *CoRR*, abs/1105.4593, 2011. → pages 1, 3, 9

[8] J. Edmonds and D. R. Fulkerson. Transversals and matroid partition constraints. *Journal of Research of the National Bureau of Standard*, 69B (3), 1965. URL https://nvlpubs.nist.gov/nistpubs/jres/69B/jresv69Bn3p147_A1b.pdf. → pages 2

[9] T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. Online stochastic max-weight matching: prophet inequality for vertex and edge arrival models. *CoRR*, abs/2002.09807, 2020. URL https://arxiv.org/abs/2002.09807. → pages 9

[10] M. Feldman, O. Svensson, and R. Zenklusen. Online contention resolution schemes. *CoRR*, abs/1508.00142, 2015. → pages 1, 4, 8

[11] H. Fu, Z. G. Tang, H. Wu, J. Wu, and Q. Zhang. Random order vertex arrival contention resolution schemes for matching, with applications. In N. Bansal, E. Merelli, and J. Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 68:1–68:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.68. URL https://doi.org/10.4230/LIPIcs.ICALP.2021.68. → pages 9

[12] N. Gravin and H. Wang. Prophet inequality for bipartite matching: merits of being simple and non adaptive. *CoRR*, abs/1902.06243, 2019. URL http://arxiv.org/abs/1902.06243. → pages 9

[13] R. Kleinberg and S. M. Weinberg. Matroid prophet inequalities. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 123–136. ACM, 2012. doi:10.1145/2213977.2213991. → pages 8

[14] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In C. Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 545–554. SIAM, 2009. URL http://dl.acm.org/citation.cfm?id=1496770.1496830. → pages 3

[15] E. Lee and S. Singla. Optimal Online Contention Resolution Schemes via Ex-Ante Prophet Inequalities. In *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. ISBN 978-3-95977-081-1. → pages 8, 9

[16] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM J. Discret. Math.*, 23(4):2053–2078, 2010. doi:10.1137/090750020. → pages 3

[17] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 67–74. ACM, 2008. → pages 3

[18] J. Vondrák. Symmetry and approximability of submodular maximization problems. *CoRR*, abs/1110.4860, 2011. URL http://arxiv.org/abs/1110.4860. → pages 3