CompSci 449T: Research Proposal

Extensions to the Online Pen Testing Problem

Harish Prabhu Shankar

1 Introduction

My honors thesis will be a theoretical computer science research manuscript. It will be centered on *online learning*, a problem setting where data becomes available in a sequential order and at any an irrevocable decision necessarily needs to be made. It can also be understood as an *optimal stopping problem.* The particular focus of my thesis is based on the framework developed in the Online Pen Testing[1] paper by my advisor Mingda Qiao and Greogry Valiant. The premise in the paper slightly alters the prototypical online learning problem by allowing a "testing" phase. Consider a sequence of pens, where we can test each pen and in doing so become aware of the fact that the pen either has more, less, or equal to the amount of ink we tested it with. However, in doing so the utility of the pen has diminished by the same amount we tested with. The paper limits itself to selecting a single pen out of a sequence of n. The aim of my thesis is to extend the framework in one of two ways: (1) select more than one pen, say k, and (2) the relation between the testing amount and utility reduction is not one-to-one.

The extension is motivated by the notion that observation and decision-making in the real world is rarely isolated to a singleton or one-to-one in terms of opportunity cost. In the first case, constraints are often combinatorial. For instance, a venture capital firm does not select a single startup to invest in; they select a portfolio of companies subject to budget constraints, sector constraints, and internal ruling procedures. Similarly, a hiring committee for a company tend to select a group of employees where the value of the team depends on the distinct skills of its members. Such constraints are modeled by Matroids. By extending the pen testing framework to Matroids, structures that generalize the notion of linear independence, my research seeks to provide algorithms that can handle "more combinatorial" scenarios, such as selecting at most k items from the sequence, while accounting for the cost of evaluation.

In the second case, the assumption of a one-to-one linear relationship between testing cost and utility loss is often too restrictive and not realized in the real world. For example, consider investing in a company. You may some period of time to observe their performance, however in that time the price of a share may have increased, and thus the return would decreases since we are buying in at a higher price. At the current moment, I am restricting myself to focusing on linear relationships were the coefficient $\alpha > 0$, since they are easy to work with. Nonetheless, in many stochastic processes, the cost of information acquisition may be non-linear. Generalizing the cost function allows the model to bridge the gap between the specific mechanics of pen testing and the broader literature on Difference Prophet Inequalities, where the goal is to maximize the difference between realized reward and paid cost.

The primary scientific challenge of this thesis lies in the dilemma between the feasibility of the selected data and the cost of verification. In standard online selection problems, such as the Matroid Secretary Problem, the algorithm observes a value upon arrival and must simply decide whether to include it in the independent set. In the combinatorial case for the Online Pen Testing problem, however, the algorithm is under tougher constraints— it must determine if the candidate is likely to improve the current independent set before knowing its true value, all while paying a cost to verify that intuition. If the algorithm tests with too high of a threshold, it "burns" the utility of the item; if it tests too conservatively, it fails to identify high-value candidates. In this instance of the problem, my objective would be to design an algorithm, likely a threshold-based one (possibility adaptive to the sequence), aiming to prove it achieves a competitive ratio (an approximation factor against an the optimal selection).

The significance of this research extends to the field of Algorithmic Game Theory and Mechanism Design. Recent literature has identified a duality between the pen testing problem and *Deferred-Acceptance Auctions*. Specifically, maximizing the residual ink in the pen testing domain is mathematically equivalent to maximizing consumer surplus in specific auction formats. By solving the combinatorial pen testing problem, this thesis contributes to the design of truthful market mechanisms where buyers face entry costs or exploration fees. Ultimately, this work aims to provide a rigorous theoretical foundation for resource allocation problems where the act of evaluation is inherently costly, moving the field toward more realistic models of sequential decision-making.

COMPSCI 449Y: Literature Review

Harish Prabhu Shankar

December 15, 2025

Contents

Chapter 1

Overview

This literature review is a “reworking”and re-implementation of problems in an online setting space, where we examine sequential decision-making, each one irrevocable in some sense. The aim is to ascertain enough information to guarantee some form of optimality. These problems arise naturally in resource allocation, auction design, and settings where decisions must be made with incomplete or asymmetric information.

1.1 Motivation

The problems studied in this review share a common structure: a decision-maker observes a sequence of items arriving one at a time, each with an associated value that may be initially unknown or partially revealed. The decision-maker must decide whether to accept or reject each item, with the goal of maximizing some objective function (typically utility maximization) subject to various constraints contingent on the problem space.

The motivating factor for the thesis is: Information Acquisition Costs. In many real-world scenarios, learning the true value of an item requires expending resources. For example, evaluating a candidate takes time, assessing a stock could lead to higher price, and testing a pen consumes its ink. There is a fine line to tread when gathering enough information to make good decisions and conserving resources for the items ultimately selected.

These problems find applications in online auctions, resource allocation, hiring decisions, and various other domains where sequential decision-making under uncertainty is fundamental.

1.2 Relationships Between Problems

The four main problems/papers covered in this review form a natural progression that aims to build foundational knowledge for the expected direction of my thesis.

1.2.1 Online Pen Testing

The Online Pen Testing problem serves as the grounding for my thesis as a whole. Here, a decision-maker must select a single item from a sequence, but learning an item’s value requires “testing” it with a threshold, which consumes part of the item’s value. This problem introduces the dilemma between exploration and exploitation: setting thresholds too high wastes value on testing, while setting them too low provides insufficient information.

The problem is studied in two main settings:

- **Prophet Setting**: Items’ values are randomly drawn from distributions (both known and unknown).
- **Secretary Setting**: Items’ values are fixed but arrive in random or adversarial order.

Key results show that competitive ratios of $O(\log n)$ are achievable.

1.2.2 Matroid Secretary Problem

The Matroid Secretary Problem generalizes the classic secretary problem by adding combinatorial constraints. Instead of selecting a single item, the algorithm must select an independent set in a matroid (e.g., at most k items). This problem foregoes the cost of testing but introduces the challenge of maintaining feasibility constraints.

The problem has been extensively studied with competitive ratios ranging from $O(\log\log r)$ for specific matroid classes to $O(\log r)$ for general matroids, where r denotes the rank of the matroid.

1.2.3 Combinatorial Pen Testing Problem

The Combinatorial Pen Testing Problem combines the constraints from both the previous papers: the cost of testing and also the combinatorial constraints.

They key insight from this work is the connection to auction theory: the problem of maximizing consumer surplus in deferred-acceptance auctions is equivalent to combinatorial pen testing. This connection allows leveraging results from mechanism design to obtain improved competitive ratios, achieving $O(\log n)$ approximations for general constraints and better bounds for specific problem settings.

1.2.4 Difference Prophet Inequalities

The Difference Prophet Inequalities goes back to the heart of the online pen testing problem, in that there is a cost for information acquisition. This problem generalizes upper bounds for a specific fixed cost function c and the number of values n.

1.3 Organization

This review is organized by problem type, with each chapter providing detailed technical results, algorithms, and analysis. Chapter 2 covers Online Pen Testing[1] in depth, including both prophet and secretary settings. Chapter 3 explores the Matroid Secretary Problem[2] and its variants. Chapter 4 presents the Combinatorial Pen Testing Problem[3] and its connections to auction theory. Finally, Chapter 5 examines Difference Prophet Inequalities[4] as a theoretical foundation for understanding sequential decision-making with costs.

Chapter 2

Online Pen Testing

2.1 Set-Up

Definition 2.1. *Online Pen Testing*: A instance of the problem is defined over $X_1, \dots, X_n \geq 0$. At each step $i \in [n]$, the player must perform two tests

1. Testing: The player selects a threshold $\theta_i \in [0, \infty)$. The player then tests X_i with the chosen θ_i. If $X_i \leq \theta_i$, then player simply observes the value of X_i. Otherwise, the pen has a remaining utility of $X_i' = X_i - \theta_i$

2. Accepting: If pen i passes the test, the player is given the choice to 'accept' or 'reject' the option irrevocably. If the player accepts the pen, the game ends and the player has a score of X_i'

Definition 2.2. *Prophet Setting*: The player is given information about the distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$ defined over $[0, \infty)$, where the values $X_1, \dots, X_n$ are drawn independently from.

The information given to the player is of two types

1. complete description of $(\mathcal{D}_i)_{i=1}^n$

2. a sample $\hat{\mathcal{X}}_i$ from each $\mathcal{D}_i$ such that the actual value X_i is independent of the sample provided

Definition 2.3. *Secretary Setting*: The player is given information about $a_1, \dots, a_n \geq 0$. The values $X_1, \dots, X_n$ will be a permutation of $a_1, \dots, a_n$, either uniformly random (where each a_i is equally likely to be found in any X_i) or arbitrarily (where the 'opponent' can chose where each a_i goes in X_i at their own discretion).

The information given to the player is of three types:

1. *Complete*: The player is provided $a_1, \dots, a_n$

2. *Optimum*: The player is given only $\max\{a_1, \dots, a_n\}$ which we will refer to as $a_{[1]}$

3. *Nothing*: The player is oblivious

2.2 Prophet i.i.d.

This is considered to be a unique case where $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are all the same distribution, which for the remainder of this section we will refer to as $\mathcal{D}$, and the player is given the complete description of it.

Using some properties omitted from this write-up (Remark 1.4 in the original paper [5]), for $\alpha \in (0, 1]$, we can define τ_α as the smallest $(1-\alpha)$-quantile— the minimal τ such that $\mathbb{P}_{\mathcal{X}\sim\mathcal{D}}[X > \tau] = \alpha$.

Upper-bounding the expected optimum $\mathbb{E}_{X_i\sim\mathcal{D}}[X^{\max}]$ for any distribution $\mathcal{D}$ and $\theta \in \mathbb{R}$

$$\begin{aligned} X^{\max} &= \theta + (X^{\max} - \theta) \\ \implies \underset{X_1,\ldots,X_n\sim\mathcal{D}}{\mathbb{E}}[X^{\max}] &= \theta + \underset{X_1,\ldots,X_n\sim\mathcal{D}}{\mathbb{E}}[X^{\max} - \theta] && \text{(linearity of expectation)} \\ &\leq \theta + \mathbb{E}\left[\max\left\{X^{\max} - \theta, 0\right\}\right]. && \text{(since } a \leq \max(a,0) \text{ for all } a\text{)} \end{aligned}$$

We can further break down the max by,

$$\begin{aligned} \max\left(X^{\max} - \theta, 0\right) &= \max\left(\max_{i\in[n]} X_i - \theta, 0\right) \\ &= \max_{i\in[n]}\left(\max\left(X_i - \theta, 0\right)\right) \\ &\leq \sum_{i=1}^{n} \max\left(X_i - \theta, 0\right). \end{aligned}$$

Thus, we get,

$$\begin{aligned} \underset{X_1,\ldots,X_n\sim\mathcal{D}}{\mathbb{E}}[X^{\max}] &\leq \theta + \underset{X_1,\ldots,X_n\sim\mathcal{D}}{\mathbb{E}}\left[\sum_{i=1}^{n}\max\left(X_i - \theta, 0\right)\right] \\ &= \theta + \sum_{i=1}^{n}\left(\underset{X_1,\ldots,X_n\sim\mathcal{D}}{\mathbb{E}}\left[\max\left(X_i - \theta, 0\right)\right]\right) \\ &\leq \theta + n \cdot \underset{X\sim\mathcal{D}}{\mathbb{E}}\left[\max\left(X - \theta, 0\right)\right]. \end{aligned}$$

Lemma 2.4. *For any distribution $\mathcal{D}$ and $\alpha \in (0, 1]$*

$$\underset{X\sim\mathcal{D}}{\mathbb{E}}\left[X - \tau_\alpha \mid X > \tau_\alpha\right] = \frac{\mathbb{E}_{X\sim\mathcal{D}}\left[(X - \tau_\alpha)\cdot \mathbb{1}\left[X > \tau_\alpha\right]\right]}{\mathbb{P}\left(X > \tau_\alpha\right)}$$

Using the fact that $(x-b)\cdot\mathbb{1}(x \geq b) \geq (a-b)\cdot\mathbb{1}(x \geq a)$ for any x and any $a > b$, we can see that $\tau_{\alpha/2} \geq \tau_\alpha$. Now checking if the value of the indicator changes we substituting τ_α for $\tau_{\alpha/2}$,

- $X < \tau_\alpha \leq \tau_{\alpha/2} \implies \mathbb{1}\left[X > \tau_\alpha\right] = 0 \wedge \mathbb{1}\left[X > \tau_{\alpha/2}\right] = 0 \implies 0_\alpha \geq 0_{\alpha/2}$
- $\tau_\alpha \leq X \leq \tau_{\alpha/2} \implies \mathbb{1}\left[X > \tau_\alpha\right] = 1 \wedge \mathbb{1}\left[X > \tau_{\alpha/2}\right] = 0 \implies 1_\alpha \geq 0_{\alpha/2}$
- $X \geq \tau_{\alpha/2} \geq \tau_\alpha \implies \mathbb{1}\left[X > \tau_\alpha\right] = 1 \wedge \mathbb{1}\left[X > \tau_{\alpha/2}\right] = 1 \implies 1_\alpha \geq 1_{\alpha/2}$

As a result,

$$\frac{\mathbb{E}_{X\sim\mathcal{D}}\left[(X - \tau_\alpha)\cdot\mathbb{1}\left[X > \tau_\alpha\right]\right]}{\mathbb{P}\left(X > \tau_\alpha\right)} \geq \frac{\mathbb{E}_{X\sim\mathcal{D}}\left[\left(\tau_{\alpha/2} - \tau_\alpha\right)\cdot\mathbb{1}\left[X > \tau_{\alpha/2}\right]\right]}{\mathbb{P}\left(X > \tau_\alpha\right)}$$

Combining the fact that $\tau_{\alpha/2} - \tau_\alpha$ is a constant and $\mathbb{E}\left[\mathbb{1}\left[\text{inequality}\right]\right] = \mathbb{P}\left(\text{inequality}\right)$,

$$\begin{aligned}\frac{\mathbb{E}_{X\sim\mathcal{D}}\left[\left(\tau_{\alpha/2} - \tau_\alpha\right)\cdot \mathbb{1}\left[X > \tau_{\alpha/2}\right]\right]}{\mathbb{P}\left(X > \tau_\alpha\right)} &= \left(\tau_{\alpha/2} - \tau_\alpha\right)\cdot \frac{\mathbb{P}\left(X > \tau_{\alpha/2}\right)}{\mathbb{P}\left(X > \tau_\alpha\right)} \\ &= \left(\tau_{\alpha/2} - \tau_\alpha\right)\cdot \frac{\frac{\alpha}{2}}{\alpha} \\ &= \frac{1}{2}\left(\tau_{\alpha/2} - \tau_\alpha\right).\end{aligned}$$

2.2.1 First Algorithm

Consider a single-threshold algorithm that tests every pen with the same threshold θ, and accepts the first pen that passes the test. We set it to $\theta = \tau_{1/n}$. By definition, we know $\mathbb{P}(X > \tau_{1/n}) = \frac{1}{n}$, which informs us that the probability that a pen doesn't pass the test is $1 - \frac{1}{n}$. So, the probability that every single pen fails is $\left(1 - \frac{1}{n}\right)^n$. Therefore, the probability that at least one pen passes is $1 - \left(1 - \frac{1}{n}\right)^n$. Taking the limit as $n \to \infty$, we know is $\frac{1}{e}$. Hence, we can claim

$$\begin{aligned}\left(1 - \frac{1}{n}\right)^n &< \frac{1}{e} \\ -\left(1 - \frac{1}{n}\right)^n &> -\frac{1}{e} \\ 1 - \left(1 - \frac{1}{n}\right)^n &> 1 - \frac{1}{e}.\end{aligned}$$

Now, conditioning on the fact the algorithm accepts the pen, the expected utility is,

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[X - \tau_{1/n} \mid X > \tau_{1/n}\right] &= \frac{\mathbb{E}_{X\sim\mathcal{D}}\left[(X - \tau_{1/n})\cdot \mathbb{1}\left[X > \tau_{1/n}\right]\right]}{\mathbb{P}(X > \tau_{1/n})} \\ &= n\cdot \mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[(X - \tau_{1/n})\cdot \mathbb{1}\left[X > \tau_{1/n}\right]\right]\end{aligned}$$

The term inside the expectation is basically the utility of the pen after testing with the threshold, given that $X > \tau_{1/n}$ otherwise it is 0. Hence, we can replace the term with,

$$\mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[\max\left(X - \tau_{1/n}, 0\right)\right]$$

Thus, we have,

$$\mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[X - \tau_{1/n} \mid X > \tau_{1/n}\right] = n\cdot \mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[\max\left(X - \tau_{1/n}, 0\right)\right]$$

Hence, the expected score is,

$$1 - \left(1 - \frac{1}{n}\right)^n \cdot n\cdot \mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[\max\left(X - \tau_{1/n}, 0\right)\right] > \left(1 - \frac{1}{e}\right)\cdot n\cdot \mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[\max\left(X - \tau_{1/n}, 0\right)\right]$$

2.2.2 Second Algorithm

Now, set $k = \lceil \log_2 n \rceil$. Then, draw α uniformly at random from $\left\{2^0, 2^{-1}, \ldots, 2^{-(k-1)}\right\}$ and based on that set the threshold to $\theta = \tau_\alpha$. Each α has a probability of $\frac{1}{k}$ of being selected. Using a similar line of reasoning as in the previous section, we end up the identical result of the probability that at least one pen passes conditioned on the choice of α is

$1-(1-\frac{1}{n})^n > 1-\frac{1}{e}$. Additionally, conditioning on one of the options is accepted, the expected score is at least $\frac{\tau_{\alpha/2}-\tau_\alpha}{2}$. Hence, this algorithm has an expected score of at least

$$\begin{aligned}\frac{1}{k}\cdot\left(1-\frac{1}{e}\right)\cdot\sum_{i=0}^{k-1}\left(\frac{\tau_{2^{-(i+1)}}-\tau_{2^{-i}}}{2}\right) &= \frac{1}{2k}\cdot\left(1-\frac{1}{e}\right)\cdot\left(\tau_{2^{-k}}-\tau_1\right)\\ &\geq \frac{\tau_{1/n}}{O(\log n)}.\end{aligned}$$

We move from the equality to the inequality by the fact that $2^k \approx n \implies 2^{-k} = \frac{1}{n}$ as well as $k = O(\log n)$ and also τ_1 is the 0th-percentile which is likely to be 0. We can obtain an $O(\log n)$ competitive algorithm by randomizing between the two algorithms.

2.3 Prophet General

Given that $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are no longer identical, for every $\alpha \in (0,1]$ we define $\tau_\alpha^{(i)}$ to be the smallest $(1-\alpha)$-quantile of $\mathcal{D}_i$. Let $X_1, \ldots, X_n$ be independently drawn samples from their corresponding distributions and that we define $X^{\max} = \max_{i\in[n]} X_i$. As a result, we can set $\tau_\alpha^{\max}$ as the smallest $(1-\alpha)$-quantile of $X^{\max}$.

We will define a single-threshold problem where we set $\theta = \tau_{1/2}$. Since $\tau_{1/2}$ is the median of the maximum which can be interpreted as some v such that 50% that the best pen out of all n pens will have more ink than the amount v.

Borrowing a lot of the logic from the prophet i.i.d case, namely the first algorithm, we can end up with the following set of inequalities

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[X^{\max}\right] &\leq \tau_{1/2} + \mathop{\mathbb{E}}_{X\sim\mathcal{D}}\left[\max\left\{X^{\max}-\tau_{1/2}, 0\right\}\right]\\ &\leq \tau_{1/2} + \sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{X_i\sim\mathcal{D}_i}\left[\max\left\{X_i-\tau_{1/2},0\right\}\right]\right).\end{aligned}$$

Let $\alpha_i = \mathbb{P}_{X_i\sim\mathcal{D}_i}\left[X_i > \tau_{1/2}\right]$ and using the definition of $\tau_{1/2}$, we have,

$$\begin{aligned}\frac{1}{2} &= \mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max}\leq\tau_{1/2}\right)\\ &= \prod_{i=1}^{n}\mathbb{P}_{X_i\sim\mathcal{D}_i}\left(X_i\leq\tau_{1/2}\right)\\ &= \prod_{i=1}^{n}\left(1-\alpha_i\right).\end{aligned}$$

Expanding the product,

$$\prod(1-\alpha_i) = 1-\sum\alpha_i+\sum_{i<j}\alpha_i\alpha_j-\sum_{i<j<k}\alpha_i\alpha_j\alpha_k+\ldots+(-1)^n\alpha_i\cdot\ldots\cdot\alpha_n.$$

By the definition of $\alpha_i \in (0,1]$ we know that every product within the sum is non-negative and that $\sum\alpha_i\alpha_j$ is larger than any following term and label that entire sum as c such that,

$$1-\sum\alpha_i+c \geq 1-\sum\alpha_i.$$

Therefore,

$$\frac{1}{2} \geq 1 - \sum_{i=1}^{n} \alpha_i$$
$$\implies \sum_{i=1}^{n} \alpha_i \geq \frac{1}{2}.$$

This basically informs us that the sum of the probability of all the pens that matter must be at least $\frac{1}{2}$. Now, setting $k = \lceil \log_2 n \rceil$ (which 'guarantees' a probability down to $\frac{1}{n}$) and partition the n distributions into $k+3$ distinct groups depending on α_i. For $j = 0, 1, \ldots, k+1$,

$$G_j = \left\{ i \in [n] \mid 2^{-(j+1)} < \alpha_i \leq 2^{-j} \right\}.$$

Working out some examples,

- G_0 is the set of pens of values $\frac{1}{2} < \alpha_i \leq 1$ which we will consider the best pens
- G_1 is the set of pens of values $\frac{1}{4} < \alpha_i \leq \frac{1}{2}$

We define one last 'bucket' that we will consider the 'leftover bucket'

$$G_{k+2} = \left\{ i \in [n] \mid \alpha_i \leq 2^{-(k+2)} \right\}.$$

Now, that we partitioned the pens into different buckets we can now iterate through them by their bucket-index first and then the actual option i inside the bucket, therefore,

$$\frac{1}{2} \leq \sum_{i=1}^{n} \alpha_i$$
$$= \sum_{j=0}^{k+1} \sum_{i \in G_j} \alpha_i + \sum_{i \in G_{k+2}} \alpha_i.$$

We know that for any bucket j that the pens inside are upperbounded by 2^{-j}; hence, we can upperbound the sum to obtain $\sum_{i \in G_j} \alpha_i \leq \frac{|G_j|}{2^{-j}}$. Since $k = \lceil \log_2 n \rceil$, we have $2^k \geq n \implies 2^{-(k+2)} \leq \frac{1}{4n}$. So, the second summation becomes $\frac{|G_{k+2}|}{4n}$. Continuing our previous inequality,

$$\sum_{j=0}^{k+1} \sum_{i \in G_j} \alpha_i + \sum_{i \in G_{k+2}} \alpha_i \leq \sum_{j=0}^{k+1} \frac{|G_j|}{2^{-j}} + \frac{|G_{k+2}|}{4n}.$$

The size of $|G_{k+2}|$ can be at most n, if every pen is found in this bucket, thus,

$$\sum_{j=0}^{k+1} \frac{|G_j|}{2^{-j}} + \frac{|G_{k+2}|}{4n} \leq \sum_{j=0}^{k+1} \frac{|G_j|}{2^{-j}} + \frac{1}{4}$$
$$\implies \frac{1}{2} \leq \sum_{j=0}^{k+1} \frac{|G_j|}{2^{-j}} + \frac{1}{4}$$
$$\implies \frac{1}{4} \leq \sum_{j=0}^{k+1} \frac{|G_j|}{2^{-j}}.$$

Using the pigeonhole principle, the average of the $k+2$ terms is at least $\frac{1}{4(k+2)}$. Therefore, at least one term must be at least the average and as such there must exists some $j^* \in \{0, \ldots, k+1\}$ so,

$$\frac{|G_{j^*}|}{2^{j^*}} \geq \frac{1}{4(k+2)} = \frac{1}{O(\log n)}.$$

2.3.1 First Algorithm

We define the following algorithm

- Draw α from some distribution over $\{1, 2^{-1}, \dots, 2^{-j^*}\}$ that will defined later
- Partition G_{j^*} into $t = \left\lceil \frac{|G_{j^*}|}{\frac{1}{\alpha}} \right\rceil$ blocks, called $B_1, \dots, B_t$
  1. Each block is of size $\frac{1}{\alpha}$ (this guarantees that at least one passing pen is found in the block)
  2. The blocks are sorted chronologically i.e. $\max B_i < \min B_{i+1}$ for every $i \in [t-1]$
- Pick B_j for $j \in [t]$, uniformly at random
- At each step i (pen i), don't test and reject it if $i \notin B_j$. Else, test with a threshold $\theta_i = \tau_\alpha^{(i)}$ (dependent on the pen's distribution) and accept pen i if it passes

Assume that $i \in G_{j^*}$ is the ι-th smallest number in its block B_j $\left(\text{note that } \iota \in \left[\frac{1}{\alpha}\right]\right)$. For i to be accepted three things must happen in the following order

i. B_j is chosen in the third step of the algorithm, which has a probability of $\frac{1}{t}$

ii. None of the previous $\iota - 1$ options passed the test, which has a probability of $(1-\alpha)^{\iota-1}$

iii. Pen ι passes the test, which has a probability of α

Combining this together, we have

$$\mathbb{P}(\text{i}) \cdot \mathbb{P}(\text{ii}) \cdot \mathbb{P}(\text{iii}) = \frac{1}{t} \cdot (1-\alpha)^{\iota-1} \cdot \alpha.$$

Using the definition of t and using the fact that $\lceil x \rceil < x + 1$,

$$\begin{aligned} t &= \left\lceil \frac{|G_{j^*}|}{\frac{1}{\alpha}} \right\rceil \\ &\geq \frac{|G_{j^*}|}{\frac{1}{\alpha}} + 1. \end{aligned}$$

Also we can upperbound $\iota \leq \frac{1}{\alpha}$. Going back to the inequality and introducing these,

$$\frac{1}{t} \cdot (1-\alpha)^{\iota-1} \cdot \alpha \geq \frac{1}{\frac{|G_{j^*}|}{\frac{1}{\alpha}} + 1} \cdot (1-\alpha)^{\frac{1}{\alpha}-1} \cdot \alpha.$$

Working out the second term in the product above,

$$\begin{aligned} (1-\alpha)^{\frac{1}{\alpha}-1} &= (1-\alpha)^{\frac{1}{\alpha}} \cdot (1-\alpha)^{-1} \\ &= (1+x)^{-\frac{1}{x}} && \text{(substituting } x = -\alpha) \\ \implies (1-\alpha)^{\frac{1}{\alpha}-1} &\geq \frac{1}{e}. \end{aligned}$$

we obtain the list line by the fact that $(1+x)^{\frac{1}{x}} \to e$ as $x \to 0 \implies \alpha \to 0$. Substituting this into the inequality,

$$\frac{1}{\frac{|G_{j^*}|}{\frac{1}{\alpha}} + 1} \cdot (1-\alpha)^{\frac{1}{\alpha}-1} \cdot \alpha \geq \frac{\alpha}{e\left(\alpha |G_{j^*}| + 1\right)}.$$

Using the result from Lemma 2.4, conditioning on that option i passes the test at $\theta_i = \tau_\alpha^{(i)}$, the expected score is at least

$$\frac{\tau_{\alpha/2}^{(i)} - \tau_\alpha^{(i)}}{2}.$$

Finally, conditioning on the choice of α, we have,

$$\frac{\alpha}{2e\left(\alpha|G_{j^*}|+1\right)} \cdot \sum_{i \in G_{j^*}} \left(\tau_{\alpha/2}^{(i)} - \tau_\alpha^{(i)}\right).$$

Remark 2.5. Examining the term $\alpha|G_{j^*}|$ in the denominator, consider the case where $\alpha|G_{j^*}| \leq 1$. Then, $2\left(\alpha|G_{j^*}|+1\right) \leq 4$, which implies $\frac{\alpha}{2e\left(\alpha|G_{j^*}|+1\right)} \geq \frac{\alpha}{4e}$. The RHS is equivalent to $\frac{\alpha|G_{j^*}|}{4e|G_{j^*}|}$. We can then substitute the numerator with $\min\{\alpha|G_{j^*}|, 1\} = \alpha|G_{j^*}|$. Alternatively, if $\alpha|G_{j^*}| > 1$, then $\alpha|G_{j^*}| + 1 \leq 2\alpha|G_{j^*}|$. This implies $\frac{\alpha}{2e\left(\alpha|G_{j^*}|+1\right)} \geq \frac{\alpha}{4e\left(\alpha|G_{j^*}|\right)}$. The RHS is equivalent to $\frac{1}{4e|G_{j^*}|}$. We can then substitute the numerator with $\min\{\alpha|G_{j^*}|, 1\} = 1$.

Using Remark 2.5, the expected score is lower bounded by

$$\frac{\alpha}{2e\left(\alpha|G_{j^*}|+1\right)} \cdot \sum_{i \in G_{j^*}} \left(\tau_{\alpha/2}^{(i)} - \tau_\alpha^{(i)}\right) \geq \frac{\min\{\alpha|G_{j^*}|, 1\}}{4e|G_{j^*}|} \cdot \sum_{i \in G_{j^*}} \left(\tau_{\alpha/2}^{(i)} - \tau_\alpha^{(i)}\right).$$

Now, we need to pick randomly over the set $\{1, 2^{-1}, \ldots, 2^{-j^*}\}$. Looking at the previous result where $2^{-j^*}|G_{j^*}| \geq \frac{1}{O(\log n)}$. So, by setting

$$Z = \sum_{j=0}^{j^*} \left(\min\left\{2^{-j^*}|G_{j^*}|, 1\right\}\right),$$

which is our normalization constant— then by the first result, $Z = O(\log n)$. This is the case as j^* is at most $k+1$. The min serves to cap off at 1 so that if $2^{-j^*}|G_{j^*}| < 1$, then j is large so expected success is less than 1. Otherwise, if $2^{-j^*}|G_{j^*}| > 1$, then j is small so expected success is greater than 1.

Setting α to 2^{-j} with probability $\frac{1/\min\left\{2^{-j^*}|G_{j^*}|,1\right\}}{Z}$ for each $j \in [j^*]$. Hence, the expected score becomes the probability that $\alpha = 2^{-j^*}$ and the score designated to such an α. So,

$$\begin{aligned}
&\geq \sum_{j=0}^{j^*} \left(\frac{1}{Z} \cdot \frac{1}{\min\{2^{-j^*}|G_{j^*}|, 1\}}\right) \cdot \left(\frac{\min\{2^{-j^*}|G_{j^*}|, 1\}}{4e|G_{j^*}|} \cdot \sum_{i \in G_{j^*}} \left(\tau_{2^{-(j+1)}}^{(i)} - \tau_{2^{-j}}^{(i)}\right)\right) \\
&= \frac{1}{4Ze|G_{j^*}|} \cdot \sum_{j=0}^{j^*} \sum_{i \in G_{j^*}} \left(\tau_{2^{-(j+1)}}^{(i)} - \tau_{2^{-j}}^{(i)}\right) \\
&= \frac{1}{4Ze|G_{j^*}|} \cdot \sum_{i \in G_{j^*}} \sum_{j=0}^{j^*} \left(\tau_{2^{-(j+1)}}^{(i)} - \tau_{2^{-j}}^{(i)}\right) \\
&\geq \frac{1}{4Ze|G_{j^*}|} \cdot \sum_{i \in G_{j^*}} \tau_{2^{-(j^*+1)}}^{(i)}.
\end{aligned}$$

So, $\tau_{2^{-(j^*+1)}}^{(i)}$ is the threshold we need to set to make the tail probability exactly $\tau_{2^{-(j^*+1)}}$. Since the probability of X_i exceeding $\tau_{1/2}$ is greater than $\tau_{2^{-(j^*+1)}}$, we must move to a higher threshold to reduce the probability down to $\tau_{2^{-(j^*+1)}}$. Therefore, for every $i \in G_{j^*}$, it must be that: $\tau^{(i)}_{2^{-(j^*+1)}} > \tau_{1/2}$. If every single term in the sum $\sum \tau$ is greater than $\tau_{1/2}$, then their average, $\frac{1}{|G_{j^*}|} \sum \tau$, must also be greater than $\tau_{1/2}$. This concludes with,

$$\begin{aligned}\frac{1}{4Ze|G_{j^*}|} \cdot \sum_{i \in G_{j^*}} \tau^{(i)}_{2^{-(j^*+1)}} &> \frac{\tau_{1/2}}{4Ze} \\ &= \frac{\tau_{1/2}}{O(\log n)}.\end{aligned}$$

2.3.2 Second Algorithm

Reusing the single-threshold by setting $\theta = \tau_{1/2}$, we can split it into three parts

1. Every pen up to option i was rejected: $\prod_{j=1}^{i-1} \mathbb{P}_{X_i \sim \mathcal{D}_i} \left(X_i \leq \tau_{1/2} \right)$
2. Pen i is accepted: $\mathbb{P}_{X_i \sim \mathcal{D}_i} \left(X_i > \tau_{1/2} \right)$
3. The expected utility given that pen i was accepted: $\mathbb{E}_{X_i \sim \mathcal{D}_i} \left[X_i - \tau_{1/2} \mid X_i > \tau_{1/2} \right]$

So, we have an expected score of

$$\sum_{i=1}^{n} \left(\left(\prod_{j=1}^{i-1} \mathbb{P}_{X_i \sim \mathcal{D}_i} \left(X_i \leq \tau_{1/2} \right) \right) \cdot \left(\mathbb{P}_{X_i \sim \mathcal{D}_i} \left(X_i > \tau_{1/2} \right) \right) \cdot \left(\mathop{\mathbb{E}}_{X_i \sim \mathcal{D}_i} \left[X_i - \tau_{1/2} \mid X_i > \tau_{1/2} \right] \right) \right).$$

For every $i \in \{1, \ldots, n\}$, the $\prod \mathbb{P}$ term is lower bounded by $\mathbb{P}_{X \sim \mathcal{D}} \left(X^{\max} \leq \tau_{1/2} \right) = \frac{1}{2}$. Let A be the event we care about, namely $(X_i \leq \tau_{1/2}) \cdot \ldots \cdot (X_{i-1} < \tau_{1/2})$ and B be the same from i to n. Combine them and all n pens less than $\tau_{1/2}$ which is the event $X^{\max} \leq \tau_{1/2}$ and $\mathbb{P}_{X \sim \mathcal{D}} \left(X^{\max} \leq \tau_{1/2} \right) = \frac{1}{2}$. Thus,

$$\mathbb{P}(A \cap B) \leq \mathbb{P}(A) \implies \frac{1}{2} \leq \prod_{j=1}^{i-1} \mathbb{P}_{X_i \sim \mathcal{D}_i} \left(X_i \leq \tau_{1/2} \right)$$

Dealing with the rest of the expected score,

$$\begin{aligned}\left(\mathbb{P}_{X_i \sim \mathcal{D}_i} \left(X_i > \tau_{1/2} \right) \right) \cdot \left(\mathop{\mathbb{E}}_{X_i \sim \mathcal{D}_i} \left[X_i - \tau_{1/2} \mid X_i > \tau_{1/2} \right] \right) &= \mathop{\mathbb{E}}_{X_i \sim \mathcal{D}_i} \left[\left(X_i - \tau_{1/2} \right) \cdot \mathbb{1} \left(X_i > \tau_{1/2} \right) \right] \\ &= \mathop{\mathbb{E}}_{X_i \sim \mathcal{D}_i} \left[\max \left\{ X_i - \tau_{1/2}, 0 \right\} \right].\end{aligned}$$

Hence, the expected score is lower bounded by,

$$\frac{1}{2} \cdot \sum_{i=1}^{n} \left(\mathop{\mathbb{E}}_{X_i \sim \mathcal{D}_i} \left[\max \left\{ X_i - \tau_{1/2}, 0 \right\} \right] \right).$$

Lastly, randomizing between the two algorithms results in an $O(\log n)$ competitive ratio.

2.4 Secretary Setting Upper-Bounds

Set $k = \lfloor \log_2 n \rfloor + 2 \implies k = O(\log n)$ such that $2^{k-1} > n$. We will use the notation $a_{[i]}$ to be the i-th largest pen. For each $j \in [k]$, define

$$n_j = \left| \left\{ i \in [n] \mid \frac{j-1}{k} a_{[1]} < a_i \leq \frac{j}{k} a_{[1]} \right\} \right|$$

as the number of pens whose value is approximately a $\frac{j}{k}$ fraction of the maximum. It should be clear that $\sum n_j \leq n$ as well as $n_k \geq 1$. We will also define the notation $n_{\geq j} = n_j + \ldots + n_k$. There exists some $j^* \in [k-1]$ such that $n_{j^*} < n_{j^*+1}$, which informs us there are more pens in the 'buckets' of pens whose value is at least $\frac{j^*+1}{k}$ of the maximum than there are pens in the bucket value whose value is at least $\frac{j^*}{k}$ of the maximum.

Assume for the sake of contradiction that for all $j \in [k-1]$ that $n_j \geq n_{j+1}$. For $j = k$, we consider $n_{\geq k}$, $n_{\geq k} = n_k$, since $a_{[1]}$ is included in the set counted by n_k, we know that $n_k \geq 1$. Since $2^{k-k} = 2^0 = 1$, we have: $n_{\geq k} \geq 1 = 2^{k-k}$. The base case holds. Assume the claim holds for $j+1$, where $1 \leq j \leq k-1$.

Assuming the definition of $n_{\geq j}$ and the assumption for contradiction: $n_{\geq j} = n_j + n_{\geq j+1}$. By the assumption for contradiction, $n_j \geq n_{\geq j+1}$, so we substitute $n_{\geq j+1}$ for n_j: $n_{\geq j} \geq n_{\geq j+1} + n_{\geq j+1} = 2 \cdot n_{\geq j+1}$. Applying the IH:

$$\begin{aligned}
n_{\geq j} &\geq 2 \cdot \left(2^{k-(j+1)}\right) \\
n_{\geq j} &\geq 2^1 \cdot 2^{k-j-1} \\
n_{\geq j} &\geq 2^{1+k-j-1} \\
n_{\geq j} &\geq 2^{k-j}.
\end{aligned}$$

Since $n_{\geq 1} = \sum_{j=1}^{k} n_j$ is the total number of elements n, we have,

$$n = n_{\geq 1} \geq 2^{k-1}$$

However, since $k = \lfloor \log_2 n \rfloor + 2$, which implies that $2^{k-1} > n$. Therefore,

$$n \geq 2^{k-1} > n$$

$$n > n$$

Since the initial assumption leads to a contradiction, it must be false. Thus, there must exist some $j^* \in \{1, 2, \ldots, k-1\}$ such that $n_{j^*} < n_{\geq j^*+1}$.

2.4.1 Random Order with Full Information

Given $a_1, \ldots, a_n$, we can easily identify such an index j^*. We can set the single-threshold algorithm at $\theta = \frac{j^*-1}{k} a_{[1]}$. By the fact that $n_{j^*} < n_{j^*+1}$ implies among all options X_i that pass, a constant fraction have values $\geq \frac{j^*}{k} a_{[1]}$. Given the values are shuffled uniformly at random, any pen that passes has equal probability to be the first to pass. Given that a constant fraction are good i.e. $\geq \frac{j^*}{k} a_{[1]}$ there is a constant probability that the first pen to pass will be good. Hence, there is at least a probability of $\frac{1}{2}$ the remaining utility of the option selected is at least

$$\frac{j^*}{k} a_{[1]} - \theta = \frac{j^*}{k} a_{[1]} - \frac{j^*-1}{k} a_{[1]} = \frac{a_{[1]}}{k}.$$

So, the expected utility is lower bounded by,

$$\frac{1}{2} \cdot \frac{a_{[1]}}{k} = \Omega\left(\frac{a_{[1]}}{\log n}\right).$$

2.4.2 Random Order with Optimum Information

We draw $j \in [k-1]$ uniformly at random and also set the single-threshold to $\theta = \frac{j-1}{k} a_{[1]}$. Conditioning on j, we accept any of the $n_{\geq j}$ options with a value greater than θ uniformly at random. So, the utility of the selected pen is at least $\frac{a_{[1]}}{k}$ with a probability of $\frac{n_{\geq j+1}}{n_{\geq j}}$. Averaging over the randomness in j lower bounds the expected score by

$$\begin{aligned}\frac{a_{[1]}}{k} \cdot \frac{1}{k-1} \cdot \sum_{i=1}^{k-1} \frac{n_{\geq j+1}}{n_{\geq j}} &\geq \frac{a_{[1]}}{k} \cdot \left(\prod_{i=1}^{k-1} \frac{n_{\geq j+1}}{n_{\geq j}} \right)^{\frac{1}{k-1}} && \text{(AM-GM inequality)} \\ &= \frac{a_{[1]}}{k} \cdot \left(\frac{n_{\geq j+1}}{n_{\geq 1}} \right)^{\frac{1}{k-1}}.\end{aligned}$$

We can lower bound this further by the fact that necessarily $n_k > 1$ and $n_1 \leq n$, so,

$$\frac{a_{[1]}}{k} \cdot \left(\frac{n_{\geq j+1}}{n_{\geq 1}} \right)^{\frac{1}{k-1}} \geq \frac{a_{[1]}}{k} \cdot \left(\frac{1}{n} \right)^{\frac{1}{k-1}}.$$

By our definition of k, we know $2^{k-1} > n$, hence,

$$\begin{aligned} 2^{k-1} &> n \\ \implies \left(2^{k-1}\right)^{\frac{1}{k-1}} &> n^{\frac{1}{k-1}} \\ \implies \frac{1}{2} &< n^{-\frac{1}{k-1}}.\end{aligned}$$

Thus, our expected score is,

$$\frac{a_{[1]}}{k} \cdot (n)^{-\frac{1}{k-1}} > \frac{a_{[1]}}{k} \cdot \frac{1}{2} = \frac{1}{O(\log n)}.$$

2.4.3 Random Order with No Information

Observe the first $m = \left\lfloor \frac{n}{2} \right\rfloor$ pens and let $\hat{a_{[1]}}$ be the largest value observed in the first m pens. The likelihood of the event where "$a_{[2]}$ appears in the first m pens" (call this E_1) and "$a_{[1]}$ appears in the remaining $n-m$ pens" (call this E_2) is

$$\mathbb{P}(E_1) = \frac{m}{n-1} \approx \frac{\frac{n}{2}}{n}, \mathbb{P}(E_2) = \frac{n-m}{n} \approx \frac{\frac{n}{2}}{n}.$$

Thus, the probability this happens is,

$$\frac{m}{n-1} \cdot \frac{n-m}{n} \approx \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$

First Algorithm

We can use the same algorithm as we did in Section 2.4.2 but now the maximal value is the one we find in the first m pens, $\hat{a_{[1]}}$. Hence, our expected score is $\frac{\hat{a_{[1]}}}{O(\log n)}$.

Second Algorithm

We can also test the remaining $n-m$ options with the found threshold $\theta = \hat{a_{[1]}}$ and accept the first pen that passes the test. The pen that passes either has a value of $a_1 \hat{a_{[1]}} = a_2$ or $a_1 > \hat{a_{[1]}} = a_2$. Regardless, the utility is exactly $a_{[1]} - a_{[2]}$.

Randomizing between the two algorithms, we get an expected score of

$$\frac{1}{4} \cdot \frac{1}{2} \left(\frac{\hat{a_{[1]}}}{O(\log n)} + \left(a_{[1]} - a_{[2]}\right) \right) = \frac{a_{[1]}}{O(\log n)}.$$

2.4.4 Arbitrary Order with Optimum Information

We will make a distinction in the pens that do pass our test given a threshold of θ. We will classify them as either "good" when their utility after testing is at least $\frac{a_{[1]}}{k}$ which implies that $a_i \geq \theta + \frac{a_{[1]}}{k}$. Consequently, we define "bad" pens as those whose value falls in the range $\left(\theta, \theta + \frac{j}{k}a_{[1]}\right)$.

Bit Sampling

An adversary decides some valid binary sequence of some possible length $m \in [n]$ where the only where the validity is contingent on the fact that the number of ones must be more than $\frac{m}{2}$. The player is aware of the maximal possible length of the sequence n but not the actual length m. The goal of the player is at any time stamp to 'commit' to the next unseen bit. If the bit is 1, then the player wins; otherwise, if the bit is 0 or if the sequence ends then the player loses.

We define a strategy to improve the probability of winning for the player to a constant chance. We will keep a running counter Δ which is the difference of 0's and 1's. The two important scenarios are when $\Delta > 0$ which informs the player that they are "due" for a 1 and when $\Delta = 0$ which might tempt the adversary to make the next bit in the sequence 1 so that they can end the game.

At each step, before seeing the bit, the player commits with a probability of $2^{-(\Delta+2)}$, otherwise the player just observes the bit and updates Δ. Assume that $n - t$ bits have been observed and that $\Delta \geq 0$, the player wins with probability at least $\frac{1}{3}\left(1 - 2^{-(\Delta+1)}\right)$. Specifying $t = n$ and $\Delta = 0$ implies that the probability is at least $\frac{1}{6}$.

Consider the base case of $t = 0$, which tells n bits have been observed. However, $\Delta \geq 0$ which informs us that there have been more 0's than 1's and thus the case can be ignored as it is not valid. Now, assume that the claim holds true for $t - 1$.

Given that $\Delta \geq 0$ and that sequence is not complete, necessarily there must be a some unseen next bit in the sequence. Examining the scenario where the bit is 1, the player accepts it with probability $2^{-(\Delta+2)}$ and thus wins the game. Assuming the player, instead, chooses to observe the game state has updated such that $\Delta := \Delta - 1$ and $t := t - 1$. If $\Delta \neq 0$, we can apply the IH and lower bound the probability by $\frac{1}{3}\left(1 - 2^{-\Delta}\right)$. If $\Delta = 0$, then trivially $\frac{1}{3}\left(1 - 2^{-\Delta}\right) = 0$. Therefore, the total probability of winning given that the next bit is 1 is

$$\begin{aligned} 2^{-(\Delta+2)} + \left(1 - 2^{-(\Delta+2)}\right) \cdot \frac{1}{3}\left(1 - 2^{-\Delta}\right) &= \frac{1}{3}\left(1 - 2^{-\Delta} + 2^{-(\Delta+1)} + 2^{-(2\Delta+2)}\right) \\ &= \frac{1}{3}\left(1 - 2^{-(\Delta+1)} + 2^{-(2\Delta+2)}\right) \\ &\geq \frac{1}{3}\left(1 - 2^{-(\Delta+1)}\right). \end{aligned}$$

Alternatively, if the next bit is 0, then the player does not lose with probability $1 - 2^{-(\Delta+2)}$. Given that the player chooses to observe, the game state has updated such that $\Delta := \Delta + 1$ and $t := t - 1$. If $\Delta \neq 0$, we can apply the IH and lower bound the probability by $\frac{1}{3}\left(1 - 2^{-(\Delta+1)}\right)$. If $\Delta = 0$, then trivially $\frac{1}{3}\left(1 - 2^{-\Delta+1}\right) = 0$. Therefore, the total probability of winning given that the next bit is 0 is,

$$\begin{aligned}\left(1-2^{-(\Delta+2)}\right)\cdot\frac{1}{3}\left(1-2^{-(\Delta+2)}\right) &= \frac{1}{3}\left(1-2^{-(\Delta+2)}-2^{-(\Delta+2)}+2^{-(2\Delta+4)}\right)\\ &= \frac{1}{3}\left(1-2^{-(\Delta+1)}+2^{-(2\Delta+4)}\right)\\ &\geq \frac{1}{3}\left(1-2^{-(\Delta+1)}\right).\end{aligned}$$

The Algorithm

Let $k = 2\lfloor \log_2 n \rfloor + 2$ and draw $j^* \in [k-1]$ uniformly at random and set the threshold $\theta = \frac{j^*-1}{k}\hat{a_{[1]}}$. For $j \in [k]$, we define $n_{\geq j}$ similar to how we did so in Section 2.4

$$n_{\geq j} = \left|\left\{ i \in [n] : a_i > \frac{j-1}{k}\hat{a_{[1]}} \right\}\right|$$

such that $n_{\geq 1} \leq n$ and $n_{\geq k} \geq 1$. With probability at least $\frac{1}{2}$, j^* satisfies $n_{\geq j^*} < 2n_{\geq j^*+1}$.

For the sake contradiction assume otherwise. So, there exists $\frac{k}{2}$ different values $1 \leq j_1 < \ldots < j_{k/2} \leq k-1$ such that $n_{\geq j} \geq 2n_{\geq 22j+1}$ holds for all $j \in \left\{ j_1, \ldots, j_{\frac{k}{2}} \right\}$. This would imply

$$\begin{aligned} n &\geq n_{\geq 1}\\ &\geq 2^{\frac{k}{2}} n_{\geq k/2}\\ &\geq 2^{\frac{k}{2}} n_{\geq k}\\ &\geq 2^{\frac{2\lfloor \log_2 n\rfloor+2}{2}}\cdot 1\\ &= 2^{\lfloor \log_2 n \rfloor + 1}\\ &> n.\end{aligned}$$

which is a contradiction.

Since $n_{\geq j^*+1} \geq \frac{n_{\geq j^*}}{2}$ by definition, amongst the $n_{\geq j^*}$ options that could pass the test given the threshold of θ, more than half of them would have a utility of at least $\frac{\hat{a_{[1]}}}{k}$ after testing, making them in our classification system— "good."

Now using the ideas presented in Section 2.4.4, we can simulate the game over the maximal sequence length of n. Each incoming pen can be tested with the threshold θ. If some pen i passes, we check whether the player in the bit-sampling games commits to the next bit. If they do, we accept pen i. Otherwise, we further test the current pen to check if its value is larger than $\frac{j^*+1}{j}\hat{a_{[1]}}$. If it is, we find the bit-sampling player 1; else, feed 0.

Given our initial claim that $n_{\geq j^*} \leq 2n_{\geq j^*+1}$, if it is the case, the probability that we obtain a score that is at least $\geq \frac{\hat{a_{[1]}}}{k}$ is lower bounded by the bit-sampling player's wining probability which we previously proved to be at least $\frac{1}{6}$. Thus, our expected score is at least

$$\frac{1}{2}\cdot\frac{1}{6}\cdot\frac{\hat{a_{[1]}}}{k} = O\left(\frac{\hat{a_{[1]}}}{\log n}\right).$$

2.4.5 An Improvement to Random Order with Complete Information

We say a "gap" appears in the set of remaining unseen pens if, there is at least one pen in the set that has a value $> B$ and if there are zero pens in the st found in the range $(A, B]$. If such a "gap" appears, then it guarantees the player a score of $B - A$, since they can simply test at the threshold $\theta = A$, and accepting the first pen that passes since we know that there is at least one pen $> B$.

Let $k \geq 1$ whose value will be determined later. As we have done previously, for every $j \in [k]$, we define

$$n_j = \left| \left\{ i \in [n] \mid \frac{j-1}{k} a_{[1]} < a_i \leq \frac{j}{k} a_{[1]} \right\} \right|$$

and also $n_{\geq j} = n_j + \ldots + n_k$.

Now, for every $i \in \{1, \ldots, n\}$, before testing X_i we review the multiset of all unseen pens. We then check if there is some $j \in [k-1]$ such that a gap $\left(\frac{j-1}{k} a_{[1]}, \frac{j}{k} a_{[1]}\right]$ exists and at least one of the pens is higher than $\frac{j}{k} a_{[1]}$. We then test all remaining pens at a threshold $\theta = \frac{j-1}{k} a_{[1]}$ and accept the first pen that passes the test. If either the gap or a "good" pen does not exist, we test with a threshold $\theta = \infty$ to observe X_i. If the condition holds at any point, the player will obtain a score of at least $\frac{a_{[1]}}{k}$.

Let $\mathcal{E}_j$ denote the even that, among $n_{\geq j}$ options with value $> \frac{j-1}{k} a_{[1]}$, that the last pen is the whose value is strictly larger than $\frac{j}{k} a_{[1]}$. In other words, $n_{\geq j}$ have been seen except one, namely the one whose value is $> \frac{j}{k} a_{[1]}$, which also allows for the existence of the gap $\left(\frac{j-1}{k} a_{[1]}, \frac{j}{k} a_{[1]}\right]$. The existence of $\mathcal{E}_j$ implies that the algorithm we defined in the paragraph above would detect the gap of size $\frac{a_{[1]}}{k}$. Since the pens are randomly ordered, $\mathcal{E}_j$ happens with probability $\frac{n_{\geq j+1}}{n_{\geq j}}$. Given that each $\mathcal{E}_i$ happens independently, the probability that none of these events occur is

$$\begin{aligned}
\prod_{j=1}^{k-1} \left(1 - \frac{n_{\geq j+1}}{n_{\geq j}}\right) &\leq \prod_{i=1}^{k-1} \exp\left(-\frac{n_{\geq j+1}}{n_{\geq j}}\right) && (\forall x : 1 - x \leq e^{-x}) \\
&\leq \exp\left(-\sum_{i=1}^{k-1} \frac{n_{\geq j+1}}{n_{\geq j}}\right) && \text{(exponentiation law)} \\
&\leq \exp\left(-(k-1) \cdot \left(\prod_{i=1}^{k-1} \left(\frac{n_{\geq j+1}}{n_{\geq j}}\right)\right)^{\frac{1}{k-1}}\right) && \text{(AM-GM ineq.)} \\
&\leq \exp\left(-(k-1) \cdot \left(\frac{1}{n}\right)^{\frac{1}{k-1}}\right) \\
&= \exp\left(-(k-1) \cdot (n)^{-\left(\frac{1}{k-1}\right)}\right).
\end{aligned}$$

Now, setting $k = \Theta\left(\frac{\log n}{\log \log n}\right)$ (to help motivate the setting of k, see Appendix A). In doing so, we know have that the probability that ever $\mathcal{E}_i$ fails to occur as,

$$\exp\left(-(k-1) \cdot (n)^{-\left(\frac{1}{k-1}\right)}\right) \leq \exp(-1).$$

This implies that the probability that at least one of these events, obtaining a score of at least $\geq \frac{a_{[1]}}{k}$, occurring is at least,

$$1 - \frac{1}{e}.$$

Hence, the expected utility is at least,

$$\left(1 - \frac{1}{e}\right) \cdot \frac{a_{[1]}}{k} = \Omega\left(\frac{\log\log n}{\log n} a_{[1]}\right).$$

Thus, the algorithm is,

$$\begin{aligned}\frac{a_{[1]}}{\left(1 - \frac{1}{e}\right) \cdot \frac{a_{[1]}}{k}} &= \frac{k}{\left(1 - \frac{1}{e}\right)} \\ &= O\left(\frac{\log n}{\log\log n}\right).\end{aligned}$$

competitive.

2.5 Secretary Setting Lower-Bounds

2.5.1 Random Order with Full Information

Set $k \geq 1$ and let $n = 2^{k+1} - 1$. Consider the sequence $(a_i)_{i=1}^n$ where each $j \in \{0, \ldots, k\}$ appears 2^{k-j} times in the sequence.

We claim that for any $\Delta \in [k]$ and $\theta \in \{1, \ldots, k - \Delta\}$, the probability that the player obtains a score of at least Δ after accepting an option that has been tested with a threshold θ is at most $4 \cdot 2^{-\Delta}$.

Fix some $\Delta \in [k]$ and some $\theta \in \{1, \ldots, k - \Delta\}$. We state that a player "wins" if they obtain a score that is at least Δ, given that they tested with a threshold of θ. WLOG the player will either

- Observe: X_i with $\theta = k$ and then reject
- Commit: X_i with θ and accept if pen passes

Similar to how we classified pens that passed in Section 2.4.4, a pen that passes the test at θ will be categorized as "good" if it has a value of $\theta + \Delta$ and any other pen as "bad." Consequently, we can state

$$G = \sum_{j=\theta+\Delta}^{k} 2^{k-j} \qquad B = \sum_{j=\theta}^{\theta-\Delta+1} 2^{k-j}$$

where G is the number of "good" pens and B is the number of "bad" pens. We also define g to be the number of "good" pens remaining and b to be the number of "bad" pens, given that there are t remaining options. We claim that the probability of accepting a good pen is at most

$$\begin{cases} 0, & b = g = 0 \\ \frac{g}{b+g}, & \text{otherwise} \end{cases}$$

Performing induction on t, if $t = 0$, necessarily $b = g = 0$ which is true. Assume that the claim holds true for $t' = t - 1$, for b, g such that $b + g \leq t$.

Consider the case where the player is committing. We know that the probability that the unseen pen fails the test for θ is $1 - \frac{b+g}{t}$ and so the player considers the next pen ignoring the one that just failed. By the IH, the probability is still at most $\frac{g}{b+g}$. Alternatively, if the next option is good that has a probability of $\frac{g}{t}$. Thus, the overall winning probability is upper bounded by

$$\left(\left(1 - \frac{b+g}{t}\right) \cdot \frac{g}{b+g}\right) + \frac{b}{g} = \frac{g}{b+g}$$

The other case is that the player chooses to observe. The probability that the current unseen pen is good is $\frac{g}{t}$ and if the player commits to the following pen, the likelihood that is good is $\frac{g-1}{b+g-1}$. Alternatively, if the next unseen pen is bad, which has a probability of $\frac{b}{t}$ and if the player commits to the following pen, the likelihood that is good is $\frac{g}{b-1+g}$. It is also possible that unseen pen fails the test with probability $\frac{t-(b+g)}{t}$ and the likelihood that the following pen is $\frac{g}{b+g}$. Thus, the overall winning probability is upper bounded by

$$\begin{aligned}\left(\frac{g}{t} \cdot \left(\frac{g-1}{b+g-1}\right)\right) + \left(\frac{b}{t} \cdot \left(\frac{g}{b-1+g}\right)\right) + \left(\frac{g}{t} \cdot \left(\frac{t-(b+g)}{t}\right)\right) &= \frac{g}{t}\left(\frac{g+b-1}{b+g-1} + \frac{t-(b+g)}{b+g}\right) \\ &= \frac{g}{b+g}.\end{aligned}$$

Therefore, the induction holds. Applying the claim at $t = n$, $b = B$ and g$= G$, the probability is at most

$$\begin{aligned}\frac{G}{B+G} &= \frac{\sum_{j=\theta+\Delta}^{k} 2^{k-j}}{\sum_{j=\theta}^{k} 2^{k-j}} \\ &\leq \frac{2^{k-\theta-\Delta+1}}{2^{k-\theta-1}} \\ &= 4 \cdot 2^{-\Delta}.\end{aligned}$$

For any $\Delta \in [k]$, the event of getting a score of at least Δ can happen with any threshold θ up to $k - \theta$, using union bound,

$$\begin{aligned}\mathbb{P}(\text{Score} \geq \Delta) &\leq \sum_{\theta=0}^{k-\Delta} \mathbb{P}(\text{Score} \geq \Delta \mid \theta) \\ &\leq \sum_{\theta=0}^{k-\Delta} 4 \cdot 2^{-\Delta} \\ &= (k - \Delta + 1)\left(4 \cdot 2^{-\Delta}\right) \\ &\leq 4k \cdot 2^{-\Delta}.\end{aligned}$$

Thus, the expected score is,

$$\begin{aligned}\mathbb{E}\left[\text{Score}\right] &= \sum_{\Delta=1}^{k} \mathbb{P}(\text{Score} \geq \Delta) \\ &\leq \sum_{\Delta=1}^{k} 4k \cdot 2^{-\Delta}.\end{aligned}$$

For small Δ, term is 1 for $\Delta \leq \log(4k)$ which implies $O(\log k)$ possible terms?? For larger Δ, the term is $4k \cdot 2^{-\Delta}$ which will converge to a constant. Hence, we can claim,

$$\begin{aligned}\sum_{\Delta=1}^{k} 4k \cdot 2^{-\Delta} &\leq \sum_{\Delta=1}^{k} \min\left\{1, 4k \cdot 2^{-\Delta}\right\} \\ &= O(\log k).\end{aligned}$$

Since, $a_{[1]} = k$, the competitive ratio is at least $\frac{k}{O(\log k)} = \Omega\left(\frac{\log n}{\log\log n}\right)$.

2.5.2 Arbitrary Order with Full Information

Let $k \geq 1$ and generate $n = \sum_{i=0}^{k} 4^i$ options by following the two outlined steps

1. Sample X from the geometric distribution that takes every $j \in \{0, 1, \ldots\}$ with probability $2^{-(j+1)}$
2. If $X \leq k$ and X has appeared less than 4^{k-X} times in the sequence, we append X to the sequence

The resulting sequence is always a permutation of the length-n sequence in which each $j \in \{0, 1, \ldots, k\}$ appears exactly 4^{k-j} times. At any step i, if you condition on the first $i-1$ pens, the distribution of the next pen is still proportional to 2^{-j} over the set of values that haven't hit their limit yet.

For any $\Delta \geq 3$, the probability of getting a score of exactly Δ is at most $c \cdot 2^{-\Delta}$ where $c = 21$.

We classify an acceptance as risky given that the player has tested with θ_i if at least one of the unseen options has value $\theta_i + 1$, otherwise the acceptance is coined as safe. Basically, $j \in \{0, \ldots, k\}$ is active if j appears at least once in $X_i, \ldots, X_n$. The construction of the sequence guarantees that conditioning on $X_1, \ldots, X_{i-1}$, the next option X_i is distributed over all values j such that probability that $X_i = j$ is proportional to 2^{-j}. Since by definition $\theta_i + 1$ is active, the conditional probability of the event $X_i = \theta_i + \Delta$ is upper bounded by

$$\begin{aligned} \frac{\mathbb{P}\left(X_i = \theta + \Delta\right)}{\mathbb{P}\left(\theta_i + 1 \text{ is active}\right)} &\propto \frac{2^{-(\theta_i+\Delta)}}{2^{-(\theta_i+1)}} \\ &= 2^{1-\Delta}. \end{aligned}$$

For $\theta \in \{0, \ldots, k-\Delta\}$, we call the options with a value of $\theta + 1$, the "θ-bad" ones, and ones with $\theta + \Delta$ as the "θ-good" ones. Let $\mathcal{E}_\theta$ be event of the last "θ-good" pen appearing after the last "θ-bad" option. for the player to obtain a score of Δ by safely accepting i, necessarily $X_i = \theta_i + \Delta$ and that $\theta_i + 1$ does not occur in the remaining set of options, which informs us that $\mathcal{E}_{\theta_i}$ must happen. We can upper bound the probability of a safe win by controlling the probability of each $\mathcal{E}_\theta$.

Similar to the classification in Section 2.5.1, we define

$$G = 4^{k-(\theta+\Delta)} \qquad B = 4^{k-(\theta+1)}$$

to be the number of θ-good and θ-bad options, respectively. Focusing on $(B+G)$ length subsequences consisting of only $\theta + 1$ and $\theta + \Delta$ values. The construction of the sequence $(X_i)_{i=1}^n$ ensure that this subsequence follows the same distribution as the output of the following procedure

- Sample $Y_1, \ldots, Y_{B+G}$ independently from the Bernoulli distribution $\mu = \frac{2^{-\Delta}}{2^{-1}+2^{-\Delta}}$
- For each $i \in \{1, \ldots, B+G\}$ if $Y_i = 0$ append $\theta + 1$, otherwise append $\theta + \Delta$. Repeat this step until either $\theta + 1$ is in the sequence B times or $\theta + \Delta$ is found G times.
- In the former case, append $\theta + \Delta$ to the end of the sequence until the sequence has length $B + G$; append $\theta + 1$ in the latter case.

Since $\mathcal{E}_\theta$ corresponds to the number of successes being less than G, the Chernoff bound is

$$\begin{aligned}\mathbb{P}\left(X \leq (1-\delta)\mu\right) &= \mathbb{P}\left(X \leq \left(1 - \left(1 - \frac{G}{\mu(B+G)}\right) \cdot \mu(B+G)\right)\right) \\ &= \mathbb{P}(X \leq G) \\ &\leq \exp\left(-\frac{1}{2} \cdot \left(1 - \frac{G}{\mu(B+G)}\right)^2 \cdot \mu(B+G)\right).\end{aligned}$$

given that we set $\delta = \left(1 - \frac{G}{\mu(B+G)}\right)$. Now substituting some of the known values,

$$-\frac{1}{2} \cdot \left(1 - \frac{4^{k-(\theta+\Delta)}\left(2^{-1}+2^{-\Delta}\right)}{2^{-\Delta}\left(4^{k-(\theta+\Delta)}+4^{k-(\theta+1)}\right)}\right)^2 \cdot \frac{2^{-\Delta}}{2^{-1}+2^{-\Delta}}\left(4^{k-(\theta+\Delta)}+4^{k-(\theta+1)}\right).$$

Working out the term $\frac{G}{\mu(B+G)}$,

$$\begin{aligned}\frac{G}{\mu(B+G)} &= \frac{G}{\mu \cdot G\left(\frac{B}{G}+1\right)} \\ &= \frac{1}{\mu\left(\frac{B}{G}+1\right)} \\ &= \frac{1}{\mu\left(4^{\Delta-1}+1\right)} \\ &= \frac{1+2^{-\Delta+1}}{2^{-\Delta+1}\left(4^{\Delta-1}+1\right)}.\end{aligned}$$

Now working out the denominator,

$$\begin{aligned}2^{-\Delta+1}\left(4^{\Delta-1}+1\right) &= 2^{-\Delta+1}\left(2^{2(\Delta-1)}+1\right) \\ &= 2^{\Delta-1}+2^{-\Delta+1}.\end{aligned}$$

Hence, we have,

$$\frac{1+2^{-\Delta+1}}{2^{\Delta-1}+2^{-\Delta+1}}.$$

Given that $\Delta \geq 3$, we can claim that,

$$\begin{aligned}\frac{1+2^{-\Delta+1}}{2^{\Delta-1}+2^{-\Delta+1}} &\leq \frac{1}{2} \\ \implies 2+2^{-\Delta+2} &\leq 2^{\Delta-1}+2^{-\Delta+1} \\ \implies 2^{\Delta-1} &\geq 2+2^{-\Delta+2}-2^{-\Delta+1} \\ &= 2+2\left(2^{-\Delta+1}\right)-2^{-\Delta+1} \\ &\geq 2+2^{-\Delta+1}.\end{aligned}$$

Given the base case of $\Delta = 3$, $2^{3-1} \geq 2+2^{-3+1}$. Then assume that it holds up to $\Delta - 1$. Then,

$$\begin{aligned}2^{\Delta-1} &= 2\left(2^{(\Delta-1)-1}\right) \\ &\geq 2+2^{-(\Delta-1)+1} \\ &= 2+2^{-\Delta+2} \\ &\geq 2+2^{-\Delta+1}.\end{aligned}$$

Hence,

$$\frac{G}{\mu(B+G)} \leq \frac{1}{2}.$$

Using this result and going back to the initial bound, we have,

$$\exp\left(-\frac{1}{2}\cdot\left(1-\frac{G}{\mu(B+G)}\right)^2\cdot\mu(B+G)\right) \leq \exp\left(-\frac{1}{8}\cdot\mu(B+G)\right).$$

Working on $\mu(B+G)$,

$$\begin{aligned}\frac{2^{-\Delta}}{2^{-1}+2^{-\Delta}}\left(4^{k-(\theta+\Delta)}+4^{k-(\theta+1)}\right) &= \frac{2^{-\Delta}}{2^{-1}+2^{-\Delta}}\cdot 4^{k-(\theta+1)}\left(1+4^{1-\Delta}\right)\\ \implies \frac{\mu(B+G)}{2^{-\Delta}\cdot 4^{k-(\theta+1)}} &= \frac{1+4^{1-\Delta}}{2^{-1}+2^{-\Delta}}.\end{aligned}$$

This is lower bound by 1 given that $\Delta \geq 3$. Continuing,

$$\begin{aligned}1+4^{1-\Delta} &\geq 2^{-1}+2^{-\Delta}\\ 2^{\Delta}+2^{\Delta}4^{1-\Delta} &\geq 2^{\Delta-1}+1\\ 2^{\Delta}+2^{2-\Delta} &\geq 2^{\Delta-1}+1.\end{aligned}$$

The last inequality is true for any $\Delta \geq 3$, thus we can conclude $\mu(B+G) \geq 2^{-\Delta}\cdot 4^{k-\theta-1}$. Reconsidering the bound,

$$\exp\left(-\frac{1}{8}\cdot\mu(B+G)\right) \leq \exp\left(-\frac{2^{\Delta}4^{k-\theta-1}}{8}\right).$$

Since $\theta+\Delta \leq k \implies k-\theta \geq \Delta$, so,

$$\exp\left(-\frac{2^{\Delta}4^{k-\theta-1}}{8}\right) \leq \exp\left(-\frac{2^{k-\theta}}{32}\right)$$

The probability that at least one of $\mathcal{E}_\theta$ happens is then upper bounded by

$$\begin{aligned}\sum_{\theta=0}^{k-\Delta}\mathbb{P}\left(\mathcal{E}_\theta\right) &\leq \sum_{\theta=0}^{k-\Delta}\exp\left(-\frac{2^{k-\theta}}{32}\right)\\ &= \sum_{j=\Delta}^{k}\exp\left(-\frac{2^{j}}{32}\right).\end{aligned}$$

Thus, considering both cases the total probability of winning for the player is at most,

$$2^{1-\Delta}+\sum_{j=\Delta}^{k}\exp\left(-\frac{2^{j}}{32}\right),$$

which is at most $21\cdot 2^{-\Delta}$ (the working produced by Gemini 2.5-pro can be seen in Appendix B). Tying this all together, the expected score is then,

$$\sum_{\Delta=1}^{k}\Delta\cdot\mathbb{P}\left(\text{Score}=\Delta\right) \leq 3+21\cdot\sum_{\Delta=3}^{\infty}\Delta\cdot 2^{-\Delta} = O(1).$$

Hence, the competitive ratio is lower bounded by $\frac{k}{O(1)} = \Omega(\log n)$.

2.5.3 Random Order with Optimum Information

Let $\mathcal{D}$ be the distribution of $\min\left\{X, \frac{\ln n}{2}\right\}$, where X is drawn from the exponential distribution with parameter 1. Thus, either $\mathcal{D}$ is either memory-less exponential distribution or a truncated mass point. We truncate the exponential by moving all the probability mass from its tail $[\frac{\ln n}{2}, \infty)$ to a point mass at $\frac{\ln n}{2}$. We draw $a_1, \ldots, a_n$ independently from $\mathcal{D}$. Intuitively, most of these values will take on $\frac{\ln n}{2}$ with high probability so there is not much information to take about $a_{[1]}$.

The probability of never getting a sample with value $\frac{\ln n}{2}$ is

$$\begin{aligned}
\mathbb{P}\left(a_i = \frac{\ln n}{2}\right) &= \mathbb{P}\left(X \geq \frac{\ln n}{2}\right) \\
&= e^{-\frac{\ln n}{2}} \\
&= \frac{1}{\sqrt{n}} \\
\implies \mathbb{P}\left(a_i \neq \frac{\ln n}{2}\right) &= 1 - \frac{1}{\sqrt{n}}.
\end{aligned}$$

Thus, the probability not a single pen has a value of $\frac{\ln n}{2}$ is,

$$\begin{aligned}
\mathbb{P}\left(a \neq \frac{\ln n}{2}\right) &= \left(1 - \frac{1}{\sqrt{n}}\right)^n \\
&\leq e^{-\sqrt{n}}. && ((1-x)^n \leq e^{-x})
\end{aligned}$$

Consequently, the probability that $a_{[1]}$ is $\frac{\ln n}{2}$ is $1 - e^{-\sqrt{n}}$ and thus the expectation is at least $\frac{\ln n}{2} \cdot \left(1 - e^{-\sqrt{n}}\right) = \Omega(\log n)$.

No player can have an expected score strictly larger than $1 + e^{-\sqrt{n}} \cdot \frac{\ln n}{2} = O(1)$. For the sake of contradiction, assume that the claim in the previous sentence is false.

Instead of providing the player with $a_{[1]}$, we lie and give them $\frac{\ln n}{2}$ as the maximal value of all pens. In the case that $a_{[1]} \neq \frac{\ln n}{2}$, which happens with probability $e^{-\sqrt{n}}$ and by the fact that the score is bounded between 0 and $\frac{\ln n}{2}$, the expected score decreases by at most $e^{-\sqrt{n}} \cdot \frac{\ln n}{2}$, and thus strictly higher than 1. However, after a random permutation (X_i) are still n independent samples from $\mathcal{D}$. So, when the player accepts an option i after testing it with $\theta_i < \frac{\ln n}{2}$, the expected score is

$$\begin{aligned}
\mathop{\mathbb{E}}_{X_i \sim D}\left[X_i - \theta_i \mid X_i > \theta_i\right] &= \mathop{\mathbb{E}}_{X \sim \mathrm{Exp}(1)}\left[\min\left\{X, \frac{\ln n}{2}\right\} - \theta_i \,\middle|\, \min\left\{X, \frac{\ln n}{2}\right\} > \theta_i\right] \\
&\leq \mathop{\mathbb{E}}_{X \sim \mathrm{Exp}(1)}\left[X - \theta_i \mid X > \theta_i\right] \\
&= 1.
\end{aligned}$$

which is a contradiction.

Therefore, the expected optimum is $\Omega(\log n)$ whereas the expected score is $O(1)$. Hence, the player's competitive ratio is $\Omega(\log n)$.

2.6 Single-Sample Algorithm for Prophet

The distribution $(\mathcal{D}_i)_{i=1}^n$ from which each corresponding $(X_i)_{i=1}^n$ is drawn from is unknown and we only get to learn about them by drawing a few samples from each $\mathcal{D}_i$. An immediate idea might be to slightly rework the algorithm introduced in section 2.3.

However, the problem is that calculating/approximating the quantiles requires $\Omega(n)$ samples from each $\mathcal{D}_i$. The most difficult quantile to estimate is when $\alpha = \frac{1}{n}$. In this case, the probability p of seeing a value greater than the $(1 - \frac{1}{n})$-quantile is exactly $1/n$. Therefore, we would need to draw approximately n samples.

Let $X^{\max}$ be the max os (X_i) given that they are drawn independently from there corresponding distribution $\mathcal{D}_i$. The CDF of $X^{\max}$ is

$$F(x) = \mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \leq x\right).$$

and is also right continuous. Consequently, we can define $A \geq 0$ as the smallest number such that $\mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \leq A\right) = F(x) \geq \frac{1}{3}$ ($\frac{1}{3}$-quantile). Similarly, we define $B \geq 0$ as the largest number such that $\mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \geq B\right) = F(x) \geq \frac{1}{3}$ ($\frac{2}{3}$-quantile). Then,

$$\begin{aligned}
\mathbb{P}_{X\sim\mathcal{D}}\left(A \leq X^{\max} \leq B\right) &\geq 1 - \mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \leq A\right) - \mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \geq B\right) \\
&\geq 1 - \frac{1}{3} - \frac{1}{3} \\
&= \frac{1}{3}.
\end{aligned}$$

Let $\hat{a_{[1]}}$ denote the maximal value among $\left(\hat{X}_i\right)_{i=1}^n$, where each $\hat{X}_i$ is a sample drawn from the respective $\mathcal{D}_i$ and provided to the player. $\hat{a_{[1]}}$ follows the same distribution as $X^{\max}$ which informs us that $\hat{a_{[1]}} \in [A, B]$ with at least probability $\frac{1}{3}$.

2.6.1 First Algorithm

Assuming that $\hat{a_{[1]}} \leq B$, then

$$\mathbb{P}_{X\sim\mathcal{D}}\left(\hat{a_{[1]}} \leq X^{\max}\right) \geq \mathbb{P}_{X\sim\mathcal{D}}\left(B \leq X^{\max}\right) \geq \frac{1}{3}.$$

Assuming that $\hat{a_{[1]}} \leq X^{\max}$, we can view $(X_i)_{i=1}^n$ as an instance of the secretary setting under the arbitrary order and $\hat{a_{[1]}}$ as the hint provided. Hence, running mentioned in Section 2.4.4, we obtain an expected score of $\frac{1}{3} \cdot \frac{\hat{a_{[1]}}}{O(\log n)} = \frac{\hat{a_{[1]}}}{O(\log n)}$.

2.6.2 Second Algorithm

Setting a single threshold with $\theta = \hat{a_{[1]}}$, the expected score is

$$\sum_{i=1}^{n}\left(\prod_{j=1}^{n-1}\left(\mathbb{P}_{X_j\sim\mathcal{D}_j}\left(X_j \leq \hat{a_{[1]}}\right)\right) \cdot \mathbb{P}_{X_i\sim\mathcal{D}_i}\left(X_i \geq \hat{a_{[i]}}\right) \cdot \mathop{\mathbb{E}}_{X_i\sim\mathcal{D}_i}\left[X_i - \hat{a_{[1]}} \mid X_i > \hat{a_{[1]}}\right]\right).$$

Examining the first term of the three-term product,

$$\begin{aligned}
\prod_{j=1}^{n-1}\left(\mathbb{P}_{X_j\sim\mathcal{D}_j}\left(X_j \leq \hat{a_{[1]}}\right)\right) &\geq \mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \leq \hat{a_{[1]}}\right) \\
&\geq \mathbb{P}_{X\sim\mathcal{D}}\left(X^{\max} \leq A\right) \\
&\geq \frac{1}{3}.
\end{aligned}$$

The product of the other two terms is,

$$\mathbb{P}_{X_i \sim \mathcal{D}_i}\left(X_i \geq \hat{a_{[i]}}\right) \cdot \underset{X_i \sim \mathcal{D}_i}{\mathbb{E}}\left[X_i - \hat{a_{[1]}} \mid X_i > \hat{a_{[1]}}\right] = \underset{X_i \sim \mathcal{D}_i}{\mathbb{E}}\left[\left(X^{\max} - \hat{a_{[1]}}\right) \cdot \mathbb{1}\left[X_i > \hat{a_{[1]}}\right]\right]$$
$$= \underset{X_i \sim \mathcal{D}_i}{\mathbb{E}}\left[\max\left\{X^{\max} - \hat{a_{[1]}}, 0\right\}\right].$$

Thus, the expected score is lower bounded by,

$$\begin{aligned}
\sum_{i=1}^{n}\left(\frac{1}{3} \cdot \underset{X_i \sim \mathcal{D}_i}{\mathbb{E}}\left[\max\left\{X^{\max} - \hat{a_{[1]}}, 0\right\}\right]\right) &= \frac{1}{3} \cdot \underset{X_i \sim \mathcal{D}_i}{\mathbb{E}}\left[\sum_{i=1}^{n} \max\left\{X^{\max} - \hat{a_{[1]}}, 0\right\}\right] \\
&\geq \frac{1}{3} \cdot \underset{X \sim \mathcal{D}}{\mathbb{E}}\left[\max_{i \in [n]} \max\left\{X^{\max} - \hat{a_{[1]}}, 0\right\}\right] \\
&= \frac{1}{3} \cdot \underset{X \sim \mathcal{D}}{\mathbb{E}}\left[\max\left\{X^{\max} - \hat{a_{[1]}}, 0\right\}\right] \\
&\geq \frac{\mathbb{E}_{X \sim \mathcal{D}}\left[X^{\max}\right] - \hat{a_{[1]}}}{3}.
\end{aligned}$$

Conditioning on each $\hat{a_{[1]}} \in [A, B]$ and uniformly selecting one of the two algorithms at random,

$$\frac{1}{2}\left(\frac{\hat{a_{[1]}}}{O(\log n)} + \frac{\mathbb{E}_{X \sim \mathcal{D}}\left[X^{\max}\right] - \hat{a_{[1]}}}{3}\right) = \frac{\mathbb{E}_{X \sim \mathcal{D}}\left[X^{\max}\right]}{O(\log n)}.$$

Hence, the algorithm is $O(\log n)$ competitive.

2.7 Appendix

A Motivating k as a $\log$ over $\log\log$ factor

$$
\begin{aligned}
1 &\approx (k-1)\cdot n^{\left(-\frac{1}{k-1}\right)} \\
\implies (k-1)^{-1} &= n^{\left(-\frac{1}{k-1}\right)} \\
(k-1) &= n^{\left(\frac{1}{k-1}\right)} \\
\implies \log(k-1) &= \frac{1}{k-1}\cdot\log(n) \\
\log(n) &= (k-1)\cdot\log(k-1)
\end{aligned}
$$

Setting $k-1=\frac{\log n}{\log\log n}$

B Upper Bounding to a Constant

We want to prove that for any $\Delta \geq 3$:

$$2^{1-\Delta} + \sum_{j=\Delta}^{k} \exp\left(-\frac{2^j}{32}\right) \leq c \cdot 2^{-\Delta}$$

for some constant c. Rearranging to solve for c:

$$c \geq \frac{2^{1-\Delta} + \sum_{j=\Delta}^{k} \exp\left(-\frac{2^j}{32}\right)}{2^{-\Delta}}$$

$$c \geq 2 + 2^{\Delta} \sum_{j=\Delta}^{k} \exp\left(-\frac{2^j}{32}\right)$$

To find a c that works for all $k \geq \Delta$, we let the sum go to infinity:

$$f(\Delta) = 2 + 2^{\Delta} \sum_{j=\Delta}^{\infty} \exp\left(-\frac{2^j}{32}\right)$$

We will find the maximum value of $f(\Delta)$ for $\Delta \geq 3$.

Step 1: Finding the Maximum of $f(\Delta)$

Let's examine the difference between consecutive terms:

$$f(\Delta+1) - f(\Delta) = 2 + 2^{\Delta+1} \sum_{j=\Delta+1}^{\infty} \exp\left(-\frac{2^j}{32}\right) - \left(2 + 2^{\Delta} \sum_{j=\Delta}^{\infty} \exp\left(-\frac{2^j}{32}\right)\right)$$

$$= 2^{\Delta+1} \sum_{j=\Delta+1}^{\infty} \exp\left(-\frac{2^j}{32}\right) - 2^{\Delta}\left[\exp\left(-\frac{2^{\Delta}}{32}\right) + \sum_{j=\Delta+1}^{\infty} \exp\left(-\frac{2^j}{32}\right)\right]$$

$$= 2^{\Delta} \sum_{j=\Delta+1}^{\infty} \exp\left(-\frac{2^j}{32}\right) - 2^{\Delta} \exp\left(-\frac{2^{\Delta}}{32}\right)$$

For $\Delta \geq 4$: The series decreases rapidly, and we want to show $f(\Delta+1) - f(\Delta) < 0$:

$$\sum_{j=\Delta+1}^{\infty} e^{-2^j/32} < e^{-2^{\Delta}/32}$$

Bound the sum:

$$\sum_{j=\Delta+1}^{\infty} e^{-2^j/32} < \frac{e^{-2^{\Delta+1}/32}}{1 - e^{-2^{\Delta+1}/32}}$$

We require

$$\frac{e^{-2^{\Delta+1}/32}}{1 - e^{-2^{\Delta+1}/32}} < e^{-2^{\Delta}/32}$$

Let $x = e^{-2^{\Delta}/32}$. The inequality is $\frac{x^2}{1-x^2} < x$, which simplifies to $x < 1 - x^2$ or $x^2 + x - 1 < 0$. The positive root is $x = (\sqrt{5}-1)/2 \approx 0.618$. So, the condition holds for $\Delta \geq 5$.

For $\Delta = 3$ and $\Delta = 4$: Compare $f(3)$ and $f(4)$. $f(4) - f(3) = 2^4 \sum_{j=4}^{\infty} e^{-2^j/32} - 2^3 \sum_{j=3}^{\infty} e^{-2^j/32}$

$$= 16\left(e^{-1/2} + e^{-1} + e^{-2} + e^{-4} + \dots\right) - 8\left(e^{-1/4} + e^{-1/2} + e^{-1} + e^{-2} + \dots\right)$$

$= 16e^{-1/2} + 16e^{-1} + ... - 8e^{-1/4} - 8e^{-1/2} - ... = 8e^{-1/2} - 8e^{-1/4} + 8e^{-1} + 8e^{-2} + ... < 0$

So the maximum occurs at $\Delta = 4$.

Step 2: Calculate an upper bound for $f(4)$

$$f(4) = 2 + 2^4 \sum_{j=4}^{\infty} \exp\left(-\frac{2^j}{32}\right) = 2 + 16\left(e^{-1/2} + e^{-1} + e^{-2} + \dots\right)$$

Bound the series:

$$\sum_{j=4}^{\infty} e^{-2^j/32} < e^{-1/2} + e^{-1} + e^{-2} + e^{-4} + \frac{e^{-8}}{1 - e^{-8}}$$

Using numerical values: $e^{-1/2} \approx 0.60653$, $e^{-1} \approx 0.36788$, $e^{-2} \approx 0.13534$, $e^{-4} \approx 0.01832$, and $\frac{e^{-8}}{1-e^{-8}} \approx 0.00034$. Therefore, $S_4 < 0.60653+0.36788+0.13534+0.01832+0.00034 \approx 1.12841$ $f(4) < 2 + 16(1.12841) = 2 + 18.05456 = 20.05456$. Since $c \geq 20.05456$, we can use $c = 21$.

Therefore, for all $\Delta \geq 3$,

$$2^{1-\Delta} + \sum_{j=\Delta}^{k} \exp\left(-\frac{2^j}{32}\right) \leq 21 \cdot 2^{-\Delta}$$

Chapter 3

Matroid Secretary Problem

3.1 Preliminary

3.1.1 Definition

Definition 3.1. A matroid is an ordered pair $M = (S, \mathcal{I})$ where S is referred to as the *ground set* and $\mathcal{I}$ is called the *independent set*, satisfying the following conditions:

1. S is a finite set.
2. If $A \in \mathcal{I}$ and $B \subset A$, then $B \in \mathcal{I}$ (Hereditary property).
3. If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists some element $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$ (Exchange/Augmentation property).

Remark 3.2. $\emptyset \in \mathcal{I}$

Remark 3.3. The ground set S can be thought of as a collection of vectors. Thus, the independent set $\mathcal{I}$ would be a subset that is linearly independent.

3.1.2 Example - Graphic Matroid

Given an undirected graph $G = (V, E)$, we can define a *graphic matroid* $M_G = (S_G, \mathcal{I}_G)$ where:

- S_G is defined as E.
- If A is a subset of E, then $A \in \mathcal{I}_G$ iff A is acyclic. In other words, a set of edges A is independent iff the subgraph $G_A = (V, A)$ is a forest.

3.1.3 Proof - Graphic Matroid

Theorem 3.4. *If $G = (V, E)$ is an undirected graph, then $M_G = (S_G, \mathcal{I}_G)$ is a matroid.*

Proof. S_G is evidently a finite set (satisfies condition 1).

Any subset of a forest is still a forest i.e. removing edges from an acyclic graph cannot create a cycle (satisfies condition 2).

Let $A, B \in \mathcal{I}_G$ with $|B| > |A|$. Then $G_A = (V, A)$ and $G_B = (V, B)$ are forests.

A forest with $|V|$ vertices and $|E|$ edges has $|V| - |E|$ trees, so G_A has more trees than G_B. Some tree T in G_B connects vertices in different trees of G_A. Pick an edge $e = (u, v) \in T$ connecting these trees. Adding e to A does not create a cycle. $\square$

3.1.4 Key Concepts

- **Basis**: A maximal independent set. A key property of matroids is that all bases have the same size.
- **Circuit**: A minimal dependent set.
- **Rank Function**: The rank of a subset of the ground set is the size of the largest independent set it contains.

Example 3.5 (Graphic Matroid)**.**

- **Basis**: The spanning trees of the graph.
- **Circuit**: The simple cycles in the graph.

3.1.5 Greedy Algorithms

Given a matroid $M = (S, \mathcal{I})$ and a weight function $\omega : S \to \mathbb{R}^+$, the greedy algorithm finds a maximum-weight basis by:

1. Sorting the elements in S in descending order of their weights.
2. Iterating through the sorted elements and adding an element to the solution if it maintains independence.

The goal is to find an independent set $A \in \mathcal{I}$ such that $\omega(A)$ is maximized— the maximal subset is coined as 'optimal.' Since $\omega(a)$ for any $a \in A$ is positive, an optimal subset is always a maximal independent subset.

3.1.6 Task Scheduling with Deadlines

Schedule unit-time (requires exactly one unit of time to complete) tasks. Each task has a deadline and a penalty if it's missed. We want to find a schedule that minimizes the total penalty.

Inputs:

- A set of n unit-time tasks $T = \{a_1, a_2, \ldots, a_n\}$.
- Integer deadlines $d_1, d_2, \ldots, d_n$, where $1 \leq d_i \leq n$.
- Non-negative penalties $w_1, w_2, \ldots, w_n$ if a task is late.

Remark 3.6. Minimizing the penalty of *late* tasks is equivalent to maximizing the sum of penalties of *early* (on-time) tasks.

The problem reduces to finding the optimal set of tasks to complete on time.

3.1.7 From Schedules to Sets: Defining Independence

Any optimal schedule can be arranged so that all early tasks come first, sorted by their deadlines.

Definition 3.7 (Independent Set of Tasks)**.** A set of tasks $A \subseteq T$ is called **independent** if there exists a schedule where all tasks in A can be completed by their deadlines.

Test for Independence: A set A is independent if and only if for all times $t = 1, \ldots, n$:

$$N_t(A) \leq t$$

where $N_t(A)$ is the number of tasks in A whose deadline is less than or equal to t.

The system $(T, \mathcal{I})$, where $\mathcal{I}$ is the collection of all independent sets of tasks, forms a matroid.

3.1.8 The Greedy Solution

Since task scheduling is a matroid, we can find the optimal set of early tasks (the one with maximum total penalty) using a simple greedy algorithm.

Greedy Task Scheduler:

1. Initialize $A \leftarrow \emptyset$
2. Sort tasks in T into a list T' by **decreasing** penalty w_i
3. For each task a_i in T':
   - If $A \cup \{a_i\}$ is independent, then $A \leftarrow A \cup \{a_i\}$
4. Return A

3.1.9 An Example

Task a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

- **Select:** $a_1(70), a_2(60), a_3(50), a_4(40)$. Current set $A = \{a_1, a_2, a_3, a_4\}$.
- **Consider** $a_5(30)$: $d_5 = 1$. The set $A \cup \{a_5\}$ is checked. $N_4(A \cup \{a_5\}) = 5 > 4$. This is **not independent**. Reject a_5.
- **Consider** $a_6(20)$: $d_6 = 4$. The set $A \cup \{a_6\}$ is checked. $N_4(A \cup \{a_6\}) = 5 > 4$. **Not independent**. Reject a_6.
- **Select** $a_7(10)$: $d_7 = 6$. The set $A \cup \{a_7\}$ is independent.

Result: Optimal early set is $\{a_1, a_2, a_3, a_4, a_7\}$. The minimized penalty is $w_5 + w_6 = 50$.

3.1.10 Online Learning

The matroid structure provides a natural framework for modeling constraints in various online learning problems, where decisions must be made sequentially under uncertainty. While the matroid secretary problem assumes that weights are revealed upon arrival, many real-world scenarios involve learning about item values through repeated interactions or exploration. This section explores several key connections between matroids and online learning paradigms.

Matroid Bandits

In the **matroid bandits** problem, a learner must repeatedly select a subset of items (or "arms") to play over multiple rounds, subject to matroid constraints. Unlike the secretary setting where each item appears once, bandit problems involve repeated interactions: the learner selects an independent set, observes rewards, and uses this feedback to improve future decisions. The challenge lies in balancing exploration (trying different independent sets to learn their values) with exploitation (selecting sets that appear promising based on current knowledge).

This problem generalizes the classic multi-armed bandit framework to combinatorial settings, where the action space consists of all independent sets in a matroid rather than individual items. The matroid structure ensures that the learner's choices remain feasible while exploring the space of possible selections. Competitive algorithms for matroid bandits typically achieve regret bounds that scale with the rank of the matroid, demonstrating how the combinatorial structure affects the learning complexity.

Submodular Maximization with Matroid Constraints

Another important connection arises in **online submodular maximization** under matroid constraints. Here, elements from a ground set arrive one by one, and the algorithm must maintain an independent set in a matroid while aiming to maximize a submodular function of the chosen set. Submodular functions capture diminishing returns, making them natural for modeling coverage, diversity, and influence maximization problems.

The online nature adds complexity: when an element arrives, the algorithm must decide whether to include it based on its marginal contribution to the current set, while ensuring the resulting set remains independent. This problem bridges the gap between the matroid secretary problem (which maximizes a linear function) and more general objective functions, requiring algorithms that can handle both the combinatorial constraints and the non-linear objective structure.

Prophet Inequalities and Matroids

The connection between matroids and **prophet inequalities** provides fundamental theoretical insights into sequential decision-making. Prophet inequalities compare the performance of online algorithms (that make irrevocable decisions as items arrive) against an omniscient offline algorithm (that sees all values in advance). For matroid constraints, prophet inequalities establish that threshold-based algorithms can achieve constant-factor competitive ratios, independent of the matroid's rank or the number of elements.

This theoretical foundation is crucial for understanding the limits of online algorithms in matroid settings. The next subsection explores the matroid prophet inequality in detail, showing how threshold strategies can be adapted to handle combinatorial constraints while maintaining strong performance guarantees.

3.1.11 Matroid Prophet Inequality

We can select a set of items, with the crucial constraint that this chosen set must be an independent set in a given uniform matroid (where any set of size at most k is independent), $M = (S, \mathcal{I})$.

- Each item i has a non-negative random value X_i, drawn independently from a known distribution $\mathcal{D}_i$.
- When an item i with value v_i is revealed, we must make an irrevocable decision: either accept i into our set Y or reject it.

- The constraint is that our set Y must remain independent at all times such that $Y \in \mathcal{I}$.

Set a threshold θ and accept any incoming item given that $v_i \geq \theta$ such that $Y \cup \{i\} \in \mathcal{I}$.

3.2 Where are we

3.2.1 The Matroid Secretary Problem

The Matroid Secretary Problem is a generalization of the classic secretary problem, incorporating combinatorial constraints. The components of a matroid secretary problem are

- a ground set $\mathcal{U}$ of size $n \geq 2$
- a family of independent sets $\mathcal{I}$, each of which are a subset of $\mathcal{U}$
- a weight function $w : \mathcal{U} \to \mathbb{R}^+$ such that it assigns a non-negative real weight $w(u)$ for every $u \in \mathcal{U}$

It should be easy to see that if $|\mathcal{U}| = 1$, then the acceptance of the single element u is contingent only on whether its inclusion in the independent set maintains independence.

The ground set of the matroid is presented in random order to the algorithm. When an element i arrives, the algorithm learns its weight $w(i)$. If adding the new element i to the current set of selected elements S maintains independence, the algorithm can choose to either accept i or discard it. Otherwise, the algorithm if forced to discard i. We make the assumption that the weights are distinct among the elements of the matroid, unless they are 0, and as a consequence there is a unique maximal weight independent set.

We will discuss algorithms related to the following types of matroids:

1. Uniform Matroid of Rank k: An independent set in this type of matroid is any subset of $\mathcal{U}$ with a size of at most k
2. Transversal Matroid: The ground set $\mathcal{U}$ is the set of nodes on one side of a bipartite graph. A subset is considered independent if a matching exists that connects it to a set of nodes on the other side of the graph.
3. Gammoids: In this case, the ground set $\mathcal{U}$ represents a set of source nodes in a graph that are all to be routed to a common sink. A subset of sources is independent if there are vertex-disjoint paths from each source in the subset to the sink.
4. Graphic Matroids: The ground set $\mathcal{U}$ corresponds to the edges of an undirected graph. A set of edges is considered independent if it does not form a cycle.
5. Partition Matroids: The ground set $\mathcal{U}$ is divided into blocks, each with an upper-bound on the number of elements that can be chosen from it. A subset is independent if it respects these upper-bounds.
6. Truncated Partition Matroids: This is a partition matroid with the additional constraint that the total number of selected elements cannot exceed a certain number k.

3.2.2 Mechanism Design Outline

In a single-value preference domain, there is a set $\mathcal{U}$ of n agents a set of possible outcomes Ω. Each agent i has a private value v_i and a set of "satisfying outcomes" A_i. The agent gains utility v_i if the final outcome is in A_i, and zero otherwise. A set of agents $S \subseteq \mathcal{U}$ is independent if there is an outcome $w \in \Omega$ that satisfies exactly the agents in S.

An agent's *type* in general is his private information— so, in the following problem the *type* is the valuation v_i. A single-value preference domain is a matroid domain if for any profile of types, the family of independent sets of agents form a matroid over the set $\mathcal{U}$ of all agents.

The paper provides a couple of examples about matroid domains

1. *Selling k Identical Items*: This scenario involves n agents, each wanting one of k items with a value of v_i. This corresponds directly to a uniform matroid of rank k, where the agents are the ground set and any set of up to k agents is independent.

2. *Selling k Non-Identical Items*: There are n agents and m distinct items, but a seller can only produce $k \leq m$ of them. Each agent can buy only one specific item (so not every item will satisfy the agent). This models a truncated partition matroid of rank k.

3. *Unit-Demand Domain*: There are n agents and m distinct items. Each agent i has a set of desired items T_i such that the agent obtains a value of v_i if they get an item $t \in T_i$. An outcome is a one-to-one matching of agents to items. This domain corresponds to a transversal matroid.

3.3 The General Matroid Domain

3.3.1 Lower Bound for General Downward-closed Set Systems

For some integer n, let $k = \lfloor \ln(n) \rfloor$ and $(\mathcal{U}, \mathcal{I})$ be the set system (which obey a weaker axiom than matroids as they may lack the 'exchange' property) where $\mathcal{U}$ consists of n elements partitioned into $m = \lceil n/k \rceil$ subsets such that each subset S_i has either $k-1$ or k elements. We say a set $A \subseteq \mathcal{U}$ is independent (i.e. $A \in \mathcal{I}$) iff its is contained in one of the partitions S_i. Thus, once an algorithm selects an element from S_i, it is restricted to only selecting other elements from S_i. Each element $u \in \mathcal{U}$ is assigned a weight $w(u)$ independently at random: $w(u) = 1$ with probability $\frac{1}{k}$, and $w(u) = 0$ with probability $1 - \frac{1}{k}$.

Lemma 3.8. *The expected weight of the maximum-weight set in $\mathcal{I}$ is $\Omega\left(\frac{\log n}{\log \log n}\right)$. For any randomized online algorithm that selects a set in $\mathcal{I}$, the expected weight of the set selected when the elements are presented to the algorithm in random order is less than 2.*

Proof. Any online algorithm must make a first selection. Suppose at some time t, it selects an element u which belongs to subset S_i. As a result, from time $t+1$, any other element selected needs to belong to S_i. Let T_i denote the subset of S_i that consists of elements that have not yet been observed at time t. Each of these remaining elements has an expected weight of $1/k$. Therefore, the total expected weight of all future selections is less than $k\left(\frac{1}{k}\right) = 1$. Hence, the total expected weight gathered by the algorithm is the weight of its first selection (which is 1) plus the expected future weight (less than 1), for a total expected weight of less than 2.

Let $j = \lfloor k/(2\ln k) \rfloor$ and consider a single partition S_i such that it contains at least j elements of weight 1. Let E_i be the event that partition S_i contains at least j elements of weight 1. The number of elements of weight 1 in S_i follows a Binomial distribution, $\text{Bin}\left(|S_i|, \frac{1}{k}\right)$. Hence, the probability of E_i is at least

$$\begin{aligned}\left(\frac{1}{k}\right)^j &\geq \left(\frac{1}{\ln n}\right)^{\frac{\ln n}{2\ln\ln n}} \\ &= \frac{1}{\sqrt{n}} && (n \geq 1)\end{aligned}$$

Then, the probability that none of the partitions S_i contains at least j elements of weight 1 is at most

$$\left(1 - \frac{1}{\sqrt{n}}\right)^m \leq e^{-\frac{m}{\sqrt{n}}}$$

Substituting back in the definition of m and k

$$e^{-\frac{m}{\sqrt{n}}} \leq e^{-\frac{\sqrt{n}}{\ln n}}$$

Taking $n \to \infty$, we know the exponent tends to $-\infty$ and thus 0. This means the probability that the optimal solution is at least j is $1 - o(1)$. The expected value of the optimal solution is therefore lower-bounded by

$$j \cdot (1 - o(1))$$

Substituting the definition of j

$$\frac{\ln n}{2\ln\ln n} \cdot (1 - o(1))$$

Hence, we have the lower bound $\Omega\left(\frac{\log n}{\log\log n}\right)$ □

3.3.2 log Competitive for Matroid Domain

The algorithm does not need to know the rank k of the matroid in advance; it only requires the total number of elements n and oracle access to the matroid structure (i.e., a way to check if a given set is independent).

Lemma 3.9. *The Threshold Weight Algorithm is an online $O(\log k)$-competitive algorithm for any matroid with (possibly unknown) rank k.*

Proof. We will assume that the rank k is sufficiently large enough, namely $k \geq 34$. The algorithm can be thought of in two phases, sampling and selection

1. Observe an initial set of elements S without selecting any. This set S is formed by including each of the n elements independently with a probability of $1/2$
2. Find a maximum-cardinality independent set within the sample S. Let k^* be four times the size of this set— serves as a rough estimate of the matroid's true rank k
3. Find the element l^* in the sample S with the highest weight. Choose an integer p uniformly at random from the set $\{0, 1, \ldots, \lceil \log k^* \rceil\}$. Set the threshold weight to $\theta = \frac{w(l^*)}{2^p}$

4. Initialize the set of selected elements B to be the empty set
5. For every subsequent element l_t that was not in the sample S, if $w(l_t) \geq \theta$ and $B \cup \{l_t\}$ is independent, then we can add l_t to B

Let $B^* = \{x_1, x_2, \ldots, x_k\}$ be the maximum-weight basis (the optimal offline solution), sorted by weights i.e. $w(x_1) > \ldots > w(x_k)$. There exists some maximal $q \leq k$ such that $w(x_q) \geq \frac{w(x_1)}{k}$. The elements of B^* that have a weight less than $\frac{w(x_1)}{k}$ must sum up to less than $w(x_1)$, as there can be at most $k-1$ elements with a weight less then $\frac{w(x_1)}{k}$. Hence, they contribute less than half the weight to B^* as a whole. Therefore,

$$\frac{1}{2} \cdot w\left(B^*\right) < w(x_q) + \ldots + w(x_1)$$

Let $A \subseteq \mathcal{U}$ be any (random) set and define $n_i(A)$ to be the number of elements of A whose weight is at least $w(x_i)$ and similarly define $m_i(A)$ to be the number of elements of A whose weight is at least $\frac{w(x_i)}{2}$. Trivially, $n_i(B^*) = i$; so, the sum of the top-q elements in B^* is

$$\begin{aligned}\sum_{i=1}^{q} w(x_i) &= \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) \cdot i + (q \cdot w(x_q)) \\ &= \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) \cdot n_i(B^*) + (n_q(B^*) \cdot w(x_q))\end{aligned}$$

Combining our previous observation and the expansion, given that B is the set produced by the algorithm outlined above, the weight of B is at least

$$\frac{1}{2}\left(\sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) \cdot m_i(B) + (m_q(B) \cdot w(x_q))\right)$$

Shifting gears, we now want to show that for any $i \in \{1, \ldots, q\}$,

$$\mathbb{E}[m_i(B)] \geq \Omega\left(\frac{i}{\log k}\right)$$

and since $n_i(B^*) = i$, this implies that $\mathbb{E}[m_i(B)]$ is within a log factor of $n_i(B^*)$.

I Guess We're Lucky

Consider the case where $i = 1$. With probability of at least $\frac{1}{4}$, the sample set S does not contain the maximal-weight element (x_1) but does contain the second-highest weight element (x_2). Conditioning on this, the likelihood the second-highest weight element becomes the threshold θ (this occurs when $p = 0$), then the algorithm is guaranteed to select the element x_1, is $\frac{1}{4+\log k}$ Therefore,

$$\mathbb{E}[m_1(B)] \geq \frac{1}{4(4 + \log k)}$$

We Got Kinda Lucky— An Informal Sketch

Consider an arbitrary element x_i from the optimal basis B^*. For the algorithm to be in a position to select an element of at least this weight, a couple of things need to happen, given that we have the actual rank k

1. We have with probability $\frac{1}{2}$ that the maximal-weight element x_1 is in the sample set
2. The randomly chosen parameter p is such that $\frac{w(x_1)}{2^p} \leq w(x_i) < \frac{w(x_1)}{2^{p-1}}$

The choice of p (and thus θ) is independent of which other elements x_j where $j \in \{2, \ldots, i\}$ fall into the sample. Considering the $i-1$ optimal elements $\{x_2, \ldots, x_i\}$, each of these arrives independently after the sample with probability $\frac{1}{2}$. By linearity of expectation, the expected number of these elements that are available for selection (i.e., not in S) is $\frac{i-1}{2}$.

The probability that the threshold is correctly set is $\Omega\left(\frac{1}{\log k}\right)$. In this event, we expect $\frac{i-1}{2}$ high-weight elements to be available for selection. The matroid exchange property guarantees that if this many high-weight, eligible elements appear, the algorithm can indeed select that many elements, resulting in an expected weight contribution proportional to $\frac{i-1}{\log k}$. This line of reasoning leads to the $O(\log k)$-competitive ratio.

We're Unlucky— A Formal Sketch

Since we had explicitly stated that algorithm does not need to be aware of the rank k and instead uses an estimate k^* which is dependent on the sample set S. Also, we must consider that if k^* is too small (specifically, $\lceil \log k^* \rceil < i$), then the algorithm may not choose a suitable p that would ensure picking enough of the elements $\{x_2, \ldots, x_i\}$. To handle the subtle conditioning, we introduce a set of formal random events:

- F_j is the event that the element x_j for $j \in \{1, \ldots, j\}$ is not in the sample
- E^*_κ is the event that the estimate k^* satisfies $\lceil \log k^* \rceil = \lceil \log k \rceil + \kappa$ where $\kappa \in \{0, 1, 2\}$
- E^* is the event that the rank estimate is 'good,' which we will define as the union of the possible events above: $E^* = E^*_0 \cup E^*_1 \cup E^*_2$, which means $\lceil \log k \rceil \leq \lceil \log k^* \rceil \leq 2 + \lceil \log k \rceil$
- E_i is the even that p sets a 'good' threshold based on the bounds $\frac{w(x_1)}{2^p} \leq w(x_i) < \frac{w(x_1)}{2^{p-1}}$

It should be immediately evident that $\mathbb{P}\left(F_j\right) = \frac{1}{2}$ for any j and that they are independent of each other.

Any subset of a matroid cannot have a rank larger than the rank of the matroid, $k^* \leq 4k$, thus $\lceil \log k^* \rceil \leq \lceil \log 4k \rceil \equiv \lceil \log k^* \rceil \leq \lceil \log k \rceil + 2$. Thus, to show that E^* occurs with high probability when can show that the converse $k^* < k$ occurs with low probability. Let $A^* = B^* \cap S$ be the elements of the optimal solution that are in the sample. Since A^* is an independent set, its size $|A^*|$ is at most the rank k and hence $k^* \geq 4|A^*|$. The expected size is $\mathbb{E}[|A^*|] = \frac{k}{2}$.

Using the Chernoff Bound and setting $\delta = 1/2$

$$\begin{aligned}
\mathbb{P}\left(|A^*| < (1-\delta)\,\mathbb{E}[|A^*|]\right) &\leq \exp\left(-\frac{\delta^2 2\,\mathbb{E}[|A^*|]}{2}\right) \\
\implies \mathbb{P}\left(|A^*| < \frac{1}{2}\cdot\mathbb{E}[|A^*|]\right) &\leq \exp\left(-\frac{\mathbb{E}[|A^*|]}{8}\right) \\
\implies \mathbb{P}\left(|A^*| < \frac{k}{4}\right) &\leq \exp\left(-\frac{k}{16}\right)
\end{aligned}$$

If $|A^*| < \frac{k}{4} \implies k^* = 4|A^*| < k$, then

$$\begin{aligned}
\mathbb{P}(k^* < k) &\leq \mathbb{P}\left(|A^*| < \frac{k}{4}\right) \\
&\leq \exp\left(-\frac{k}{16}\right)
\end{aligned}$$

Conditioning on $\neg F_1$ (the event that x_1 is in the sample), then $l^* = x_1$ such that the threshold is $\theta = \frac{w(x_1)}{2^p}$. Hence, the event E_i occurs iff $p = \left\lceil \log\left(\frac{w(x_1)}{w(x_i)}\right)\right\rceil$. Also conditioning on a specific good-estimate event E^*_κ. This means p is chosen uniformly from $\{0, 1, \ldots, \lceil \log k\rceil + \kappa\}$. This set has $\lceil \log k\rceil + \kappa + 1$ elements. The paper uses the looser upper bound of $4 + \log k$ for simplicity. The value $p = \left\lceil \log\left(\frac{w(x_1)}{w(x_i)}\right)\right\rceil$ is a valid candidate. Since $w(x_i) \geq w(x_k)$, we have $\frac{w(x_1)}{w(x_i)} \leq k$, so $\log\left(\frac{w(x_1)}{w(x_i)}\right) \leq \log k$. The required p is within the range of possible choices. Therefore, the probability of picking exactly the right p is at least $1/(4 + \log k)$. Hence,

$$\mathbb{P}\left(E_i \mid E^*_\kappa \cap \neg F_1\right) \geq \frac{1}{4 + \log k}$$

Whenever $E_i \cap E^* \cap \neg F_1$ happens, any element $j \leq i$ not in the sample is eligible for choosing by the algorithm in the following sense: j would be picked if it were the first element after the sample, and the only reason why it wouldn't be picked is that the algorithm instead picked an element with weight at least half that of j. That is, $\mathbb{P}\left(F_j \cap \neg F_1 \cap E^* \cap E_\kappa\right)$; thus,

$$\begin{aligned}
\mathbb{P}\left(F_j \cap \neg F_1 \cap E^* \cap E_i\right) &= \sum_{\kappa=0}^{2} \mathbb{P}\left(F_j \cap \neg F_1 \cap E^*_\kappa \cap E_i\right) \\
&= \sum_{\kappa=0}^{2} \mathbb{P}\left(F_j \cap \neg F_1 \cap E_i \mid E^*_\kappa\right) \cdot \mathbb{P}\left(E^*_\kappa\right) \\
&= \sum_{\kappa=0}^{2} \mathbb{P}\left(F_j \cap \neg F_1 \mid E^*_\kappa\ \right) \cdot \mathbb{P}\left(E_i \mid E^*_\kappa\right) \cdot \mathbb{P}\left(E^*_k\right)
\end{aligned}$$

The event E_i depends on p. The distribution of p depends only on k^*. The event E^*_κ fixes the value of $\lceil \log k^*\rceil$. So, conditioned on E^*_κ, the distribution of p is fixed. since we already conditioned on E^*_κ, the event $F_j, \neg F_1$ doesn't change the distribution of p further.

Therefore, E_i is conditionally independent given E^*_κ. Continuing,

$$\begin{aligned}\sum_{\kappa=0}^{2} \mathbb{P}\left(F_j \cap \neg F_1 \mid E^*_\kappa\ \right) \cdot \mathbb{P}\left(E_i \mid E^*_\kappa\right) \cdot \mathbb{P}\left(E^*_k\right) &\geq \frac{1}{4+\log k} \cdot \sum_{\kappa=0}^{2} \mathbb{P}\left(F_j \cap \neg F_1 \mid E^*_\kappa\ \right) \cdot \mathbb{P}\left(E^*_k\right) \\ &= \frac{1}{4+\log k} \cdot \mathbb{P}\left(F_j \cap \neg F_1 \cap E^*\right) \\ &\geq \frac{1}{4+\log k} \cdot \left(\mathbb{P}\left(F_j \cap \neg F_1\right) - \mathbb{P}\left(\neg E^*\right)\right) \\ &\geq \frac{1}{4+\log k} \cdot \left(\frac{1}{4} - \exp\left(-\frac{k}{16}\right)\right) \\ &\geq \frac{1}{8(4+\log k)}\end{aligned}$$

For each $x_j \in \{x_2, \ldots, x_i\}$, the probability of it being eligible is at least $\frac{1}{8(4+\log k)}$. By linearity of expectation, the total expected number of such candidates from this set is $\frac{i-1}{8(4+\log k)}$ which is at least $\frac{i}{16(4+\log k)}$. Hence,

$$\mathbb{E}[m_i(B)] \geq \frac{1}{16(4+\log k)} = \frac{n_i(B^*)}{16(4+\log k)}$$

The previous argument relied on an asymptotic property ($\exp(-k/16)$ being small). For small, constant k, this doesn't hold. Consider the event where x_1 is not in the sample, but $x_2$is. This happens with probability $1/4$. In this case, $l^* = x_2$. The algorithm chooses $p = 0$ with probability at least $1/(4 + \log k)$. When the element x_1 arrives, $w(x_1) > \theta$. $B \cup x_1$ must be independent if B contains elements chosen so far. The algorithm is guaranteed to select x_1. The probability of this scenario is at least $(1/4) \cdot (1/(4 + \log k)$. So,

$$\mathbb{E}[w(B)] \geq \frac{w(x_1)}{4(4+\log k)}$$

The total optimal weight $w(B^*)$ is at most $k \cdot w(x_1)$. Since $k < 34$, $w(B^*) \leq 33 \cdot w(x_1)$. Hence, the competitive ratio is bounded by a constant $33 \cdot 4 \cdot (4 + \log 33)$, which is $O(1)$. $\square$

3.4 Uniform Matroids

The algorithm that will be outlined below, will require to sample some of the elements that will be presented. Thus, we define a timestamp s and importantly $s < n$ such that the algorithm simply observes s elements and does not select any of them. We will refer to s as the *threshold time*. The algorithm can then accept up to k elements in the remaining $n - s$ options presented, as it sees fit. The elements observed during s will be referenced as the *sample set*, S.

At any given time t, the algorithm maintains a *reference set*, denoted as R_t. This set consists of the k elements with the highest weights observed up to time $t-1$. We will denote the smallest weight element as θ_t, formally defined as $\theta_t = \text{argmin}_{i \in R_t} w(i)$ and coined as the *threshold element*. The composition of R_t and the identity of θ_t are determined solely by the sequence of observed elements and are independent of any selections made by the algorithm.

By what we have stated above, no elements are selected during the first s steps. Now, consider a timestamp t such that $s < t \leq n$. We will abuse notation and also claim that the element at timestamp t is also t. The algorithm selects t if either

- $|R_t| < k$, i.e. fewer than k elements have been reserved
- $w(\theta_t) < w(t)$ and $\theta_t \in S$, i.e. the threshold element was in the sample, and i has a higher weight than it

We should recognize that the algorithm never selects more than k elements. First, if $s < k$, then until time k, all elements are picked, for a total of $k - s$ elements. Then, $|R_{k+1} \cap S| = s$. When $s \geq k$,there are at least k elements to choose from, so the reference set will pick k elements such that $|R_{s+1}| = k$. Each selection strictly decreases the cardinality of the set $R_t \cap S$. Thus, at most s (respectively, k) elements can be picked by the algorithm, for a total of k in either case.

Theorem 3.10. *For every $n \in \mathbb{Z}^+$, the competitive ratio of the algorithm above where $s = \left\lfloor \frac{n}{e} \right\rfloor$ is less than e*

Proof. Let's consider one of the top k elements, which we'll call i. The probability that element i arrives at any specific time t is exactly $1/n$. For i to be selected upon its arrival at time t (where $t > s$), the threshold element at that time, θ_t must have been part of the sample set S. The conditional probability of this is $\frac{s}{t-1}$, as the first $t-1$ elements are in a uniformly random order. Therefore, the total probability of selecting element i is the sum of the probabilities of it being selected at each possible time t after the sampling period

$$\sum_{t=s+1}^{n} \frac{1}{n} \cdot \frac{s}{t-1} = \frac{s}{n} \cdot \sum_{t=s+1}^{n} \frac{1}{t-1}$$
$$= \frac{s}{n} \cdot \sum_{j=s}^{n-1} \frac{1}{j}$$

We define the m-th Harmonic Number as $H_m = \sum_{i=1}^{m} \frac{1}{i}$; hence, we can rewrite the sum as

$$\frac{s}{n} \cdot \sum_{j=s}^{n-1} \frac{1}{j} = \frac{s}{n} \left(H_{n-1} - H_{s-1} \right)$$

We now claim that for $n \geq 3$ and $s = \left\lfloor \frac{n}{e} \right\rfloor$ that

$$\frac{s}{n}\left(H_{n-1} - H_{s-1}\right) > \frac{1}{e}$$

Since $x \mapsto \frac{1}{x}$ is convex, an immediate fact is

$$\begin{aligned}\int_{t-1}^{t} \frac{1}{x} &< \frac{1}{2}\left(\frac{1}{t-1} + \frac{1}{t}\right) \\ &= \frac{1}{t-1} - \frac{1}{2}\left(\frac{1}{t-1} - \frac{1}{t}\right)\end{aligned}$$

Now applying to the harmonic sum difference from above,

$$\begin{aligned}\left(H_{n-1} - H_{s-1}\right) &= \sum_{t=s}^{n-1} \frac{1}{t} \\ &> \int_{s}^{n} \frac{1}{x}\,\mathrm{d}x + \frac{1}{2}\left(\frac{1}{s} - \frac{1}{n}\right) \\ &= \ln\left(\frac{n}{s}\right) + \frac{1}{2}\left(\frac{1}{s} - \frac{1}{n}\right)\end{aligned}$$

Multiplying by $\frac{s}{n}$, we have

$$\begin{aligned}\frac{1}{e} &< \frac{s}{n}\left(\ln\left(\frac{n}{s}\right) + \frac{1}{2}\left(\frac{1}{s} - \frac{1}{n}\right)\right) \\ &= \frac{s}{n}\ln\left(\frac{n}{s}\right) + \frac{1}{2n} - \frac{s}{2n^2}\end{aligned}$$

Now defining $f(s,n) = \frac{s}{n}\ln\left(\frac{n}{s}\right) + \frac{1}{2n} - \frac{s}{2n^2}$. The relationship $s = \left\lfloor \frac{n}{e} \right\rfloor$ implies $es \leq n < e(s+1)$. We analyze $f(s,n)$ for a fixed s as n varies in this range. The derivative of $f(s,n)$ with respect to n is

$$f'(n) = \frac{1}{n^2}\left(s\left(1 - \ln\frac{n}{s}\right) - \frac{1}{2} + \frac{s}{n}\right)$$

Since $n \geq es$, we have $\frac{n}{s} \geq e$, so $\ln(n/s) \geq 1$. This makes the term $1 - \ln(n/s)$ non-positive, and thus, $f'(n)$ is negative. Hence, we need only check the inequality at the minima, specifically $e(s+1)$. Evaluating at $(s, e(s+1))$, using the fact $\ln(x+1) \geq \frac{x}{1+x}$, we have

$$\begin{aligned}\ln\left(\frac{n}{s}\right) &= \ln\left(\frac{e(s+1)}{s}\right) \\ &= 1 + \ln\left(1 + \frac{1}{s}\right) \\ &\geq 1 + \frac{\frac{1}{s}}{1 + \frac{1}{s}} \\ &= 1 + \frac{1}{s+1} \\ &= \frac{s+2}{s+1}\end{aligned}$$

Working out the entire inequality

$$\begin{aligned}
f(s, e(s+1)) &= \frac{s}{e(s+1)} \ln\left(\frac{e(s+1)}{s}\right) + \frac{1}{2e(s+1)} - \frac{s}{2(e(s+1))^2} \\
&\geq \left(\frac{s}{e(s+1)} \cdot \frac{s+2}{s+1}\right) + \frac{1}{2e(s+1)} - \frac{s}{2e^2(s+1)^2} \\
&= \frac{s^2+2s}{e(s+1)^2} + \frac{1}{2e(s+1)} - \frac{s}{2e^2(s+1)^2} \\
&= \frac{(s^2+2s)\cdot 2e}{2e^2(s+1)^2} + \frac{e(s+1)}{2e^2(s+1)} - \frac{s}{2e^2(s+1)^2} \\
&= \frac{2es^2+4es+es+e-s}{2e^2(s+1)^2} \\
&= \frac{2es^2+4es+2e}{2e^2(s+1)^2} + \frac{es-s-e}{2e^2(s+1)^2} \\
&= \frac{2e(s^2+2s+1)}{2e^2(s^2+2s+1)} + \frac{s(e-1)-e}{2e^2(s+1)^2} \\
&= \frac{1}{e} + \frac{s(e-1)-e}{2e^2(s+1)^2}
\end{aligned}$$

This clearly greater than e^{-1} whenever $s \geq 2$. Considering the case of $s = 1$, we have we have $e \leq n < 2e$, so n can be 3, 4, or 5. Considering the worst case of $n = 5$,

$$\begin{aligned}
\frac{1}{5}(H_4) &= \frac{1}{5}\left(\frac{25}{12}\right) \\
&= \frac{5}{12} \\
&> \frac{1}{e}
\end{aligned}$$

Hence, this is true for any $s \in \mathbb{Z}^+$. We have shown that the probability of selecting any single element from the optimal set is strictly greater than $\frac{1}{e}$.

Since we have proven that each of the k highest-weight elements has a probability greater than $\frac{1}{e}$ of being selected, it immediately follows that the expected weight of our chosen set is also greater than $\frac{1}{e}$ times the maximum possible weight— confirming that the algorithm is e-competitive, guaranteeing its performance is always within a constant factor of the optimal solution. $\square$

3.5 Unit-Demand Domain

3.5.1 From Unit-Demand to Transversal Matroids

Consider a market composed of two different sets

- a set of n agents, denoted by L
- a set of m non-identical items, denoted by R

(the naming of the set will become apparent later). Each agent i has a non-empty specific subset of items $T_i \subseteq R$ that they desire. If an agent receives any item from their desired set T_i, they obtain a value of $v_i > 0$; otherwise, their value is 0. Since, the agent values each item the same, we will refer to this as the "single-value" domain. We assume that there is a constant d such that $|T_i| \leq d$ for all i; that is, each agent demands one of at most d items.

An outcome in this market is a matching between agents and items, ensuring that no item is assigned to more than one agent and no agent receives more than one item. The objective is to find a matching that maximizes the total value of the satisfied agents. It is assumed that each agent's valuation v_i is private to them, but it is possible that the set of desirables T_i is also private, and such a scenarios is known as the unknown unit-demand domain.

The unit-demand domain is a specific instance of a matroid domain, where the independent sets of agents form a transversal matroid. A transversal matroid is constructed from a bipartite graph $G = (L \cup R, E)$, where L represents one set of vertices and R the other. In our context:

- L (set of agents) form the left side of a bipartite graph
- R (set of times) form the right side of a bipartite graph
- An edges exists from an agent l to an item r iff r is in T_l
- The weight of the edges is v_l

$S \subseteq L$ is independent if there exists a matching in the bipartite graph that covers all agents in S. In other words, every agent in S can be simultaneously assigned a unique item from their desired set. The constraint that each agent desires at most d items, informs us that the maximal degree of any 'left vertex' is at most d. This is referred to as a transversal matroid of bounded left-degree.

3.5.2 The Price-Sampling Algorithm

The algorithm defined below creates a truthful online mechanism given that the values of the agents are revealed online, the structure of the bipartite graph is known in advance.

Observer the first $s = \left\lceil \frac{n}{2} \right\rceil$ agents that arrive without assigning them an item. The set of observed agents is the sample, denoted by S. For each item $r \in R$, set a price $p(r)$. This price is determined by the maximum value among the agents in the sample S who desire item r. Formally, let $l^*(r)$ be the agent in S with the highest value who wants r; thus, $p(r) = v_{l^*(r)}$. If no agent desires r, then $p(r) = 0$. For any following agent l after the sample period, identify the set of items $R^*(l)$ that agent l desires and can afford, meaning the item's price is less than the agent's value i.e $p(r) < v_l$. If this set $R^*(l)$ is not empty, match agent l to the item in $R^*(l)$ with the lowest price (this is the online decision).

Otherwise, agent l is not matched to any item and the next agent is given.

Proof. Let OPT be the maximal weight matching of the bipartite graph. For all $r \in R$, let $m(r)$ denote the agent matched to item r. If r is unmatched, then $m(r) = \emptyset$. Let $h(r)$ be the agent who values r the highest, and $s(r)$ be the agent who values r the second-highest. Finally, define $H = \bigcup_{r \in R}\{h(r)\}$ i.e. the set of all the "highest-value" agents for every item.

By definition, for any item r, the value of the agent matched to it in the optimal solution, $w(m(r))$, cannot be greater than the value of the highest-value agent desiring that item, $w(h(r))$. That is,

$$\begin{aligned} w(\mathrm{OPT}) &= \sum_{r \in R} w(m(r)) \\ &\leq \sum_{r \in R} w(h(r)) \end{aligned}$$

$\sum_{r \in R} w(h(r))$ can be rewritten by summing over the agents in the set H . Since each agent $l \in H$ is constrained to desiring at most d items, it can be the highest-value agent for at most d different items. Hence,

$$\sum_{r \in R} w(h(r)) \leq d \cdot w(H)$$

Now, shifting gears, let's fix an agent, say $l \in H$. Necessarily, there exists at least one item, say r, such that $h(r) = l$. Furthermore, define E to be $E = \{s(r) \in S \wedge l \notin S\}$. In words, E is the event that the second-highest value agent i.e. $s(\cdot)$ for item r is in the sample, while the highest-value agent i.e. $h(\cdot)$, namely l, is not.

The probability that $s(r)$ is in S is

$$\frac{|S|}{n} = \frac{\lceil \frac{n}{2} \rceil}{n}$$

Conditioning on $s(r) \in S$, the probability that l is not is

$$\frac{n - |S|}{n-1} = \frac{n - \lceil \frac{n}{2} \rceil}{n-1} = \frac{\lfloor \frac{n}{2} \rfloor}{n-1}$$

Hence, we have

$$\begin{aligned} \mathbb{P}(E) &= \frac{\lceil \frac{n}{2} \rceil}{n} \cdot \frac{\lfloor \frac{n}{2} \rfloor}{n-1} \\ &= \frac{\lceil \frac{n}{2} \rceil \cdot \lfloor \frac{n}{2} \rfloor}{n(n-1)} \end{aligned}$$

Consider the case where n is even say $n = 2k$,

$$\begin{aligned} \mathbb{P}(E) &= \frac{\lceil \frac{2k}{2} \rceil \cdot \lfloor \frac{2k}{2} \rfloor}{2k(2k-1)} \\ &= \frac{k^2}{4k^2 - 2k} \\ &= \frac{k}{4k-2} \\ &> \frac{k}{4k} \\ &= \frac{1}{4} \end{aligned}$$

Otherwise, if n is odd, say $n = 2k+1$

$$\begin{aligned}
\mathbb{P}(E) &= \frac{\left\lceil \frac{2k+1}{2} \right\rceil \cdot \left\lfloor \frac{2k+1}{2} \right\rfloor}{2k+1(2k+1-1)} \\
&= \frac{(k+1)k}{4k^2+2k} \\
&= \frac{k+1}{4k+2} \\
&> \frac{k+1}{4k+4} \\
&= \frac{1}{4}
\end{aligned}$$

Thus, we can claim

$$\mathbb{P}(E) > \frac{1}{4}$$

Given that E occurs, we know that $s(r) \in S$ and that $h(r) \notin S$; therefore, the price of r is $p(r) = w(s(r))$, the second-highest valuation. By definition, $w(h(r)) > w(s(r))$, so the item goes to player $l = h(r)$. Therefore, $R^*(l)$ is not empty, and the algorithm will match agent l to some item (either r or another even cheaper item). So, if E occurs, agent l is guaranteed to be matched. Since $\mathbb{P}(E) > 1/4$, the probability of matching any agent $l \in H$ is more than $1/4$.

The expected weight is

$$\begin{aligned}
\mathbb{E}\left[w(\mathrm{ALG})\right] &= \sum_{l \in L} w(l) \cdot \mathbb{P}(l \text{ is matched}) \\
&\geq \sum_{l \in H} w(l) \cdot \mathbb{P}(l \text{ is matched}) \\
&> \sum_{l \in H} \frac{w(\mathrm{OPT})}{4d}
\end{aligned}$$

□

3.5.3 Truthfulness is the Only Way

An agent in the sample is never matched and receives a utility of 0, regardless of what they declare their value or desired items to be. Therefore, they have no incentive to lie.

An agent arriving after the sample has a fixed price for the items. Among the items they truly desire, find the one with the lowest price. If that price is less than their true value, they are matched to it and get a positive utility. If they lie, there are two scenarios:

- If they report a lower value, they might not be matched to an item they could have afforded, resulting in a lower or zero utility. If they report a higher value, it doesn't change which items are affordable, so their outcome remains the same.

- If they report a different set of desired items, they might be matched to an item they don't actually want, resulting in zero utility. They cannot gain by adding items to their desired set because the prices are fixed.

3.6 Graphic Matroids

Given an undirected graph $G = (V, E)$, where $n = |E|$, we can construct a bipartite graph $G' = (L, R, E')$.

1. For every node in $v \in V$, we create a corresponding vertex on the right side of our new bipartite graph such that $v \in R$ such that $V = R$.
2. For every edge $e = (u, v) \in E$, we create a corresponding node for the left side of the bipartite graph, call it $l_{uv} \in L$ such that $L = E$.
3. An edge exists in G' between a left-node l_{uv} and a right-node $r \in R$ iff the edge (u, v) was incident on r. That is, for every left-node l_{uv} there is an edge to both u and v.

This construction results in a bipartite graph G' where every vertex in L has a degree of exactly 2— this is a transversal matroid with a bounded left-degree of $d = 2$.

Any forest in G can be mapped to a matching in G'. Root the forest arbitrarily. For each edge (u, v) direct it away from the root (e.g., if u is 'closer' to the root, the edge becomes $u \to v$). In the bipartite graph G', include the edge (l_{uv}, v) in the matching. Since each vertex in the directed forest has at most one incoming edge, each right-side vertex $r \in R$ will be matched at most once.

However, the converse is not true. Consider the following example where

$$G = (\{1, 2, 3\}, \{(1, 2), (2, 3), (3, 1)\} \qquad \text{(a triangle)}$$

The corresponding bipartite graph would be made up of the two vertex sets $L = \{l_{12}, l_{23}, l_{31}\}$ and $R = \{1, 2, 3\}$. The set of edges would be $E' = \{(l_{12}, 1), (l_{12}, 2), (l_{23}, 2), (l_{23}, 3), (l_{31}, 3), (l_{31}, 1)\}$. It should be immediately apparent that there is a cycle, which is not an independent set.

This leads to an issue when applying the Price-Sampling algorithm from Section 3.5.2. To account for this, we append an extra rule: The edge l_{uv} is only matched if it is affordable and if it does not create a cycle.

Proof. Let $h(r)$, $s(r)$, $m(r)$, and H be defined as in the previous proof, but now with respect to the bipartite graph G' where $r \in R$ is a vertex from the original graph G. Here, $h(r)$ represents the highest-weight edge incident to vertex r. As we are dealing with a transversal matroid with a bounded left-degree of $d = 2$, we get the following bound

$$w(\text{OPT}) \leq 2 \cdot w(H)$$

Now, let's fix an edge $l = l_{uv} \in H$. By definition, this edge has the highest weight for at least one of its endpoints. WLOG, let us assume $l = h(u)$. For our algorithm to select l_{uv}, two conditions must be met: it must be affordable, and it must not create a cycle with the edges already selected.

ß

Let $s(u)$ be the second-highest weight edge incident to vertex u. Furthermore, let L' be the highest-weight edge incident to vertex v, excluding l_{uv}. Formally, if $l_{uv} = h(v)$, then $L' = s(v)$; otherwise, $L' = h(v)$. We define the event $\mathcal{E}$ as

$$\mathcal{E} = \{s(u) \in S \wedge L' \in S \wedge l_{uv} \notin S\}$$

In words, $\mathcal{E}$ is the event where the edges $s(u)$ and L' are both in the initial sample S, while our target edge l_{uv} is not. These three edges, l_{uv}, $s(u)$, and L', are distinct. We can calculate the probability of this event by considering the uniformly random permutation of the n edges.

$$\begin{aligned}\mathbb{P}(\mathcal{E}) &= \mathbb{P}(s(u) \in S) \cdot \mathbb{P}(L' \in S | s(u) \in S) \cdot \mathbb{P}(l_{uv} \notin S | s(u), L' \in S) \\ &= \frac{\lceil n/2 \rceil}{n} \cdot \frac{\lceil n/2 \rceil - 1}{n-1} \cdot \frac{\lfloor n/2 \rfloor}{n-2}\end{aligned}$$

Consider if n is even such that $n = 2k$

$$\begin{aligned}\mathbb{P}(\mathcal{E}) &= \frac{k}{2k} \cdot \frac{k}{2k-2} \cdot \frac{k-1}{2k-1} \\ &> \frac{1}{2} \cdot \frac{k}{2k} \cdot \frac{k-1}{2(k-1)} \\ &= \frac{1}{8}\end{aligned}$$

Otherwise, if n is odd such that $n = 2k+1$

$$\begin{aligned}\mathbb{P}(\mathcal{E}) &= \frac{k+1}{2k+1} \cdot \frac{k}{2k-1} \cdot \frac{k}{2k} \\ &> \frac{k+1}{2(k+1)} \cdot \frac{k}{2k} \cdot \frac{1}{2} \\ &= \frac{1}{8}\end{aligned}$$

Thus, we can claim

$$\mathbb{P}(\mathcal{E}) > \frac{1}{8}$$

1. **Pricing:** Since $s(u)$ and L' are in the sample, the prices of the endpoints u and v are set based on the highest-value incident edges seen in the sample. Specifically, $p(u) = w(s(u))$ and $p(v) = w(L')$.

2. **Arrival:** Because $l_{uv} \notin S$, it arrives during the second phase of the algorithm.

3. **Affordability:** By our initial assumption, $l_{uv} = h(u)$, so its weight $w(l_{uv})$ is greater than $w(s(u))$. This means $w(l_{uv}) > p(u)$, making the edge affordable.

4. **Cycle Check:** Because the prices $p(u)$ and $p(v)$ were set by other edges ($s(u)$ and L'), no edge incident to u or v could have been selected by the algorithm before l_{uv} arrives. If an edge incident to u had been selected, its weight would have had to exceed $p(u) = w(s(u))$, but the only such edge is $l_{uv} = h(u)$, which has not arrived yet. We can apply a similar train of thought when considering vertex v. Therefore, at the moment l_{uv} arrives, vertices u and v belong to separate components in the forest being constructed by the algorithm. Adding the edge (u, v) will connect these two components without creating a cycle.

Thus, if event $\mathcal{E}$ occurs, the edge l_{uv} is guaranteed to be selected by the modified algorithm. Since $\mathbb{P}(\mathcal{E}) > 1/8$, the probability of selecting any given edge $l \in H$ is at least $1/8$.

Finally, we can lower-bound the expected weight of the algorithm's solution, denoted

$w(\text{ALG})$.

$$\begin{aligned}\mathbb{E}\left[w(\text{ALG})\right] &= \sum_{l\in L} w(l)\cdot \mathbb{P}(l \text{ is selected})\\ &\geq \sum_{l\in H} w(l)\cdot \mathbb{P}(l \text{ is selected})\\ &\geq \sum_{l\in H} w(l)\cdot \frac{1}{8}\\ &= \frac{w(H)}{8}\end{aligned}$$

Using our initial bound, $w(H) \geq \frac{w(\text{OPT})}{2}$, we can conclude:

$$\mathbb{E}\left[w(\text{ALG})\right] \geq \frac{w(\text{OPT})/2}{8} = \frac{w(\text{OPT})}{16}$$

The mechanism is trivially truthful, as the edges (the "agents") have no private information to misreport, and the pricing and cycle-check rules are deterministic. □

3.7 Truncation of Matroids

Truncation is an operation that decreases the rank of a matroid by discarding all independent sets whose cardinality exceeds a specified limit— the outcome set is modified by eliminating all outcomes that satisfy more than a specified number of agents.

Consider an obvious problem: a company is looking to hire n new employees given that no two hires have the same technical expertise. This can be modeled by an truncated partition matroid, where the ground set $\mathcal{U}$ is partitioned into subsets, and the independent sets are those that have at most n elements while also intersecting each partition in at most one element.

Define $M = (\mathcal{U}, \mathcal{I})$ be a matroid of rank k, and let $r \leq k$. The truncation of M, denoted as $\tau_r(M) = (\mathcal{U}, \tau_r(\mathcal{I}))$, is a matroid where the collection of independent sets $\tau_r(\mathcal{I})$ consists of all sets in $\mathcal{I}$ that have at most r elements. If D is a matroid domain with an outcome set Ω, then $\tau_r(D)$ is the domain obtained by removing all outcomes $\omega \in \Omega$ that satisfy more than r agents.

3.7.1 The Reduction Strategy

The goal is to transform any c-competitive algorithm for a matroid M into a constant-factor competitive algorithm for its truncation $\tau_r(M)$.

1. The algorithm first observes a large fraction of the elements (about 3/4 of them) without making any selections. This set of observed elements forms a sample, S. From this sample, it computes a baseline solution— a maximum-weight independent set $B \subseteq S$ of size at most r. This set B serves as a 'test' for the elements that arrive later.

2. For the remaining 1/4 of the elements, the algorithm "filters" them. An arriving element x is evaluated based on whether it can improve the current baseline solution B. If x improves the solution, it is presented to the inner algorithm, ALG, with its true weight. If it does not offer an improvement, it is assigned a weight of zero when presented to ALG.

3. The sample elements from S are not processed separately from the new elements. Instead, the algorithm randomly interleaves the elements from S (presented to ALG with weight 0) with the new, unseen elements.

3.7.2 The Truncated Matroid Algorithm

Let ALG be a c-competitive algorithm for the secretary problem on matroid M. WLOG, assume the algorithm for the truncated matroid $\tau_r(M)$, $\tau_r(\text{ALG})$, never selects an element of weight 0.

1. If $r \leq 144$, the problem is considered to have a 'small rank'. In this case, we ignore ALG and instead run Dynkin's standard secretary algorithm to pick the single maximum-weight element, which is guaranteed to be found with probability at least $1/e$.

2. Before the process begins, generate a sequence of n independent Bernoulli random variables, $z_1, z_2, \ldots, z_n$, where $\mathbb{P}(z_i = 1) = 3/4$ for all i. Let s be the total number of i for which $z_i = 1$. An element arriving at time i will be part of the "sample phase" if $z_i = 1$ and the "decision phase" if $z_i = 0$.

3. Observe the first s elements corresponding to the indices i where $z_i = 1$. Place these elements into a sample set S.

4. Compute B, the maximum-weight independent set within S that has a cardinality of at most r.

5. For each time step $i = 1, \ldots, n$:
   - If $z_i = 1$
     - Draw a uniformly random element x from the remaining elements in S and remove it from S.
     - Present this element x to the inner algorithm ALG, but with a modified weight $\hat{w}(x) = 0$. This effectively hides the element from ALG's consideration for selection.
   - If $z_i = 0$
     - Let x be the next previously unobserved element from the online sequence.
     - If x improves B, present x to ALG with its true weight, $\hat{w}(x) = w(x)$. If ALG decides to select x, then our algorithm $\tau_r(\text{ALG})$ also selects x.
     - If x does not improve B, present x to ALG with a modified weight of $\hat{w}(x) = 0$.

For the case where $r \leq 144$, Dynkin's algorithm finds the best single element with probability at least $1/e$. The optimal solution in $\tau_r(M)$ has a value of at most $r \cdot w(x_1)$, where x_1 is the maximum weight element. The expected value from Dynkin's is at least $\frac{1}{e}w(x_1)$. Hence, the competitive ratio is at most

$$\begin{aligned} \frac{r \cdot w(x_1)}{\frac{w(x_1)}{e}} &= re \\ &\leq 144e \\ &< 400 \end{aligned}$$

Otherwise, if $r > 144$, let OPT_r be the maximum-weight basis of $\tau_r(M)$. The elements of OPT_r have the property that they improve any independent set they are not part of.

Therefore, every element $x \in \mathrm{OPT}_r$ will improve our baseline set B. This means that any element of the optimal solution that appears during the decision phase ($z_x = 0$) will be passed to ALG with its true weight.

Let A be the set of all elements presented to ALG with their true weight. An element $x \in \mathrm{OPT}_r$ ends up in A if $z_x = 0$ (with probability $1/4$) and ends up in the initial sample pool S if $z_x = 1$ (with probability $3/4$). Thus, $\mathrm{OPT}_r \subseteq A \cup S$.

Let us define $\mathcal{E}$ as the event where all of the following three conditions hold:

1. $|A \cap \mathrm{OPT}_r| \geq r/6$
2. $|S \cap \mathrm{OPT}_r| \geq r/2$
3. $|(A \cup S) \cap \mathrm{OPT}_r| \leq 3r/2$

The number of elements from OPT_r that fall into A follows a binomial distribution $\mathrm{Bin}(r, 1/4)$, with mean $\mu = r/4$. Using the Chernoff Bound, the probability is

$$\begin{aligned}\mathbb{P}(|A \cap \mathrm{OPT}_r| < r/6) &= \mathbb{P}(|A \cap \mathrm{OPT}_r| < (1 - 1/3)\mu) \\ &\leq e^{-\mu(1/3)^2/2} \\ &= e^{-r/72} \\ &\leq e^{-2}\end{aligned}$$

Similarly, the number of elements from OPT_r in S has mean $\mu = 3r/4$. The probability of the second condition failing is:

$$\begin{aligned}\mathbb{P}(|S \cap \mathrm{OPT}_r| < r/2) &= \mathbb{P}(|S \cap \mathrm{OPT}_r| < (1 - 1/3)\mu) \\ &\leq e^{-\mu(1/3)^2/2} \\ &= e^{-r/24} \\ &\leq e^{-6}\end{aligned}$$

Theorem 3.11 (Karger's matroid sampling theorem [6]). *Let M be a matroid with ground set $\mathcal{U}$ in which each element has been assigned a real-valued weight. Let $p > 0$, and obtain $\mathcal{U}'$ from $\mathcal{U}$ by including each element of U independently with probability p. Let $\mathcal{I}$ be a maximum-weight independent subset of $\mathcal{U}'$. Then, for any number $\varepsilon > 0$, the probability that more than $\frac{(1+\varepsilon)r}{p}$ elements improve $\mathcal{I}$ is at most*

$$\exp\left(\frac{-\varepsilon^2 r}{2(1+\varepsilon)}\right)$$

Using Karger's matroid sampling theorem by setting $\varepsilon = 1/8$,

$$\begin{aligned}\exp\left(-\frac{\epsilon^2 r}{2(1+\epsilon)}\right) &= \exp\left(-\frac{(1/8)^2 r}{2(1+1/8)}\right) \\ &= \exp\left(-\frac{r/64}{2(9/8)}\right) \\ &= \exp\left(-\frac{r/64}{9/4}\right) \\ &= \exp\left(-\frac{r}{144}\right) \\ &\leq \exp(-1)\end{aligned}$$

Using the union bound, the probability that any of these events don't happen is

$$e^{-2} + e^{-6} + e^{-1} < 0.51$$

Thus, the event $\mathcal{E}$ occurs with probability $\mathbb{P}(\mathcal{E}) > 1 - 0.51 = 0.49$. Since $|S \cap \mathrm{OPT}_r| \geq r/2$ and B is the max-weight r-subset of S, $|B|$ must also be at least $r/2$. Furthermore, $|A \cup B| \leq |(A \cup S) \cap \mathrm{OPT}_r| \leq 3r/2$ which implies $|A| \leq r$.

Conditioning on $\mathcal{E}$, elements of A have positive modified weight, $|A| \leq r$, and ALG only picks elements of positive modified weight by assumption— the output must be an independent set of at most size r. Thus, the output of $\tau_r(\mathrm{ALG})$ is a valid independent set in $\tau_r(M)$. Let R be the maximum-weight independent set within A. The expected weight of R conditioned on $\mathcal{E}$ is at least the expected weight of the portion of the optimal solution that fell into A.

$$\begin{aligned}
\mathbb{E}[w(R) \mid \mathcal{E}] &\geq \mathbb{E}[w(A \cap \mathrm{OPT}_r) \mid \mathcal{E}] \\
&\geq \frac{r/6}{r} w(\mathrm{OPT}_r) \\
&= \frac{w(\mathrm{OPT}_r)}{6}
\end{aligned}$$

The random interleaving by the z_i variables ensures that, from ALG's perspective, the sequence of elements it sees (the set A) arrives in a uniformly random order. Since ALG is c-competitive, its expected score is at least $\frac{1}{c}$ of the optimal solution within that stream.

$$\begin{aligned}
\mathbb{E}[\text{weight of ALG's output} \mid \mathcal{E}] &\geq \frac{\mathbb{E}[w(R) \mid \mathcal{E}]}{c} \\
&\geq \frac{w(\mathrm{OPT}_r)}{6c}
\end{aligned}$$

The expected weight of the set selected by $\tau_r(\mathrm{ALG}))$ is

$$\begin{aligned}
\mathbb{E}[w(\tau_r(\mathrm{ALG}))] &\geq \mathbb{P}(\mathcal{E}) \cdot \mathbb{E}[w(\tau_r(\mathrm{ALG})) \mid \mathcal{E}] \\
&\geq 0.49 \cdot \frac{w(\mathrm{OPT}_r)}{6c} \\
&\geq \frac{w(\mathrm{OPT}_r)}{13c}
\end{aligned}$$

Chapter 4

Combinatorial Pen Testing Problem

4.1 Set Up

Definition 4.1. *Combinatorial Pen Testing*: A combinatorial pen testing instance is described by a set $N = \{1, \dots, n\}$ of pens with a 'hidden' ink level $v_1, \dots, v_n$, where each v_i is drawn independently from a distribution $\mathcal{D}_i$, and a subset $\mathcal{P}$ of the powerset of pens, representing the feasible subsets of pens. At each step $i \in [n]$, the player must perform two tests:

1. Testing: The player selects a threshold $\theta_i \in [0, \infty)$. The player then tests i with the chosen θ_i. If $v_i \leq \theta_i$, then player simply observes the value of v_i. Otherwise, the pen has a residual ink of $u_i = v_i - \theta_i$.

2. Accepting: If pen i passes the test, the player is given the choice to 'accept' or 'reject' the option irrevocably. If the player accepts the pen, then it gets added to the collection of pens they have accepted.

The goal is to output some feasible subset $P \in \mathcal{P}$ of pens, maximizing

$$\sum_{i \in P} u_i.$$

The non-specificity of $\mathcal{P}$ allows for a variety of constraints like cardinality-based, knapsack, or matroids.

There is a parallel between a combinatorial pen testing problem over n pens and a feasibility constraint $\mathcal{P}$ and an n-agent auction given the same feasibility constraint. The original levels of ink v_i correspond to the value each agent gets upon being allocated. The ink spent through testing is analogous to the prices in the auction. So, maximizing total residual ink is equivalent to optimizing consumer surplus (the sum of the values of the winning agents minus the sum of all payments).

An immediate pitfall in these incomplete parallelization is that in the pen testing environment, the information that the player obtains is whether a pen i has more ink than some decided threshold θ_i, and in doing so, irrevocably expend ink up to the threshold. The ascending-price auction fits the formulation, where the auctioneer irreversibly increases the price faced by each agent. As a result, the agent is only informed if they have a valuation at least the newly hiked price.

Definition 4.2. *Deferred-Acceptance Auctions*: A deferred-acceptance auction is composed of n stages. At each stage t, there is a set of active bidders A_t such that $A_1 \supseteq A_2 \supseteq \dots A_n$ given that A_1 is the set of all bidders in the auction. The auction is characterized by a pricing rule $\vec{p}$ which maps the history of the auction at a stage t to prices for each agent satisfying $p_i(H_t) \geq p_i(H_{t-1})$, i.e. the prices are non-decreasing for every agent over all steps. Agents can opt to drop out once the prices are updated in stage t. A_{t+1} is the set of agents in A_t that did not drop out in stage t. The mechanism terminates at stage t when A_t becomes feasible. The agents in A_t are charged based on $\vec{p}(H_t)$.

Given a stage in the DAA algorithm, we have a price p_i for any agent i. At each stage, it uses those prices as test thresholds, performing a test for a threshold $\theta_i = p_i$ for each pen i.

We will use the following benchmarks as metrics to evaluate the deferred-acceptance mechanisms against

- **Omniscient Benchmark**: Consider an omniscient algorithm/oracle in the pen testing problem that is aware of the ink levels of all pens, and as a result, does not need to test inks as a way to evaluate them— they can just choose the best pens. In the reduction to the auction environment, the omniscient benchmark corresponds to a mechanism that neither loses out on the payments nor has to stick to an increasing trajectory of prices. Thus, the omniscient benchmark corresponds to the surplus optimal sealed-bid mechanism ($\sum v_i$), since a sealed-bid format is not constrained by a gradually increasing price. The auctioneer can just see all the bids and determine the optimal allocation. The *omniscient approximation* $\pi(n)$ corresponds to the ratio between the omniscient benchmark and the performance of a given pen testing algorithm in the pen testing setting, and the omniscient benchmark and the consumer surplus of a given deferred-acceptance auction in the auction environment.
- **Standard Benchmark**: The standard benchmark corresponds to the consumer-surplus-optimal sealed-bid mechanism ($\sum v_i - \theta_i$). The goal of the auctioneer is to not to maximize total value, but to maximize the amount of value left over for the bidders after they've paid. The *standard approximation* $\gamma(n)$ is the ratio between the standard benchmark and the consumer surplus of a candidate deferred-acceptance mechanism. However, the standard benchmark does not have a natural correspondence with the pen testing problem. In the pen testing problem, you cannot separate the "price" (ink lost) from the process of discovery. The standard benchmark is a tighter upper bound to the optimal pen testing algorithm than the omniscient benchmark. The total surplus is the consumer surplus plus the auctioneer revenue. Since the ink lost in testing is "burned" and not collected by anyone, the revenue is zero in the pen testing analogy. As a result, total surplus is at as large as the consumer surplus, and this $\gamma(n) \geq \pi(n)$.

Theorem 4.3. *Consider a combinatorial pen testing environment with n pens, the analogous auction environment with an optimal omniscient approxiamtion $\zeta(n)$, and a deferred-acceptance mechanism with a standard approximation equal to $\gamma(n)$. Then, there is a pen testing algorithm that is a γ-approximation to the standard benchmark and $\pi(n) = \gamma(n)\zeta(n)$-approximation to the omniscient benchmark.*

4.1.1 Consumer Surplus Optimization & Virtual Valuations

Let us being in a simplified environment with a single agent and one item. The agent's valuation of the good is v and is drawn from some distribution $\mathcal{D}$ with a CDF $F(x) = \mathbb{P}(v \leq x)$. We define the quantile q of an agent with a value of $v \sim \mathcal{D}$ as $q = 1 - F(v)$. Necessarily, $q \in [0, 1]$ and thus we also define $v(q)$ to be the inverse demand function of F such that $F(v(q)) = 1 - q$. In other words,

$$\mathbb{P}_{\hat{v} \sim \mathcal{D}}(\hat{v} > v(q)) = q.$$

We motivate $v(q)$ given that $v = v(q)$, such that

$$\begin{aligned} \mathbb{E}\left[g(v)\right] &= \int_{v_0}^{v_1} g(v)\, dF(v) \\ &= \int_{v_0}^{v_1} g(v) f(v)\, dv, \end{aligned}$$

where $v_0 = \min v$ and $v_1 = \max v$. If $v = v_0$, then $q = 1 - F(v) = 1$ and similarly for $v = v_1$, we have $q = 0$. Continuing the work on the integral,

$$\begin{aligned}\mathbb{E}[g(v)] &= \int_{v_0}^{v_1} g(v)f(v)\,dv \\ &= -\int_1^0 g(v(q))\,dq \\ &= \int_0^1 g(v(q))\,dq,\end{aligned}$$

given that differentiating q by v, we get $dq = -f(v)\,dv$. Also, we can understand $v(q)$ as the market price at which a q-th fraction would be willing to buy the good.

We will make the following assumption for the remaining part of the paper: (i) $v(1) = 0$ and (ii) $\int_0^0 v(t)\,dt = 0$.

Proving (i): Let an agent's value be $\hat{v}$ drawn from $\mathcal{D}$ where the minimum value is v_0. So, $\hat{v}(1) = v_0$. The agent's utility for obtaining the good with an allocation probability y and paying p is $U = \hat{v}y - p$. Consequently, the consumer surplus is $CS = \mathbb{E}[\hat{v}y - p]$. Define another variable $v = \hat{v} - v_0$, which is drawn a 'shifted' distribution where the minima is 0. So, $v(1) = 0$. Hence, we can re-express $\hat{v}$ as $v + v_0$. As a result, we get the new definition of utility

$$U = (v + v_0)y - p = vy + v_0y - p.$$

Suppose we have an incentive-compatible mechanism (y, p) for the original problem (with value v_0). Now, let's define a new payment $p' = p - v_0y$ for the new mechanism (y, p') designed for the simplified problem (with value v). The agent's utility in this new mechanism would be

$$U' = vy - p' = vy - (p - v_0y) = vy + v_0y - p.$$

Since the agent's utility function is identical in both settings ($U = U'$), their strategic behavior will be identical. An allocation rule y is incentive-compatible in the original problem iff it is incentive-compatible in the simplified problem. The consumer surplus in the simplified problem is

$$\begin{aligned}CS' &= \mathbb{E}\left[vy - p'\right] \\ &= \mathbb{E}[vy + v_0y - p] \\ &= \mathbb{E}[\hat{v}y - p] \\ &= CS.\end{aligned}$$

Consider a seller who posts the price of a good at p. An agent with a valuation of v will buy the good if $v \geq p$. If the seller chooses to post a price $p = v(q)$, then agents with valuations of $v \geq v(q)$— which is exactly those agents with quantiles $t \leq q$. The probability of a sale is q. We define $V(q)$ to be expected surplus from posting a price $v(q)$ to the agent, or more formally the *price-posting surplus curve*. Thus,

$$\begin{aligned}V(q) &= \mathbb{P}(q) \cdot \mathbb{E}[v \mid v \geq v(q)] \\ &= q \cdot \frac{1}{q}\int_0^q v(t)\,dt \\ &= \int_0^q v(t)\,dt.\end{aligned}$$

A consequence of this definition is that $V'(q) = v(q)$.

Consumer surplus is the difference in how much an agent is willing to pay v and what they actually pay q. For a posted price $p = v(q)$, the consumer surplus for a winning agent is $v - v(q)$. The expected consumer surplus $U(q)$ is the total expected consumer surplus, or more formally the *price-posting consumer surplus curve.* Thus,

$$\begin{aligned} U(q) &= \mathbb{P}(q) \cdot \mathbb{E}\left[v - v(q) \mid v \geq v(q)\right] \\ &= \mathbb{P}(q) \cdot \mathbb{E}\left[v \mid v \geq v(q)\right] - \mathbb{P}(q) \cdot v(q) \\ &= V(q) - q \cdot v(q). \end{aligned}$$

The term $q\,v(q)$ is the expected revenue.

Let $u(q)$ be the *marginal price-posting consumer surplus curve*, i.e. it represents the marginal change in expected consumer surplus given a change in the probability of allocation q. Therefore,

$$u(q) = U'(q).$$

Furthermore,

$$\begin{aligned} U'(q) &= \frac{d}{dq}\left(V(q) - q \cdot v(q)\right) \\ &= V'(q) - v(q) - q \cdot v'(q) \\ &= v(q) - v(q) - q \cdot v'(q) \\ &= -q \cdot v'(q). \end{aligned}$$

Since $v(q)$ is non-increasing, its derivative $v'(q)$ is non-positive. Hence, $u(q) \geq 0$ for all q and U is monotone non-decreasing.

Theorem 4.4 (Myerson, 1981)**.** *In a Bayesian incentive-compatible mechanism with allocation rule y and payment rule p (over quantiles), the expected consumer surplus of an agent satisfies*

$$\begin{aligned} \mathop{\mathbb{E}}_{q \sim U[0,1]}\left[v(q)y(q) - p(q)\right] &= \mathop{\mathbb{E}}_{q \sim U[0,1]}\left[u(q)y(q)\right] \\ &= \mathop{\mathbb{E}}_{q \sim U[0,1]}\left[-U(q)y(q)\right] + U(0)y(0). \end{aligned}$$

A mechanism is Bayesian incentive-compatible if it is in every agent's best interest to report their value truthfully, given that other agents are also acting truthfully. The allocation is rule is the probability that an agent with quantile q wins the item, which we denote as $y(q)$. The theorem above allows us to convert the problem of maximizing expected consumer surplus (LHS) into a simpler problem of maximizing $u(q)y(q)$— $u(q)$, the marginal price-posting consumer surplus, plays the role of a "virtual value," so mechanism designer can now proceed as if the agent's value for the purpose of calculating consumer surplus is $u(q)$, not $v(q)$.

However, Myerson[7] also states than "an allocation rule can be implemented as a truthful auction if and only if the allocation rule is monotone non-increasing in quantile space." This is equivalent to being monotone non-decreasing in value v—an agent with a higher value must have at least as high a probability of winning as an agent with a lower value.

So, now all we have to do is maximize

$$\int_0^1 u(q)y(q)\,dq$$

given that $y(q)$ is a monotonic non-increasing function. This maximization problem is easiest when $u(q)$ is also non-increasing— as it guarantees that $U(q)$ is a concave function. In such cases, where $u(q)$ is non-increasing, the distribution is called *consumer surplus regular*. For these "regular" distributions, the problem of finding the optimal mechanism is straightforward— a pointwise optimization $\max_q\{u(q)y(q)\}$.

However, many common distributions like uniform and normal distributions are not consumer surplus regular: in fact, u is monotone non-decreasing for these distribution. Consider a uniform distribution for v on $[0, 1]$

- The CDF is $F(v) = v$
- The quantile is $q = 1 - F(v) = 1 - v$.
- The inverse demand is $v(q) = 1 - q$.
- The derivative is $v'(q) = -1$.
- The marginal consumer surplus is $u(q) = -q * v'(q) = q$.

The function $u(q) = q$ is strictly increasing, and thus we have violated the monotonicity constraint. Intuitively speaking, we would want to allocate the item to agents with the highest $u(q)$, meaning those with q close to 1. This would correspond to giving the item to the agents with the lowest values.

To solve for this, Myerson[7] developed an "ironing" procedure to optimize for virtual surplus subject to monotonicity.

1. Construct a concave hull $\overline{U}$ of the price-posting consumer surplus curve U. Given that $u(q)$ is not non-increasing, by consequence $U(q)$ will not be concave. $\overline{U}(q)$ is the tightest possible concave function that is everywhere greater than or equal to $U(q)$, i.e. for any q necessarily $\overline{U}(q) \geq U(q)$. As a result of this construction, for any part of $U(q)$ that is convex, $\overline{U}(q)$ will be a straight line.
2. Define $\overline{u}(q) = \overline{U}'(q)$ which by the concavity of $\overline{U}$, now guarantees $\overline{u}$ is non-increasing.
3. Find the virtual surplus optimal mechanism using $\overline{u}$

Acknowledging the multi-agent environment, the interim allocation rule for a particular agent is the single-agent allocation rule is based on the expectation over the quantiles of all other agents (models the fact that every agent is only aware of their valuation before reveled preferences).

Theorem 4.5. *Given that y is the interm allocation rule for some agent such that $\frac{d}{dq}y(q) = 0$ for all q and $U(q) \neq \overline{U}(q)$, then*

$$\mathop{\mathbb{E}}_q [u(q)y(q)] = \mathop{\mathbb{E}}_q [\overline{u}(q)y(q)]$$

In essence, given that the allocation rule stays the same between the agents over the convex approximation, then convex-approximation virtual surplus can be used to compute the consumer surplus instead of the actual virtual surplus.

Theorem 4.5 states that the expected surplus is the same irrespective of whether it was calculated using the normal or ironed marginal price-posting consumer surplus curve.

Theorem 4.6 (Alaei et al., 2012 [8]). *Let y be some allocation rule with interim allocation rule y_i monotonously non-increasing for each agent i. Then, there exists a mechanism $\overline{y}$ with interim allocation rule $\overline{y_i}$ for agent i such that the expected consumer surplus for agent i equals*

$$\mathop{\mathbb{E}}_{q \sim U[0,1]} [u_i(q)\overline{y_i}(q)] = \mathop{\mathbb{E}}_{q \sim U[0,1]} [\overline{u_i}(q) y_i(q)]$$

In other words, the expected consumer surplus of $\overline{y}$ is the consumer surplus of the original mechanism if the virtual values were given by $\overline{u_i}$ for agent i instead of u_i.

4.1.2 Near-Optimal Deferred-Acceptance Mechanism for Consumer Surplus

In Section 4.1.1, we showed that maximizing the expected consumer surplus is equivalent to maximizing the expected ironed virtual surplus. The problem is that a lot of the literature on auction design is focused on mechanism design focused on maximizing total surplus. We spend the following section providing a generic transformation for any surplus-optimal mechanism to a near-optimal consumer surplus mechanism., so that we can help analyze the combinatorial pen testing problem in the context of auction theory.

Definition 4.7 (Virtual-Pricing Transformation of a Deferred-Acceptance Mechanism)**.** Let DA^V be the DAA mechanism that optimizes for surplus. Let v_i be the inverse demand function and u_i be the virtual value function for some agent i. The virtual-price transformation on DA^V, results in DAA mechanism DA^U that optimizes for consumer-surplus, by implementing DA^V in the ironed virtual-value space $\overline{U}$. That is, whenever DA^V posts a price $\hat{v}_i$ to agent i, DA^U posts a price $v_i(\theta_i)$ given that

$$\theta_i = \sup\{\theta \mid \overline{u}_i(\theta) \geq \hat{v}_i\}.$$

When the mechanism DA^V posts a price of $\hat{v}_i$, the transformed mechanism DA^U determine a quantile θ_i and the posts the price $v_i(\theta_i)$. Thus, the agent stays in for the round if $v_i(q_i) \geq v_i(\theta_i) \implies q_i \leq \theta_i$. Consequently, this is also equivalent to $\overline{u}_i(q_i) \geq \overline{u}_i(\theta_i)$.

Some preliminary observations about the transformation

1. Behavior in Ironed Regions: Suppose an agent's true virtual valuation $u(q)$ was increasing, then based on our construction of $\overline{U}$, then this has been "ironed" out in $\overline{u}(q)$ such that it is now a constant over the interval of quantiles for which it was found to be increasing. Given by the definition of θ_i, "the agent faces the smallest price needed to differentiate itself from the virtual values smaller than the threshold set by the mechanism."[3] Thus, the transformed mechanism will never post a price corresponding to a quantile θ_i from the middle of an ironed interval— the price will always correspond to a boundary. As a result, the mechanism does not distinguish between the values found in the same "ironed" interval. When an allocation rule is constant across an ironed region. the expected consumer surplus equals the expected ironed virtual surplus.

2. If the mechanism DA^V is a γ-approximation to the optimal surplus, in expectation over all product distributions, then the transformed mechanism DA^U is also a γ-approximation to the optimal ironed virtual surplus, and consequently, the optimal consumer surplus.

3. DA^U is actually a DAA— by its non-decreasing price posting.

Proving observation 2: We are given that DA^V is a γ-approximation to the optimal surplus. This means for any set of agent value distributions, the following holds

$$\frac{\mathbb{E}\left[\text{surplus}(\mathrm{DA}^V)\right]}{\mathrm{OPT}^V} \geq \frac{1}{\gamma}.$$

Given the assumption that DA^V is γ-approximate for any agent value distribution, it must necessarily also hold over $\overline{u}$ such that we have

$$\begin{aligned}\frac{\mathbb{E}\left[\sum \overline{u}(q)\right]}{\mathrm{OPT}^{\overline{U}}} &= \frac{\mathbb{E}\left[\text{surplus}(\mathrm{DA}^{\overline{U}})\right]}{\mathrm{OPT}^{\overline{U}}} \\ &\geq \frac{1}{\gamma}.\end{aligned}$$

Using theorem 4.6, we know that the expected consumer surplus of a mechanism is equal to its expected ironed surplus. Then, the optimal possible consumer surplus is equal to the optimal possible ironed surplus. Thus, we have

$$\begin{aligned}\frac{\mathbb{E}\left[\text{surplus}(\mathrm{DA}^{\overline{U}})\right]}{\mathrm{OPT}^{\overline{U}}} &= \frac{\mathbb{E}\left[\text{surplus}(\mathrm{DA}^{U})\right]}{\mathrm{OPT}^{U}} \\ &\geq \frac{1}{\gamma}.\end{aligned}$$

This is exactly DA^U being a γ-approximation to the optimal ironed virtual surplus, and hence, to the optimal consumer surplus.

Proving observation 3: The mechanism DA^V posts a non-decreasing sequence of virtual prices $\hat{v}_i$. By the rule $\theta_i = \sup\{\theta \mid \overline{u}_i(\theta) \geq \hat{v}_i\}$, a higher $\hat{v}_i$ means the set of θ's becomes smaller. Since $\overline{u}_i$ is non-increasing, θ_i must decrease. A smaller quantile θ_i corresponds to a higher real value $v_i(\theta_i)$: so as the price $\hat{v}_i$ increases, the posted price $v_i(\theta_i)$ non-decreases. Hence, the mechanism is a valid DAA.

Using the following definitions and observation 2

$$\frac{\mathrm{OPT}^U}{\mathrm{DA}^U} \leq \gamma(n) \implies \mathrm{DA}^U \geq \frac{\mathrm{OPT}^U}{\gamma(n)}$$

and

$$\zeta(n) = \frac{\mathrm{OPT}^V}{\mathrm{OPT}^U} \implies \mathrm{OPT}^U = \frac{\mathrm{OPT}^V}{\zeta(n)},$$

we can conclude

$$\begin{aligned}\mathrm{DA}^U &\geq \frac{\mathrm{OPT}^U}{\gamma(n)} \\ &= \frac{\mathrm{OPT}^V}{\gamma(n)\zeta(n)}.\end{aligned}$$

4.2 Consumer Surplus vs. Surplus

The first subsection will focus on the single-agent environment, where we will show that the optimal surplus is at most $1 - \ln q$ times the consumer surplus where the ex-ante probability of allocation is constrained to be at most q. The second subsection will focus on a multi-agent environment and will show a $(1 + o(1)) \ln n$ approximation between the optimal surplus and consumer surplus over a n-agent environments with general feasibility constraints.

4.2.1 Single-Agent Problem

Theorem 4.8. *For an ex-ante allocation probability $q \in [0, 1]$, the ratio of the optimal surplus $V(q)$ to the optimal consumer surplus $\overline{U}(q)$ is at most $1 - \ln q$.*

We will start for proving the case for a consumer surplus regular distribution, which if you can recall is one where $u(q)$ is non-increasing.

Lemma 4.9. *For any consumer surplus regular distribution and an ex-ante allocation probability $q \in [0, 1]$, the ratio of the optimal surplus $V(q)$ to the optimal consumer surplus $U(q)$ is at most $1 - \ln q$.*

Proof.

$$\begin{aligned}
V(q) &= U(q) + q \cdot v(q) \\
&= \int_0^q u(t)\, dt + q \int_1^q v'(t)\, dt \\
&= \int_0^q u(t)\, dt - q \int_q^1 v'(t)\, dt \\
&= \int_0^q u(t)\, dt + q \int_q^1 \frac{u(t)}{t}\, dt. && (u(q) = -q \cdot v'(q))
\end{aligned}$$

Finding the minimum consumer surplus distribution becomes the following program

$$\min \int_0^q u(t)\, dt$$

subject to the following constraints

$$\begin{gathered}
V(q) = \int_0^q u(t)\, dt + q \int_q^1 \frac{u(t)}{t}\, dt \\
u(t) \geq 0, u(t) \text{ is monotonously non-increasing} \\
v(1) = 0, V(0) = 0.
\end{gathered}$$

To make $U(q) = \int_0^q u(t)\, dt$ as small as possible, we should make the function $u(t)$ as low as possible over the interval $[0, q]$. The non-increasing constraint $u(t) \geq u(q)$ for $t \leq q$ provides a floor. To minimize the integral, we should set $u(t)$ to this floor: $u(t) = u(q)$ for all $t \leq q$. Let$u(0) = u(t) = u(1) = u$. For a constant marginal price-posting consumer surplus curve,

$$\begin{aligned}
U(q) &= \int_0^q u(t)\, dt \\
&= uq \\
\implies V(q) &= \int_0^q u(t)\, dt + q \int_q^1 \frac{u(t)}{t}\, dt \\
&= uq - uq \cdot \ln q \\
&= uq\,(1 - \ln q)\,.
\end{aligned}$$

So, for any consumer regular distribution

$$\begin{aligned}\frac{V(q)}{U(q)} &\leq \frac{uq\,(1-\ln q)}{uq}\\ &= 1-\ln q.\end{aligned}$$

The constant marginal consumer surplus curve that achieves this ratio corresponds to the exponential distribution. □

Lemma 4.10. *Let the consumer surplus curves $U, \hat{U}$, satisfying $\hat{U}(0) = 0$ and $\hat{U}(t) \geq U(t)$ for all $t \in [0, 1]$. Then, for the corresponding surplus curves $\hat{V}(q) \geq V(q)$.*

This lemma states that if you pointwise increase the consumer surplus curve, the total surplus curve also pointwise increases.

Proof. Using a pervious statement, we have the following using integration by parts

$$\begin{aligned}V(q) &= \int_0^q u(t)\,dt + q\int_q^1 \frac{u(t)}{t}\,dt\\ &= U(q) + q\left[\frac{U(t)}{t}\right]_{t=q}^1 + q\int_q^1 \frac{U(t)}{t^2}\,dt\\ &= q \cdot U(1) + q\int_q^1 \frac{U(t)}{t^2}\,dt.\end{aligned}$$

Since we assume $\overline{U}(t) \geq U(t)$ for all $t \in [0, 1]$, the integrand $\frac{\overline{U}(t)}{t^2} \geq \frac{U(t)}{t^2}$. Therefore, the integral term for $\overline{V}$ is at least as large as the integral term for V, which implies that $\overline{V}(q) \geq V(q)$. □

Proof for theorem 4.8. Let U and V be the consumer surplus and surplus curves for any distribution. The optimal consumer surplus achievable with ex-ante allocation probability q is given by the concave hull, $\overline{U}(q)$. Let $\overline{V}$ be the surplus curve that corresponds to the ironed consumer surplus curve $\overline{U}$. Finally, we have

$$\begin{aligned}\frac{V(q)}{\overline{U}(q)} &\leq \frac{\overline{V}(q)}{\overline{U}(q)}\\ &\leq 1-\ln q.\end{aligned}$$

The first inequality, holds because we proved in Lemma 4.10 that $\overline{V}(q) \geq V(q)$. The second inequality is a direct application of Lemma 4.9. □

4.2.2 Multi-Agent Environment

The goal of this section is to prove a general upper bound on the optimal omniscient approximation, $\zeta(n) = \mathrm{OPT}^V/\mathrm{OPT}^U$, for any n-agent environment with arbitrary feasibility constraints. The strategy is to leverage our single-agent result from Theorem 4.8.

At the moment, consider just a single agent i out of all n. The interim allocation rule, $y_i(q)$, is the probability that agent i wins the item when their quantile is q, averaged over all possible values of the other $n-1$ agents. The expected surplus from agent i is

$$\mathbb{E}\left[\text{surplus}_i\right] = \int_0^1 y_i(q) v_i(q)\,dq.$$

Using integration by parts, where $f = y_i(q)$ and $g' = v_i(q) = V_i'(q)$,

$$\mathbb{E}\left[\text{surplus}_i\right] = \left[y_i(q)V_i(q)\right]_{q=0}^{1} - \int_0^1 y_i'(q)V_i(q)\, dq.$$

Assuming $y_i(1) = 0$ (the lowest-value agent has zero chance of winning) and also $V_i(0) = 0$, we get a simpler equation

$$\mathbb{E}\left[\text{surplus}_i\right] = -\int_0^1 y_i'(q)V_i(q)\, dq.$$

Now, applying the bound from theorem 4.8, $V_i(q) \leq \overline{U}_i(q)(1 - \ln q)$. Since $y_i(q)$ must be non-increasing for an implementable mechanism, $y_i'(q) \leq 0$, and so $-y_i'(q) \geq 0$, and thus the following inequality is valid

$$\mathbb{E}\left[\text{surplus}_i\right] \leq -\int_0^1 y_i'(q)(1 - \ln q)\overline{U}_i(q)\, dq.$$

However, it should be apparent that this upper-bound is not useful as $q \to 0$, $\ln q \to -\infty$, and so the integral diverges. Thus, we need to show that the loss from ignoring the small (strong) quantiles is not much.

To circumvent this divergence, we use the ϵ-buffering rule from Hartline and Taggart (2019)[9]. We apply this rule to the optimal mechanism OPT^V to create a new, near-optimal mechanism $\hat{y}$. The crucial property of $\hat{y}$ is that its interim allocation rule $\hat{y}_i'(q) = 0$ for $q \in [0, \epsilon]$. Thus, $\hat{y}_i'(q) \cdot V_i(q) = \hat{y}_i'(q) \cdot U_i(q) = 0$. For $q \in [\epsilon, 1]$, we can use the previous inequality, that is

$$V_i(q) \leq (1 - \ln \epsilon) \cdot \overline{U}_i(q).$$

So, ultimately

$$\text{surplus}(\hat{y}) \leq \text{OPT}^U \cdot (1 - \ln \epsilon)$$

Definition 4.11 (Hartline and Taggart, 2019; ϵ-buffering rule)**.** Given an allocation rule $\vec{y} = \langle y_1, \ldots, y_n \rangle$ and a quantile $\epsilon \in [0, 1]$, the ϵ-buffering rile for $\vec{y}$ simulates y with quantiles transformed on each agent as follows

- Top inflate: for any $q_i \in [0, \epsilon]$, return 0
- For any $q_i \in [\epsilon, 1 - \epsilon]$ return $(q_i - \epsilon)/(1 - 2\epsilon)$
- Bottom deflate: for any $q_i \in [1 - \epsilon, 1]$, return 1

The top inflate transformation makes the ϵ-buffering rule treat all agents with a small quantile as if they had quantile 0 and thus, $y_i'(q) = 0$ for $q \in [0, \epsilon]$.

Of course, modifying the optimal mechanism OPT^V to create $\hat{y}$ is not without cost; the buffered mechanism will necessarily generate slightly less surplus.

Theorem 4.12. *Let $\hat{y}$ be the ϵ-buffering rule of the surplus optimal mechanism OPT^V. Then,*

$$OPT^V \leq \frac{1}{\left(1 - \frac{\varepsilon}{1-\epsilon}\right)(1 - \epsilon)(1 - 2n\epsilon)} \cdot surplus(\hat{y})$$

Using the bound on surplus$(\hat{y})$ and the bound relating surplus$(\hat{y})$ back to OPT^V, we can prove the following claim.

Theorem 4.13. *Consider an n-agent environment with an arbitrary feasibility constraint. The optimal omniscient approximation is at most*

$$\frac{1-\ln\epsilon}{\left(1-\frac{\varepsilon}{1-\epsilon}\right)(1-\epsilon)(1-2n\epsilon)}$$

Proof. The total surplus is the sum of expected surpluses from each agent i:

$$\mathrm{surplus}(\hat{y}) = \sum_{i=1}^{n}\left(-\int_0^1 \hat{y}_i'(q)V_i(q)\,dq\right).$$

Because $\hat{y}_i'(q) = 0$ for $q \in [0,\epsilon]$, the integral is non-zero for $q \in [\epsilon, 1]$. For $q \geq \epsilon$, necessarily $(1-\ln q) \leq (1-\ln\epsilon)$, and so

$$\begin{aligned}\mathrm{surplus}(\hat{y}) &= \sum_{i=1}^{n}\left(-\int_\epsilon^1 \hat{y}_i'(q)V_i(q), dq\right)\\ &\leq \sum_{i=1}^{n}\left(-\int_\epsilon^1 \hat{y}_i'(q)(1-\ln\epsilon)\overline{U}_i(q), dq\right)\\ &= (1-\ln\epsilon)\sum_{i=1}^{n}\left(-\int_0^1 \hat{y}_i'(q)\overline{U}_i(q), dq\right).\end{aligned}$$

The inner term is the expected ironed consumer surplus for agent i, so

$$\begin{aligned}\mathrm{surplus}(\hat{y}) &\leq (1-\ln\epsilon)\cdot\mathbb{E}\left[\mathrm{CS}(\hat{y})\right]\\ &\leq (1-\ln\epsilon)\cdot\mathrm{OPT}^U.\end{aligned}$$

Thus, we can conclude

$$\begin{aligned}\mathrm{OPT}^V &\leq \frac{1}{\left(1-\frac{\varepsilon}{1-\epsilon}\right)(1-\epsilon)(1-2n\epsilon)}\cdot\mathrm{surplus}(\hat{y})\\ &\leq \frac{1-\ln\epsilon}{\left(1-\frac{\varepsilon}{1-\epsilon}\right)(1-\epsilon)(1-2n\epsilon)}\cdot\mathrm{OPT}^U\end{aligned}$$

□

Corollary 4.14. *In an n-agent environment with an arbitrary feasibility constraint, the optimal omniscient approximation is $\zeta(n) \leq (1+o(1))\ln n$.*

Proof. Setting $\epsilon = \frac{1}{n\ln n}$, we have

$$\begin{aligned}\zeta(n) &\leq \frac{1-\ln\frac{1}{n\ln n}}{\left(1-\frac{\frac{1}{n\ln n}}{1-\frac{1}{n\ln n}}\right)\left(1-\frac{1}{n\ln n}\right)\left(1-2n\cdot\frac{1}{n\ln n}\right)}\\ &= \frac{1+\ln n+\ln\ln n}{\left(1-\frac{1}{n\ln n-1}\right)\left(1-\frac{1}{n\ln n}\right)\left(1-\frac{2}{\ln n}\right)}\\ &= \left(\frac{n\ln n-2}{n\ln n-1}\right)\left(\frac{n\ln n-1}{n\ln n}\right)\left(\frac{\ln n-2}{\ln n}\right)(1+\ln n+\ln\ln n)\\ &= \left(\frac{n\ln n-2}{n\ln n-1}\right)\left(\frac{n\ln n-1}{n\ln n}\right)\left(\frac{\ln n-2}{\ln n}\right)\left(\frac{1+\ln n+\ln\ln n}{\ln n}\right)\cdot\ln n\end{aligned}$$

Take the limit as $n \to \infty$

$$\lim_{n\to\infty}\left(\frac{n\ln n - 2}{n\ln n - 1}\right) = 1$$
$$\lim_{n\to\infty}\left(\frac{n\ln n - 1}{n\ln n}\right) = 1$$
$$\lim_{n\to\infty}\left(\frac{\ln n - 2}{\ln n}\right) = 1$$
$$\lim_{n\to\infty}\left(\frac{1+\ln n + \ln\ln n}{\ln n}\right) = 1$$

Hence, we conclude

$$\zeta(n) \le (1+o(1))\ln n.$$

□

4.2.3 The k-Identical Goods Environment

Although we have found a tight bound, we can find better approximations for specific environments. For example, in the k-identical goods environment, we will show optimal omniscient approximation is either $(1+o(1))\ln\frac{n}{k}$ or $2.27(1+o(1))(0.577+\ln\frac{n}{k})$ depending on k.

Theorem 4.15. *In the n-agent k-identical goods environment, the optimal omniscient approximation satisfies*

1. $\zeta(n) \le (1+o(1))\ln\frac{n}{k}$, *when* $k = o(n)$
2. $\zeta(n) \le 2.27(1+o(1))(0.577+\ln\frac{n}{k})$, *when* $k = \Theta(n)$.

Lemma 4.16. *In the n-agent k-identical goods environment, the optimal omniscient approximation is*

$$\zeta(n) \le \frac{1}{\left(1-\frac{1}{\sqrt{2\pi k}}\right)}\cdot(1-\ln\epsilon)\left(1+\epsilon\left(\frac{n}{k}-1\right)\right)$$

for any $\epsilon > 0$.

The lemma above is particularly useful when $k = \Theta(n)$. For a constant, ϵ, $\frac{1}{\left(1-\frac{1}{\sqrt{2\pi k}}\right)}$ is $(1+o(1))$ while the remaining part of the product in the inequality above is a constant.

Proving Theorem 4.15. We split the proof based on three different cases

1. $k = \Theta(1)$
2. $k = \omega(1)$ but $o(n)$ (k is growing but slower than n)
3. $k = \Theta(n)$

Case 1: $k = \Theta(1)$
Using corollary 4.14, where we should that for any n-agent environment that

$$\zeta(n) \le (1+o(1))\ln n,$$

then by the fact that k is a constant,

$$(1+o(1))\ln n = (1+o(1))\ln\frac{n}{k}.$$

Case 2: $k = \omega(1)$ but $o(n)$
Using lemma 4.16 and setting $\epsilon = \frac{1}{n/k(1+\ln n/k)}$ which implies that $\ln \epsilon = -\ln(n/k) - \ln(1+\ln(n/k))$, we get

$$\begin{aligned}
\zeta(n) &\leq \frac{1}{\left(1 - \frac{1}{\sqrt{2\pi k}}\right)} \cdot (1 - (-\ln(n/k) - \ln(1 + \ln(n/k)))) \cdot \left(1 + \frac{1}{\frac{n}{k}(1 + \ln \frac{n}{k})}\left(\frac{n}{k} - 1\right)\right) \\
&= \frac{1}{\left(1 - \frac{1}{\sqrt{2\pi k}}\right)} \cdot \left(1 + \ln \frac{n}{k} + \ln\left(1 + \ln \frac{n}{k}\right)\right) \cdot \left(1 + \frac{1 - \frac{k}{n}}{1 + \ln \frac{n}{k}}\right).
\end{aligned}$$

Taking the limit of each of these terms— since $k = \omega(1)$, as $k \to \infty$, the first terms goes to 1, so we get $(1 + o(1))$. For the second term, consider $n \to \infty$ since $k = o(n)$ such that then, necessarily, $n/k \to \infty$, then $\ln n/k \to \infty$ and also $\ln(1 + \ln n/k) \to \infty$ but the first term grows significantly faster than the second one. So, we can say $(1 + o(1)) \ln \frac{n}{k}$, or more formally

$$\ln \frac{n}{k}\left(1 + \frac{\ln(1 + \ln \frac{n}{k})}{\ln \frac{n}{k}} + \frac{1}{\ln \frac{n}{k}}\right),$$

where the second and third term tend to 0. Finally, the third term, taking $n \to \infty$, since $k = o(n)$, we obtain $k/n \to 0$. Therefore, the entire terms tends to 0, and thus we have $(1 + o(1))$. Putting all of this together, we conclude

$$\begin{aligned}
\zeta(n) &\leq (1 + o(1)) \cdot \left((1 + o(1)) \ln \frac{n}{k}\right) \cdot (1 + o(1)) \\
&= (1 + o(1)) \ln \frac{n}{k}.
\end{aligned}$$

Case 3: $k = \Theta(n)$
We do the same as we did in case 3, and set $\epsilon = \frac{1}{n/k(1+\ln n/k)}$. However, given that $k = \Theta(n)$, both n/k and consequently $\ln n/k$ are now constants. Nonetheless, for at least the first term, there is only a k that still tends to ∞, so it remains $(1 + o(1))$. As a result of n/k evaluating to a constant, the second and third term are now just constants. Thus, we can claim

$$\zeta(n) \leq (1 + o(1)) \cdot \kappa,$$

where κ is some constant. □

Using Fact 1.5, in the online pen testing paper[1], we know that for a single item selection and given that the values are drawn from an exponential distribution $F(v) = 1 - e^{-v}$, the expected maximum value is $H_n = \ln n$, while the optimal consumer surplus (the expected score) is 1. We know aim to extend this for k-goods.

Lemma 4.17. *Consider the n-agent k-identical goods environment with values drawn IID from the exponential distribution with mean 1. Assume, for convenience, that n/k is an integer. The optimal omniscient approximation is*

$$\zeta(n) \geq H_{n/k} \geq \left(0.577 + \ln \frac{n}{k}\right).$$

Proof. Using properties of the exponential distribution, we know that the inverse demand is $v(q) = \ln \frac{1}{q}$, the price-posting consumer surplus curve is $U(q) = q(1 - \ln q) - q \cdot v(q) = q$, and the marginal consumer surplus is $u(q) = 1$. Thus, the optimal mechanism selects the k highest ink pens. The total expected consumer surplus is k.

Now lower-bounding this expectation. consider partitioning the n pens into k disjoint groups such that the size of each group is $\lfloor n/k \rfloor$. So, we select the highest pen in each group. Reinterpreting this problem, we have a single item environment with n/k agents, and thus the consumer surplus (expected score) is $H_{n/k}$. Hence, doing this over k groups, the consumer surplus of the entire problem is $k \cdot H_{n/k}$. Finally, working out the ratio we have

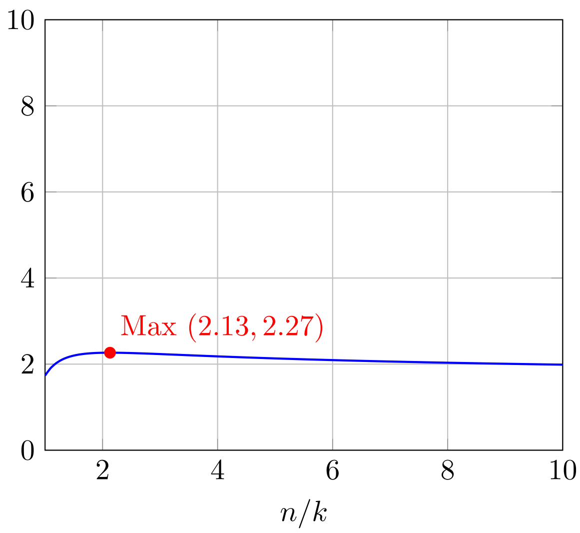
$$\begin{aligned}\zeta(n) &\geq \frac{k \cdot H_{n/k}}{k} \\ &= H_{n/k} \\ &\approx \ln \frac{n}{k} + \gamma,\end{aligned}$$

where γ is the Euler-Mascheroni constant ($\gamma = 0.577$). $\square$

Now working on the upper-bound, we have

$$\frac{\left(1 + \ln \frac{n}{k} + \ln\left(1 + \ln \frac{n}{k}\right)\right) \cdot \left(1 + \frac{1 - \frac{k}{n}}{1 + \ln \frac{n}{k}}\right)}{0.577 + \ln \frac{n}{k}}.$$

Plotting this, we have



So, the "gap" between the algorithm's performance and the optimal solution is at most 2.27. Thus, in regards to Theorem 4.15, we have shown it is tight up to $(1 + o(1))$ terms when $k = o(n)$ and up to a multiplicative $2.27(1 + o(1))$ when $k = \Theta(n)$.

We now need to prove Lemma 4.16. Increasing the allocation to an agent by some small quantile so that $y_i(q_i) = y_i(0)$ for $q_i \in [0, \epsilon]$, also considerably increases the consumer surplus of agent i without visibly decreasing the surplus of the mechanism (using the top-inflation in 4.11). We consider online price-posted mechanisms with the constraint $y_i'(q_i) = 0$ whenever $q_i \leq \epsilon$ for all agents.

We will approximate the performance of the mechanism against the optimal ex-ante relaxed mechanism EAR. Let Δ be the set of all allocation vectors $\vec{q} = \langle q_1, \ldots, q_n \rangle$ such that $q_i \in [0, 1]$ and $\sum q_i \leq k$. The value $\text{EAR}(\vec{q}) = \sum V_i(q_i)$, that is the surplus generated by allocating to agent i with probability q_i. In other words, $\text{EAR}(\vec{q})$ be the ex-ante relaxed mechanism that allocates to the q_i smallest quantiles of agent i. Let EAR$= \max_{\vec{q} \in \Delta} \text{EAR}(\vec{q})$.

Note that $\mathrm{OPT}^V \leq \mathrm{EAR}$ since any feasible mechanism satisfies the ex-ante constraint.

Yan[10] showed that the following sequential posted-price mechanism $\mathrm{SP}(\vec{q})$ has a surplus comparable to $\mathrm{EAR}(\vec{q})$ for all $\vec{q} \in \Delta$.

1. Order agents in decreasing order of $\frac{V_i(q_i)}{q_i}$, the expected surplus conditioned on selling to agent i
2. In the order above, offer agent i a price $v_i(q_i)$ until all n agents interact with the mechanism or until all k goods are sold

Definition 4.18 (Yan, 2019)**.** For $\vec{q} \in \Delta$,

$$\mathrm{SP}(\vec{q}) \geq \left(1 - \frac{1}{\sqrt{2\pi k}}\right) \mathrm{EAR}(\vec{q}).$$

For some $\vec{q} \in \Delta$, let the ϵ-inflated sequential posted-price mechanism $\mathrm{SP}(\vec{q}, \epsilon)$ denote $\mathrm{SP}(\vec{p})$, where $p_i = \max\{q_i, \epsilon\}$. If $\mathrm{SP}(\vec{q}, \epsilon)$ has some an unsold good when interacting with agent i, it will always sell the good to the agent given that it ha a quantile $q_i \leq \epsilon$. Thus, the interim allocation rule y_i is a constant function in $[0, \epsilon]$ and $y_i'(q) = 0$. This inflation ensures that for every agent, the allocation probability is at least ϵ— if an agent is allocated with probability at least ϵ , the ratio of their Surplus to Consumer Surplus is bounded by $1 - \ln \epsilon$.

Lemma 4.19. *For all $\vec{q} \in \Delta$,*

$$EAR(\vec{q}) \leq \frac{1 + \epsilon\left(\frac{n}{k} - 1\right)}{1 - \sqrt{2\pi k}} \cdot SP(\vec{q}, \epsilon).$$

Proof. The are two parts to this proof. First, for $\vec{p}$ we want to show that it is not too far away from the set of ex-ante feasible allocations Δ. Then, we want to show that the projection $\vec{t}$ of $\vec{p}$ onto Δ satisfies

$$\mathrm{EAR}(\vec{q}) \leq \mathrm{EAR}(\vec{p}) \leq \frac{k + \epsilon(n-k)}{k} \cdot \mathrm{EAR}(\vec{t})$$

and

$$\mathrm{SP}(\vec{t}) \leq \mathrm{SP}(\vec{p}) = \mathrm{SP}(\vec{p}, \epsilon).$$

Step 1: Bounding Inflation

We can assume (WLOG) $\sum_n q_i = k$. Increasing q_i so that the sum is k would not decrease the value $\sum p_i$. Hence, we can show that the maxima of $\sum(p_i - q_i)$ is at most $\epsilon(n-k)$.

The equality $p_i - q_i = \epsilon$ occurs exactly when $q_i = 0$. Hence, whenever $q_i < \epsilon$, decreasing them to 0 would increase the value of $\sum(p_i - q_i)$. However, in doing so would invalidate the constraint $\sum q_i = k$. If $q_i \geq \epsilon$, then necessarily $p_i - q_i = 0$. Thus, we could increase q_i to 1 and it would not change $\sum(p_i - q_i)$ and would also allow the setting of q_i to 0 such that the constraint $\sum q_i = k$ remains intact. Thus, it should be easy to see that $\sum(p_i - q_i)$ is maximized when at least k agents are inflated to 1 and $n - k$ agents are inflated to 0. Thus,

$$\sum(p_i - q_i) \leq \epsilon(n-k).$$

Step 2: Constructing the projection $\vec{p}$

Order the agents in order the agents in decreasing order of their generated surplus $V_i(p_i)$. Define a valid allocation $\vec{t}$ by greedily filling the knapsack capacity k using the values from $\vec{p}$ in this sorted order. Specifically, there exists some index $m \leq n$ such that we assign $t_i = p_i$ for $i < m$, a partial amount for m, and 0 for $i > m$, such that $\sum t_i = k$.

The surplus of $\vec{t}$ captures a large fraction of the surplus of $\vec{p}$ since we prioritized high-value agents. The ratio of the total mass of $\vec{t}$ (which is k) to the total mass of $\vec{p}$ (which is $\leq k + (n-k)\epsilon$) bounds the loss. Thus,

$$\begin{aligned}\mathrm{EAR}(\vec{t}) &\geq \frac{k}{\sum p_i}\mathrm{EAR}(\vec{p}) \\ &\geq \frac{k}{k+(n-k)\epsilon}\mathrm{EAR}(\vec{p}).\end{aligned}$$

Step 3: Connecting bounds

1. Note that $\mathrm{EAR}(\vec{p}) \geq \mathrm{EAR}(\vec{q})$ because $p_i \geq q_i$ and surplus V_i is monotone increasing in probability
2. The sequential posted price mechanism $SP(\vec{t})$ allocates to the top agents exactly as $\vec{t}$ specifies (until items run out). The mechanism $SP(\vec{p})$ attempts to allocate even more. Trivially, $SP(\vec{p}) \geq SP(\vec{t})$
3. By Yan's Theorem (Theorem 2.9), $SP(\vec{t}) \geq (1 - \frac{1}{\sqrt{2\pi k}})\mathrm{EAR}(\vec{t})$

Hence,

$$\begin{aligned}\mathrm{OPT}^V &\leq \mathrm{EAR}(\vec{q}) \\ &\leq \mathrm{EAR}(\vec{p}) \\ &\leq \left(1 + \left[\frac{n}{k} - 1\right]\epsilon\right)\mathrm{EAR}(\vec{t}) \\ &\leq \left(1 + \left[\frac{n}{k} - 1\right]\epsilon\right)\frac{1}{1-\frac{1}{\sqrt{2\pi k}}}SP(\vec{t}) \\ &\leq \left(1 + \left[\frac{n}{k} - 1\right]\epsilon\right)\frac{1}{1-\frac{1}{\sqrt{2\pi k}}}SP(\vec{p}).\end{aligned}$$

Step 4: Don't Forget About Consumer Surplus

The mechanism $SP(\vec{p})$ posts prices $v_i(p_i)$. Since $p_i \geq \epsilon$ for all agents, the interim allocation rule $y_i(q)$ is constant for $q \in [0, \epsilon]$. As proven in Theorem 2.6, the ratio of Surplus to Consumer Surplus is bounded by $(1 - \ln\epsilon)$. Therefore,

$$SP(\vec{p}) \leq (1 - \ln\epsilon)\cdot \mathrm{CS}(SP(\vec{p})) \leq (1-\ln\epsilon)\mathrm{OPT}^U.$$

Consequently,

$$\mathrm{OPT}^V \leq \frac{1 + \epsilon(\frac{n}{k} - 1)}{1 - \frac{1}{\sqrt{2\pi k}}}(1-\ln\epsilon)\mathrm{OPT}^U.$$

Therefore,

$$\zeta(n) \leq \frac{1}{1-\frac{1}{\sqrt{2\pi k}}}\cdot\left(1 + \epsilon\left(\frac{n}{k} - 1\right)\right)\cdot(1-\ln\epsilon).$$

$\square$

4.3 Pen Testing Corollaries from Deferred-Acceptance Mechanisms

Using Theorem 4.3, and the results we have proven any good deferred-acceptance mechanism that are known to good, imply good pen testing algorithms.

Corollary 4.20. *For a pen testing environment with a matroid feasibility constraint, there exists a pen testing algorithm with an omniscient approximation ratio* $(1 + o(1)) \ln n$.

This is by consequence of a generalized english auction, where the auction stops, in this case, when the remaining set of active bidders forms a basis. This is a deferred-acceptance auction that maximizes the ex-post consumer surplus. Hence, we have

$$\pi(n) \leq 1 \cdot \zeta(n) \leq (1 + o(1)) \ln n.$$

Extrapolating this result to the problem discussed above of choosing k out of n pens that are feasible, then we know there exists a pen testing algorithm such that

$$\pi(n) \leq (1 + o(1)) \ln \frac{n}{k}.$$

when $k = o(1)$, and

$$\pi(n) \leq 2.27(1 + o(1)) \left(0.557 + \ln \frac{n}{k}\right).$$

Moving beyond simple k-identical items or matroids, we look at general downward-closed constraints. Feldman et al. (2022) provide a deferred-acceptance mechanism for these settings that achieves an approximation of $O(\log \log m)$ to the optimal surplus, where m is the number of feasible sets. Since $m \leq 2^n$, this results in a poly-logarithmic approximation factor.

However, the pen testing problem allows us to extend this result to general combinatorial constraint, not just downward-closed ones. In an auction, an agent cannot be forced to pay if they are not allocated the item. However, in pen testing, the "payment" is the ink burned during the test. If we test a set of pens P that turns out to be infeasible (but was a subset of a feasible set $\mathcal{P}$), we can simply discard the excess pens until we reach a feasible subset $\overline{P}$. Given that the objective is to maximize residual ink, and ink must remain non-negative, throwing away pens is a valid operation. Thus, near optimal pen testing algorithms for downward-closed constraints can be extended to give near optimal algorithms for general combinatorial constraints, giving the same performance guarantee as the downward-closed environment.

Corollary 4.21. *For any combinatorial pen testing environment, there exists a pen testing algorithm with an omniscient approximation ratio* $O(\log n \log \log m) = O(\log^2 n)$.

It is worth noting that this reduction does not work in the reverse direction: we cannot use pen testing logic to design auctions for general constraints because "allocating then dropping" in an auction would require charging bidders who receive nothing, breaking individual rationality.

For the specific case of Knapsack constraints, we can achieve much tighter bounds than the general case. Milgrom and Segal (2014) showed that this selection process can be implemented via a deferred-acceptance auction with $\gamma(n) = 2$. Applying our reduction framework:

Corollary 4.22. *For a pen testing environment with a knapsack feasibility constraint, there exists a pen testing algorithm with an omniscient approximation ratio* $2(1 + o(1)) \ln n$.

4.3.1 Online Pen Testing

There are two models to consider

- Sequential: The algorithm can choose the order in which to test the pens.
- Oblivious: The order of pens is determined by an adversary.

In the sequential setting, the problem relates to the correlation gap. Chawla et al. (2010) and Yan (2011) demonstrated that for single-item and matroid environments, sequential posted pricing achieves a standard approximation of $\gamma(n) = \frac{e}{e-1}$.

Corollary 4.23. *In the online sequential pen testing problem with a matroid feasibility constraint, there exists a pen testing algorithm that achieves an omniscient approximation ratio* $\frac{e}{e-1} \cdot (1 + o(1)) \ln n$.

In the oblivious setting, we lose the power to order the pens. This scenario mirrors prophet inequalities, where we must set a threshold to capture a high-value agent from a random stream. The result from Samuel-Cahn (1984) and Kleinberg and Weinberg (2012) provides a 2-approximation for single items and matroids using adaptive prices.

Corollary 4.24. *In the online oblivious pen testing problem (for single selection or matroid constraints), there exists a pen testing algorithm that achieves an omniscient approximation ratio* $2(1 + o(1)) \ln n$.

These corollaries represent a significant improvement over the previous bounds by Qiao and Valiant (2022) for online pen testing, which were bounded by $O(\log n)$.

4.4 The Online IID Environment

We now consider a specific instance of the online pen testing problem where the ink in each pen is drawn independently from the same distribution. We show a tighter bound distinct from the reduction in 4.3.

Theorem 4.25. *In online single-item environments with IID agents, there exists a price-posting strategy with an omniscient approximation at most* $(1 + o(1)) \ln n$. *Equivalently, in the online pen testing problem with IID ink levels, there exists an algorithm that achieves* $\pi(n) \leq (1 + o(1)) \ln n$.

Before proceeding with the proof, it is known that in an online environment with IID agents that $\gamma(n) = e/(e-1)$. So using the framework established above and Theorem 4.3, we would have

$$\frac{e}{e-1} \cdot (1 + o(1)) \ln n.$$

as the bound. We will show a tight

$$H_n + 1 = (1 + o(1)) \ln n.$$

bound up to an additive factor of 1.

Proof. Let $v_{\max}$ be the random variable denoting the value of the winner, so

$$\mathbb{P}\left(v_{\max} \leq v(t)\right) = (1 - t)^n.$$

As a result, the expected surplus of the surplus optimal mechanism is

$$\begin{aligned}
\int_0^1 v(t)\cdot n(1-t)^{n-1}\,dt &= \int_0^1 \left(\int_t^1 \frac{u(r)}{r}\,dr\right)\cdot n(1-t)^{n-1}\,dt \\
&= \int_0^1 \frac{u(r)}{r} - \left(\int_0^r n(1-t)^{n-1}\,dt\right)\,dr \\
&= \int_0^1 \frac{u(r)}{r}\left(1-(1-r)^n\right)\,dr \\
&= U(1) + \int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t^2} U(t)\,dt.
\end{aligned}$$

The integral measures the total area under the marginal consumer surplus curve, weighted by the probability that at least one agent has a quantile better than x.

The optimal strategy to maximize consumer surplus is a single posted price. Setting the threshold to q, then the corresponding probability that we accept a specific pen is q. The expected consumer surplus per acceptance is $\frac{U(q)}{q}$. Therefore, the total expected consumer surplus for a threshold at quantile q is

$$\frac{1-(1-q)^n}{q}\cdot U(q).$$

The goal is to find the distribution that maximizes $\frac{\mathrm{OPT}^V}{\mathrm{ALG}}$. WLOG, assume that the maximum achievable consumer surplus through anonymous price posting is at most 1. Thus, we want to maximize OPT^V subject to that $\mathrm{ALG} \leq 1$ for any feasible quantile q. Hence, the program is

$$\max U(1) + \int_o^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t^2} U(t)\,dt,$$

subject to

$$U(q) \leq \frac{q}{1-(1-q)^n}\ \forall q \in [0,1]$$
$$U \text{ is concave}$$

Let $C(q) = \frac{q}{1-(1-q)^n}$ is convex. The goal is to fit U "inside" C. Let $\overline{C}_q$ be the tangent to C at q. $\overline{C}_{\hat{q}}$ is feasible solutiuon for U for all $\hat{q} \in [0,1]$. Additionally, given that U intercepts C at $\hat{q}$, $U(t) \leq \overline{C}_{\hat{q}}(t)$ for all $t \in [0,1]$. The optimal solution to the program is achieved at $U = \overline{C}_{\hat{q}}$.

Let $U(q) = q + a$. So the surplus is

$$1+a+\int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t}\,dt + \left(a\cdot\int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t}\,dt\right) = H_n + an.$$

By posting a price $v(q)$, we generate a consumer surplus

$$\frac{1-(1-q)^n}{q}\cdot U(q) = (1+(1-q)+\ldots+(1-q)^{n-1})\cdot(q+a).$$

If $a \geq \frac{1}{n}$, the ratio between the surplus and the consumer surplus is at most H_n+1 by setting $q=0$. If $a < \frac{1}{n}$, the surplus is less than H_n+1, and by setsting $q=1$, the consumer surplus is at least 1. Hence, the worst case ratio between surplus and consumer surplus is at most $H_n + 1 = (1 + o(1)) \ln n$. We can rewrite the optimal surplus as

$$\int_0^1 v(t) \cdot n(1-t)^{n-1} = \left[V(t) \cdot n(1-t)^{n-1}\right]_0^1 + \int_0^1 V(t) \cdot n(n-1)(1-t)^{n-2}\, dt.$$

Thus, the optimal surplus increases with a pointwise increase in V.

$\overline{U}$ is pointwise larger than U, and using Lemma 4.10 we can further claim $\overline{V}$ is pointwise larger than V. Hence, the optimal surplus is larger in the distribution with price-posting consumer surplus curve $\overline{U}$. But we know from Theorem 4.6 he consumer surplus from posting prices is identical to both the distributions. Therefore, the consumer surplus irregular distribution has a better ratio between surplus and consumer surplus than the consumer surplus regular distribution. Thus, the approximation ratio is bounded by the harmonic number

$$\pi(n) \leq H_n = (1 + o(1)) \ln n$$

for all distributions. □

Chapter 5

Difference Prophet Inequalities

5.1 Introduction

Let $X_1, X_2, \ldots$ be i.id. random variables such that $0 \leq X_i \leq 1$ and let $c \geq 0$ be a fixed sampling cost to observe any X_i. Let the net reward for stopping at option i be the value observed minus the accumulated cost

$$Y_i = X_i - ic.$$

We will compare two strategies for optimizing the return from this sequence over some n.

The Prophet M

The prophet has complete foresight of the 'stream' of random variables X_i. However, by the structure of the problem, the still must pay the cost of observing the option in order to select it. Thus, the prophet's strategy is just choosing the appropriate i that maximizes the net reward Y_i.

$$M(Y_1, \ldots, Y_n) = \mathbb{E}\left[\max_{1 \leq i \leq n} Y_i\right].$$

The Statistician V

The statistician is oblivious to the 'stream' of random variables, that is they are unaware of X_i unless the pay the cost to observe it,

$$V(Y_1, \ldots, Y_n) = \sup_{\tau \in C^n} \mathbb{E}(Y_\tau),$$

where C^n denotes the set of stopping rules for $Y_1, \ldots, Y_n$.

For any real number m, let $[m]$ denote the largest integer strictly smaller than m. Samuel-Chan [11] proved the following results:

Let $X_1, X_2, \ldots$ be i.i.d. random variables, $0 \leq X_1 \leq 1$

a) For $0 \leq c \leq 1$ fixed and all $n \geq 1$,

$$M(Y_1, \ldots, Y_n) - V(Y_1, \ldots, Y_n) \leq \left[\frac{1}{c}\right] \cdot c(1-c)^{\left[\frac{1}{c}\right]+1}$$

b) For $n \geq 1$ fixed and all $c \geq 0$,

$$M(Y_1, \ldots, Y_n) - V(Y_1, \ldots, Y_n) \leq \left(1 - \frac{1}{n}\right)^{n+1}$$

c) For all $c \geq 0$ and all $n \geq 1$

$$M(Y_1, \ldots, Y_n) - V(Y_1, \ldots, Y_n) < e^{-1}$$

In his thesis, Harten[12] shows that inequality in a) fails to hold for $c > 0.5$ and b) fails to hold for $n \geq 2$. Consequently, c) fails as it is contingent on b). Harten provides the correct upperbound for part a) and puts forth a new proof for c). Thus, we actually have

Theorem 5.1. *Let $X_1, X_2, \ldots$ be i.i.d. random variables, $0 \leq X_i \leq 1$*

a) For $0 \leq c \leq 1$ fixed and all $n \geq 1$,

$$M(Y_1, \ldots, Y_n) - V(Y_1, \ldots, Y_n) \leq \begin{cases} \left[\frac{1}{c}\right] \cdot c(1-c)^{\left[\frac{1}{c}\right]+1}, & \text{for } c \leq \frac{1}{2} \\ (1-c)/4, & \text{for } c \geq \frac{1}{2} \end{cases} \tag{5.1}$$

b) For $n \geq 1$ fixed and all $c \geq 0$,

$$M(Y_1, \ldots, Y_n) - V(Y_1, \ldots, Y_n) \leq \frac{(n-1)\left(1 - \frac{1}{n+1}\right)^n}{n+1} \tag{5.2}$$

c) For all $c \geq 0$ and all $n \geq 1$

$$M(Y_1, \ldots, Y_n) - V(Y_1, \ldots, Y_n) < e^{-1} \tag{5.3}$$

The author makes a couple of remarks that are of note

Remark 5.2.

1. The inequalities in parts a) and c) remain true for infinite sequences of $[0, 1]$-valued i.i.d.. random variables.
2. The ratio prophet inequality $M(Y)/V(Y)$ is unbounded for all $n > 1$ and all $0 < c < 1$
3. The case for $c = 0$ has been previously shown, where $M(Y)/V(Y) \leq a_n$ and $M(Y) - V(Y) \leq b_n$ for all $n = 2, 3, \ldots$ such that $1.1 < a_n < 1.6$ and $0 < b_n < 0.25$. The results are significantly different that those derived for the $c > 0$ case and seemingly cannot be obtained from them. The second inequality is a vital for the proof below.

5.2 Proving Theorem 5.1

5.2.1 Proof for a)

This has been shown in [12].

5.2.2 Proof for b)

If $n = 1$, , the prophet and the statistician have the exact same strategy of observing X_1 and paying c, such that

$$M(Y_1) = \mathbb{E}[X_1 - c]$$
$$V(Y_1) = \mathbb{E}[X_1 - c].$$

Evidently, the difference is 0 and the upperbound is 0. Thus, $0 \leq 0$; the inequality holds. We will assume $n \geq 2$.

Let Y be the sequence of $Y_1, \ldots, Y_n$ such that we can define $D(Y) = M(Y) - V(Y)$ as well as $d_n = (n-1)(1 - 1/(n+1))^n/(n+1)$.

Lemma 5.3. *For all $c \geq 0$ and i.i.d. Bernoulli variables $X_1, \ldots, X_n$, given that $X_i \in \{0,1\}$, $D(Y) \leq d_n$.*

Proof.

1. Case $c = 0$ (becomes classical difference prophet inequality)
   The prophet sees all values and picks the max. If any $X_i = 1$, Prophet gets 1. Prophet only gets 0 if all are 0.
   The statistician can secure the same value using the following stopping rule $\tau = \inf\{i \mid X_i = 1 \vee i = n\}$ which is basically taking the first option that has a value of 1 or if none of the pens up to n have been 1, select the last option n.
   It is easy to see that in this case $D(Y) = 0 < d_n$.

2. Case $c \geq 1$ ($X_i - ic \leq 0$ and usually negative)
   The prophet will always accept the first option as it is guaranteed to be the best option as the net-reward is at most 0 and the incremental cost of selecting an option after it will necessitate it being "more negative."
   The statistician can secure the same value using the following stopping rule $\tau = 1$ which is the exact same policy.
   It is easy to see that in this case $D(Y) = 0 < d_n$.

3. Case $0 < c < 1$
   This has been shown in [12].

$\square$

Lemma 5.4. *If $c = 0$, then $D(Y) < d_n$ for any i.i.d. $[0,1]$-valued variables.*

Proof. Hill and Kertz[13] established that for $c = 0$, the difference is bounded by a constant b_n,

$$D(Y) = D(X_1, \ldots, X_n) \leq b_n,$$

for certain constants $0 < b_n < 1/4$. Hence, all we need to do is guarantee that $b_n < d_n$. For $n = 5$, $d_n > 1/4$ and we know by definition $b_n < 1/4$, therefore, $b_n < d_n$. We can work out the specific values for $n = 2, 3, 4$. For $n = 2$, $b_2 = 0.063 < 0.148 = d_2$; for $n = 3$, $b_3 = 0.077 < 0.211 = d_3$; for $n = 4$, $b_4 = 0.085 < 0.246 = d_4$. $\square$

Chow, Robbins and Siegmund[14] had previously shown

$$V(Y_1, \ldots, Y_i) = \mathbb{E}\left[\max\{X_1, V(Y_1, \ldots, Y_{i-1})\}\right] - c$$

for all $i > 1$ (this is basically bellman's equation?). The paper also introduces the baylage technique, pushing probability mass from the center of a distribution to the boundaries (0 and 1) while preserving the mean. For an integrable random variable Y and any $-\infty < a < b < \infty$, let Y_a^b denote a random variable with $Y_a^b = Y$ for $Y \notin [a,b]$,

$$\mathbb{P}(Y_a^b = a) = (b-a)^{-1} \int_{Y \in [a,b]} (b - Y)\, dP, \qquad \mathbb{P}(Y_a^b = b) = (b-a)^{-1} \int_{Y \in [a,b]} (Y - a)\, dP.$$

As a result of this construction, $\mathbb{E}[Y] = \mathbb{E}\left[Y_a^b\right]$ and if X is any random variable independent of Y and Y_a^b,

$$\mathbb{E}\left[\max\{X, Y\}\right] \leq \mathbb{E}\left[\max\{X, Y_a^b\}\right].$$

Proof for 5.2). The case for when $c = 0$, has already been shown in lemma 5.4. We will now consider the scenario where $c > 0$ and $\mathbb{E}[X_1] \leq c$, that is the expected value of a single observation is less than or equal to the cost of obtaining it. Let $\tilde{X}_1, \ldots, \tilde{X}_n$ be i.i.d. random variables, where $\tilde{X}_i = (X_i)_0^1$ is the 0-1 baylage of X_i and consequently define $\tilde{Y}_i = \tilde{X}_i - ic$. Then, by a result I assume derived by Harten[12], we can claim $V(\tilde{Y}_1, \ldots, \tilde{Y}_i) = V(Y_1, \ldots, Y_i)$ for all $i = 1, \ldots, n$ and $M(\tilde{Y}_1, \ldots, \tilde{Y}_n) \geq M(Y_1, \ldots, Y_n)$ (the intuition for the second inequality would be that increasing the variance but keeping the mean constant increases the expectation of the maximum). Putting this together, we have

$$\begin{aligned} D(Y) &= M(Y) - V(Y) \\ &\leq M(\tilde{Y}) - V(\tilde{Y}) \\ &= D(\tilde{Y}) \\ &< d_n. \end{aligned}$$

Considering the alternate, $c > 0$ and $\mathbb{E}[X_1] > c$, we cannot directly reduce to Bernoulli variables.

Lemma 5.5. *Let $X_1, \ldots, X_n$ be i.i.d. $[0,1]$-valued random variables and let $c > 0$. Then, there exists i.i.d. $[0,1]$-valued random variables $\tilde{X}_1, \ldots, \tilde{X}_n$ with*

$$\begin{aligned} \tilde{x}_* &= \inf\left\{\tilde{x} \in \mathbb{R} \mid I\!P\left(\tilde{X}_1 \leq \tilde{x}\right) > 0\right\} = 0 \\ \tilde{x}^* &= \sup\left\{\tilde{x} \in \mathbb{R} \mid I\!P\left(\tilde{X}_1 \leq \tilde{x}\right) < 1\right\} = 0. \end{aligned}$$

and $\tilde{c} > 0$ such that $D(Y) \leq D(\tilde{Y})$, where $\tilde{Y}_i = \tilde{X}_i - i\tilde{c}$ for all $i = 1, \ldots, n$.

All these lemma says WLOG, is that the support of X can be extended such that it touches 0 and 1.

Proof. Define x_* and x^* as

$$\begin{aligned} x_* &= \inf\{x \in \mathbb{R} \mid \mathbb{P}(X_1 \leq x) > 0\} = 0, \\ x^* &= \sup\{x \in \mathbb{R} \mid \mathbb{P}(X_1 \leq x) < 1\} = 1. \end{aligned}$$

For $x_* = x^*$, necessarily $M(Y) - V(Y) = 0$. Choosing i.i.d. random variables $\tilde{X}_1, \ldots, \tilde{X}_n$ with $\mathbb{P}(\tilde{X}_i = 1) = 1/2 = 1 - \mathbb{P}(\tilde{X}_i = 0)$ and $\tilde{c} = c$ guarantees $D(Y) \leq D(\tilde{Y})$.

If $x_* < x^*$, consider random variables $\tilde{X}_1, \ldots, \tilde{X}_n$ defined by

$$\tilde{X}_i = \frac{X_i - x_*}{x^* - x_*},$$

which are evidently $[0,1]$-valued i.i.d. random variables with $\tilde{x}_* = 0$ and $\tilde{x}^* = 1$. Setting $\tilde{c} = c/(x^* - x_*)$, we also have

$$\begin{aligned} M(\tilde{Y}) - V(\tilde{Y}) &= \frac{M(Y) - x_*}{x^* - x_*} - \frac{V(Y) - x_*}{x^* - x_*} \\ &\geq M(Y) - V(Y). \end{aligned}$$

Hence, the random variables $\tilde{X}_i$ have the necessary properties. □

In the reduction above, it is possible that we get from $c > 0$ and $\mathbb{E}[X_1] > c$ to $\tilde{c} > 0$ and $\mathbb{E}[\tilde{X}_1] \leq \tilde{c}$. If that is the case, the first part of the proof yields the assertion. All that is left to consider is $\tilde{c} > 0$ and $\mathbb{E}[\tilde{X}_1] > \tilde{c}$.

Lemma 5.6. *Let $X_1, \ldots, X_n$ be i.i.d. $[0,1]$-valued random variables satisfying the Support condition and let $c > 0$ as well as suppose that $\mathbb{E}[X_1] > c$. Let $v_i = V(Y_1, \ldots, Y_i)$ for $i = 1, \ldots, n-1$. Then, $1 > v_{n-1} > v_{n-2} > \ldots > v_2 > v_1 > 0$ and there exists i.i.d. $[0,1]$-valued random variables $\tilde{X}_1, \ldots, \tilde{X}_n$ with*

$$\begin{aligned} I\!P\left(\tilde{X}_1 \in \{1, v_{n-1}, v_{n-2}, \ldots, v_2, v_1, 0\}\right) &= 1, \\ I\!P(\tilde{X}_1 = 1) &> 0, \\ I\!P(\tilde{X}_1 = 0) &> 0, \end{aligned}$$

and

$$V\left(\tilde{Y}_1, \ldots, \tilde{Y}_i\right) = v_i = V\left(Y_1, \ldots, Y_i\right) \quad \textit{for all } i = 1, \ldots, n-1,$$

such that

$$D\left(Y\right) \leq D\left(\tilde{Y}\right),$$

where $\tilde{Y}_i = \tilde{X}_i - ic$.

We are simplifying the distribution X by transforming it to a discrete distribution $\tilde{X}$.

Proof. Necessarily, $v_i \leq 1-c \leq 1$ for all $i = 1, \ldots, n-1$. By assumption, $v_i = \mathbb{E}\left[X_1\right] - c > 0$. Additionally, if $v_i > v_{i-1}$ golds for some $i \in [n-2]$, the backward reward equation yields

$$\begin{aligned} v_{i+1} &= \mathbb{E}\left[\max\left\{X_1, v_i\right\}\right] - c \\ &> \mathbb{E}\left[\max\left\{X_1, v_{i-1}\right\}\right] - c \\ &= v_i, \end{aligned}$$

where the inequality follows from the assumption that $x_* = 0$. We then perform the baylage technique on each X_i such that we obtain

$$\left(\left(\cdots\left(\left(X_i\right)_0^{v_1}\right)_{v_1}^{v_2}\cdots\right)_{v_{n-2}}^{v_{n-1}}\right)_{v_{n-1}}^{1},$$

which is a recursive definition that initially looks at the interval $[0, v_1]$ and sweeps all the probability mass inside the interval to the endpoints 0 and v_1. Immediately after, it looks at the interval $[v_1, v_2]$ and sweeps all the probability mass inside the interval to the endpoints v_1 and v_2. It continues doing this, until the last interval $[v_{n-1}, 1]$ and sweeps all the probability mass inside the interval to the endpoints v_{n-1} and 1. We are left with a new random variable $\tilde{X}_i$ that has zero probability mass between the points— all the probability is concentrated exactly at the points $0, v_1, v_2, \ldots, v_{n-1}, 1$. By construction, necessarily

$$\begin{aligned} \mathbb{P}\left(\tilde{X}_1 \in \{1, v_{n-1}, v_{n-2}, \ldots, v_2, v_1, 0\}\right) &= 1, \\ \mathbb{P}(\tilde{X}_1 = 1) &> 0, \\ \mathbb{P}(\tilde{X}_1 = 0) &> 0, \end{aligned}$$

where the inequalities hold given the assumptions of $x^* = 1$ and $x_* = 0$. Using, the same logic in Hill and Kertz[13], we can conclude

$$\begin{aligned} V\left(\tilde{Y}_1, \ldots, \tilde{Y}_i\right) &= V\left(Y_1, \ldots, Y_i\right) \\ M\left(\tilde{Y}_1, \ldots, \tilde{Y}_n\right) &\geq M\left(Y_1, \ldots, Y_n\right), \end{aligned}$$

and consequently

$$D(Y) \leq D\left(\tilde{Y}\right)$$

□

This construction leaves the expectation unchanged, so we remain in the case of $c > 0$ and $\mathbb{E}[X_1] > c$. Because we chose the sweep points to exactly match the points (v_k) of the statistician, the statistician is indifferent to the change.

We now embed the random variables Y into a whole family of random variables $Y(\beta)$ such that

1. we can bound the difference $D(Y(\beta))$ from above for two special values of the parameter β
2. the resulting bounds lead to an upper bound for the original difference $D(Y)$.

Lemma 5.7. *Let $X_1, \dots, X_n$ be i.i.d. $[0,1]$-valued random variables and let $c > 0$ such that $\mathbb{E}[X_1] > c$ and*

$$\begin{aligned} I\!P(X_1 \in \{1, v_{n-1}, v_{n-2}, \dots, v_2, v_1, 0\}) = 1, \\ I\!P(X_1 = 1) > 0, \\ I\!P(X_1 = 0) > 0, \end{aligned}$$

are satisfied. Then,

$$c' = c - I\!P(X_1 = 1) < 0.$$

We are establishing a property of the cost c relative to the probability of seeing a 1.

Proof. Assume for the sake of contradiction that $c' \geq 0$, that is,

$$\mathbb{P}(X_1 = 1) \leq c.$$

We know for certain that $X_1 \in \{0, v_1, \dots, v_{n-1}, 1\}$, and as a result

$$v_{n-1} = \mathbb{E}[\max\{X_1, v_{n-2}\}] - c$$

We expand the expectation over the discrete support. Since X only takes values of $0, v_1, \dots, v_{n-1}, 1$, the max implies we ignore values below v_{n-2}. So, we have

$$v_{n-1} = v_{n-2} \cdot \mathbb{P}(X_1 < v_{n-1}) + v_{n-1} \cdot \mathbb{P}(X_1 = v_{n-1}) + \mathbb{P}(X_1 = 1) - c$$

Using $v_{n-2} < v_{n-1}$(strict monotonicity proved in Lemma 5.6), we get:

$$\begin{aligned} v_{n-1} &\leq v_{n-1} \cdot \mathbb{P}(X_1 < v_{n-1}) + v_{n-1} \cdot \mathbb{P}(X_1 = v_{n-1}) + \mathbb{P}(X_1 = 1) - c \\ &< v_{n-1} \cdot (1 - \mathbb{P}(X_1 = 1)) + \mathbb{P}(X_1 = 1) - c \\ &= v_{n-1} - v_{n-1} \cdot \mathbb{P}(X_1 = 1) + \mathbb{P}(X_1 = 1) - c \\ \implies 0 &< \mathbb{P}(X_1 = 1)(1 - v_{n-1}) - c \end{aligned}$$

Since $v_{n-1} < 1$, the term is $1 - v_{n-1}$ is positive. If $\mathbb{P}(X_1 = 1) \leq c$, the RHS is negative. Therefore, we have a contradiction and thus can claim $\mathbb{P}(X_1 = 1) > c$. □

We now construct the family of random variables mentioned earlier. Let $X_1, \dots, X_n$ be i.i.d. $[0,1]$-valued random variable and let $c > 0$ such that the conditions we outlined in lemma 5.7 are met. Let

$$\beta^* = -\frac{\mathbb{P}(X_1 = 1)}{c'} \quad (> 0).$$

We now construct a family of random variables

$$\{X_1(\beta), \dots, X_n(\beta)\}_{\beta \in [0, \beta^*]}$$

with corresponding sampling costs

$$\{c(\beta)\}_{\beta \in [0,\beta^*]} \,.$$

For all $\beta \in [0, \beta^*]$, let $v_i(\beta) = \beta \cdot v_i$ for $i = 1, \ldots, n-1$,

$$X_h(\beta) = \begin{cases} 1, & \text{on } \{X_h = 1\}\,, \\ v_i(\beta), & \text{on } \{X_h = v_i\} \ (h = 1, \ldots, n), \\ 0, & \text{on } \{X_h = 0\}\,. \end{cases}$$

and

$$\begin{aligned} c(\beta) &= \beta \cdot c' + \mathbb{P}(X_1 = 1) \\ &= \beta\,(c - \mathbb{P}(X_1 = 1)) + \mathbb{P}(X_1 = 1) \end{aligned}$$

Finally, let

$$Y_h(\beta) = X_h(\beta) - h \cdot c(\beta)$$

for $h = 1, \ldots, n$.

Given the definition β is the uniquely determined zero of the strictly decreasing function $\beta \mapsto \beta \cdot c' + \mathbb{P}(X_1 = 1)$.

$$c(\beta^*) = 0 \implies \beta^* = \frac{\mathbb{P}(X_1 = 1)}{\mathbb{P}(X_1 = 1) - c}$$

Since $\mathbb{P}(X_1 = 1) > c$, the denominator is positive and smaller than the numerator. Thus, $\beta^* > 1$. This means the original problem of $\beta = 1$ lies strictly inside the interval $[0, \beta^*]$. Furthermore, $c(\beta) \geq 0$ for all $\beta \in [0, \beta^*]$, with equality holding iff $\beta = \beta^*$.

Lemma 5.8. *Given the construction above we have*

(a) For $\beta = 1$, the $X_i(\beta)$ and $c(\beta)$ coincide with X_i and c.

(b) For all $\beta \in (0, \beta^*)$, $X_1(\beta), \ldots, X_n(\beta)$ are i.i.d. $[0,1]$-valued random variables such that the second condition in lemma 5.7 is satisfied with respect to the sampling cost $c(\beta)$.

(c) For $\beta = 0$, $X_1(\beta), \ldots, X_n(\beta)$ are i.i.d. random variables with

$$\mathbb{P}\,(X_1(\beta) = 1) = 1 - \mathbb{P}\,(X_1(\beta) = 0)$$

and $c(\beta) = \mathbb{P}(X_1 = 1)$.

(d) For $\beta = \beta^*$, $X_1(\beta), \ldots, X_n(\beta)$ are i.i.d. $[0,1]$-valued random variables and $c(\beta) = 0$.

Proof. We will begin with proving the claim (a), as it is trivial. For any value of $X \in \{0, v_1, \ldots, v_{n-1}, 1\}$, necessarily $X(\beta) = X(1) = X$. Using the cost formula,

$$\begin{aligned} c(1) &= 1(c - \mathbb{P}(X_1 = 1)) + \mathbb{P}(X_1 = 1) \\ &= c \end{aligned}$$

So, our claim for (a) holds true.

Let $\tilde{\beta} = \sup\{\beta \in [0, \beta^*] \mid v_{n-1}(\beta) < 1\}$. Since, $v_{n-1} < 1$, the necessarily $\tilde{\beta} > 1$. Furthermore,

$$0 \leq v_1\,(\beta) \leq \ldots \leq v_{n-1}\,(\beta) \leq 1$$

for all $\beta \in [0, \tilde{\beta}]$, where the first $n-1$ equalities hold exactly for $\beta = 0$ and the last equality holds at most for $\beta = \tilde{\beta}$.

We now claim

$$V(Y_1(\beta), \ldots, V_h(\beta)) = v_h(\beta) = \beta \cdot V(Y_1, \ldots, Y_h)$$

for all $h = 1, \ldots, n-1$ and prove by induction. This will establish that statistician's value function scales linearly with β. It should be noted that $X_1(\beta), \ldots, X_n(\beta)$ are i.i.d. random variables and that with probability 1,

$$X_1(\beta) = \beta \cdot X_1 \cdot \mathbb{1}_{\{X_1<1\}} + \mathbb{1}_{\{X_1=1\}}.$$

Setting $v_0 = 0$ and $v_0(\beta) = 0$ for $h = 1$ and using the IH

$$V(Y_1(\beta), \ldots, Y_{h-1}(\beta)) = v_{h-1}(\beta),$$

for $h = 2, \ldots, n-1$, we get

$$V(Y_1(\beta), \ldots, Y_h(\beta)) = \mathbb{E}\left[\max\left\{X_1(\beta), v_{h-1}(\beta)\right\}\right] - c(\beta)$$

that is the the player pays the cost $c(\beta)$ to observe the first value $X_1(\beta)$ and we either take it or the expected of the remaining $h-1$ values, $v_{h-1}(\beta)$. Then,

$$\begin{aligned}\mathbb{E}\left[\max\left\{X_1(\beta), v_{h-1}(\beta)\right\}\right] - c(\beta) = \mathbb{E}&\left[\max\left\{X_1(\beta), v_{h-1}(\beta)\right\} \cdot \mathbb{1}_{\{X_1<1\}}\right] \\ &+ \mathbb{P}(X_1 = 1) - \beta \cdot c' - \mathbb{P}(X_1 = 1).\end{aligned}$$

We split the expectation into two possible cases $X_1 = 1$ or $X_1 < 1$. If $X_1 = 1$, we know $v_{h-1} < 1$ and since $\beta \leq 1$, then necessarily $v_{h-1}(\beta) < 1$. So, $\max\{1, v_{h-1}(\beta)\} = 1$. Therefore, the expectation is $1 \cdot \mathbb{P}(X_1 = 1)$, which is the sum term. The case for $X_1 < 1$, is left as is with the max function and indicator function. Lastly, expand the cost function. Following that,

$$\begin{aligned}&\mathbb{E}\left[\max\left\{X_1(\beta), v_{h-1}(\beta)\right\} \cdot \mathbb{1}_{\{X_1<1\}}\right] + \mathbb{P}(X_1 = 1) - \beta \cdot c' - \mathbb{P}(X_1 = 1) \\ &= \beta \cdot \left(\mathbb{E}\left[\max\left\{X_1, v_{h-1}\right\} \cdot \mathbb{1}_{\{X_1<1\}}\right] - c' + \mathbb{P}(X_1 = 1) - \mathbb{P}(X_1 = 1)\right).\end{aligned}$$

Here, since $X_1 < 1$, we know $X_1(\beta) = \beta \cdot X_1$ and from the IH that $v_{h-1}(\beta) = \beta \cdot v_{h-1}$. The probability terms cancel out, so we can multiply both by β. All of that to say, the β can be taken as a common factor. Penultimately,

$$\begin{aligned}\beta \cdot &\left(\mathbb{E}\left[\max\left\{X_1, v_{h-1}\right\} \cdot \mathbb{1}_{\{X_1<1\}}\right] - c' + \mathbb{P}(X_1 = 1) - \mathbb{P}(X_1 = 1)\right) \\ &= \beta \cdot \left(\mathbb{E}\left[\max\left\{X_1, v_{h-1}\right\}\right] - c\right).\end{aligned}$$

We regroup the expectation by incorporating the positive probability term. Recall, we previously defined $c' = c - \mathbb{P}(X_1 - 1) \implies -c = -c' - \mathbb{P}(X_1 = 1)$. Finally,

$$\begin{aligned}\beta \cdot \left(\mathbb{E}\left[\max\left\{X_1, v_{h-1}\right\}\right] - c\right) &= \beta \cdot V(Y_1, \ldots, Y_h) \\ &= \beta \cdot v_h \\ &= v_h(\beta),\end{aligned}$$

by the definition of equation 5.2.2. □

By definition, it is clear that $\tilde{\beta} \leq \beta^*$. For the sake of contradiction assume $\tilde{\beta} < \beta^*$. Then by monotonicity and continuity

$$0 < v_1(\tilde{\beta}) < \ldots < v_{n-1}(\tilde{\beta}) = 1$$

and $c(\tilde{\beta}) > 0$, we obtain

$$\begin{aligned} 1 &= v_{n-1}(\tilde{\beta}) \\ &= V(Y_1(\tilde{\beta}), \ldots, Y_{n-1}(\tilde{\beta})) \\ &\leq 1 - c(\tilde{\beta}) \\ &< 1 \end{aligned}$$

which is a contradiction. Hence, $\beta^* = \tilde{\beta}$. It follows that

$$0 \leq v_1(\beta) \leq \ldots \leq v_{n-1}(\beta) \leq 1$$

holds for all $\beta \in [0, \beta^*]$.

Lemma 5.9. *The function*

$$\beta \mapsto M(\beta) = M(Y_1(\beta), \ldots, Y_n(\beta))$$

is convex.

Proof. For almost all $\omega \in \Omega$, the "path"

$$\beta \mapsto Y^*(\beta; \omega) = \max\{Y_1(\beta; \omega), \ldots, Y_n(\beta; \omega)\}$$

is convex, since it is the maximum of the affine-linear functions $\beta \mapsto Y_i(\beta; \omega)$. Thus, the function $\beta \mapsto M(\beta) = \mathbb{E}[Y^*(\beta)]$ is also convex. $\square$

Finally concluding, let $X_1, X_2, \ldots$ and let c be such that $c > 0$, $\mathbb{E}[X_1] > c$. Assuming lemma 5.2.2 and 5.6, we get the conditions of 5.7. Then using the construction, we can claim that the function $\beta \mapsto D(\beta) = M(\beta) - V(\beta)$ is convex, since M is convex and V is linear. Since D is defined over a compact interval and by its convexity we know it must assume its maxima at the boundary of its interval, it follows that

$$\begin{aligned} D(Y) &= D(1) \\ &\leq \max\{D(0), D(\beta^*)\}. \end{aligned}$$

We have Bernoulli variables for $\beta = 0$ which inform us that $D(0) \leq d_n$. We also have $c(\beta) = 0$ for $\beta = \beta^*$, which tells us that $D(\beta^*) < d_n$. Hence, the maxima is bounded by d_n. $\square$

5.2.3 Proof for c)

This proof has been shown in [12] and is dependent on the proof for part a).

Bibliography

[1] M. Qiao and G. Valiant, "Online pen testing," 2022. [Online]. Available: https://arxiv.org/abs/2210.00655

[2] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg, "Matroid secretary problems," *J. ACM*, vol. 65, no. 6, Nov. 2018. [Online]. Available: https://doi.org/10.1145/3212512

[3] A. Ganesh and J. Hartline, "Combinatorial pen testing (or consumer surplus of deferred-acceptance auctions)," *arXiv preprint arXiv:2301.12462*, 2023.

[4] H. Kösters, "Difference prophet inequalities for [0,1]-valued i.i.d. random variables with cost for observations," *The Annals of Probability*, vol. 32, no. 4, Oct. 2004. [Online]. Available: http://dx.doi.org/10.1214/009117904000000496

[5] A. Author and B. Author, "Original paper title," *Journal Name*, YYYY, tODO: Add full citation details.

[6] D. R. Karger, "Random sampling in matroids, with applications to graph connectivity and minimum spanning trees," *SIAM Journal on Computing*, vol. 24, no. 3, pp. 502–517, 1995.

[7] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[8] S. Alaei, H. Fu, N. Haghpanah, J. Hartline, and A. Malekian, "Bayesian optimal auctions via multi-to single-agent reduction," *arXiv preprint arXiv:1203.5099*, 2012.

[9] J. Hartline and S. Taggart, "Sample complexity for non-truthful mechanisms," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, ser. EC '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 399–416. [Online]. Available: https://doi.org/10.1145/3328526.3329632

[10] Q. Yan, "Mechanism design via correlation gap," 2010. [Online]. Available: https://arxiv.org/abs/1008.1843

[11] E. Samuel-Cahn, "A difference prophet inequality for bounded iid variables, with cost for observations," *The Annals of Probability*, pp. 1222–1228, 1992.

[12] F. Harten, *Prophetenregionen bei zeitlichen Bewertungen im unabhängigen und im iid-Fall*, ser. Skripten zur Mathematischen Statistik [Lecture Notes on Mathematical Statistics]. Gesellschaft zur Förderung der Mathematischen Statistik, Münster, 1996, vol. 27, dissertation, Universität Münster, Münster, 1996.

[13] T. P. Hill and R. P. Kertz, "Comparisons of stop rule and supremum expectations of iid random variables," *The Annals of Probability*, pp. 336–345, 1982.

[14] Y. S. Chow, H. Robbins, and D. Siegmund, *The theory of optimal stopping*, ser. Dover books on advanced mathematics. Dover, 1991. [Online]. Available: https://cir.nii.ac.jp/crid/1971430859757620142

2 Methodology

2.1 Theoretical Framework and Analysis

As this thesis is found within the field of Theoretical Computer Science, the primary methodology is mathematical analysis and algorithm design instead of empirical data collection. My research process can differentiated into three parts:

1. Problem Formulation and Reduction
2. Algorithm Design
3. Algorithm Analysis

A lot of this can only be realized will working out the mathematics. I can draw inspiration from the existing literature, the ones I have reviewed for 499Y and the endless ones I did not have the time to. While there are techniques that could be deployed, this intuition may not be guaranteed and so I must spend time in building the problem entirely before declaring the techniques and frameworks I will use.

2.2 Computational Simulation

While the primary deliverable is a set of mathematical proofs, I will utilize computational simulation as a verification tool. I plan to develop scripts in Python to simulate the proposed algorithms against both random input sequences (the Prophet setting) and adversarial input sequences (the Secretary setting). This will provide empirical evidence as to whether my algorithms work or not.

2.3 Resources and Training

The resources required for this research are primarily informational and computational. I will rely on the existing body of literature regarding Online Algorithms, Mechanism Design, and Matroid Theory, accessed through university library databases (e.g., ACM Digital Library, arXiv).

2.4 Specialized Training and Compliance

This research is entirely theoretical and computational. It involves no interaction with human subjects, animal subjects, or hazardous materials.

3 Evaluation

The successful completion of this honors thesis will be evaluated based on the production of a formal research manuscript and the validity of the theoretical results derived therein.

My committee expects the following deliverables by the end of the semester:

1. Problem Definition: A precise mathematical formulation of the Combinatorial Pen Testing and the Generalized Cost environments
2. Algorithm Design: Novel online algorithms for the Prophet and Secretary settings under these constraints
3. Proofs: A rigorous derivation for the bounds of the algorithms design as to determine the "quality" of them

It is also entirely possible, due to the nature of thesis, that I do not create a new algorithm. However, it is also in the realm, to examine particular settings and draw novel conclusions, that are not currently known. This too would be a meaningful thesis, in that it has produced something new to this problem space.

3.1 Feedback and Assessment

Progress will be monitored through weekly 30-minute meetings with Prof. Mingda Qiao, focused on proof sketches and technical roadblocks. I will also provide monthly updates to my second committee member. The committee will assess the project based on the correctness of the proofs.

4 Communication

- Faculty Meetings: I will meet with my faculty sponsor, Professor Mingda Qiao, for 30 minutes weekly. These meetings will focus on reviewing proof sketches, addressing technical roadblocks, and tracking timeline progress.
- Committee Updates: I will provide monthly email updates to my second committee member and convene the full committee at least once prior to the final defense.
- Time Commitment: I plan to dedicate approximately 10–12 hours per week

5 Timeline

- **February 15:** Submission of detailed thesis outline and formalized problem definitions for the Matroid and Generalized Cost settings.
- **March 13:** Submission of First Draft. This will include the Introduction, Literature Review, and core theoretical results (theorems and proof sketches).
- **April 20:** Submission of Second Draft. This will be the complete manuscript including refined proofs, simulation results, and conclusion.
- **May 1:** Submission of the Final Written Component, incorporating all committee feedback.

- **May 4 – May 8:** Final Presentation and Oral Defense. The specific date and setting will be coordinated with the committee members later in the semester.

References

[1] M. Qiao and G. Valiant, “Online pen testing,” 2022. [Online]. Available: https://arxiv.org/abs/2210.00655