Beyond Pandora's Box: Algorithm and Mechanism Design with Costly Information Acquisition

by

Robin Bowers

B.A., Oberlin College, 2020

M.S., University of Colorado Boulder, 2024

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2026

Committee Members:

Bo Waggoner, Co-Chair

Rafael Frongillo, Co-Chair

Robert Kleinberg

Shuchi Chawla

Sriram Sankaranarayanan

Bowers, Robin (Ph.D., Computer Science)

Beyond Pandora's Box: Algorithm and Mechanism Design with Costly Information Acquisition

Thesis directed by Prof. Bo Waggoner and Prof. Rafael Frongillo

In many real-world decision-making settings, the true values of alternatives are not immediately known. Instead, a decisionmaker must invest in acquiring *costly information* about various alternatives before making a decision. How should that information be dynamically acquired to make the best decisions, while wasting value performing unnecessary inspections? This work considers the *Pandora's Box* model of costly information acquisition and develops a constructive understanding of boxes of arbitrary depth. We further extend this model to develop more complex *Markov Search Processes*, in which a decisionmaker can choose from multiple methods of information discovery at each step.

In the matching setting, we prove a price of anarchy result for the Marshallian Match mechanism both with and without costly information, when all values are nonnegative. We then consider the case of negative values, and find that the problem requires a high level of coordination across agents even when all agents are perfectly truthful, i.e. in the algorithmic setting.

Finally, we develop algorithms with a constant approximation factor for selection over Markov Search Processes (Combinatorial Markov Search). We provide an adaptivity gap-like result and use a novel prophet inequality construction to provide an efficient, $1/2 - \varepsilon$ algorithm which implies a truthful mechanism with a matching price of anarchy.

"I am losing my illusions, perhaps to acquire new ones."

Virginia Woolf, *Orlando: A Biography* (1928)

Acknowledgements

There are so many people who have helped me in completing this dissertation. Thank you first of all to my advisors, Bo and Raf, for giving me free rein to explore the things I've found meaningful. Thank you to my committee your encouragement and for seeing value in my work.

My family has been incredibly supportive of me throughout my academic career, cheering me on wherever my choices have taken me. Thank you to my mom Amy for walking me through my first experiences in computer science, and to my dad Jim for listening to all my PhD woes. Thank you to my grandparents, Kaarli and Gil, for your happiness in everything I've chosen to do with my life. Thank you to Alice for only biting me when I deserve it.

My amazing lab and friends have meant so much to be over the past five years. Elias, so much of this work was only possible with your help and enthusiasm. Jessie, your mentorship has helped me grow in so many ways. Thank you especially to Robby for spending the past year and a half with me, and Melody for getting strong with me and making things much too small. Thank you to Rebecca and Carolyn for many years of reliable games and friendship, and to Kelsey for your joy and wisdom. Thank you to Claudia for some truly excellent television and conversations over the years. Thank you Ky for your support and friendship, our spontaneous shenanigans and study sessions have gotten me through the past year. I am so grateful to have all of you in my life.

Finally, a few people have mentored me beyond what I could ever have asked for. Thank you George Schnurle for showing me the beauty in a different kind of mechanism design. Thank you to my undergraduate advisor (and secret third PhD advisor) Sam Taggart for introducing me to this field. Your enthusiasm, support, and belief in me far beyond my own has meant the world to me.

Contents

Chapter

Figures

Figure

Chapter 1

Introduction

Many real-world algorithmic problems involve unknown information that must be acquired to make good decisions. For example, a company looking to hire a new employee must first gather information on and interview candidates before making a decision. Similarly, an employee may need to do some research into a potential company before applying for a position. Alternately, a researcher may be deciding which of several approaches to take to a problem, each with different costly equipment which must be purchased. Choosing a promising direction must be balanced against the cost of investing in equipment for every possible course of action. We consider decisionmaking under these forms of costly information across several settings, adopting the Pandora's Box Model of Weitzman [1979].

Chapter 2 gives a high-level introduction to the tools and techniques of Pandora's Box, including developing a constructive approach to sequential inspection of "nested" boxes, or *bandits*. We warm up by using these tools for a simple proof of Weitzman [1979]'s algorithm, using the techniques of Kleinberg et al. [2016].

1.1 Matching

We then turn to applications: Chapter 3 considers the job market matching setting, and analyzes the *Marshallian Match* mechanism for two-sided matching. We prove price of anarchy guarantees for several related mechanisms in similar settings when values are positive, before developing an understanding of why the crucial negative-values regime fails. Chapter 4 takes an

algorithmic approach to the two-sided matching problem and obtains a number of negative results for various classes of algorithms, demonstrating the difficulty of matching under such settings. Finally, we give an approximation algorithm based on a randomization of edges.

We then move on to another decisionmaking setting, capturing the researcher investing in equipment for a lab.

1.2 Combinatorial Markov Search

In much recent work growing out of the Pandora's Box literature [Beyhaghi and Cai, 2024], a decisionmaker is presented with a set of alternatives of uncertain value. She must dynamically acquire costly information and choose a valuable alternative or subset of alternatives. We propose a general model for this problem in which the decisionmaker's n alternatives $\mathcal{M}_1, \ldots, \mathcal{M}_n$ are each a **Markov Search Process (**MSP**)**, a variant of a Markov Decision Process.[1] Each MSP $\mathcal{M}_i$ is defined by a choice of actions, each associated with a cost and a stochastic transition to a new state; the new state has a new choice of costly actions, and so on until a terminal state is reached, revealing an available reward for choosing alternative $\mathcal{M}_i$.

In the **Combinatorial Markov Search (**CMS**)** problem, the decisionmaker is given a full description of $\mathcal{M}_1, \ldots, \mathcal{M}_n$. She interacts by repeatedly selecting any $\mathcal{M}_i$, taking an action, paying the incurred cost, and observing its state transition. She then selects another $\mathcal{M}_i$ (or possibly the same one), and so on. At any point, she may decide to stop and claim some subset of the MSPs, obtaining their associated revealed rewards. The accepted subset F must belong to a given feasibility system $\mathcal{F} \subseteq 2^{\{1,\ldots,n\}}$, such as a matroid, and we call the associated problem CMS($\mathcal{F}$). For example, in **single selection**, $\mathcal{F} = \{\emptyset, \{1\}, \ldots, \{n\}\}$, denoting that at most one of the MSPs can be claimed. The **welfare** of the algorithm is the sum of rewards claimed minus the sum of all costs incurred. The algorithm is an α-approximation if its expected welfare on every instance is at least an α fraction of the expected welfare of the optimal, fully-adaptive, computationally

[1] An MSP has a finite directed acyclic graph structure; no discount factor; actions only generate costs (not rewards); and has terminal states that are associated with an available reward.

unconstrained algorithm.

CMS models problems such as a company investing in the development of its next product, which may be one of n potential options. Each option is a Markov Search Process involving research and development choices that require resources and may open the door to further choices. The company may interactively investigate and develop the potential options in parallel, e.g. abandoning some when they become less promising, and eventually choosing one to bring to market.

1.3 Related Work

This work primarily focuses on extensions of the Pandora's Box Problem, but draws on techniques from several strands of literature.

1.3.1 Pandora's Box & Bandits

Chapter 2 introduces the Pandora's Box Problem of Weitzman [1979], and develops generalization tools for what we refer to as *bandits*. Our terminology and machinery for the problem follow Kleinberg et al. [2016], who used the problem in an auction setting, as well as a one-sided matching setting (i.e. matching with a single Pandora box on each edge).

The Weitzman index approach and optimal descending algorithm was shown to extend to the bandit case by Kleinberg et al. [2016, Appendix G]. The treatment there is quite general (in particular, handling countably infinite stages of inspection), but it is also quite complex and nonconstructive. In Section 2.3, we include a concrete and direct extension of the Weitzman index machinery to the nested-boxes setting. We present the key features in the main body of the paper because, in addition to enabling our final results, they do provide some novel points of utility and interest:

- A simple and direct construction with a much shorter, self-contained treatment. Given an example (e.g. a Bernoulli value whose bias is selected over the course of the inspections), it is not necessarily clear from Kleinberg et al. [2016] how exactly to compute the Weitzman indices; here we give a direct formula.

- Illustration of several subtle points that arise in the analysis, such as a precise and constructive formulation of the *non-exposed* condition (also called exercise in the money, claim above the cap, etc.).

- Simple and concrete machinery – Weitzman indices (also called strike prices or reservation values) capped values (also called covered call values or deferred values), and a general "amortization lemma" – that is more amenable to adaptation to different environments.

Gupta et al. [2019] provides a generalization of Singla [2018]'s frugal algorithms to a multistage inspection model for combinatorial optimization. They consider optimization over boxes represented by Markov chains with costs for advancement, where a chain must be advanced to a terminal state in order for the box to be added to the solution. Gupta et al. [2019] derives a parallel to the Weitzman index machinery of Kleinberg et al. [2016] in the discrete Markov decision process setting. Their approach is essentially a generalization of Section 2.3, except with a focus on compatibility with frugal algorithms.

Several other works study variants of the bandit problem, but they are much farther from our setting. Ke and Villas-Boas [2019] study a model in which a decisionmaker gathers information about each of a set of alternatives in continuous time for continuous cost, and stops at any time to choose an alternative. (In particular, full inspection is not obligatory.) Aouad et al. [2020] consider a model with two stages of inspection, partial and full. However, the algorithm does not have to undertake partial inspection of an option prior to a full inspection. (Full inspection is still obligatory in order to claim.) In this setting, Aouad et al. [2020] show that a descending index-based policy is no longer optimal, studying features of the optimal solution and approximation algorithms. Boodaghians et al. [2020b] study a somewhat-related setting that can be viewed as Pandora's box problem where boxes must be opened in a particular order, particularly when the ordering follows a linear or DAG structure.

There has been a variety of other recent work on variants of the Pandora's Box Problem, including when inspection is non-obligatory to claim the item [Doval, 2018, Beyhaghi and Kleinberg,

2019, Beyhaghi and Cai, 2023a, Fu et al., 2023] and when the values in the boxes are correlated [Chawla et al., 2020a]. All of the above settings and a number of others are discussed in the recent survey of Beyhaghi and Cai [2023b].

1.3.2 Matching

Chapters 3 and 4 focus on two-sided matching problems, a setting that has been extensively studied.

Kelso and Crawford [1982] explores a a mechanism for two-sided matching with money under full information, inspired by job markets, through a process reminiscient of both the deferred-acceptance mechanism and an ascending-price auction. They provide stability results for this mechanism given truthful behavior from all agents, but do not analyze the welfare of these outcomes. Our approach, in contrast, utilizes descending-price dynamics and emphasizes welfare over stability.

Probably closest to our work is Immorlica et al. [2021], which considers design of a platform to coordinate two-sided matchings with inspection costs and quantifiable welfare guarantees. Motivated by e.g. matching platforms for romantic dating, the paper studies agents coming from specific populations with a known distribution of types. The mechanism uses its knowledge of the type distributions to compute strategies for directing inspection stages (e.g. first dates). The computational problem is challenging and intricate. Immorlica et al. [2021] is able to show the structure of equilibria and use this to give welfare guarantees, all in a setting without money. In contrast, we are interested in monetary mechanisms, motivated (eventually) by e.g. labor markets. We consider a very simple descending-price mechanism that has no access to knowledge about the agent types or distributions.

Beyond Immorlica et al. [2021] there is an extensive literature on matching marketplaces and dynamics. Work in that literature involving money (transferable utility) historically often takes a Walrasian equilibrium perspective, while ours is in the tradition of auction design. We refer to the survey of Chade et al. [2017] for more on this literature.

A number of recent works study matching markets with information acquisition, but generally

consider variants of deferred acceptance without money. Works of this kind include He and Magnac [2020], Che and Tercieux [2019], Ashlagi et al. [2020], Chen and He [2021], Fernandez et al. [2021], Immorlica et al. [2020a], Hakimov et al. [2021].

Among these, Immorlica et al. [2020a] uses a lens of optimal search theory similar to ours. It studies matching of students to schools. Its matching problem is almost one-sided, in the sense that schools have known and fixed preferences. The focus is on coordinating efficient acquisition of information by students. Unlike our social-welfare perspective, that work focuses on the more standard criterion of stability and introduces regret-free stable outcomes. In particular, it does not involve money. Similarly, Hakimov et al. [2021] study a serial-dictator mechanism for coordinating student inspection without money. We refer to Immorlica et al. [2020a] for an extensive discussion of further work related to information acquisition in matching markets.

Our notions of stability are naturally closely related to others in the literature. Ex post stability is only a quantitative version of stability in matching; a more sophisticated version of it is used by Immorlica et al. [2020a], for example. Fernandez et al. [2021] also utilizes a similar notion of stability, in a setting of incomplete information.

Pandora matching. Our work differs from literature in the Pandora tradition (e.g. Singla [2018]'s frugal algorithms) which also deals with matching constraints in that it has weights on both endpoints rather than one weight per edge. This change means that, in general, frugal algorithms will fail in our setting. We explore this failure in the impossibility result for vertex-based algorithms in Section 4.1, which implies that no variant of the frugal framework can solve our variant of the matching problem.

1.3.3 Combinatorial Markov Search

Chapter 5 considers a generalization in which each box in a series of nested boxes can be opened in different ways. One such well-studied variation on Pandora's Box is "non-obligatory inspection," first posed by Doval [2018]. Beyhaghi and Kleinberg [2019] provides a $1-\frac{1}{e}$-approximation for the problem using techniques from Asadpour and Nazerzadeh [2016], and Scully and Doval [2025]

propose an approximation algorithm via a randomized "committing" algorithm. Fu et al. [2023] established that non-obligatory Pandora's Box is an NP-hard problem, unlike the original and multistage versions of the problem and, along with Beyhaghi and Cai [2023a] independently, provide a PTAS. In Appendix H, we extend the results of Fu et al. [2023] to prove hardness of CMS. Other extensions of Pandora's Box have been studied such as combinatorial selection constraints in Singla [2018], order constraints in Boodaghians et al. [2020a], correlations between boxes in Chawla et al. [2020b], and applications to matching and mechanism design in Immorlica et al. [2020a], Bowers and Waggoner [2024], and Bowers and Waggoner [2026].

1.3.3.1 Prophets

Esfandiari et al. [2019] considers a problem where Pandora boxes arrive online and apply prophet inequality techniques. Their closest result to our prophet work involves a setting inspired by reinforcement learning in which there are a number of rounds, and in each round, a number of Pandora boxes arrive; only one can be opened per round. This setting is special case of our COB Selection model with each bandit consisting only of a Pandora box. Esfandiari et al. [2019] makes several additional assumptions: the set of arriving boxes have the same parameters (i.e. cost and value distributions) in every round, and the benchmark "optimal" algorithm is restricted and not fully adaptive as in our paper. Their constraint, that at most one box of each "type" can be claimed, is generally incomparable to our matroid constraint results, but in other respects their result can be viewed as a highly special case of Theorem 5.0.2. Our key technical tools, such as our "matroid prophet inequality" for the COR Selection setting, our reductions from CMS to COB Selection, our use of the SAUP subroutine and threshold-based algorithms, and our extensive use of the *ex-ante* benchmark, are not related to any of the techniques in Esfandiari et al. [2019].

Independently to this work, Chawla et al. [2026] introduce the *Costly Information Combinatorial Selection (CICS)* problem, which is analogous to our CMS problem while also encompassing minimization problems. They use a similar thresholding approach, but focus on bounding the "commitment gap" of CMS, a quantity studied in several costly information settings. Chawla et al.

[2026] does not consider prophet inequalities at all.

Finally, in subsequent work Chawla et al. [2025] extend the techniques of our paper's Section 5.3 to achieve improved bounds on the commitment gap of the CMS problem.

1.3.3.2 Superprocesses and index theorems

Index theorems are a broad technique for proving guarantees about algorithms for certain problems. As in Weitzman [1979]'s solution to the Pandora's Box Problem, index theorems show that a given problem can be assigned by maintaining some form of indices across a number of different elements, and always interacting with or accepting the element with the highest index.

A superprocess or *alternative decision process* consists of a set of Markov Decision Processes (MDPs), which generate costs or rewards with each action taken [Nash, 1973]. This differs from our MSPs in the presence of *rewards* as well as costs for advancement, rather than having a final reward for selection of a process. Nevertheless, this work is closely related conceptually to work on superprocesses, which generally focuses on sufficient conditions for index theorems (efficient local rules that enable globally optimal algorithms). The most famous is the Gittins index [Gittins, 1979], but many others along this line have been proposed and studied, some allowing for limited decisionmaking or other relaxations of the bandit setting [Glazebrook, 1976, Whittle, 1980, Nash, 1980, Glazebrook, 1982, Granot and Zuckerman, 1991, Glazebrook, 1993, Dumitriu et al., 2003, Keller and Oldale, 2003]. Scully and Terenin [2025]'s recent tutorial on the Gittins index also highlights the close connection to the Pandora's Box problem.

Applications in AI and reinforcement learning include [Brown and Smith, 2013, Hadfield-Menell, 2015]. The main question of this chapter is relevant to this literature: we ask, in a somewhat different setting, how to construct robust and approximate local strategies when index theorems are unavailable.

Guha and Munagala [2013] considers generalizations of the Gittins setting in which an index theorem does not apply, obtaining constant-factor approximations using techniques such as linear programming relaxations. Ma [2018] obtains constant-factor approximation algorithms for broad

structures of superprocesses in a finite-horizon setting. These results can be related to versions of our problem with single selection and where actions always yield rewards, rather than costs as in our model.

1.4 Preliminaries

We now introduce some standard notation and concepts we will use throughout this work.

1.4.1 Matching

Chapters 3 and 4 deal with two-sided matching problems. In these problems, we seek a matching $\mathcal{M}$ between elements or agents, such that the total weight of values in the match is maximized.

1.4.2 Mechanism Design

Several portions of this work consider *mechanism design*. Mechanism design deals with agents who would like to influence the decision made in their own favor and attempts to coordinate an overall beneficial solution by managing agents' self-interested behavior. We introduce several important concepts within the field of mechanism design.

A *mechanism* takes some strategy profiles $b_1, \ldots, b_n$ from each of n agents and outputs some choice. Throughout this work, strategy profiles will largely consist of bids on outcomes and outputs will be either matchings between agents or selections of sets of winning agents.

Agents have underlying values for various matching outcomes: we denote the value of an agent i for matching to agent j by v_{ij}.

1.4.2.1 Equilibria & Social Welfare

A *Nash equilibrium* of a mechanism with given types $\{v_{ij}\}$ is a strategy profile b, consisting of a plan for how to set bids at each moment in time, where each participant maximizes their expected

utility, i.e.

$$\mathbb{E}\, u_i(b) \geq \mathbb{E}\, u_i(b_{-i}, b_i')$$

for all i and for any other strategy b_i' of i, where the randomness is taken over the strategies. A *Bayes-Nash equilibrium* is defined in exactly the same way, but in a setting consisting of a joint distribution over types. In that case, the randomness is taken over both types and strategies (which are maps from an agent's type to a plan for bidding over time).

We use $\text{Welf}(b)$ to denote the expected social welfare, i.e. sum of utilities and payments:

$$\text{Welf}(b) = \mathbb{E}\left[\sum_i v_i(b) + \sum_j v_j(b)\right],$$

where the expectation is over all randomness. In the setting with inspection costs, utility and social welfare also accounts for the loss of utility from the inspection processes; this will be formalized at the relevant point, Section 3.4.

$\text{Welf}(\text{OPT})$ refers to the optimal social welfare. In the Nash setting,

$$\text{Welf}(\text{OPT}) = \max_M \sum_{\{i,j\}\in M} v_{ij} + v_{ji},$$

where the maximum is over all matchings M. In the Bayes-Nash setting,

$$\text{Welf}(\text{OPT}) = \mathbb{E}\left[\max_M \sum_{\{i,j\}\in M} v_{ij} + v_{ji}\right],$$

where the expectation is over the realizations of types.

The *price of anarchy* measures the worst-case ratio of $\text{Welf}(b)$ to $\text{Welf}(\text{OPT})$ in any equilibrium b. For Nash equilibrium, we have

$$\text{PoA} = \min \frac{\text{Welf}(b)}{\text{Welf}(\text{OPT})},$$

where the minimum is taken over all settings (i.e. all types of the participants) and all Nash equilibria b. The Bayes-Nash price of anarchy is defined in exactly the same way, but the minimum is now over all Bayes-Nash settings (i.e. joint distributions on types) and all Bayes-Nash equilibrium strategy profiles b. Note that a Nash equilibrium is a special case of Bayes-Nash where the type

distributions are degenerate. Therefore, a Bayes-Nash price of anarchy result immediately implies a Nash price of anarchy.

Chapter 2

Pandora's Box Tools and Extensions

This chapter introduces the basic model of costly information considered throughout this work. We use the Pandora's Box modle introduced by Weitzman [1979], and rely on related proof techniques from Kleinberg et al. [2016]. amongst others.

Pandora has a number of boxes, each of which contains some unknown valuable reward, but is also costly to open. She may open as many boxes as she wishes and incur the cost for opening each, but may ultimately claim the reward from only one box. In what order should she open her boxes, and when should she stop and take a value from a box she's already opened?

This problem may seem complicated, but Weitzman [1979] presents a simple algorithm (Algorithm 2.1.1). The key challenge is balancing the cost of additional information with exploiting the information and access already acquired.

This section introduces the Pandora's Box Problem and Weitzman's solution formally (Section 2.1), before introducing amortization tools that we will use throughout in our analysis of various algorithms (Section 2.2). We prove the optimality of Weitzman's algorithm as a warm-up to the techniques of the rest of this work, before moving on to new contributions.

In Section 2.3, we provide a constructive extensions to "bandits," processes in which a box contains additional boxes nested with in it which must be opened in sequence before revealing a final value. The techniques developed for bandits will then be used in Chapters 4 and 5.

Finally, we will develop the concept of *descending procedures* in Section 2.4, which we will use to design algorithms in Chapter 4.

2.1 Weitzman's Pandora's Box

In the traditional Pandora's Box Problem, Pandora is presented with n boxes. Formally, a *Pandora box* is a pair (D, c) consisting of the known distribution D of the value $v \in \mathbb{R}$ inside the box along with the cost $c \in \mathbb{R}_{\geq 0}$ to open the box and observe the value. Pandora can pay the costs and open as many boxes as she wishes, but can only claim the value from one box.

The optimal algorithm for the original problem of Weitzman [1979] relies on indices that we will call σ_i for each box i.

Definition 2.1.1 (Weitzman Index & Capped Value)**.** The *Weitzman index* of a Pandora box (D_i, c_i) is the unique σ_i such that

$$\mathop{\mathbb{E}}_{v_i \sim D_i}\left[(v_i - \sigma_i)^+\right] = c_i,$$

and the *capped value* of the box is the random variable $\kappa_i = \min(v_i, \sigma_i)$, where $(\cdot)^+ := \max\{\cdot, 0\}$.

Discussions, such as a financial-options interpretation, can be found in other works such as Kleinberg et al. [2016], Beyhaghi and Cai [2024].

Informally, Weitzman's algorithm sorts all boxes by their Weitzman indices σ_i. Then, as long as the largest observed value from an opened box is smaller that σ_{i^*} for some box i^*, open box i^*. When the largest value seen is bigger than all remaining indices, claim that value. If all remaining indices are non-positive, leave with nothing. Formally, the algorithm is as follows.

Algorithm 2.1.1 (Weitzman's Pandora's Box Algorithm, Weitzman [1979])**.**

Given n Pandora's boxes (D_i, c_i).

- Compute σ_i for all boxes and set $O = \emptyset$.
- While $\max_{i \in O} v_i < \max_{i \in [n] \setminus O} \sigma_i$ and $0 < \max_{i \in [n] \setminus O} \sigma_i$,
  - Open box $i^* = \operatorname{argmax}_{i \in [n] \setminus O} \sigma_i$ and add i^* to O.
- If $\max_{i \in O} v_i < 0$, stop and take no boxes. Otherwise, take the opened box $i^* = \operatorname{argmax}_{i \in O} v_i$.

We introduce some formal notation to describe the outcome of an algorithm for the Pandora's Box problem. Let $\mathbb{I}_i \in \{0,1\}$ be an indicator random variable which is 1 in the event that the algorithm opens box i and $\mathbb{A}_i \in \{0,1\}$ be an indicator random variable for the event that the algorithm makes box i. In the obligatory inspection setting we consider, we assume that $\mathbb{I}_i \geq \mathbb{A}_i$ always, i.e. any box must be opened if it is claimed. Thus, we can write the expected reward of an algorithm as

$$\mathbb{E}[\sum_i (\mathbb{A}_i v_i - \mathbb{I}_i c_i)].$$

We note, importantly, that these indicator variables do not capture the sequence of choices made by an algorithm but only the totality of decisions at the end of a run: if boxes i and j are both opened, we cannot distinguish from the indicator variables alone whther i was opened before j or the other way around. This notation will be helpful in discussing amortization below and will be augmented throughout the rest of this work for more complicated settings.

2.2 Amortization

Weitzman's original algorithm was re-proved using an amortization approach by Kleinberg et al. [2016], who introduce the concept of *exposure* in algorithms.

Definition 2.2.1. An algorithm is *exposed* on box i if there is a nonzero probability that $\mathbb{I}_{ij} = 1$, $v_i > \sigma_i$, and $\mathbb{A}_i = 0$. Otherwise, the algorithm is *non-exposed* on box i.

In other words, an algorithm is exposed on a box if it opens the box, finds that the value is larger than the index, and ultimately fails to claim that box. We call an algorithm *non-exposed* if it is non-exposed on all boxes.

This leads to the key lemma of Kleinberg et al. [2016] bounding the value of any algorithm for Pandora's Box by the value of an algorithm in an amortized, costless world based on capped vaules κ.

Lemma 2.2.1 (Kleinberg et al. [2016])**.** *For any algorithm, for any Pandora box i,*

$$\mathbb{E}\left[\mathbb{A}_i v_i - \mathbb{I}_i c_i\right] \leq \mathbb{E}\left[\mathbb{A}_i \kappa_i\right],$$

with equality if and only if the policy is non-exposed on box i.

Conceptually, this lemma will allow us to "amortize" the costs in the costly information settings away in two important ways. Firstly, we will focus on designing non-exposed algorithms where the equality allows us to analyze algorithm performance in the amortized κ world. Secondly, the lemma allows us to upper-bound the performance of any (possibly exposed) algorithm, and in particular the performance of the optimal algorithm, by its performance in the amortized world.

We recreate the proof of this lemma here to build intuition, as we will develop conceptually similar but more complicated lemmas for several applications throughout this work.

Proof. By defintion of σ_i, $\mathbb{E}\left[\mathbb{A}_i v_i - \mathbb{I}_i c_i\right] = \mathbb{E}\left[\mathbb{A}_i v_i - \mathbb{I}_i \mathbb{E}[(v_i - \sigma_i)^+]\right]$. We observe that the choice to inspect is independent of the realized value, and use linearity of expectation to combine the expectations:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{A}_i v_i - \mathbb{I}_i c_i\right] &= \mathbb{E}\left[\mathbb{A}_i v_i - \mathbb{I}_i (v_i - \sigma_i)^+\right] \\
&\leq \mathbb{E}\left[\mathbb{A}_i v_i - \mathbb{A}_i (v_i - \sigma_i)^+\right] && (2.1) \\
&= \mathbb{E}\left[\mathbb{A}_i \min(v_i, \sigma_i)\right] && (2.2) \\
&= \mathbb{E}\left[\mathbb{A}_i \kappa_i\right].
\end{aligned}$$

The inequality 2.1 comes from the fact that we must inspect before we can consider accepting a box, i.e. $\mathbb{I}_i \geq \mathbb{A}_i$ pointwise. Note that this inequality is an equality exactly when, for every realization where $(v_i - \sigma_i)^+ > 0$ and $\mathbb{I}_i = 1$, $\mathbb{A}_i = \mathbb{I}_i$. This is exactly the condition for non-exposure on box i. The equality 2.2 then follows by a pointwise case analysis of v_i as larger or smaller than σ_i. □

This amortization approach will form the backbone of our results. Our techniques for proving optimality or approximation results throughout this work will broadly proceed as follows: a) introduce an algorithm b) prove the algorithm is non-exposed, c) compare the amortized value of

our algorithm to the amortized value of the optimal algorithm, which upper bounds the true value of the (possiby exposed) optimal algorithm.

As a warm-up to using these techniques, we will prove that Weitzman's algorithm (Algorithm 2.1.1) is optimal.

Proposition 2.2.1. *Algorithm 2.1.1 maximizes the expected reward* $\mathbb{E}\left[\sum_i (\mathbb{A}_i v_i - \mathbb{I}_i c_i)\right]$ *of any algorithm.*

Proof. We break this proof into three parts: a) showing that Weitzman's algorithm is non-exposed, b) upper-bounding the value of the optimal algorithm, and c) computing its amortized reward.

For convenience of notation, we assume there is a box with $c_i = 0$, $v_i = 0$ deterministically and disregard the possibility of selecting no boxes.

Non-exposure. Suppose by way of contradiction that Weitzman's algorithm is exposed. Then, for some realization of values and actions by the algorithm, there is some box j for which $v_j > \sigma_j$ and $1 = \mathbb{I}_j > \mathbb{A}_j$. Since box j was opened, at the time it was opened $\sigma_j \geq \sigma_i$ for all $i \in [n] \setminus O$. Similarly, $v_i < \sigma_j$ for all $i \in O$, as otherwise we would have claimed some other opened box i. Therefore, at the next step, $v_j > \sigma_j \geq \max_{i \in [n] \setminus O \cup \{j\}} \sigma_i, \max_{iO} v_i$. At this step, then, box j should have been claimed, contradicting the assumption that $\mathbb{A}_j = 0$.

Optimal Upper Bound. Let $\mathbb{I}^*$ and $\mathbb{A}^*$ denote the indicator variables for some optimal algorithm. By applying Lemma 2.2.1 to each term i in the sum, the reward of the optimal algorithm is upper bounded by

$$\begin{aligned} \mathrm{OPT} &= \mathbb{E}\left[\sum_i \mathbb{A}_i^* v_i - \mathbb{I}_i^* c_i\right] \\ &\leq \mathbb{E}\left[\sum_i \mathbb{A}_i^* \kappa_i\right]. \end{aligned}$$

Since $\sum_i \mathbb{A}_i^* \leq 1$ pointwise for every realization, an optimistic upper bound on this sum in the amortized world is then

$$\mathrm{OPT} \leq \mathbb{E}\left[\max_i \kappa_i\right],$$

equating to the algorithm which sees every value κ_i and simply picks the largest. We next use the fact that Weitzman's algorithm is non-exposed to show that it exactly matches this optimistic amortized upper bound on OPT.

Algorithm Upper-Bound. As Algorithm 2.1.1 is non-exposed, Lemma 2.2.1 states that its performance is characterized as

$$\text{ALG} = \mathbb{E}\left[\sum_i \mathbb{A}_i \kappa_i\right].$$

We argue that if $\mathbb{A}_{i^*} = 1$ for some i^* then $i^* \in \text{argmax}_i \kappa_i$, and reduce $\sum_i \mathbb{A}_i \kappa_i$ to $\max_i \kappa_i$. If $\mathbb{A}_{i^*} = 1$ then for all other boxes i, $\sigma_{i^*} \geq \min(v_i, \sigma_i)$ and $v_{i^*} \geq \min(v_i, \sigma_i)$. The first claim follows as i^* was selected to be opened: for every opened box i, i^*'s index was higher than the revealed value v_i and for every unopened box i', i^*'s index was higher than i's index. The second claim follows as i^* was claimed using the same logic over the sets of opened an unopened boxes available at the time. Thus, $\kappa_{i^*} = \min(v_{i^*}, \sigma_{i^*}) \geq \min(v_i, \sigma_i) = \kappa_i$ for all other boxes i, giving that $\text{ALG} = \mathbb{E}[\max_i \kappa_i]$, matching the upper bound on OPT and completing the proof. □

We now begin building on these existing results with novel extensions.

2.3 Bandit Extensions

Our first contribution is a constructive index for a more complicated version of the Pandora's Box problem, the *bandit* model. While these results are a special case of the multistage inspection results in Appendix G of Kleinberg et al. [2016] and can be solved using the Gittins index [Gittins, 1979], we note that our proofs are constructive and provide intuition for why this index behaves the same was as the original Weitzman index. This model and adjacent ones have been studied in a number of works, including Kleinberg et al. [2016], Gupta et al. [2019], Chawla et al. [2026, 2025], Gittins [1979]. It has alternately been called the *nested boxes problem*, *multistage inspection*, or *bandit processes*.

2.3.1 Bandit Model

We use the nested boxes language to motivate the bandit model. In the traditional Pandora's Box model Pandora is presented with n boxes, and each box contains a reward. When Pandora chooses to inspect box i, she pays the cost c_i, and finds a reward v_i. We consider a setting in which once Pandora opens the outermost box i, she finds another box $i^{(2)}$. Upon opening box $i^{(2)}$ for cost $c_i^{(2)}$, she finds box $i^{(3)}$ and so on, until finally opening box $i^{(d)}$ and revealing v_i.

Bandit Processes. This nesting of boxes is similar to the notion of bandit processes with known distributions, where every "pull" of a bandit's arm changes the state of the bandit and gains some reward/cost. We refer to the "opening" of a box as advancing a bandit, and refer to the *depth* of the bandit as the number of boxes nested. We adopt the notation of bandit processes for the remainder of this work, and formalize a bandit process as follows.

Definition 2.3.1 (Bandit Process)**.** A *bandit process* i, defined by a pair (D_i, c_i), contains a set of $d \geq 1$ boxes. For $\ell = 1, \ldots, d-1$, advancing the ℓth box in bandit i provides the algorithm with a signal $s_i^{(\ell)}$ regarding the value v_i of the bandit. Opening the dth box reveals value v_i. The signals and final value are drawn jointly from a known distribution, i.e. $(s_i^{(1)}, \ldots, s_i^{(d-1)}, v_i) \sim D_i$. The sequence of costs $c_i = (c_i^{(1)}, \ldots, c_i^{(d)})$ to advance each box is also known.

We refer to the most-recently opened box ℓ in a bandit as the current *stage* of a bandit: When we have not advanced a bandit at all, we are in stage zero. After advancing through all d stages, we are on the dth stage, and after claiming the item, the $d+1$st.

We modify the indicator variables for algorithms $\mathbb{A}$ and $\mathbb{I}$ to accommodate bandits. $\mathbb{A}_i$ continues to denote the claiming of bandit i, but now the indicator variable for advancing from the ℓth stage in bandit i becomes $\mathbb{I}_i^{(\ell)}$. We will let $\mathbb{I}_i^{(d+1)} = \mathbb{A}_i$ for notational convenience in inductive statements. We may also use the term *advancing* a bandit to refer to taking the reward once all stages have been passed through.

All previous stages $k = 1, \ldots, \ell-1$ in the bandit must be advanced through before advancing stage ℓ, and all d stages in a bandit must have been advanced through to claim the bandit. Formally,

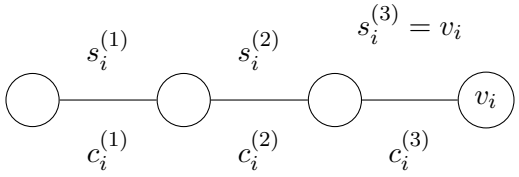


Figure 2.1: Three-stage Bandit

A Three-stage bandit i, with a signal obtained from each advancement. The final signal is also equal to the realized value for convenience.

we require for any algorithm that $\mathbb{I}_i^{(1)} \geq \cdots \geq \mathbb{I}_i^{(d)} \geq \mathbb{I}_i^{(d+1)} = \mathbb{A}_i$.

Optimization problems. In the spirit of Singla [2018], we design notation for bandit processes in general optimization problems. In such a problem we are given a set of bandit processes, each specified by (D_i, c_i), along with a constraint on which subsets of bandit processes may be claimed. An algorithm algorithm performs any sequence of steps in which it selects a bandit and advances it, and eventually stops and claims a subset of fully-advanced bandits that satisfies the constraint. In particular, in the Single-Selection Bandit problem, the algorithm may claim at most one bandit process. This is the simplest generalization of the Pandora's Box problem, in which each box is simply replaced with a series of nested boxes. The *welfare contribution* of bandit i is $\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} c_i^{(\ell)}$. The expected welfare of an algorithm is the sum of the expected welfare contributions of the bandits.

Remarks. The model and results in this section can be generalized so that the "depth" d of each bandit may be a different integer, and so that the costs of inner stages are random variables, with each $c_i^{(\ell)}$ a function of the signals $s_i^{(1)}, \ldots, s_i^{(\ell-1)}$. Essentially no change to the below analysis is required. We will consider these generalizations further in Chapter 5. The case of possibly-infinite depth is treated in Kleinberg et al. [2016].

2.3.2 Amortization for Bandits

Classic Pandora's Box analyses depend on the proper definitions of the index and capped value, as well as on the key lemma (Lemma 2.2.1). We need to generalize these definitions correctly

to bandit processes and prove the analogous lemma. Kleinberg et al. [2016] give more general but nonconstructive definitions, and fleshing out these details reveals a few interesting subtleties and useful intuitions.

Conceptually, when defining an index $\sigma_i^{(\ell)}$ for the stage ℓ, we would like to capture the usefulness of its contents, stage $\ell+1$, by some "proxy value" $\hat{v}_i^{(\ell+1)}$, such that the index could be assigned in the classical way as the solution to $\mathbb{E}[(\hat{v}_i^{(\ell+1)} - \sigma_i^{(\ell)})^+] = c_i^{(\ell)}$. In fact, this turns out to be possible, where the "proxy value" is none other than the capped value of the box at stage $\ell+1$. This allows us to define $\kappa_i^{(\ell)}$ and $\sigma_i^{(\ell)}$ together via backward induction.

Definition 2.3.2 (Bandit Weitzman Index & Capped Value)**.** The *Weitzman index* for box i at the ℓth stage of inspection is the random variable $\sigma_i^{(\ell)}$, a function of $s_i^{(1)}, \ldots, s_i^{(\ell-1)}$, that uniquely solves

$$\mathbb{E}\left[\left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+ \,\middle|\, s_i^{(1)}, \ldots, s_i^{(\ell-1)}\right] = c_i^{(\ell)},$$

where $\kappa_i^{(\ell)}$ is the *capped value* $\kappa_i^{(\ell)} = \min\{\sigma_i^{(\ell)}, \kappa_i^{(\ell+1)}\}$ for $1 \le \ell \le d$, and we define $\kappa_i^{(d+1)} = v_i$. We also define $\sigma_i^{(d+1)} = v_i$.

We also adapt the notion of non-exposure to fit the bandit setting. Intuitively, potential for exposure occurs when the algorithm advances at stage ℓ in bandit i and receives a "good news" signal, resulting in a high Weitzman index for the next stage. To avoid exposure, the algorithm should continue by advancing at the next stage of this bandit as well. Given our claim that $\kappa_i^{(\ell)}$ represents the value of the ℓth stage, we might like to define exposure as a case where $\kappa_i^{(\ell+1)} > \sigma_i^{(\ell)}$, and the algorithm has advanced bandit i to stage ℓ, yet the algorithm does not advance it to stage $\ell+1$. Consider a case, however, where $\sigma_i^{(1)} < \kappa_i^{(3)} < \sigma_i^{(2)}$ and we have advanced to stage 2. In this case, not advancing to stage 3 would cause an exposure *relative to the decision to advance to stage* 2, because $\kappa_i^{(3)} > \sigma_i^{(1)}$, even if there is no exposure relative to $\sigma_i^{(2)}$. This intuition motivates our definition of non-exposure.

Let $\gamma_i^{(\ell)} = \min_{1 \le \ell' \le \ell} \sigma_i^{(\ell')}$, the minimum of the first ℓ Weitzman indices for bandit i. We note that $\gamma_i^{(d+1)} = \kappa_i^{(1)} = \min_\ell \sigma_i^{(\ell)}$. See Figures 2.2 and 2.3 for a visual depiction.

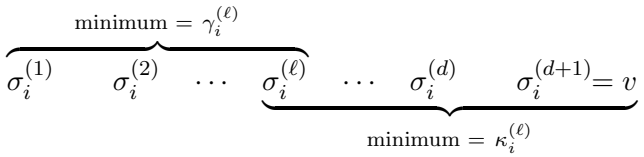


Figure 2.2: γ and κ for bandit processes

The definitions of $\kappa_i^{(\ell)}$ and $\gamma_i^{(\ell)}$ in terms of the bandit indices $\sigma_i^{(\ell)}$ of a bandit i. Note that $\gamma_i^{(1)}, \gamma_i^{(2)}, \ldots$ is a weakly decreasing sequence, while $\kappa_i^{(1)}, \kappa_i^{(2)}, \ldots$ is weakly increasing (see Figure 2.3).

Definition 2.3.3 (Bandit Exposure)**.** An algorithm is *exposed* on bandit i if there is nonzero probability that there exists $\ell \in \{1, \ldots, d\}$ such that $\mathbb{I}_i^{(\ell)} > \mathbb{I}_i^{(\ell+1)}$ but $\gamma_i^{(\ell)} < \kappa_i^{(\ell+1)}$. Otherwise, it is *non-exposed* on bandit i. An algorithm is non-exposed if it is non-exposed on all bandits.

Definition 2.3.3 raises an apparent obstacle. In the classic Pandora's Box problem, it is obvious how to keep an algorithm non-exposed: if the value is higher than the index, claim it. But here, at an arbitrary stage ℓ, $\kappa_i^{(\ell+1)}$ is not visible to the algorithm, as its realization depends on what would happen if the basket continued to advance. Therefore, it is not obvious how to guarantee non-exposure. The resolution comes from observing that $\kappa_i^{(\ell+1)} \leq \sigma_i^{(\ell+1)}$. Therefore, we can prevent exposure by always continuing to advance if the next stage's Weitzman index is larger than $\gamma_i^{(\ell)}$, the minimum of the indices so far. It is necessary to compare to $\gamma_i^{(\ell)}$ rather than $\sigma_i^{(\ell)}$ by the previous discussion.

In the classic setting, the key lemma (Lemma 2.2.1) upper-bounds the expected welfare contribution of a Pandora box, with equality iff non-exposure. It can also be referred to as an amortization lemma, as it relates the average welfare to the average capped value. In our nested setting, the analogue is the following.

Lemma 2.3.1. *For any algorithm, the expected welfare contribution of a bandit process i satisfies*

$$\mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} c_i^{(\ell)}\right] \leq \mathbb{E}\left[\mathbb{A}_i \kappa_i^{(1)}\right],$$

with equality if and only if the algorithm is non-exposed on bandit i.

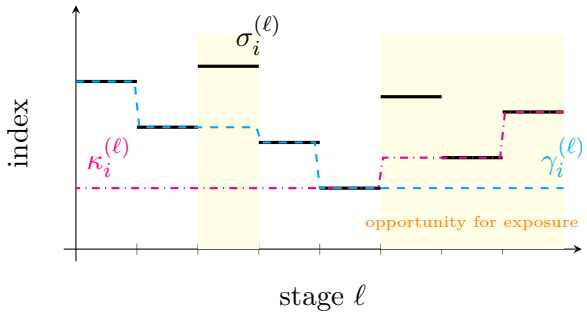


Figure 2.3: Realization of bandit indices and capped values over time

One realization of the Weitzman indices $\sigma_i^{(1)}, \ldots$ when advancing bandit i. The minimum index so far is $\gamma_i^{(\ell)}$ and the minimum yet to come (including the final value) is the capped value $\kappa_i^{(\ell)}$. Yellow regions represent possible opportunities for exposure: if the index $\sigma_i^{(\ell)}$ of the next stage under consideration is higher than the current $\gamma_i^{(\ell-1)}$, it is possible that $\kappa_i^{(\ell)} > \gamma_i^{(\ell-1)}$, in which case failing to continue opening would cause exposure.

We prove Lemma 2.3.1 in Appendix A.

It is not immediately clear that non-exposed algorithms exist, as the conditions in Definition 2.3.3 rely on information unavailable to a decision-maker at the time of each choice.

2.4 Descending Procedures

In this section, we develop a template for desiging non-exposed algorithms, which we categorize as *descending procedures*.

Definition 2.4.1. Call an algorithm a *descending procedure* if: (a) it maintains a non-increasing subset of baskets, the *eligible* set, (b) at each step it advances the eligible basket with the largest Weitzman index, and (c) it never removes a basket from the eligible set in the same step in which the basket was advanced.

The idea of the eligible subset is that in optimization problems such as matching, the constraints often cause an algorithm to permanently rule out some basket from further consideration. This approach is quite similar to the *frugal* algorithms of Singla [2018], but focuses chooses permanent exclusions rather than permanent inclusions. This shift turns out to be critical to the

matching setting.

Lemma 2.4.1. *Any descending procedure is non-exposed.*

We will crucially use Lemma 2.4.1 to show our main positive result for Pandora's Matching Problem.

We introduce some notation and helpful lemmas before proving Lemma 2.4.1. We write $\ell_i(t) \geq 0$ to denote the current stage of bandit i at the beginning of step t. For notational convenience, let $\sigma_i^t := \sigma_i^{(\ell_i(t)+1)}$, the bandit index of stage we are **considering** advancing to in bandit i at time t; Similarly, let $\kappa_i^t := \kappa_i^{(\ell_i(t)+1)}$ and $\gamma_i^t := \gamma_i^{(\ell_i(t)+1)}$. We refer to σ_i^t as the **current Weitzman index** of bandit i at step t.

The proof of Lemma 2.4.1 is not as immediate as one might expect. As mentioned above, there can be cases in which the index σ_i^t decreases after advancing bandit i, yet to avoid non-exposure, we must continue advancing. A key observation is that the descending procedure always advances a bandit with maximum γ_i^t. However, this is not quite enough on its own either, as an increase in σ_i^t does not correspond to an increase in γ_i^t. The second key observation is that the descending procedure only switches away from a bandit i when $\gamma_i^t = \sigma_i^t$. These points are captured in the following lemma.

Lemma 2.4.2. *At each time t, a descending procedure satisfies the following: it always advances an eligible bandit i with maximum γ_i^t, and for all other eligible $j \neq i$, $\gamma_j^t = \sigma_j^t$.*

Lemma 2.4.2. We argue by induction on the step of the algorithm. At $t = 1$, we have for all eligible i that $\sigma_i^t = \sigma_i^{(1)} = \gamma_i^{(1)} = \gamma_i^t$, and the descending procedure selects the maximum index. Assume the claim holds up to and including step $t-1$, when we advanced bandit i'. Now let i be the bandit that is advanced in step t. Observe that, for any currently eligible bandit $j \notin \{i, i'\}$, j was eligible in step $t-1$ by the nonincreasing property, and because j was not advanced, we have $\gamma_j^t = \gamma_j^{t-1} = \sigma_j^{t-1} = \sigma_j^t$ as required.

There are two cases: $i = i'$ and $i \neq i'$. In the first case, because the algorithm chose to advance i on steps $t-1$ and t, we have $\gamma_i^{t-1} \geq \gamma_j^{t-1} = \gamma_j^t$ and $\sigma_i^t \geq \sigma_j^t = \gamma_j^t$ for all eligible $j \neq i$.

This implies $\gamma_i^t = \min\{\gamma_i^{t-1}, \sigma_i^t\} \geq \gamma_j^t$. So γ_i^t is maximum and we are done. In the second case, we have for all eligible $j \neq i$ that $\gamma_j^t \leq \sigma_j^t \leq \sigma_i^t = \gamma_i^t$, so γ_i^t is maximum. Furthermore, in particular $\sigma_{i'}^t \leq \sigma_i^t \leq \gamma_{i'}^{t-1}$, so $\gamma_{i'}^t = \min\{\gamma_{i'}^{t-1}, \sigma_{i'}^t\} = \sigma_{i'}^t$. This completes the induction proof. □

We now use Lemma 2.4.2 to prove Lemma 2.4.1.

Proof of Lemma 2.4.1. To prove the lemma, suppose that $\mathbb{I}_i^{(\ell)} = 1$ and $\gamma_i^{(\ell)} < \kappa_i^{(\ell+1)}$. We must show that $\mathbb{I}_i^{(\ell+1)} = 1$. Let t be the step on which the algorithm opened basket i's box ℓ. We will show that i is eligible at time $t+1$ and that $\sigma_i^{t+1} > \sigma_j^{t+1}$ for all eligible $j \neq i$, implying that the descending procedure advances bandit i on step $t+1$, so $\mathbb{I}_i^{(\ell+1)} = 1$ as required.

First, because bandit i was advanced at step t, by definition of a descending procedure, i is eligible at step $t+1$. Now consider any bandit j that is eligible at step $t+1$. Because eligible sets are nonincreasing in time, j was eligible at time t. Because j was not advanced at time t, Lemma 2.4.2 give us $\sigma_j^{t+1} = \gamma_j^{t+1} = \gamma_j^t$. And, because the descending procedure selected i at time t, Lemma 2.4.2 also gives $\gamma_i^t \geq \gamma_j^t$. Finally, we supposed $\gamma_i^t < \kappa_i^{t+1} \leq \sigma_i^{t+1}$. This gives $\sigma_i^{t+1} > \sigma_j^{t+1}$ for all $j \neq i$, completing the proof. □

Bandit Single-Selection Sketch. As an introduction to using the descending procedure framework, we sketch a proof that the natural descending algorithm is optimal for single-selection of bandits, the simplest extension of Weitzman's original Pandora's Box problem. The descending algorithm always advances the basket i with maximum current Weitzman index σ_i^t, until one has been claimed or all indices are negative. Its capped value is pointwise maximum, i.e. $\sum_i \mathbb{A}_i \kappa_i^{(1)} = \max_i(\kappa_i^{(1)})^+$ in all realizations. The key point in the proof is that at each time basket i was advanced a stage, its Weitzman index was larger than all other baskets', so its minimum index exceeds the minimum of the others, i.e. $\kappa_i^{(1)} \geq \kappa_j^{(1)}$ for all j. Now, the algorithm is indeed a *descending procedure*, so it is non-exposed by Lemma 2.4.1. By Lemma 2.3.1 then, (a) its welfare is equal to its expected capped value $\mathbb{E}[\max_i(\kappa_i^{(1)})^+]$, and (b) no algorithm can do better.

We will ultimately apply this lemma extensively in Section 4.2 to derive our approximation algorithms for matching with costly information on negative values.

Chapter 3

Matching Mechanisms

After introducing mechanical tools and generalizations of Weitzman's Pandora's Box problem, we turn to applying the model more broadly. Singla [2018] replaces the single-selection constraint of Weitzman's original problem with a broad class of combinatorial constraints and considers algorithmic solutions based on Weitzman indices and introduces *frugal* algorithms, a similar concept to greedy algorithms. We are interested in the matching problem where each endpoint of each edge is a Pandora box and both must be opened to match an edge. This setting is not addressed by his results in because greedy max-weight matching, with a weight on each endpoint of an edge, is not a frugal algorithm: it needs to consider the contributions of pairs of items (both endpoints of an edge), while a frugal algorithm considers one at a time. In particular, our results in Section 4.1 imply that no variant of the frugal framework can solve our problem.

Our two-sided matching problem models, for example, job interviewing: a matching is an assignment of employees to available jobs, where both sides of every match must invest in learning their value for a given match. Two models of interest here are the algorithmic and mechanism design versions of this problem: the algorithmic design focuses on a central planner coordinating all choices, while the mechanism design problem directly models the interests of individual vertices making choices. The mechanism design version of this problem was originally introduced by Waggoner and Weyl [2019], who proposed the *Marshallian match* mechanism for matching agents. This chapter considers focuses on the the mechanism design setting and provides welfare guarantees for the Marshallian match mechanism, while Chapter 4 addresses the algorithmic questions.

3.1 Related Work

Related works in the market design literature generally focus more on modeling of matching marketplaces than we do here. Immorlica et al. [2021] studies design of a platform to facilitate two-sided matching with inspection costs. There is a continuous population of agents and matching occurs in continuous time, with the platform directing agents of a given type to "meet" and inspect counterparts of some other type. The authors give a 1/4-approximation to the problem of finding an optimal flow process, despite adhering to incentive-compatibility constraints. Immorlica et al. [2020a] and Hakimov et al. [2023] both study the college application and admissions process with costly information acquisition through a theoretical lens; Immorlica et al. [2020a] include a case where each student is modeled as a Pandora box (two-sided matching with one-sided information acquisition). Other works such as Ashlagi et al. [2020] study the cost of learning preferences through a lens of communication complexity. Chade et al. [2017] surveys prior approaches in the economics literature to matching markets with information acquisition, but the models and techniques are quite different from our problem.

We begin by focusing on the setting without the Pandora model of costly information, focusing on a mechanism for matching where agents can exchange money with one another and with the mechanism.

3.2 Model

While most matching applications focus on bipartite settings, we do not place this restriction on our mechanism at this point. We also assume general values, which may give agents a negative value for a match. Later sections will consider specific restrictions. We formalize the setting as follows.

There is a finite set of n agents, forming vertices of an undirected graph $G = (\{1, \ldots, n\}, E)$. For now, we do not assume that G is bipartite. The presence of an edge $\{i, j\} \in E$ represents that it is feasible to match agents i and j. In this case, agent i has a possibly negative *value* $v_{ij} \in \mathbb{R}$

for being matched to j, and symmetrically, j has a value v_{ji} for being matched to i. If i and j are neighbors, we let $s_{ij} = v_{ij} + v_{ji}$ denote the *surplus* of the edge $\{i, j\}$.

An agent i's *type* consists of their values v_{ij} for each feasible partner j. In the *Nash* setting, each agent has a fixed type, and types are common knowledge. In the better-motivated *Bayes-Nash* setting, types are drawn from a common-knowledge joint distribution $\mathcal{D}$ and each agent observes their own type.

3.2.1 the Marshallian Match

We focus our work on a related mechanism to that proposed by Waggoner and Weyl [2019], and define a generalization of both below.

In the α-Marshallian Match (MM), a global price $p(t)$ begins at $+\infty$, i.e. $p(0) = \infty$, and descends continuously in time until it reaches zero at time 1, i.e. $p(1) = 0$.[1] Each agent i maintains, for all neighbors j, a bid $b_{ij}(t)$ at each time t. The mechanism can observe all bids at all times, but agents cannot observe any bids except their own. For convenience, we may drop the dependence on t from the bid notation. When the sum of bids on any edge exceeds the global price, i.e. $b_{ij} + b_{ji} \geq p(t)$, then i and j are immediately matched to each other. Each agent pays their respective bid to the mechanism.

The mechanism keeps 2α-fraction of this total payment for $\alpha \in [0, 1/2]$ and returns an α-fraction to each player.

Two specific variants of interest are the $\alpha = 0$ variant with no rebate, which is the original mechanism proposed by Waggoner and Weyl [2019], and the $\alpha = 1/4$-rebate Marshallian Match, on which we focus our analysis. For the classic zero-rebate mechanism we prove a price of anarchy of 1/8 in Appendix B, matching the results of the 1/4-rebate Marshallian Match in this chapter. Additionally, we prove that the full-rebate Marshallian Match mechanism has unbounded price of anrachy in Appendix C when graphs are non-bipartite.[2]

[1] This can be accomplished in theory by letting the price be e.g. $p(t) = \frac{1}{2t}$ for $t \in [0, 1/2]$ and $p(t) = 2 - 2t$ for $t \in [1/2, 1]$.

[2] While this mechanism is budget-balanced and makes no profit, this result does not immediately follow from

Intuition for the mechanism. Why might this mechanism be good, and how might participants strategize? We briefly describe some intuition, referring the reader to Waggoner and Weyl [2019] for more detailed discussion.

First, the edges of the graph are in competition with each other to match first. When an edge $\{i, j\}$ is matched, it produces a surplus of $s_{ij} = v_{ij} + v_{ji}$. The first-best solution, i.e. the optimal solution for a central planner who holds all information, is to select a maximum matching where s_{ij} are the edge weights. However, as discussed in Kleinberg et al. [2016], algorithms for maximum matching are complex and appear to interact poorly with inspection stages. More robust is an approximate first-best solution: the *greedy* matching, where the highest-weight edge is matched first, and so on down. This procedure obtains at least half of the optimal welfare, and can be simulated by a Marshallian Match in which all participants bid truthfully. In Kleinberg et al. [2016], this fact was used to obtain a constant price of anarchy for matching people to items, including in the presence of inspection costs.

However, two-sided matching introduces new strategic considerations. "Within" an edge $\{i, j\}$, there is a competition or bargaining for how to split the surplus generated. An agent i with many edges (many outside options) may be able to underbid significantly, while her counterpart j with very few outside options must offer a very high bid along that edge. The question is whether this strategizing and competition between i and j destroys the cooperation incentives for the overall bid on the edge $\{i, j\}$.

3.2.2 Notation, strategies, and welfare

A strategy b_i for player i consists of a plan[3] $b_{ij}(t)$ for how to set bids over time, for each neighbor j.

A full strategy profile is denoted $b = (b_1, \ldots, b_n)$. We let $v_i(b)$ be i's value for their match when

payment-free mechanism results as agents may still make positive payments to one another.

[3] We observe that usual nuances around equilibrium in dynamic games, such as non-credible threats, refinements such as subgame perfect equilibrium, etc., do not arise here. In our variant of the MM, each agent i observes nothing until they are matched, after which they can no longer affect the game. So without loss of generality, i commits in advance to a plan $\{b_{ij}(t)\}$ and follows it until matched.

b is played (zero if none), a random variable depending on the randomization in the strategies and, in the Bayes-Nash setting, on the random draw of the types. Next, $\pi_i(b)$ denotes i's net payment to the mechanism, i.e. bid minus rebate. Finally, $p_i(b)$ denotes the *total* net payment on the edge that i is matched along, i.e. $p_i(b) = \pi_i(b) + \pi_j(b)$ when i is matched to j.

We assume all agents are Bayesian, rational, and have preferences quasilinear in payment. That is, given a mechanism and a particular strategy profile b, the *utility* of a participant i is the random variable

$$u_i(b) = v_i(b) - \pi_i(b).$$

For example, if under profile b, i matches to j at time t, then $\pi_i(b) = b_{ij}(t) - \frac{p(t)}{4}$ and $u_i(b) = v_{ij} - b_{ij}(t) + \frac{p(t)}{4}$.

We next consider a restriction of the general setting where all values v_{ij} are nonnegative. First, we show that the 1/4-rebate Marshallian Match achieves a constant approximation of optimal welfare for matching. Second, we extend the result to the group formation (i.e. matchings on hypergraphs) setting. Finally, we extend the result in a different direction to the case where participants do not initially know their valuations and can choose to expend effort to discover them.

3.3 Approximation Guarantee of the Marshallian Match

Here, we take the general model of Section 3.2 and assume that each valuation satisfies $v_{ij} \geq 0$. We modify the Marshallian Match to require all bids b_{ij} to be nonnegative at all times.

Intuition. The nonnegative MM is similar to running multiple interlocking descending-price unit-demand auctions simultaneously.[4] This parallel is most obvious in the bipartite case: any bidder i competes against other bidders in the same set for her favorite matches. Extending this perspective, each bidder i could hypothetically bid as though her neighbors j were simply items with no preferences. By analyzing the failure of this hypothetical strategy, we find in the MM that

[4] Although general price of anarchy results are available for these kinds of auctions for goods, e.g. Lucier and Syrgkanis [2015], Feldman et al. [2016a], we do not know of any that apply to two-sided matching.

it is connected with a different high-welfare event, namely j matching early. This intuition underlies our smoothness lemma, discussed next.

3.3.1 Smoothness for two-sided matching

We give a smoothness lemma that powers our price of anarchy result. Recall (e.g. Roughgarden et al. [2017]) that smoothness proofs of PoA proceed by guaranteeing high-welfare events in a counterfactual world where i deviates to a less-preferred strategy. One challenge is that in a dynamic mechanism that proceeds over time, a deviation could cause chain reactions that make it impossible to reason about the outcomes. Here, we rely on our variant of MM that does not leak any information about bids or strategies. The only piece of information the agent receives from the mechanism comes at the moment they are matched, after which they cannot react.

A second key challenge is that in a matching market, i may not match j *and* the prices that i and j each pay may still be low, obstructing an adaptation of a standard smoothness proof. In the MM there is, however, a high *total* price paid for an edge that obstructs the match. Recall that while $\pi_i(b)$ is i's payment, $p_i(b)$ is the *total* payment on the edge containing i that is matched (zero if i is unmatched).

The deviation and smoothness lemma. Let b be any strategy profile. For any bidder i, define the deviation strategy b'_i as $b'_{ij}(t) = v_{ij}$ for all feasible neighbors j and all times t. That is, the deviation strategy is simply truthful bidding.

Recall that for the 1/4-rebate MM, i's utility in deviation when matching to j at bid $b'_{ij}(t)$ and price $p(t)$ is

$$\begin{aligned} u_i(b_{-i}, b'_i) &= v_{ij} - b'_{ij}(t) + \frac{p(t)}{4} \\ &= \frac{p(t)}{4}. \end{aligned}$$

Lemma 3.3.1. *In the nonnegative values setting, for any feasible pair $\{i, j\}$, any strategy profile b, and any realization of types,*

$$u_i(b_{-i}, b'_i) + \frac{p_i(b)}{4} + \frac{p_j(b)}{4} \geq \frac{v_{ij}}{8}. \tag{3.1}$$

Proof. All three terms on the left-hand side are nonnegative. Therefore, if $p_j(b) \geq v_{ij}/2$ or $p_i(b) \geq v_{ij}/2$, the result is immediate.

Otherwise, under b, neither i nor j is matched on an edge with net payment at least $v_{ij}/2$. So they are both unmatched by the time the price has dropped to $p(t) = v_{ij}$. The behavior of the mechanism and all participants is identical under b and (b_{-i}, b'_i) until i matches, because the two profiles cannot be distinguished. So if i is still unmatched under (b_{-i}, b'_i) when $p(t) = v_{ij} = b'_{ij}(t)$, then i matches to j at this price. We conclude that i matches at some price $p(t) \geq v_{ij}$. Therefore, $u_i(b_{-i}, b'_i) \geq p(t)/4 \geq v_{ij}/4$. □

Theorem 3.3.1. *In the nonnegative values setting, the* 1/4*-rebate Marshallian Match has a Bayes-Nash price of anarchy of at least* 1/8*.*

Proof. Letting b be any Bayes-Nash equilibrium and M^* be the optimal matching (a random variable), we have the following. We will use that, in equilibrium, i prefers b_i to b'_i; that M^* contains at most every participant, and utilities and payments are nonnegative; and Lemma 3.3.1.

$$\begin{aligned}
\text{Welf}(b) &= \mathbb{E}\sum_i \left(u_i(b) + \frac{p_i(b)}{2}\right) \\
&\geq \mathbb{E}\sum_i \left(u_i(b_{-i}, b'_i) + \frac{p_i(b)}{2}\right) \\
&\geq \mathbb{E}\left[\sum_{\{i,j\}\in M^*} \left(u_i(b_{-i}, b'_i) + u_j(b_{-j}, b'_j) + \frac{p_i(b)}{2} + \frac{p_j(b)}{2}\right)\right] \\
&= \mathbb{E}\left[\sum_{\{i,j\}\in M^*} \left(u_i(b_{-i}, b'_i) + \frac{p_i(b)}{4} + \frac{p_j(b)}{4} + u_j(b_{-j}, b'_j) + \frac{p_i(b)}{4} + \frac{p_j(b)}{4}\right)\right] \\
&\geq \mathbb{E}\left[\sum_{\{i,j\}\in M^*} \left(\frac{v_{ij}}{8} + \frac{v_{ji}}{8}\right)\right] = \frac{1}{8}\text{Welf}(\text{OPT}).
\end{aligned}$$

□

3.4 Matching with Costly Information

We now extend our welfare result for graphs to a model with Pandora's boxes representing information acquisition costs, following e.g. Kleinberg et al. [2016]. For each edge $\{i, j\}$, there is a

Pandora box for endpoint i and endpoint j, which we treat as agents. The type of i consists of, for each feasible partner j, a cost $c_{ij} \geq 0$ to open the box on endpoint i of edge $\{i, j\}$ and a distribution D_{ij} over the nonnegative reals. Our setting is Bayes-Nash, i.e. all types are drawn jointly from a common-knowledge prior. We refer equivalently to opening a box and "inspecting" an edge on which a box sits. When i inspects an edge $\{i, j\}$, they incur a cost of c_{ij} and observe a value $v_{ij} \sim D_{ij}$ independently of all other randomness in the game. The cost c_{ij} can model a financial investment, or the cost of time or effort required for i to learn their value v_{ij}. Recalling Section 2.3.2, this induces a capped value and Weitzman index for each box, κ_{ij} and σ_{ij}, respectively.

We augment the opening and claiming notation from Section 2.3.2 for edges, letting $\mathbb{I}_{ij} \in \{0, 1\}$ be the random variable indicator that i inspects j and let $\mathbb{A}_{ij} \in \{0, 1\}$ be the indicator that i is matched to j. We adopt the standard assumption (although recent algorithmic work of Beyhaghi and Kleinberg [2019] has weakened it) that i must inspect j prior to being matched to j; i.e. if the match occurs, i must inspect and incur cost c_{ij} if they haven't yet. In other words, $\mathbb{A}_{ij} \leq \mathbb{I}_{ij}$ pointwise. We also assume that, in the game, inspection is instantaneous with respect to the movement of the price $p(t)$.

An agent i's utility is their value for their match (if any) minus the sum of all inspection costs and their net payment. Formally, we have $u_i(b) = \sum_j (\mathbb{A}_{ij} v_{ij} - \mathbb{I}_{ij} c_{ij}) - \pi_i(b)$, where the sum is over feasible neighbors. Social welfare is the sum of all agent utilities and revenue, i.e. $\text{Welf}(b) = \sum_i (u_i(b) + \pi_i(b))$.

A *matching process* is any procedure that involves sequentially inspecting some of the potential matches and making matches, according to any algorithm or mechanism. A matching procedure is non-exposed on an ordered pair (i, j) according to Definition 2.2.1 if the box associated with end i is non-exposed. We can immediately extend Lemma 2.2.1, which applies to single-selection settings, to matching processes, where for mechanisms the matching algorithm is a function of agents' chosen strategies.

Lemma 3.4.1 (Immediate extension of Kleinberg et al. [2016])**.** *For any feasible partners* $\{i, j\}$*,*

any fixed types of all agents, and any matching process,

$$\mathbb{E}\left[\mathbb{A}_{ij}v_{ij} - \mathbb{I}_{ij}c_{ij}\right] \leq \mathbb{E}\left[\mathbb{A}_{ij}\kappa_{ij}\right],$$

with equality if and only if the matching process is non-exposed on ordered pair (i, j)*.*

We say agent i's strategy is a non-exposed strategy if, for every other strategy of every other agent, the induced matching algorithm is not exposed on any pair (i, j). It is not immediately clear that i alone can enforce her own non-exposed strategy, but we will construct such a deviation strategy.

Proof. Using the definitions, independence of $v_{ij} \sim D_{ij}$ from the variable $\mathbb{I}_{ij}$, and the assumption $\mathbb{A}_{ij} \leq \mathbb{I}_{ij}$,

$$\begin{aligned}
\mathbb{E}\left[\mathbb{A}_{ij}v_{ij} - \mathbb{I}_{ij}c_{ij}\right] &= \mathbb{E}\left[\mathbb{A}_{ij}v_{ij} - \mathbb{I}_{ij}\mathop{\mathbb{E}}_{v'_{ij}\sim D_{ij}}(v'_{ij} - \sigma_{ij})^+\right] \\
&= \mathbb{E}\left[\mathbb{A}_{ij}v_{ij} - \mathbb{I}_{ij}(v_{ij} - \sigma_{ij})^+\right] \\
&\leq \mathbb{E}\left[\mathbb{A}_{ij}v_{ij} - \mathbb{A}_{ij}(v_{ij} - \sigma_{ij})^+\right] \\
&= \mathbb{E}\left[\mathbb{A}_{ij}\min\{\sigma_{ij}, v_{ij}\}\right].
\end{aligned}$$

We observe that the inequality is strict if and only if there is positive probability of the event that $\mathbb{I}_{ij} = 1$, $\mathbb{A}_{ij} = 0$, and $v_{ij} > \sigma_{ij}$ all occur, i.e. the algorithm is exposed on ordered pair (i, j). □

Informally, Lemma 3.4.1 states that the utility of any strategy is upper-bounded by the capped value of the same strategy, with equality if and only if the strategy is non-exposed. This lemma provides two useful insights. Firstly, the utility of any inspection strategy can be upper-bounded by the utility of the same strategy in a modified setting with no inspections, which we will use to upper-bound OPT. Secondly, we can analyze the utility of any agent playing a strategy which is non-exposed by instead analyzing their corresponding inspection-free utility in capped values.

3.4.1 Smoothness for costly inspection

To analyze the mechanism using the smoothness framework, we again propose and analyze the utility of a deviation strategy. To make this strategy easier to analyze, we construct a non-exposed strategy and use 3.4.1 to instead consider the strategy's utility in the associated Pandora-free setting of κ_{ij}s.

The deviation strategy. Define the deviation strategy b'_i for agent i as follows: Initially bid 0 on each neighbor j; when the clock reaches σ_{ij}, inspect neighbor j and update bid to $\kappa_{ij} = \min\{\sigma_{ij}, v_{ij}\}$. For a strategy profile b, define the random variable $\kappa_i(b)$ to be the capped value of i for its match in profile b, i.e. $\min\{\sigma_{ij}, v_{ij}\}$ when i is matched to j. Let $\bar{u}_i(b)$ denote i's "covered call utility", i.e.

$$\bar{u}_i(b) = \kappa_i(b) - \pi_i(b).$$

Lemma 3.4.2. *The deviation strategy b'_i is non-exposed and ensures $\bar{u}_i(b_{-i}, b'_i) \geq 0$, for any b_{-i}.*

Proof. (Non-exposure.) We consider the two possible scenarios where i inspects a neighbor j, i.e. when $\mathbb{I}_{ij} = 1$. If the inspection occurs because the mechanism has just matched i to a previously-uninspected neighbor j, then $\mathbb{A}_{ij} = 1$ and the requirement of non-exposure is satisfied. Otherwise, the clock $p(t)$ has reached σ_{ij}, and after inspecting, i updates that bid to κ_{ij}. If $v_{ij} \geq \sigma_{ij}$, then $b'_{ij} \geq p(t)$ and the match occurs immediately, so $\mathbb{A}_{ij} = 1$.

(Nonnegative capped value.) Recall that the rebate is nonnegative, so i's net payment π_i is always at most i's bid. If i is matched to some previously-uninspected j, then $b_{ij} = 0$, so $\pi_i(b_{-i}, b'_i) \leq 0$, and $\bar{u}_i(b_{-i}, b'_i) \geq 0$. If i is matched to some j that i has already inspected, then $\pi_i(b_{-i}, b'_i) \leq b'_{ij} = \kappa_{ij}$, so $\bar{u}_i(b_{-i}, b'_i) \geq \kappa_{ij} - \kappa_{ij} \geq 0$. Finally, if i is unmatched, then $\bar{u}_i(b_{-i}, b'_i) = 0$. □

Lemma 3.4.3 (Capped value smoothness)**.** *For any feasible neighbors $\{i, j\}$ and any strategy profile b, for all realizations of types and values:*

$$\bar{u}_i(b_{-i}, b'_i) + \frac{p_i(b)}{4} + \frac{p_j(b)}{4} \geq \frac{\kappa_{ij}}{8}.$$

Proof. Fix a realization of all types and values. The quantities $p_i(b)$, $p_j(b)$, and $\bar{u}_i(b_{-i}, b'_i)$ are all nonnegative. If $p_i(b) \geq \kappa_{ij}/2$ or $p_j(b) \geq \kappa_{ij}/2$, then we are already done. So suppose neither holds. Then in profile b, both i and j are not yet matched at price $p(t) = \kappa_{ij} \geq \kappa_{ij}/2$. Observe that in (b_{-i}, b'_i), all other agents' bids and behavior are unchanged until i is matched, because they continue playing their strategies in b and no information available to them indicates the change in i's strategy.

Therefore, in profile (b_{-i}, b'_i), i matches at some $p(t) \geq \kappa_{ij}$, because if the price reaches κ_{ij} without i being matched, then i will match to j. Let j' be the partner i is matched to. Then by construction, i is bidding $b'_{ij'}(t) \leq \kappa_{ij'}$, so $\bar{u}_i(b_{-i}, b'_i) \geq \kappa_{ij'} - \kappa_{ij'} + \frac{p(t)}{4} \geq \frac{p(t)}{4} \geq \frac{\kappa_{ij}}{4}$. □

Theorem 3.4.1. *In the inspection setting with nonnegative values, the* 1/4*-rebate Marshallian Match guarantees a Bayes-Nash price of anarchy of at least* 1/8*.*

Proof. Let the random variable M^* be the max-weight matching where the weight on edge $\{i, j\}$ is $\kappa_{ij} + \kappa_{ji}$. Lemma 3.4.1 also implies that Welf(OPT) is at most the total weight of M^*, as follows. Here $\mathbb{I}_{ij}, \mathbb{A}_{ij}$ are the indicators under the OPT procedure, and notice $\mathbb{A}_{ij} = \mathbb{A}_{ji}$ in any matching.

$$\begin{aligned}
\text{Welf(OPT)} &= \mathbb{E} \sum_{\{i,j\} \in E} [(\mathbb{A}_{ij} v_{ij} - \mathbb{I}_{ij} c_{ij}) + (\mathbb{A}_{ji} v_{ji} - \mathbb{I}_{ji} c_{ji})] \\
&\leq \mathbb{E} \sum_{\{i,j\}} (\mathbb{A}_{ij} \kappa_{ij} + \mathbb{A}_{ji} \kappa_{ji}) && \text{Lemma 3.4.1} \\
&\leq \mathbb{E} \sum_{\{i,j\} \in M^*} (\kappa_{ij} + \kappa_{ji}) && \text{the solution must form a matching.}
\end{aligned}$$

(We note that we have no idea what OPT actually is in this setting, or if it can even be computed in polynomial time; nevertheless, this upper bound cannot be too loose, since the MM approximates it.) Lemma 3.4.2 states that the deviation strategy b'_i is non-exposed, so Lemma 3.4.1 implies

$\mathbb{E}\, u_i(b_{-i}, b'_i) = \mathbb{E}\, \bar{u}_i(b_{-i}, b'_i)$. Therefore,

$$\begin{aligned}
\text{Welf}(b) &= \mathbb{E} \sum_i \left[u_i(b) + \frac{p_i(b)}{2} \right] \\
&\geq \mathbb{E} \sum_i \left[u_i(b_{-i}, b'_i) + \frac{p_i(b)}{2} \right] \\
&= \mathbb{E} \sum_i \left[\bar{u}_i(b_{-i}, b'_i) + \frac{p_i(b)}{2} \right] \\
&\geq \mathbb{E} \sum_{\{i,j\} \in M^*} \left(\left[\bar{u}_i(b_{-i}, b'_i) + \frac{p_i(b)}{4} + \frac{p_j(b)}{4} \right] + \left[\bar{u}_j(b_{-j}, b'_j) + \frac{p_i(b)}{4} + \frac{p_j(b)}{4} \right] \right) \\
&\geq \mathbb{E} \sum_{\{i,j\} \in M^*} \left[\frac{\kappa_{ij}}{8} + \frac{\kappa_{ji}}{8} \right] \quad \geq \frac{1}{8} \text{Welf}(\text{OPT}).
\end{aligned}$$

□

3.5 Group Matching Extensions

We now extend to the problem of coordinating formation of groups of size up to k, for now ignoring costly inspection. the approach is similar to the matching case above, i.e. the special case where $k = 2$ and groups of size one are disallowed. Additionally, Appendix D proves an approximation guarantee matching the results of this section, but in the costly inspection setting.

Instead of a graph, we are given a hypergraph where agents are vertices and a hyperedge S represents a feasible group, i.e. subset of agents of size at most k. The value of agent i for being assigned to group S is $v_{iS} \geq 0$. A "matching" or assignment M consists of a subset of the hyperedges such that no agent is in two different groups $S, S' \in M$. The surplus of a group S is $s_S = \sum_{i \in S} v_{iS}$. The social welfare of an assignment M is $\sum_{S \in M} s_S$.

The 1/4-rebate Marshallian Match for this setting is modified as follows to a "$\frac{1}{2k}$-rebate MM". Participant i maintains bids b_{iS} for all of their feasible groups S. When the descending price matches the total sum of bids on any group, i.e. $p(t) = \sum_{i \in S} b_{iS}$, that group is matched and drops out of the mechanism. All members pay their bids, and each receives a rebate of $\frac{p(t)}{2|S|}$, with the mechanism keeping $\frac{p(t)}{2}$.

This modification of the Marshallian Match has a Bayes-Nash price of anarchy at least

$1/(2k^2)$.

Theorem 3.5.1. *In the hyperedge matching (group formation) setting with nonnegative values and group size up to k, the Marshallian Match has a Bayes-Nash price of anarchy of at least $\frac{1}{2k^2}$.*

Again consider the truthful deviation b'_i where $b'_{iS}(t) = v_{iS}$ for all feasible S and all t. When i matches in group S at price $p(t)$ in deviation,

$$\begin{aligned} u_i(b_{-i}, b'_i) &= v_{iS} - b'_{iS}(t) + \frac{p(t)}{2|S|} \\ &\geq \frac{p(t)}{2k}. \end{aligned}$$

Lemma 3.5.1. *For any strategy profile b, realizations of types, agent i, and hyperedge S containing i,*

$$u_i(b_{-i}, b'_i) + \frac{1}{k^2} \sum_{j \in S} p_j(b) \geq \frac{v_{iS}}{2k^2}.$$

Proof. All terms are nonnegative. If under b there exists $j \in S$ with $p_j(b) \geq \frac{v_{iS}}{2}$, then we are done. Otherwise, in (b_{-i}, b'_i) all members of S are unmatched at least until i matches, which at the latest occurs at $p(t) = v_{iS}$, yielding i utility at least equal to its rebate of $\frac{p(t)}{2|S|} \geq \frac{v_{iS}}{2k} \geq \frac{v_{iS}}{2k^2}$. □

Proof of Theorem 3.5.1. Letting b be a Bayes-Nash equilibrium and M^* the optimal assignment (a random variable), we have the following. The first line follows because groups have size at most k, so the total payment of a group S is at least $\sum_{i \in S} p_i(b)/k$.

$$\begin{aligned} \text{Welf}(b) &\geq \mathbb{E} \sum_i \left(u_i(b) + \frac{p_i(b)}{k} \right) \\ &\geq \mathbb{E} \sum_i \left(u_i(b_{-i}, b'_i) + \frac{p_i(b)}{k} \right) \\ &\geq \mathbb{E} \sum_{S^* \in M^*} \sum_{i \in S^*} \left(u_i(b_{-i}, b'_i) + \frac{p_i(b)}{k} \right) \\ &\geq \mathbb{E} \sum_{S^* \in M^*} \sum_{i \in S^*} \left(u_i(b_{-i}, b'_i) + \sum_{j \in S^*} \frac{p_j(b)}{k^2} \right) \\ &\geq \mathbb{E} \sum_{S^* \in M^*} \sum_{i \in S^*} \frac{v_{iS^*}}{2k^2} \quad = \frac{\text{Welf(OPT)}}{2k^2}. \end{aligned}$$

□

It remains to be seen if the factor can be improved to $\Omega(1/k)$. We appear to lose one factor of k because the greedy algorithm (e.g. the MM where all participants are truthful) is only a $1/k$ approximation to optimal, and then another factor from strategic behavior.

3.6 Failure of the Marshallian Match Under Negative Valuations

It is natural to consider negative values and bids in a matching market. Negative values model *costs* incurred for a match, such as in a job market. When a worker is matched as an employee to a company, she experiences some cost for which she must be compensated. The goal of the matching market is to find an efficient price for her labor. However, negative costs complicate matters because they also introduce negative bids. A participant can make it difficult for a match to occur by bidding an arbitrary negative amount. The result is that equilibrium is no longer sufficient for good welfare, even in bipartite graphs.

In this section, we first formalize the failure of equilibrium. We then turn to *stability* as a possible saviour. We observe that approximate *ex post* stability indeed implies good welfare in *any mechanism*, and that the MM satisfies approximate ex post stability when participants are truthful.

However, ex post stability is a very strong requirement. We formulate an alternative, *ex ante* stability, and show that the MM has approximately optimal welfare in any ex-ante stable *Nash* equilibrium. Finally, however, we observe that this result does not naturally extend to Bayes-Nash equilibrium and illustrate the apparent difficulty involving coordinated communication.

We make one restriction on the general model of Section 3.2: we assume the graph is bipartite. This is primarily for notational and narrative convenience. To make our presentation more intuitive, we adopt terminology in which the two sides of the bipartite market are asymmetric: One side (e.g. employers) are *bidders*, while the other side (e.g. workers) are *askers*. The bidders are indexed by i. They have values v_{ij} and make bids b_{ij}.

The askers are indexed by j. We assume they have *costs* c_{ij} and make *asks* a_{ij}. The costs

and asks are simply the negative of their values and bids under the previous section's model. Now, for example the surplus of a match between i and j is $s_{ij} = v_{ij} - c_{ij}$. We still use π_j to denote the net payment made by an asker. So the utility of an asker j for being matched to i is $u_j = -c_{ij} - \pi_i$, and so on.

We generally picture values, bids, costs, and asks all as positive numbers, so a bidder has a positive value for matching to an asker, who incurs a positive cost from the match. However, our results are all fully general and would allow for any value, bid, cost, or ask to be either positive or negative.

Failure of equilibrium. When asks are allowed, equilibrium becomes insufficient to provide welfare guarantees. Participants in the mechanism can place bids and asks such as to effectively refuse matches with one another, by asking above value or bidding below cost. If two players both "refuse" matches with one another, neither can unilaterally fix the situation. We prove this result for the MM with 1/4 rebate, but the same profile is also a zero welfare equilibrium with no rebate.

Proposition 3.6.1. *In the bipartite setting with general values and costs, there always exists a Nash equilibrium of the Marshallian Match with zero welfare.*

Proof. Let $x = \max_{i,j} \max\{|v_{ij}|, |c_{ij}|\}$. Consider the strategy profile where every bidder i bids $b_{ij} = -2x$ on all neighbors j, and every asker j asks $a_{ij} = 2x$ on all neighbors i. This is an equilibrium in which no matches occur. For a unilateral deviation to cause a match to occur, e.g. a bidder would have to change a bid to at least $b_{ij} = 2x$, resulting in a net payment of at least $2x$ after the rebate, giving negative utility. The asker's case is analogous. □

In this example, any single player cannot deviate alone to improve her welfare. However, any pair of bidders sharing an edge can coordinate a deviation together and guarantee themselves higher welfare. This suggests that the bad equilibrium profile lacks *stability*, a key concept in matching algorithm design.

3.7 alternative notions of equilibrium

This simple failure motivates our exploration of alternative definitions of "equilibrium" in matching, incorporating notions of stability from traditional matching literature. In classical "stable matching" problems Gale and Shapley [1962], the goal is that, once the mechanism produces a final matching, no two participants i, j both prefer to leave their assigned partners and switch to matching each other instead.

3.7.1 ex post stability

We first introduce *ex post* stability, a notion that requires any final matching produced by a mechanism has stability properties. We use ex post to refer to the fact that this evaluation occurs after all randomness and the matching's outcome have been realized. In our setting, if i and j chose to match each other, they would obtain a net utility of their surplus $s_{ij} = v_{ij} - c_{ij}$. If this amount is larger than their total utility in the mechanism, then they could switch to each other and split the surplus so as to make them both better off. On the other hand, making this switch presumably involves some amount of friction. Therefore, we introduce an approximate version of stability, as a more lenient requirement of a mechanism.

Definition 3.7.1 (Approximate ex post stability)**.** A strategy profile (b, a) in a mechanism is k-ex post stable if, for all realizations of costs and values and for all feasible pairs of bidder i and asker j,

$$u_i(b, a) + u_j(b, a) \geq \frac{1}{k} s_{ij}.$$

We note that ex post stability is an extremely strong notion. For any matching mechanism, an ex post stable strategy profile produces an approximately optimal matching.

Observation 3.7.1. *For any matching mechanism, any k-ex post stable strategy (if one exists) is a $\frac{1}{k}$-approximation of the first-best welfare.*

Proof. Recall that in the Bayes-Nash setting agents' types are drawn from from a joint distribution $\mathcal{D}$. Let (b, a) be a k-ex post stable strategy profile for some matching mechanism, and M the

associated matching. The expected welfare over all realizations of types and strategies is at least the total utility of participants,

$$\text{Welf}(b,a) \geq \mathbb{E}\left[\sum_{\{i,j\}\in M} u_i(b,a) + u_j(b,a)\right] = \mathbb{E}\left[\sum_{\{i,j\}\in M^*} u_i(b,a) + u_j(b,a)\right],$$

where M^* is the maximal matching by surplus. Ex post stability gives a lower-bound on utility for all pairs, regardless of realization, so

$$\text{Welf}(b,a) \geq \mathbb{E}\left[\sum_{\{i,j\}\in M^*} \frac{s_{ij}}{k}\right] = \frac{1}{k}\text{Welf}(\text{OPT})$$

□

Observation 3.7.1 also arises directly from a primal-dual analysis of the linear program for bipartite matching, in which the dual variables are the utilities of the agents and k-approximate satisfaction of the dual constraints is precisely k-ex post stability. We note that while there exist deferred-acceptance style "stable" matching mechanisms that technically involve money, such as matching with contracts Hatfield and Milgrom [2005], we have not ascertained if they can be made to satisfy approximate ex post stability in our sense.

Ex post stability of the Marshallian Match under truthfulness. Ideally, a matching mechanism would admit equilibria in ex post stable strategies, so that welfare would be high and participants would adhere to the outcomes of the mechanism. But as a weaker requirement, we would like an indication of whether approximately ex post stable strategies might be reasonable in a mechanism. We present a simple 4-ex post stable strategy profile for the 1/4-rebate MM, for all realizations of costs and values.

Proposition 3.7.1. *The truthful strategy in which all participants bid or ask their true value or cost is* 4*-ex post stable in the* 1/4*-rebate Marshallian Match.*

Proof. When all participants truthfully report their values and costs, Marshallian Match produces the greedy maximum weighted matching M, where edge weights are supluses $s_{ij} = v_{ij} - c_{ij}$ (and negative edges are discarded). Let (b^*, a^*) denote the strategy profile in which all agents are truthful.

Consider any pair $\{i, j\}$. We wish to show that $u_i(b^*, a^*) + u_j(b^*, a^*) \geq \frac{s_{ij}}{4}$. If $\{i, j\} \in M$, then i and j each obtain utility exactly $s_{ij}/4$, as their value (cost) is canceled by their bid (ask) and they are left with the rebate. If $\{i, j\} \notin M$, then either i or j must have already matched before the descending price reached s_{ij}. Without loss of generality, say i matched to k before s_{ij}, with $s_{ik} > s_{ij}$. By truthfulness, i obtains welfare $u_i(b^*, a^*) = s_{ik}/4 > s_{ij}/4$. □

Again, once one connects the truthful MM to greedy matching, Proposition 3.7.1 follows from a basic primal-dual analysis combined with the rebate payment rule.

While truthful reporting from all participants produces approximately optimal social welfare (and the participants capture half of it), truthfulness is not in general even an approximate equilibrium. We provide an example in the 1/4-rebate MM setting in which a player can increase her welfare arbitrarily by overstating her true value for a match. If players can make significant gains by deviating from the truthful strategy, it is likely that they will not adhere to an ex post stable profile.

Example 3.7.1 (Figure 3.1)**.** *Consider the bipartite graph with three participants: A and B place bids on matching to C. C has cost $c_{CA} = c_{CB} = 0$ for both matches, A has value $v_{AC} = k + 1$ and B has value $v_{BC} = k$ for matching with C. In the truthful profile, A is matched to C, and B goes unmatched with welfare* 0*. If B deviates to the non-truthful strategy of bidding $b'_{BC} = k+2$, B would be matched with C and would receive utility $u_B(b^*_{-B}, a^*, b') = k - (k + 2) + (k + 2)/4 = k/4 - 3/2$. Thus, picking k appropriately, B can benefit arbitrarily by deviating from the truthful strategy.*

Drawbacks of ex post stability. If one can prove that a mechanism is ex post stable in equilibrium, that is an ideal result as a welfare guarantee immediately follows. However, without such a result, the value of ex post stability is questionable.

Consider the following somewhat subtle point. Intuitively, it may seem reasonable to view stability as a sort of equilibrium refinement. In particular, if strategy profile b is not approximately stable, then (one would think) there are two participants i and j who would prefer to jointly switch their strategies so as to match to each other. However, *ex post* stability does not give this kind

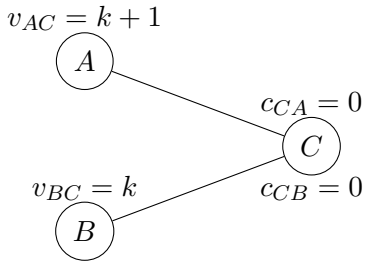


Figure 3.1: Arbitrary gains from deviating from the truthful strategy

In the 1/4-rebate MM, B is incentivized to deviate from truthfulness to the non-truthful bid $k+2$.

of guarantee. It can only tell i and j whether they are satisfied after the mechanism happens. To capture the above intuition, we turn to *ex ante* stability.

3.7.2 ex ante stability

We now consider a model of stability that generalizes equilibrium by supposing no *pair* of participants has an incentive to *bilaterally* deviate. Importantly, the incentive is relative to expected utility, so it involves an *ex ante* calculation by the participants rather than ex post.

Definition 3.7.2 (ex ante stability)**.** A strategy profile (b, a) is k-ex ante stable if, for all feasible pairs of bidder i and asker j, for all strategies b'_i of i and a'_j of j,

$$\mathbb{E}[u_i(b,a) + u_j(b,a)] \geq \frac{1}{k}\,\mathbb{E}[u_i(b_{-i}, b'_i, a_{-j}, a'_j) + u_j(b_{-i}, b'_i, a_{-j}, a'_j)],$$

with randomness taken over realizations of types and strategies.

That is, a profile is k-ex ante stable if there exists no deviation for any pair of players by which they could expect to increase their collective welfare by a factor of more than k. There are two primary differences between ex ante and ex post stability. First, ex ante applies to preferences "before the fact", i.e. in expectation, while ex post applies to preferences "after the fact". Second, ex post stability postulates the ability of two participants to completely bypass the rules of the mechanism and match to each other. In ex ante stability, the participants are limited to deviating to strategies actually allowed by the mechanism.

We observe that an ex ante stable profile is by definition a Bayes-Nash equilibrium, as one can in particular consider profiles where only one of the two participants deviates. We show that for deterministic values and costs, ex ante stability is in fact sufficient to guarantee an approximation of optimal welfare.

Smoothness and deviation strategies. We define a pairwise deviation of $\{i, j\}$ to truthfulness for bids and asks as follows: $b'_{i\ell} = v_{i\ell}$ for all feasible neighbors ℓ, and similarly $a'_{j\ell} = c_{j\ell}$ for all neighbors ℓ. For any pairwise deviation to truthfulness, we show that that pair collectively achieves a constant fraction of their welfare in the 1/4-rebate MM.

Lemma 3.7.1. *For any pair of feasible neighbors i and j, for any strategy profile (b, a),*

$$u_i(b_{-i}, b'_i, a_{-j}, a'_j) + u_j(b_{-i}, b'_i, a_{-j}, a'_j) \geq \frac{s_{ij}}{4}.$$

Proof. If i and j are still unmatched in deviation when the price reaches s_{ij}, then they will be matched and each get utility $s_{ij}/4$, since their bid-ask spread is s_{ij}.

If i matches to some $\ell \neq j$ before s_{ij}, we claim i still achieves high utility. Since i is truthful, she obtains utility $u_i(b_{-i}, b'_i, a_{-j}, a'_j) = v_{i\ell} - b'_{i\ell} + (b'_{i\ell} - a_{\ell i})/4 = (b'_{i\ell} - a_{\ell i})/4 > s_{ij}/4$, since in deviation the bid-ask spread on the edge (i, ℓ) is greater than that on edge (i, j).

If j matches to some $\ell \neq i$ before s_{ij}, then j achieves high utility. Since j is truthful, she obtains utility $u_j(b_{-i}, b'_i, a_{-j}, a'_j) = a'_{\ell j} - c_{\ell j} + (b_{\ell j} - a'_{j\ell})/4 = (b_{\ell j} - a'_{j\ell})/4 > s_{ij}/4$, again since (ℓ, j) has a higher bid-ask spread in deviation than (i, j). □

Theorem 3.7.1. *For deterministic values and costs, and strategy profile (b, a) that is k-ex ante stable, the 1/4-rebate Marshallian Match achieves a $\frac{1}{4k}$-approximation of the optimal expected welfare.*

Surprisingly, in this result, the contribution of payments to overall welfare can be ignored. We will actually obtain that total participant surplus is a constant fraction of the optimal welfare.

Proof of Theorem 3.7.1. Let M denote the matching produced by the strategy profile (b, a), and

M^* denote the first-best welfare matching. The welfare of M is

$$\begin{aligned}\textsc{Welf}(b,a) &= \mathbb{E}\left[\sum_{\{i,j\}\in M} u_i(b,a) + u_j(b,a) + p_i(b,a)\right] \\ &\geq \mathbb{E}\left[\sum_{\{i,j\}\in M^*} u_i(b,a) + u_j(b,a)\right] \\ &\geq \mathbb{E}\left[\frac{1}{k}\sum_{\{i,j\}\in M^*} u_i(b_{-i},b'_i,a_{-i},a'_j) + u_j(b_{-i},b'_i,a_{-i},a'_j)\right]\end{aligned}$$

since the values and costs of agents are deterministic and common knowledge, each agent can compute her partner in the first-best welfare matching, and can coordinate to deviate as a pair, guaranteeing a lower bound on welfare for both. Applying Lemma 3.7.1,

$$\textsc{Welf}(b,a) \geq \frac{1}{k}\,\mathbb{E}\left[\sum_{\{i,j\}\in M^*} \frac{s_{ij}}{4}\right] = \frac{1}{4k}\textsc{Welf}(\textsc{Opt})$$

□

This result relies heavily on participants' ability to calculate the fixed matching with optimal social welfare, and coordinate deviations with their partner. We exploit the deterministic costs and values considered to fix the optimal matching M^* across realizations of potentially mixed strategies.

3.8 Failure of Ex Ante Approach to Bayes-Nash Equilibria

We would like to extend the positive results of our ex ante Nash equilibria to the Bayes-Nash setting. However, a subtle issue arises with a direct extension. In the Bayes-Nash setting, the distributions over values and costs are common knowledge, but their realizations are not. As a result, the optimal matching M^* is dependent on the realizations of players' types. This causes our proof of welfare approximation under ex ante stability to fail in an interesting way. In fact, even the definition of ex ante stability (Definition 3.7.2) has nuanced implications.

Consider a complete bipartite $n \times n$ graph with a "star-crossed lovers" distribution on types[5] : there is a uniformly random perfect matching M^* in which the edges have positive surplus, while

[5] One can create a version where type distributions are independent that makes roughly the same point.

all edges not in M^* have high costs and low values. Even if we take a very bad mechanism, such as one that always assigns the same matching M regardless of types, it can be ex ante stable: any particular pair are getting low utility, but if they are able to jointly deviate to matching with each other deterministically, they also get low utility in expectation over the type distribution. They are unlikely to be fated for each other once the types are realized.

Similarly, recall that the proof of Theorem 3.7.1 used ex ante stability in the following step:

$$\mathbb{E}\left[\sum_{(i,j)\in M^*} u_i(b,a) + u_j(b,a)\right] \geq \mathbb{E}\left[\frac{1}{k}\sum_{(i,j)\in M^*} u_i(b_{-i}, b'_i, a_{-i}, a'_j) + u_j(b_{-i}, b'_i, a_{-i}, a'_j)\right].$$

In the Nash setting, M^* was fixed, and so were the deviation strategies b'_i, a'_j which depended on M^*. So the expectation could move inside the sum, followed by an application of ex ante stability. But in a Bayes-Nash setting, M^* is a random variable depending on the realizations of the types. We cannot move the expectation inside the sum.

A tempting fix is some sort of **ex interim stability** assumption, where the deviation strategies of i and j can depend on the types of both players. In the star-crossed lovers example, this is appealing: the pairs with high surplus know this fact from their types and can easily coordinate a deviation. So it is reasonable to assume that strategy profiles played in the mechanism are not much worse than such coordinated deviations.

However, the amount of coordination required grows significantly if type distributions are more complicated. For example, suppose each value and cost comes from an independent power-law distribution. Finding blocking joint deviations seems to require significant knowledge by the agents, perhaps of global properties of the type space (such as who would be matched under the greedy matching or optimal matching). To assume agents play strategies that eliminate such deviations appears to unjustly relieve the mechanism of responsibility to coordinate agents' information and decisionmaking.

Chapter 4

Algorithmic Approaches to Negative Valuations

Chapter 3 focuses on the design of mechanisms for matching agents together, but Section 3.8 leaves open the problem of devising a mechanism for matching in the presence of negative valuations.

This chapter takes a step back from the strategic setting, and asks the question: does the difficulty in matching with negative valuations come from the strategic aspects of the problem, or from the underlying algorithmic problem? We address this by turning to the purely algorithmic problem with a central planner. We consider a number of possible algorithms in Section 4.1, each with mechanism design-motivated interpretations, and provide impossibility results for each class in the negative results setting.

In Section 4.3, we then leverage the bandit tools introduced in Section 2.3 to design a simple approximation algorithm for negative valuations, but without a corresponding interpretation in the mechanism design setting.

Throughout this chapter, we will continue to use the term "welfare" to refer to the total value minus cost achieved by the algorithm, despite the absence of agent-based terminology.

Triviality of the Free-Information Setting. When we remove costly information from the algorithmic setting, a central planner sees all values on all endpoints. She can then simply sum values along edges to obtain a classic edge-weighted matching instance, and run a matching algorithm of her choice to find the optimal matching. Thus, this chapter focuses exclusively on matching algorithms with costly inspection, with boxes on both endpoints of each edge. We find quickly that this problem is much more difficult.

4.1 Initial Algorithms

We first consider two natural classes of algorithms with intuitive interpretations, and demonstrate that neither of them can achieve welfare guarantees when values may be negative.

4.1.1 Bundled-Box Algorithms

Singla [2018] provides a general welfare approximation guarantee for the one-sided matching setting, in which each edge has only one Pandora box on it. Can our two-box setting be reduced to this setting, by simply forcing boxes on an edge to always be opened simultaneously? This approach parallels the simple reduction to edge-weighted matching in the free-information setting, and provides some enticing properties for a mechanism-based interpretation. In many real-world settings such as interviewing, where a company and employee must both invest simultaneously to reveal their values, it is natural to require the algorithm to "bundle" the two boxes incident to an edge. Formally, a *bundled-box algorithm* is one that, if it opens one box on an edge, always opens the other next. In particular, it satisfies $\mathbb{I}_{ij} = \mathbb{I}_{ji} \quad (\forall \{i,j\} \in E)$.

A bundled-box algorithm functionally reduces each edge to a single Pandora box, with value the sum of individual values and cost the sum of costs. We model the bundled box on edge $\{i,j\}$ as having a combined value $v'_{\{i,j\}} = v_{ij} + v_{ji}$, and we define the new value distribution $v'_{\{i,j\}} \sim D'_{\{i,j\}}$, the convolution of D_{ij} and D_{ji}.

If we let OPT' denote the optimal algorithm for this setting, then Kleinberg et al. [2016] and Singla [2018] both give 1/2-approximations to OPT', extending Weitzman's descending procedure: inspect the edges from largest index downward, matching any edge once it has been inspected and its value is larger than all remaining indices, and removing edges from consideration once they are not legal to add to the matching.

The power of bundled boxes. The key question is then whether OPT' can approximate OPT, i.e. whether bundling comes at a high cost. On one hand, OPT clearly has significant additional power over simultaneous inspection in choosing inspections. On the other, edges are

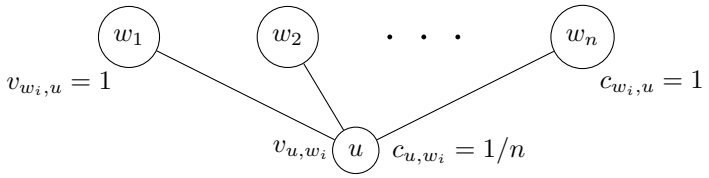


Figure 4.1: Bundled Star Graph

The "bundled star graph" with vertices $w_1, \ldots, w_n$ surrounding u. Each leaf w_i has a constant value and cost of 1. The cost for u to inspect any of its values is $c_{u,w_i} = 1/n$, and the value is either $v_{u,w_i} = n$ with probability $1/n$ or 0 otherwise. A bundled-box algorithm performs arbitrarily worse on this example than an algorithm which can inspect endpoints independently.

ultimately bundled when claimed by a matching. Here, we show that bundling the edges can perform arbitrarily poorly relative to algorithms that inspect asynchronously.

Proposition 4.1.1. *No bundled-box algorithm can guarantee better than an $O(\frac{1}{n})$-approximation factor for Pandora's Matching Problem on n vertices.*

The proof of Proposition 4.1.1 uses the bundled star graph given in Figure 4.1. The key observation is that a bundled-box algorithm is forced to pay to open boxes with unhelpful information, which can be avoided by an algorithm inspecting endpoints independently. This result may have implications for the design of matching platforms such as labor markets, where it suggests enabling at least some asynchronous inspection.

Proof. We combine the values along each edge of the bundled star graph and note that exactly one edge can be chosen by a matching: we can claim exactly one pair of boxes. We denote the combined boxes on each edge $\{i, j\}$ as having a value $v'_{\{i,j\}} = v_{ij} + v_{ji}$ with new value distribution $v'_{\{i,j\}} \sim D'_{\{i,j\}}$, the convolution of D_{ij} and D_{ji}.

Using Lemma 2.2.1 from Kleinberg et al. [2016], the expected value of the optimal algorithm is

$$\mathbb{E}\left[\sum_i \mathbb{A}_{\{u,w_i\}} v'_{\{u,w_i\}} - \mathbb{I}_{\{u,w_i\}}\left(c_{\{u,w_i\}} + c_{\{u,w_i\}}\right)\right] = \mathbb{E}\left[\left(\max_i \kappa_{\{u,w_i\}}\right)^+\right],$$

where $\kappa_{\{u,w_i\}} = \min(\sigma_{\{u,w_i\}}, v'_{\{u,w_i\}})$ and $\sigma_{\{u,w_i\}}$ is the index for the combined boxes. We can compute the index for every edge of the example as

$$\begin{aligned}\frac{1}{n} + 1 &= \mathbb{E}\left[(v_{u,w_i} + 1 - \sigma_i)^+\right] \\ &= \frac{1}{n}(n + 1 - \sigma_i) + \left(1 - \frac{1}{n}\right)(1 - \sigma_i) \\ &= 2 - \sigma_i \\ \implies \quad \sigma_i &= 1 - \tfrac{1}{n}.\end{aligned}$$

By Lemma 2.2.1 then, the optimal bundled-box algorithm's welfare is at most

$$\mathbb{E}[(\max_i \min(\sigma_{\{u,w_i\}}, v'_{\{u,w_i\}}))^+] \leq 1.$$

However, with independent inspections allowed, a simple algorithm does much better: inspect every edge from u's side. If for any i the realized value is $v_{u,w_i} = 1/n$, then inspect $v_{w_i,u}$ and match that edge. This algorithm has expected welfare $(1 - (1 - 1/n)^n)(n + 1 - 1) - 1 \geq n(1 - e^{-1}) - 1 = \Omega(n)$. □

4.1.2 Vertex-Based Algorithms

We next consider algorithms which treat the boxes independently of one another, making decisions to open or permanently accept boxes independently of any other boxes. A natural approach is to directly extend Weitzman [1979]'s optimal index-based approach to this setting. For example, one could inspect the boxes from largest Weitzman index downward, matching an edge once both of its endpoints are inspected and the sum of the values is large enough. Approaches like this match the *frugal algorithm* framework of Singla [2018]. We find that, when values must be positive, the mechanism of Bowers and Waggoner [2024] yields a 1/4-approximation algorithm, but that no vertex-based algorithm can achieve a constant approximation for general values.

Definition 4.1.1. An algorithm is *vertex-based* if, for each ordered pair (i, j), the algorithm uses only Weitzman's index σ_{ij}, the inspection cost of a vertex c_{ij}, and (when revealed) the value v_{ij}, but does not have access to the distribution D_{ij}.

To enable analysis, Definition 4.1.1 formalizes vertex-based algorithms by limiting the information available to the algorithm. A vertex-based algorithm knows everything about the setting except for the distribution of values D_{ij} at each endpoint. We replace the knowledge of D_{ij} with direct access to the Weitzman index σ_{ij} instead.

4.1.2.1 Positive Values

We have one positive approximation result for vertex-based algorithms, in the positive valuation only setting considered in Chapter 3. This result follows very closely from the Marshallian Match considered there, and simply leverages the deviation strategies considered in Section 3.3.

Recall that in the *positive-values* setting of Pandora's Matching Problem, value distributions D_{ij} are supported on the positive reals and $\mathbb{E}[v_{ij}] \geq c_{ij}$ for all (i, j). This implies $\sigma_{ij} \geq 0$ and $\kappa_{ij} \geq 0$ as well.

This setting is already nontrivial, as our impossibility result of Proposition 4.1.1 holds even in the positive-valued setting, showing that no bundled-box algorithm can obtain a better-than-$O(1/n)$ approximation ratio. We can, however, convert the Marshallian Match mechanism into a vertex-based algorithm for matching in the positive values regime.

Proposition 4.1.2. *There exists a vertex-based algorithm which achieves a* $1/4$*-approximation in the positive-values setting of Pandora's Matching Problem.*

We define our algorithm derived from the Marshallian Match mechanism of Chapter 3 here, and prove it obtains a $1/4$-approximation in the positive-values setting.

Algorithm 1 (Vertex-Based Descending Procedure)**.** While there is a feasible edge that is not matched, consider the highest σ_{ij} of unopened boxes and the highest $v_{k\ell}$ of opened boxes. If $\sigma_{ij} > v_{k\ell}$, open box (i, j). Otherwise, match the edge $\{k, \ell\}$, opening the (ℓ, k) endpoint if necessary.

This algorithm is the direct result of assuming every agent bids truthfully on every edge at all times, and then running the Marshallian Match. While we do not prove this, we not that Algorithm 1 is in fact a Descending Procedure as defined in Section 2.4.

We will show that the vertex-based descending procedure is a 1/4-approximation for the positive-valued setting, but first we prove several helpful intermediate facts.

Observation 4.1.1. *The Vertex-Based Descending Procedure (Algorithm 1) is non-exposed on every ordered pair* (i, j).

Proof. If the algorithm inspects (i, j) and finds that $v_{ij} \geq \sigma_{ij}$, the pair is added back at the top of the list as σ_{ij} was previously the largest entry in the list. At the next step, the algorithm will examine the same pair again, find that it has already been inspected, and add it to the matching. □

We compare the welfare of this algorithm to the greedy matching algorithm on edges with weights $w_{\{i,j\}} = \max(\kappa_{ij}, \kappa_{ji})$. We call this the max-$\kappa$-Greedy matching and denote it by the random variable $\mathbb{A}^G$. It is unclear whether there exists a box-opening algorithm which achieves this matching, since the greedy algorithm relies on revealed capped values κ which are not initially accessible to an algorithm. We show that the vertex-based matching algorithm, in fact, produces a matching which is exactly the greedy-by-max-kappa matching.

Claim 4.1.1. *With probability one, the vertex-based descending procedure (Algorithm 1) produces the same matching as max-*κ*-Greedy for all realizations of values.*

Proof. Fix some realization of values, and suppose the vertex-based algorithm Algorithm 1 matches the edge $\{i, j\}$. Without loss of generality, assume the edge was matched from the (i, j) endpoint, so $\kappa_{ij} = \max(\kappa_{ij}, \kappa_{ji})$.

We show that $\kappa_{ij} \geq \kappa_{k\ell'}$ for all other feasible pairs (k, ℓ).

Assume by way of contradiction that there exists another ordered pair (k, ℓ) such that $\kappa_{ij} < \kappa_{k\ell}$. Then either $\sigma_{ij} < \kappa_{k\ell}$ or $v_{ij} < \kappa_{k\ell}$.

If $\sigma_{ij} < \kappa_{k\ell}$, then $\sigma_{ij} < \sigma_{k\ell}$ and the algorithm would have inspected (k, ℓ) before inspecting (i, j). After inspecting, the algorithm would find $v_{k\ell} > \sigma_{ij}$ and would match $\{k, \ell\}$ before matching $\{i, j\}$, a contradiction.

Similarly, if $v_{ij} < \kappa_{k\ell}$, then $v_{ij} < \sigma_{k\ell}$ and the algorithm would have inspect (k, ℓ) before matching $\{i, j\}$. After inspecting, the algorithm would reinsert (k, ℓ) into the list with value $v_{k\ell} > v_{ij}$, and would match $\{k, \ell\}$ before matching $\{i, j\}$, also a contradiction.

The algorithm adds the feasible edge endpoint $\{i, j\}$ with the largest κ_{ij} as the next edge to the matching. Since and edge is listed twice by (i, j) and (j, i), the next added edge then maximizes $\max(\kappa_{ij}, \kappa_{ji})$, and the matching created is exactly max-κ-Greedy. □

Finally, we use these facts about Algorithm 1 to prove our 1/4-approximation result.

Proof of Proposition 4.1.2. We denote the optimal matching produced by any box-opening algorithm $\mathbb{A}^{\mathrm{OPT}}$. By Lemma 2.2.1, the value of the optimal policy is $\mathrm{OPT} \leq \mathbb{E}[\sum_{(i,j)\in E} \mathbb{A}_{ij}^{\mathrm{OPT}}(\kappa_{ij}+\kappa_{ji})]$. Since the algorithm is non-exposed with positive valuations, the expected value of the algorithm is

$$\begin{aligned}
\mathbb{E}\left[\sum_{(i,j)} \mathbb{A}_{ij} v_{ij} - \mathbb{I}_{ij} c_{ij}\right] &= \mathbb{E}\left[\sum_{\{i,j\}\in E} \mathbb{A}_{ij}(\kappa_{ij} + \kappa_{ji})\right] \\
&\geq \mathbb{E}\left[\sum_{\{i,j\}\in E} \mathbb{A}_{ij} \max(\kappa_{ij}, \kappa_{ji})\right] \\
&= \mathbb{E}\left[\sum_{\{i,j\}\in E} \mathbb{A}_{ij}^{\mathrm{G}} \max(\kappa_{ij}, \kappa_{ji})\right],
\end{aligned}$$

where $\mathbb{A}_{ij}^{G}$ is the indicator for max-κ-Greedy. Since the greedy matching algorithm gives a 1/2-approximation of the optimal matching on the given edge weight, this is at least half the value of the optimal matching $\mathbb{A}^*$ on max-kappa-weighted edges. The welfare optimal matching on max-κ-weighted edges is, in turn, an upper bound on the welfare of the optimal matching on edges with different weights, particularly the weights $w_{\{i,j\}} = \kappa_{ij} + \kappa_{ji}$. Combining these steps mathematically

then,

$$\mathbb{E}\left[\sum_{(i,j)\in E}\mathbb{A}^{\mathrm{G}}_{ij}\max(\kappa_{ij},\kappa_{ji})\right] \geq \frac{1}{2}\,\mathbb{E}\left[\sum_{(i,j)\in E}\mathbb{A}^{*}_{ij}\max(\kappa_{ij},\kappa_{ji})\right]$$
$$\geq \frac{1}{2}\,\mathbb{E}\left[\sum_{(i,j)\in E}\mathbb{A}^{\mathrm{OPT}}_{ij}\max(\kappa_{ij},\kappa_{ji})\right]$$
$$\geq \frac{1}{4}\,\mathbb{E}\left[\sum_{(i,j)\in E}\mathbb{A}^{\mathrm{OPT}}_{ij}(\kappa_{ij}+\kappa_{ji})\right] \geq \frac{\mathrm{OPT}}{4}.$$

□

This reliance on the max-endpoint greedy matching causes the Vertex-Oriented Descending Procedure of Algorithm 1 to immediately fail on instances where values may be negative.

4.1.2.2 General Values and Failure of Vertex-Based Algorithms

We find that the problems of Algorithm 1 are not fixable for any vertex-based algorithm when values may be negative.

Proposition 4.1.3. *No vertex-based algorithm can guarantee any positive approximation factor for Pandora's Matching Problem.*

The proof of Proposition 4.1.3 relies on a single-edge counterexample. In this example, a vertex-based algorithm cannot distinguish the endpoints to pick an inspection ordering but an optimal policy must inspect them in a specific order.

The idea will be that the edge's endpoints are indistinguishable to vertex-based algorithms: they have the same cost of inspection and Weitzman index. However, to achieve nontrivial welfare, it is crucial to inspect the endpoints in the correct order based on their differing distributions of values.

Example 4.1.1 (Indistinguishable Edge)**.** *The graph consists of a single edge $\{i,j\}$. The Pandora box (j,i) has deterministic value $v_{ji}=2$ and cost $c_{ji}=1$. The Pandora box (i,j) has value $v_{ij}=9$ with probability $1/8$ and $v_{ij}=-3$ otherwise, and has cost $c_{ij}=1$.*

We find that an optimal algorithm can achieve a positive value by correctly ordering the inspection of endpoints, while a vertex-based algorithm cannot distinguish the endpoints to choose the correct order.

Similar to the proof of Proposition 4.1.1, the key is to observe that one direction of inspection obtains information immediately while the other pays an extra cost for unhelpful information. Randomizing the inspection order is not enough to obtain a constant approximation because one direction has negative welfare.

Faced with the possibility of inspecting the wrong order, any inspecting vertex-based algorithm expects to receive a negative welfare and would rather do nothing. This will hold even if the vertex-based algorithm is randomized. We break this proof down into an analysis of the optimal solution, the claim that a vertex-based algorithm cannot distinguish the endpoints, and combine the two for a proof of Proposition 4.1.3.

Claim 4.1.2. OPT *on Example 4.1.1 achieves welfare* 1/4.

Proof. We consider the options available to an algorithm: starting its inspection with vertex i and with j. (The algorithm can also do nothing, for welfare zero.)

If it inspects vertex i first, then it either finds $v_{ij} = 9$ or $v_{ij} = -3$. If $v_{ij} = -3$, it should terminate immediately as the value from v_{ji} will not cover the loss from v_{ij}. If $v_{ij} = 9$, however, it should inspect (j, i) and claim the edge for a value of 11. Thus by inspecting vertex i first, and then inspecting and matching j if and only if $v_{ij} = 9$, the algorithm gets expected value $\mathbb{E}[\mathbb{A}(v_{ij} + v_{ji} - c_{ji})] - c_{ij} = (9 + 2 - 1)/8 - 1 = 1/4$.

If the algorithm instead inspects j first, it gains no information about i for a cost of 1. If it chooses not to inspect i, then its net welfare is -1. On the other hand, if it inspects i, it has a 1/8 chance of finding a high value worth claiming the edge. This gives an overall value of $\mathbb{E}[\mathbb{A}(v_{ij} + v_{ji})] - c_{ji} - c_{ij} = -5/8$. While better than not performing the second inspection, it is still better to perform no inspections at all than to start from (j, i). □

Lemma 4.1.1. *Prior to inspection, any vertex-based algorithm cannot distinguish the endpoints of the edge in Example 4.1.1.*

Proof. It is sufficient to show that the values that the algorithm has access to prior to inspection $(c_{ij},\ \sigma_{ij})$ are the same for both endpoints. The costs c_{ij}, c_{ji} are both deterministically 1 and therefore indistinguishable, so we compute the indices σ_{ij}.

For (j,i), σ_{ji} is the solution to $1 = \mathbb{E}[(v_{ji} - \sigma_{ji})^+] = (2 - \sigma_{ji})^+$, so $\sigma_{ji} = 1$. For the endpoint (i,j), we solve $1 = \mathbb{E}[(v_{ij} - \sigma_{ij})^+] = \frac{1}{8}(9 - \sigma_{ij})^+$, so $\sigma_{ij} = 1$ as well. □

Proof of Proposition 4.1.3. Given an algorithm, we present it with Example 4.1.1, uniformly randomizing the order of the endpoints (i.e. the identities of i and j).

Any algorithm which performs no inspections gets value 0.

Otherwise, since the algorithm cannot distinguish the endpoints due to our randomization, it has a 1/2 probability of beginning its inspection with vertex i and 1/2 with j. The best any algorithm can do, if it does inspect, is the average of the best outcomes if it began inspection from i or from j:

$$\frac{1}{2}\left(\mathbb{E}[\mathbb{A}(v_{ij} + v_{ji} - c_{ji})] - c_{ij}\right) + \frac{1}{2}\left(\mathbb{E}[\mathbb{A}(v_{ij} + v_{ji})] - c_{ji} - c_{ij}\right) = \frac{1}{8} - \frac{5}{16} = -3/16.$$

Therefore, the optimal vertex-based algorithm on this instance performs no inspections. □

This negative result has implications for platform and mechanism design: an algorithm must coordinate a nontrivial amount of information-sharing in order to guarantee high welfare in general. In Section 4.3 we will turn to algorithm which consider edges as a whole, rather than the individual indices of the endpoints.

The correct way to combine endpoints' information. If we must combine endpoints' information to make good decisions, how should we do so? An edge in the Pandora's Matching Problem involves *multiple stages* of information gathering: first we inspect one endpoint and learn something about the edge's total value, and later we inspect the other. This insight allows us to apply the tools of Section 2.3 to treat an edge as an ordered sequence of inspections.

4.2 Pandora's Matching as Bandits

We begin with the simple observation that, for a single edge, if any boxes on that edge are opened, one box must be opened first. Once we have chosen which box to open first, the edge behaves as a nested box within a box, or a two stage *bandit* (Definition 5.2.2). This allows us to leverage the tools built in Section 2.3, particularly the concepts of *bandits* and *descending procedures*. This observation reframes the matching problem to choosing, for each edge, between opening two, mutually-exclusive, bandits and selecting a feasible set of fully-advanced bandits.

We begin by considering an easier problem in which the graph is restricted to require the boxes for each edge be opened in a given order, a reasonable modeling assumption for many settings.

4.2.1 Orientations

Suppose that each edge is restricted to a particular order of inspection, e.g. (i, j) must be inspected prior to (j, i). This is a natural setting for some applications. A job-search platform may require candidates to submit resumes before companies can screen them: inspections must begin on the candidate side before moving to the employer side of the match. Formally, we capture the predetermined order of inspection with an orientation of the graph.

Definition 4.2.1. An *orientation* $\mathcal{O}$ of an undirected graph $G = (V, E)$ is a set of ordered pairs of vertices such that, for every edge $\{i, j\} \in E$, $\mathcal{O}$ contains exactly one of (i, j) or (j, i).

We refer to an ordered pair $(i, j) \in \mathcal{O}$ as an *oriented edge*. Given an instance and an orientation $\mathcal{O}$ of the edges, we say an algorithm is $\mathcal{O}$*-oriented* if for all realizations its sequence of inspections respects the orientation, i.e. if $(i, j) \in \mathcal{O}$, then endpoint (i, j) must be inspected prior to (j, i), if (j, i) is inspected. If an algorithm is $\mathcal{O}$-oriented for some $\mathcal{O}$, we call it a *fixed-orientation algorithm*.

Comparison to committing policies. Fixed-orientation algorithms may be reminiscent of *committing policies* appearing in related work on Pandora's Box with Non-obligatory Inspection, e.g. Beyhaghi and Kleinberg [2019]. Committing policies fix the sequence of box inspections in

advance, though they may choose their stopping time dynamically. Although a fixed-orientation algorithm "commits" to an orientation $\mathcal{O}$, it will in general adaptively decide which edge, i.e. associated bandit, to advance next, depending on information revealed at previous stages. In other words, it only commits to order of inspection *within* each edge. Another difference is that, in the nonobligatory setting, we can reduce WLOG to a set of $n+1$ commitments with n boxes; it is not obvious how to reduce the $2^{|E|}$ orientations of a Pandora's Matching Problem instance.

Orientations and Bandits. For fixed-orientation algorithms, we represent the inspections required for an edge as a two-stage bandit. Advancing the first stage reveals the value of one endpoint, and the second stage reveals the full value of the edge. For example, if $(i,j) \in \mathcal{O}$, then edge $\{i,j\}$ is a bandit where the first stage costs c_{ij} to advance and reveals signal v_{ij}, and the second stage costs c_{ji} and reveals the combined value $v_{ij}+v_{ji}$. To advance to the third stage, or "claim the bandit", is to add the edge to our matching.

We now consider optimizing expected welfare when restricted to a fixed orientation $\mathcal{O}$. Even under the fixed-orientation restriction, the optimal matching algorithm is unclear. The problem is more general than matching with a single Pandora box on each edge, a problem suspected to be NP-hard. But for a given orientation $\mathcal{O}$, the natural descending procedure achieves a 1/2-approximation to the optimal $\mathcal{O}$-oriented algorithm.

Algorithm 2 ($\mathcal{O}$-Oriented Descending Procedure)**.** Given a fixed orientation $\mathcal{O}$, for steps $t = 1, \ldots$:

- For each $(i,j) \in \mathcal{O}$, let σ_{ij}^t denote the current Weitzman index of bandit (i,j).
- Advance the bandit (i,j) with largest σ_{ij}^t.
    - If advancing results in "claiming" the bandit, then add edge $\{i,j\}$ to the matching. Delete any edges incident to i or j from $\mathcal{O}$.
- If at any point the largest σ_{ij}^t is negative or $\mathcal{O}$ is empty, stop.

This is indeed a *descending procedure* (Definition 2.4.1), as it maintains a nonincreasing

eligible set (technically, $\mathcal{O}$ union the matched edges) and always advances and retains the largest-index bandit in that set. Lemma 2.4.1 therefore gives the following.

Corollary 4.2.1. *For any orientation $\mathcal{O}$, the $\mathcal{O}$-Oriented Descending Procedure is non-exposed.*

We can continue with an analysis analogous to that of Kleinberg et al. [2016] for the case of matching with a bandit on each edge. Given $\mathcal{O}$, let *$\mathcal{O}$-κ-Greedy* refer to the following matching (used only for analysis): for each oriented edge $(i,j) \in \mathcal{O}$, let $x_{\{i,j\}} = \kappa_{ij}^{(1)}$, the capped value for bandit (i,j); take the matching produced by the weighted greedy algorithm with edge weights $x_{\{i,j\}}$.

Lemma 4.2.1. *For any orientation $\mathcal{O}$, with probability one, the $\mathcal{O}$-Oriented Descending Procedure 2 produces the same matching as $\mathcal{O}$-κ-Greedy.*

Proof. Fix a realization of all random variables. Suppose edge $e = \{i,j\}$ is matched by the $\mathcal{O}$-Oriented Descending Procedure. We abuse terms by referring to e as a bandit, where it is oriented according to $\mathcal{O}$. We show that $\kappa_e^{(1)} \geq \kappa_{e'}^{(1)}$ for any edge e' that was available (i.e. legal to match) when e is matched.

Since bandit e has been fully advanced, all of its Weitzman indices $\sigma_e^{(1)}, \sigma_e^{(2)}, \sigma_e^{(3)} = v_e$ were at some point the largest current Weitzman index among all available bandits. In particular, since e' is available when e is matched, each index $\sigma_e^{(k)}$ is larger than $\sigma_{e'}^{(\ell)}$ for some ℓ, so $\kappa_e^{(1)} = \min_k \sigma_e^{(k)} \geq \min_\ell \sigma_{e'}^{(\ell)} = \kappa_{e'}^{(1)}$. □

Using the characterization of the $\mathcal{O}$-Oriented Descending Procedure's matching as the greedy matching, we show that the algorithm achieves a constant factor of the optimal welfare.

Proposition 4.2.1. *The $\mathcal{O}$-Oriented Descending Procedure (Algorithm 2) achieves at least $1/2$ the expected welfare of the optimal $\mathcal{O}$-oriented algorithm.*

Proof. Let $\mathbb{A}_{ij}^{\mathrm{OPT}}$ be the indicator that edge $\{i,j\}$ is claimed by the optimal $\mathcal{O}$-oriented algorithm. For any realization of all values, consider a graph with edge weights $\{\kappa_{ij}^{(1)}\}$. In this graph, let $\mathbb{A}_{ij}^{G}$

be the indicator for $\{i, j\}$ being included in the greedy matching, i.e. in $\mathcal{O}$-κ-Greedy; and let $\mathbb{A}^*_{ij}$ be the corresponding indicator for the maximum weighted matching. The welfare of the $\mathcal{O}$-Oriented Descending Procedure satisfies

$$\begin{aligned}
\text{Welf}(\text{ALG}) &= \mathbb{E} \sum_{(i,j)\in\mathcal{O}} \mathbb{A}_{ij}\kappa^{(1)}_{ij} && \text{Corollary 4.2.1} \\
&= \mathbb{E} \sum_{(i,j)\in\mathcal{O}} \mathbb{A}^{G}_{ij}\kappa^{(1)}_{ij} && \text{Lemma 4.2.1} \\
&\geq \frac{1}{2}\,\mathbb{E} \sum_{(i,j)\in\mathcal{O}} \mathbb{A}^{*}_{ij}\kappa^{(1)}_{ij} && \text{weighted matching fact} \\
&\geq \frac{1}{2}\,\mathbb{E} \sum_{(i,j)\in\mathcal{O}} \mathbb{A}^{\text{OPT}}_{ij}\kappa^{(1)}_{ij} && \text{optimality of } \{\mathbb{A}^*_{ij}\} \\
&\geq \frac{1}{2}\text{Welf}(\text{OPT}) && \text{Lemma 2.3.1.}
\end{aligned}$$

□

4.3 Simple Approximation Algorithms

We now finally provide two approximation algorithms for Pandora's Matching Problem, both of which first pick an orientation $\mathcal{O}$ and then run the $\mathcal{O}$-Oriented Descending Procedure.

The first algorithm we consider is randomized, and orients each edge independently. We will first present and analyze the randomized algorithm, before presenting a deterministic algorithm using concepts developed in the randomized setting.

Algorithm 3 (Randomized Pandora Matching Algorithm)**.** Construct an orientation $\mathcal{O}$ by orienting each edge $\{i, j\}$ as either (i, j) or (j, i) uniformly and independently at random. Run the $\mathcal{O}$-Oriented Descending procedure.

Note that by construction, $\mathcal{O}$ is drawn from the uniform distribution over all orientations for the graph.

Theorem 4.3.1. *In Pandora's Matching Problem, Algorithm 3 achieves at least a* $1/4$*-approximation of the optimal welfare.*

Before proving the approximation guarantee, we first define some helpful notation and an upper-bound on the value of an optimal algorithm. We will also use these tools in defining our deterministic algorithm.

Definition 4.3.1. The *reverse* of an orientation $\mathcal{O}$ is $\hat{\mathcal{O}} = \{(j,i) \mid (i,j) \in \mathcal{O}\}$.

The key step in the analysis of the randomized algorithm is an upper bound on the welfare of the optimal algorithm relative to an orientation and its reverse. The crux of this bound is that any realization of an algorithm induces an orientation on edges it inspects, based on the order of inspection. The welfare an edge generates in an algorithm, then, can be "charged" across any orientation $\mathcal{O}$ and its reverse, according to which orientation matches the direction in which the edge is inspected.

Lemma 4.3.1. *For any orientation $\mathcal{O}$ and it s reverse $\hat{\mathcal{O}}$, the welfare of* OPT *on Pandora's Matching Problem is upper-bounded by*

$$\text{Welf}(\text{OPT}) \leq 2 \cdot \text{Welf}(\mathcal{O}\text{-}\textit{desc.}) \quad + \quad 2 \cdot \text{Welf}(\hat{\mathcal{O}}\text{-}\textit{desc.}).$$

Proof. First, we decompose the welfare of the optimal algorithm into the contributions from $\mathcal{O}$ and $\hat{\mathcal{O}}$. Each edge that is inspected in the optimal algorithm is inspected in the order given by one of the two orientations. Given an edge $\{i,j\}$, let W_{ij}^{OPT} be the welfare contribution of the edge $\{i,j\}$ (i.e. total value claimed minus inspection cost paid) if box (i,j) is inspected prior to (j,i), else $W_{ij}^{\text{OPT}} = 0$. Define W_{ji}^{OPT} analogously, so that the total welfare contribution of the edge is $W_{ij}^{\text{OPT}} + W_{ji}^{\text{OPT}}$. For each $(i,j) \in \mathcal{O}$, let $W_{ij}^{\mathcal{O}}$ (respectively $W_{ji}^{\hat{\mathcal{O}}}$) be the welfare contribution of bandit (i,j) (respectively, (j,i)) when running the optimal $\mathcal{O}$-oriented algorithm (respectively, $\hat{\mathcal{O}}$-oriented). Let Welf($\mathcal{O}$-desc.) be the expected welfare of the $\mathcal{O}$-Oriented Descending Procedure,

and analogously for Welf($\hat{\mathcal{O}}$-desc.).

$$\begin{aligned}
\text{Welf}(\text{OPT}) &= \mathbb{E} \sum_{\{i,j\}\in E} \left(W_{ij}^{\text{OPT}} + W_{ji}^{\text{OPT}}\right) \\
&= \mathbb{E} \sum_{(i,j)\in\mathcal{O}} W_{ij}^{\text{OPT}} \quad + \quad \mathbb{E} \sum_{(j,i)\in\hat{\mathcal{O}}} W_{ji}^{\text{OPT}} \\
&\leq \mathbb{E} \sum_{(i,j)\in\mathcal{O}} W_{ij}^{\mathcal{O}} \quad + \quad \mathbb{E} \sum_{(j,i)\in\hat{\mathcal{O}}} W_{ji}^{\hat{\mathcal{O}}} \\
&\leq 2\text{Welf}(\mathcal{O}\text{-desc.}) \quad + \quad 2\text{Welf}(\hat{\mathcal{O}}\text{-desc.}) \qquad \text{Proposition 4.2.1}
\end{aligned}$$

□

This bound on the optimal welfare of any algorithm for Pandora's Matching Problem allows a quick proof of Theorem 4.3.1.

Proof of Theorem 4.3.1. Since the orientation used by the randomized algorithm is drawn from the uniform distribution, the expected welfare is

$$\begin{aligned}
\text{Welf}(\text{ALG}^{\text{rand}}) &= \mathop{\mathbb{E}}_{\mathcal{O}\sim \text{ unif}} [\text{Welf}(\mathcal{O}\text{-desc.})] \\
&= \mathop{\mathbb{E}}_{\mathcal{O}\sim \text{ unif}} \left[\frac{1}{2}\text{Welf}(\mathcal{O}\text{-desc.}) \quad + \quad \frac{1}{2}\text{Welf}(\hat{\mathcal{O}}\text{-desc.})\right] \\
&\geq \frac{1}{2} \mathop{\mathbb{E}}_{\mathcal{O}\sim \text{ unif}} \left[\frac{1}{2}\text{Welf}(\text{OPT})\right] \qquad \text{Lemma 4.3.1} \\
&= \frac{1}{4}\text{Welf}(\text{OPT}).
\end{aligned}$$

□

Given the heavy reliance of Algorithm 3 on randomization of orientation, we also present a deterministic algorithm based on determining an orientation and then running the associated descending procedure. A similar "best-of-two-worlds" technique is used in Beyhaghi and Kleinberg [2019] in a different context.

Algorithm 4 (Best-of-Two-Worlds Algorithm)**.** Given an instance of Pandora's Matching Problem, pick an arbitrary orientation $\mathcal{O}$. Compute the expected welfare of the $\mathcal{O}$-Oriented Descending Procedure and the $\hat{\mathcal{O}}$-Oriented Descending Procedure. Run the one with larger expected welfare.

This algorithm differs from the randomized algorithm in that the final orientation is determined globally. In the randomized algorithm we select the orientation of each edge independently, but the Best-of-Two-Worlds algorithm compares two orientations over the entire graph and selects one.

To show the approximation guarantee of Theorem 4.3.1, we first prove the upper-bound on the optimal welfare of Lemma 4.3.1.

Finally, we can use this upper-bound to prove an approximation guarantee for the randomized algorithm.

Theorem 4.3.2. *In Pandora's Matching Problem, the Best-of-Two-Worlds algorithm achieves at least a* $1/4$*-approximation of the optimal welfare.*

We leverage Lemma 4.3.1 again to prove the approximation guarantee of Theorem 4.3.2.

Proof. The welfare of the Best-of-Two-Worlds algorithm is

$$\begin{aligned}\max\left\{\text{Welf}(\mathcal{O}\text{-desc.}) \ , \ \text{Welf}(\hat{\mathcal{O}}\text{-desc.})\right\} &\geq \frac{1}{2}\text{Welf}(\mathcal{O}\text{-desc.}) \ + \ \frac{1}{2}\text{Welf}(\hat{\mathcal{O}}\text{-desc.}) \\ &\geq \frac{1}{4}\text{Welf}(\text{OPT}) \qquad \text{Lemma 4.3.1}\end{aligned}$$

□

Extension to correlated-within-edges model. We can extend these positive results to a model where each pair of values v_{ij}, v_{ji} on an edge are arbitrarily correlated with each other and independent of the other edges. In fact, we can extend this a bit further: in the *correlated-within-edges model*, we suppose that each Pandora box on an endpoint (i, j) reveals for cost c_{ij} an arbitrary signal s_{ij}, the total value on an edge is an arbitrary measurable function $f_{\{i,j\}}(s_{ij}, s_{ji})$, and for each edge, the pair (s_{ij}, s_{ji}) are drawn jointly from a distribution $D_{\{i,j\}}$, independently of all other pairs.

In this setting, the definition of an algorithm is unchanged, i.e. it adaptively pays to open boxes and eventually outputs a matching of fully-opened edges. The proofs of Proposition 4.2.1 and

Lemma 4.3.1 go through unchanged, and the proofs of both algorithms hold with values correlated along edges. In the proof of Lemma 4.3.1, an edge is modeled as a pair of arbitrary bandits of ordered endpoints (i, j) and (j, i), where an algorithm must pick one of the bandits at the time of first inspection of the edge. For any orientation $\mathcal{O}$, i.e. a commitment on each edge to one of its two possible bandits, the proof then cites Proposition 4.2.1. And Proposition 4.2.1 addresses the problem of matching with a single, arbitrary Pandora bandit per edge, utilizing our generic results for bandits in Section 2.3.

Observation 4.3.1. *The* 1/4*-approximations of the Randomized Pandora Matching and Best-of-Two-Worlds algorithms continue to hold in the correlated-within-edges model.*

4.4 Dessert?

We have finally presented two constant-factor approximation algorithms for Pandora's Matching Problem, one randomized and one deterministic. The extreme simplicity of our randomized algorithm's edge orientation choice hints that a simpler deterministic approach should be possible, potentially following the structure of the randomized algorithm: for each edge, pick an orientation independently of any other edge orientations. This would greatly simplify our deterministic approximation algorithm, and such an algorithm would be a nice "dessert". In particular, a natural approach might be to orient each edge according to which associated bandit has a higher initial Weitzman index, then run the Oriented Descending Procedure. We consider the following generalization of this approach:

Definition 4.4.1. An *edge-based fixed-orientation algorithm* is one that, given an instance of Pandora's Matching Problem, constructs an orientation $\mathcal{O}$ by applying a deterministic rule to each edge $\{i, j\}$ that depends only on the parameters of that edge, $D_{ij}, c_{ij}, D_{ji}, c_{ji}$, and then runs some $\mathcal{O}$-oriented algorithm.

Unfortunately, we have the following result.

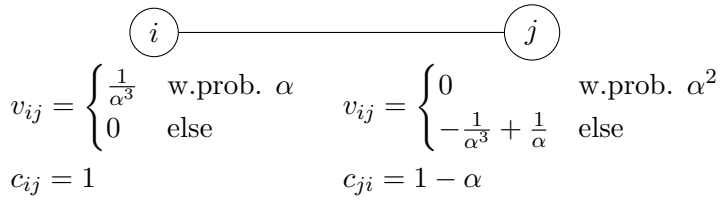


Figure 4.2: An edge which cannot be oriented without context

An edge used to prove Theorem 4.4.1. Without an outside option, it is optimal to inspect i before j (Instance 1), but with an outside option of $\frac{1}{\alpha}$, it is optimal to inspect j before i (Instance 2).

Theorem 4.4.1 ("No dessert")**.** *No edge-based fixed-orientation algorithm guarantees any positive approximation for Pandora's Matching Problem.*

Intuition. The proof relies on a particular edge, shown in Figure 4.2. Every edge-based fixed orientation algorithm either orients that edge in one direction or the other. If there is no outside option, then the (i, j) orientation is arbitrarily better, because we want to match the edge if and only if $v_{ij} > 0$, and we discover this immediately, saving the cost of inspecting (j, i) when $v_{ij} = 0$. But if there is an outside option of $\frac{1}{\alpha}$, then the bandit is worthwhile if and only if both boxes contain their high value (a "success"). Here the (j, i) orientation is arbitrarily better, as the overall probability of a success is the same but we pay a second inspection cost with only α^2 probability instead of α.

For the skeptical reader, we also give an alternative proof of Theorem 4.4.1 in Appendix E.1 that does not use any Pandora box machinery.

Proof. Recall that $\sigma_{ij}^{(1)}$ is the initial Weitzman index for the bandit corresponding to orientation (i, j), and $\kappa_{ij}^{(1)}$ is the capped value, a random variable. By Lemma 2.3.1, the expected welfare contribution of a bandit (i, j) is upper-bounded $\mathbb{E}[\mathbb{A}_{ij}\kappa_{ij}^{(1)}]$, with equality if the algorithm is non-exposed. In particular, since $\kappa_{ij}^{(1)} \leq \sigma_{ij}^{(1)}$, we can upper-bound the welfare contribution of a box under any algorithm by the maximum $\sigma_{ij}^{(1)}$ over any bandit available to it.

We record the following (calculations appear in Appendix E):

- $\sigma_{ij}^{(1)} = \frac{1}{\alpha} - 1$.

- $\kappa_{ij}^{(1)} = \frac{1}{\alpha} - 1$ with probability α, otherwise negative.
- $\sigma_{ji}^{(1)} = \frac{1}{\alpha^2} - \frac{1}{\alpha}$.
- $\kappa_{ji}^{(1)} = \frac{1}{\alpha^2} - \frac{1}{\alpha}$ with probability α^3, otherwise nonpositive.

Instance 1. We consider a graph consisting of just the edge in Figure 4.2. An algorithm can orient the edge as a bandit (i, j) or as a bandit (j, i). In each case, the descending procedure is optimal and achieves welfare equal to expected capped value (Section 2.3). With a single (i, j) bandit, the descending procedure obtains welfare $\mathbb{E}[(\kappa_{ij})^+] = \alpha\left(\frac{1}{\alpha} - 1\right) = 1 - \alpha$. With (j, i), the welfare is $\mathbb{E}[(\kappa_{ji}^{(1)})^+] = \alpha^3\left(\frac{1}{\alpha^2} - \frac{1}{\alpha}\right) = \alpha - \alpha^2$. So the orientation (j, i) first is an approximation of $\frac{\alpha-\alpha^2}{1-\alpha} = \alpha$. This holds for arbitrarily small $\alpha > 0$, so no algorithm that orients the edge as (j, i) can guarantee a positive approximation.

Instance 2. We will consider a star graph with i at the center. There is a special edge $\{i, k\}$ with $v_{ik} = \frac{1}{\alpha}$, $c_{ik} = 0$, $v_{ki} = 0$, $c_{ki} = 0$. In other words, i has an "outside option" $\frac{1}{\alpha}$ it can match to at any time. Every other edge in the graph $\{i, j_1\}, \ldots, \{i, j_m\}$ is an independent copy of the edge $\{i, j\}$ in Figure 4.2. The number of copies will be $m = \omega(\frac{1}{\alpha^3})$. A edge-based fixed-orientation algorithm must orient each copy in the same direction, as it applies the same deterministic rule to each edge.

After orienting every edge in the (j, i) direction, we have an instance of Nested Pandora's Boxes with $m + 1$ bandits. The optimal welfare, achieved by the descending procedure, is equal to the expected maximum capped value of a bandit. With $\omega(\frac{1}{\alpha^3})$ bandits, some bandit has capped value $\frac{1}{\alpha^2} - \frac{1}{\alpha}$ with probability $1 - o(1)$; so the algorithm's expected welfare is $(\frac{1}{\alpha^2} - \frac{1}{\alpha})(1 - o(1))$.

After orienting every edge in the (i, j) direction, the same argument holds, except that the capped value of every bandit is always at most $\frac{1}{\alpha} - 1$. These are actually less than the capped value of the outside option, which is deterministically $\frac{1}{\alpha}$, which is the maximum capped value and therefore the best any algorithm can do.

The ratio of the optimal (i, j) orientation welfare to (j, i) is, asymptotically, $\frac{1/\alpha}{1/\alpha^2} = \alpha$.

This holds as $\alpha \to 0$, so no algorithm that orients $\{i, j\}$ as (i, j) has a positive approximation guarantee. □

The analysis of instance 1 provides an interesting corollary.

Corollary 4.4.1. *Even with just a single edge, inspecting according to the orientation* (i, j) *with the larger Weitzman index* $\sigma_{ij}^{(1)}$ *can be arbitrarily suboptimal.*

Chapter 5

Beyond Bandits: Combinatorial Markov Search

This chapter develops a model which generalizes the bandit processes of Section 2.3. In a bandit process, the only option is to *advance* the bandit. However, there are many situations in which we may have some choice over *how* we advance a bandit.

For instance, consider the nonobligatory extension of the Pandora's Box problem introduced by Doval [2018]. In this setting, the initial single-stage bandit (Pandora's box) is augmented with a second approach, in which the box can be taken without first paying the cost to open it for reward equal to the expected value of the box. We develop the model of *Markov Search Processes* to capture this and other significantly broader generalizations of the Pandora's Box Problem, in which we make dynamic choices beyond simply to advance the bandit.

This chapter introduces the *Combinatorial Markov Search Problem*, equivalent to the Pandora's Box Problem with generalized feasible sets beyond single-selection. CMS models problems such as a company investing in the development of its next product, which may be one of n potential options. Each option is a Markov Search Process involving research and development choices that require resources and may open the door to further choices. The company may interactively investigate and develop the potential options in parallel, e.g. abandoning some when they become less promising, and eventually choosing one to bring to market.

More formally, in the Combinatorial Markov Search (CMS) problem a decisionmaker is given a full description of $\mathcal{M}_1, \ldots, \mathcal{M}_n$. She interacts by repeatedly selecting any $\mathcal{M}_i$, taking an action, paying the incurred cost, and observing its state transition. She then selects another $\mathcal{M}_i$ (or

possibly the same one), and so on. At any point, she may decide to stop and claim some subset of the MSPs, obtaining their associated revealed rewards. The accepted subset F must belong to a given feasibility system $\mathcal{F} \subseteq 2^{\{1,\dots,n\}}$, such as a matroid, and we call the associated problem CMS($\mathcal{F}$). For example, in *single selection*, $\mathcal{F} = \{\emptyset, \{1\}, \dots, \{n\}\}$, denoting that at most one of the MSPs can be claimed. The *welfare* of the algorithm is the sum of rewards claimed minus the sum of all costs incurred. The algorithm is an α-approximation if its expected welfare on every instance is at least an α fraction of the expected welfare of the optimal, fully-adaptive, computationally unconstrained algorithm.

Combinatorial Markov Search appears to require highly interactive, adaptive algorithms. We first ask whether this is truly the case by considering a restricted class of *online* algorithms[1] as approximations to the optimal offline algorithm. These algorithms must fully interact with $\mathcal{M}_i$ and "take it or leave it" before beginning any interactions with $\mathcal{M}_{i+1}$. In the special case of our problem where each $\mathcal{M}_i$ has no actions at all and simply reveals a stochastic reward, existence of an online algorithm with approximation guarantee α is known as an α *prophet inequality.*

Our first main result is that a prophet inequality for feasibility constraint $\mathcal{F}$ can be amplified into an online algorithm for CMS($\mathcal{F}$) with the same approximation guarantee.

Theorem 5.0.1 (Online Approximation)**.** *Let $\mathcal{F}$ be a nonempty downward-closed set system. If there is an ex-ante*[2] *α prophet inequality for $\mathcal{F}$, then there is an online α-approximation algorithm, not necessarily computationally efficient, for Combinatorial Markov Search($\mathcal{F}$).*

For single selection, the optimal prophet inequality is $\frac{1}{2}$ [Krengel and Sucheston, 1978, Samuel-Cahn, 1984], meaning that the "fully-adaptive" algorithm that simply picks the largest of n rewards can perform at most twice as well as an online algorithm. The $\frac{1}{2}$ guarantee extends to matroid feasibility constraints $\mathcal{F}$ even against the ex-ante benchmark [Lee and Singla, 2018], so Theorem 5.0.1 implies that even for matroids, even when all alternatives may be highly dynamic Markov

[1] We emphasize that, as with prophet inequalities, our online algorithms are given the full problem description in advance.

[2] An approximation guarantee is *ex-ante* if the benchmark is a fully-adaptive optimal algorithm that is only required to satisfy the feasibility constraints in expectation. The approximation algorithm is still required to always be feasible. See Section 5.2.2.

Search Processes involving sequences of costly decisions and stochastic transitions, the benefit of sharing information across MSPs is no higher: the benefit of full adaptivity is still only a factor of two. Theorem 5.0.1 is similar in spirit to an *adaptivity gap* in stochastic probing problems (e.g. Gupta et al. [2016]).

The approximation algorithms for CMS in Theorem 5.0.1 are not computationally efficient. By substituting our own "matroid prophet inequality" and designing additional efficient subroutines, we obtain our second main result.

Theorem 5.0.2 (Efficient Approximation)**.** *Let $\mathcal{F}$ be a matroid. There exists an algorithm, running in time polynomial in the input size and $\frac{1}{\epsilon}$, that for any $\epsilon > 0$ achieves a $(\frac{1}{2} - \epsilon)$-approximation for Combinatorial Markov Search($\mathcal{F}$). Moreover, the algorithm is online.*

We thus show that there exists an efficient constant-factor approximation algorithm for CMS, not only for single selection but also for any matroid constraint. Furthermore, the efficient algorithm is "non-adaptive" in that it interacts with the MSPs one at a time online.

Mechanism Interpretation. It is well-known that prophet inequalities are *incentive-compatible (IC)*: they can be converted to mechanisms for e.g. selling items to arriving buyers [Lucier, 2017]. The thresholds used by prophets algorithms are interpreted as posted prices, to which the buyers best-respond, implementing the algorithm.

We define the CMS($\mathcal{F}$)-Auction problem where an $\mathcal{F}$-feasible subset of buyers may be selected as winners, and each buyer interacts privately with their own MSP to discover their value for winning. Notably, our generic reduction in Theorem 5.0.1, despite beginning with an IC prophet inequality, converts it to an algorithm that is *not* IC. However, an essential component of our efficient algorithm for Theorem 5.0.2 is incentive compatibility. The algorithm behaves on each MSP as if it were an agent attempting to maximize the profit from that MSP alone subject to a posted price. Converting the algorithm to a mechanism yields the following result.

Corollary 5.0.1. *If $\mathcal{F}$ is a matroid, there exists a mechanism for the* CMS($\mathcal{F}$)*-Auction problem with Price of Anarchy $\frac{1}{2}$. Furthermore, for any $\epsilon > 0$, there exists a polynomial-time mechanism*

for the CMS($\mathcal{F}$)*-Auction problem with Price of Anarchy* $\frac{1}{2} - \epsilon$.

In subsequent work, Chawla et al. [2025] use a *commitment gap* approach to improve the guarantee to $1 - 1/e$.

5.1 Prophet inequalities

We introduce the tool of *prophet inequlities* for this section, and give a brief section of related work here.

There is a substantial existing literature on prophet inequalities. Krengel and Sucheston [1978] first proposed prophet inequalities, and Samuel-Cahn [1984] first proposed a thresholding algorithm. Various extensions allowing for the selection of multiple items, particularly under matroid constraints, have been studied by Chawla et al. [2010], Yan [2011], Kleinberg and Weinberg [2012], Feldman et al. [2016b], Dutting et al. [2020], and Chawla et al. [2024]. Lee and Singla [2018] used ex ante relaxations to analyze prophet inequalities with matroid selection constraints, a technique we employ heavily. Other extensions of prophet inequalities have also been analyzed. Rubinstein and Singla [2017] studies combinatorial valuations of the items, Ezra et al. [2023] studies a different objective (expected ratio rather than ratio of expectations), and Immorlica et al. [2020b] studies correlated values.

Prophet inequalities have natural applications to incentive-compatible mechanism design, in particular posted price auctions (Chawla et al. [2010] and subsequent literature; see Lucier [2017]). Alaei et al. [2022] and Kleinberg et al. [2016] study connections to descending price auctions. Many of the aforementioned papers discuss these economic connections, and Lucier [2017] provides a survey.

5.2 Additional Models

We now introduce several important models and concepts for this chapter, from Markov Search Processes and the CMS model to ex ante approximation and online approximations.

5.2.1 Markov Search Processes

Our main problem involves interacting with and selecting among n Markov Search Processes (MSPs), each defined as follows.

Definition 5.2.1 (Markov Search Process)**.** A *Markov Search Process* is a tuple $\mathcal{M} = (S, A, P, C, V)$. S is a finite set of states and A a finite set of actions. $P : S \times A \times S \to [0, 1]$ is the *transition function*, where $P(s'; a, s)$ represents the probability of transitioning to state s' given that the process is currently in state s and action a is taken. There is a set of *legal actions* $A^s \subseteq A$ for each state s. For all $a \in A^s$, $\sum_{s' \in S} P(s'; a, s) = 1$. $C : A \times S \to \mathbb{R}_{\geq 0}$ is the *cost function*. $C(a, s)$ is the cost incurred by taking action $a \in A^s$ when the current state is s. Finally, $V : S \to \mathbb{R}_{\geq 0}$ is the *reward function*, representing an available reward at state s.

Given an MSP $\mathcal{M}$, the *state graph* of $\mathcal{M}$ contains a directed edge (s, s') if $P(s'; a, s) > 0$ for some a. We assume the state graph is a directed acyclic graph (DAG). The *start state* of $\mathcal{M}$ is the unique state s^* with no incoming edges. A *terminal state* of $\mathcal{M}$ is a state s with no outgoing edges, i.e. $A^s = \emptyset$. We assume that rewards are only positive on terminal states, i.e., $V(s) = 0$ unless s is a terminal state.

MSPs are a generalization of bandit processes (Definition 2.3.1) with the slight restriction that the support of each signal is finite, translating to the state transitioned to after each advancement. It will be useful throughout to refer to bandit processes as special instances of MSPs. Formally, we reframe a bandit process with finite support for signals at each stage in the MSP setting as follows.

Definition 5.2.2 (Bandit Process)**.** An MSP $\mathcal{M} = (S, A, P, C, V)$ is a *bandit process* if $|A^s| \leq 1$ for all $s \in S$.

In fact, the most general case of CMS that is known to have an efficient optimal algorithm is the case where every $\mathcal{M}_i$ is a bandit and $\mathcal{F}$ is a matroid [Gupta et al., 2019], which follows an index-based approach.[3]

[3] This model closely resembles the Bayesian Multi-Armed-Bandits problem, a discounted infinite-horizon version

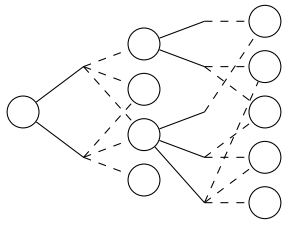


Figure 5.1: MSP Visualization

An example of a general Markov Search Process (MSP). States are given by nodes in the graph. Solid lines represent available actions, while dotted lines represent possible randomized state transitions given the chosen action.

We will update the definitions and terminology used in bandit processes in the rest of this section to capture the complexity of the MSP structure.

Definition 5.2.3 (Combinatorial Markov Search($\mathcal{F}$))**.** Let $\mathcal{F} \subseteq 2^{[n]}$ be downward-closed.[4] An instance $\mathcal{I}$ of the *Combinatorial Markov Search*($\mathcal{F}$) or CMS($\mathcal{F}$) problem consists of a set of n independent MSPs $\mathcal{M}_1, \ldots, \mathcal{M}_n$. Each $\mathcal{M}_i$ begins in its start s_i^*. An algorithm begins with an empty solution set $F = \emptyset$. At any time, the algorithm may either:

(1) Choose any MSP $\mathcal{M}_i$ in a non-terminal state s_i and take some action $a_i \in A_i^{s_i}$. The algorithm incurs cost $C_i(a_i, s_i)$ and $\mathcal{M}_i$ transitions to a new state s_i' drawn independently according to $P_i(\,\cdot\,; a_i, s_i)$.

(2) *Claim* an MSP $\mathcal{M}_i$ in a terminal state s_i, meaning to add i to F. The algorithm receives reward $V_i(s_i)$.

(3) Halt.

In line with out bandit terminology, We refer to either (1) or (2) as *advancing* $\mathcal{M}_i$. The algorithm is *ex-post feasible* if $F \in \mathcal{F}$ with probability 1. Unless otherwise stated, all algorithms are assumed to be ex-post feasible.

of our problem where there is no selection. In this problem, when each alternative is a bandit, the Gittins Index Theorem gives an optimal algorithm. However, if the alternatives are generalized from bandits to general Markov Decision Processes, no approximation algorithm is known [Nash, 1980].

[4] We write $[n]$ for $\{1, \ldots, n\}$. $\mathcal{F}$ is *downward-closed* if $F \in \mathcal{F}, F' \subseteq F \implies F' \in \mathcal{F}$.

Welfare and performance. We update the notation of bandit selection algorithms to the MSP setting. Given an algorithm and an instance $\mathcal{I} = (\mathcal{M}_1, \ldots, \mathcal{M}_n)$, with $\mathcal{M}_i = (S_i, A_i, P_i, C_i, V_i)$, we write $\mathbb{A}_{i,s}$ as the indicator variable for claiming $\mathcal{M}_i$ in state s and $\mathbb{I}_{i,a,s}$ as the indicator for advancing $\mathcal{M}_i$ from s by taking action a. The *performance* of an algorithm on a particular MSP $\mathcal{M}_i$ is the net value, i.e. reward claimed (if any) minus the sum of all costs incurred.

$$\text{Perf}(\mathcal{M}_i) = \sum_{s \in S_i} \mathbb{A}_{i,s} V_i(s) - \sum_{s \in S_i, a \in A_i} \mathbb{I}_{i,a,s} C_i(a, s).$$

The *welfare* of the algorithm on the instance is

$$\text{Welf}(\mathcal{I}) = \sum_{i=1}^{n} \text{Perf}(\mathcal{M}_i).$$

We sometimes write Perf^{ALG} and Welf^{ALG} to clarify reference to a particular algorithm ALG. We may similarly write $\mathbb{A}^{\text{ALG}}_{i,s}$ and $\mathbb{I}^{\text{ALG}}_{i,a,s}$. We write $\mathbb{A}_i := \sum_{s \in S_i} \mathbb{A}_{i,s}$.

The objective in CMS is to maximize expected welfare. An algorithm ALG is an α-approximation if, for all algorithms OPT and all instances $\mathcal{I}$, $\mathbb{E}[\text{Welf}^{\text{ALG}}(\mathcal{I})] \geq \alpha\, \mathbb{E}[\text{Welf}^{\text{OPT}}(\mathcal{I})]$, with expectation taken over randomness of the instance's transitions and the algorithms themselves.

Our algorithms, though given the entire problem description in advance, will interact *online*.

Definition 5.2.4 (Online, Fully-Adaptive)**.** An algorithm is *online* if, for each $i = 1, \ldots, n$, the algorithm must complete all interactions with $\mathcal{M}_i$ and decide irrevocably whether to claim $\mathcal{M}_i$ or not before it can take any action in $\mathcal{M}_{i+1}$. An algorithm that is not required to be online is *fully-adaptive*. In either case, the full description of each $\mathcal{M}_i$ is given.

5.2.2 Ex-ante CMS and Policies

We will compare our algorithms, which are always ex-post feasible, to a stronger benchmark: the optimal algorithm for CMS that is only required to produce a feasible set "in expectation."

Definition 5.2.5 ($\mathcal{P}_{\mathcal{F}}$, ex-ante feasible)**.** Given $F \subseteq [n]$, let $1_F \in [0,1]^n$ denote the indicator vector for F, i.e. $1_F(i) = \mathbb{1}[i \in F]$. Given $\mathcal{F} \subseteq 2^{[n]}$, define $\mathcal{P}_{\mathcal{F}}$ to be the convex hull of $\{1_F : F \in \mathcal{F}\}$.

An algorithm OPT is termed *ex-ante feasible* if the vector Q defined by $Q_i = \Pr[i \in \text{OPT}]$ satisfies $Q \in \mathcal{P}_{\mathcal{F}}$. The probability is taken over both the algorithm's coins and the random transitions.

We also refer to $\mathcal{P}_{\mathcal{F}}$ as the *feasible polytope*. For example, in single selection, $\mathcal{P}_{\mathcal{F}}$ consists of nonnegative vectors whose entries sum to at most one. In that case, an example of an ex-ante feasible algorithm that is not ex-post feasible is one that claims each $\mathcal{M}_i$ with probability $\frac{1}{n}$ independently.

We say an algorithm ALG *is an ex-ante α-approximation* if, for all ex-ante feasible algorithms OPT and all instances $\mathcal{I}$, $\mathbb{E}[\text{Welf}^{\text{ALG}}(\mathcal{I})] \geq \alpha\, \mathbb{E}[\text{Welf}^{\text{OPT}}(\mathcal{I})]$. Because every ex-post feasible OPT is also ex-ante feasible, this implies that ALG simply guarantees an α-approximation as well.

Policies and independence. We use the term *policy* for a "local algorithm" that is defined on a single MSP $\mathcal{M}_i = (S_i, A_i, P_i, C_i, V_i)$. Formally, a policy ρ is a possibly-randomized function from a *history* $h = (s^*, a_1, s_1, \ldots, a_t, s_t)$ of states and actions in $\mathcal{M}_i$ to either "halt", "claim", or a legal next action $\rho(h) = a \in A^{s_t}$. When executing a policy ρ on an instance $\mathcal{M}_i$, we define the indicators $\mathbb{A}^{\rho}_{i,s}$ and $\mathbb{I}^{\rho}_{i,a,s}$ and the performance $\text{Perf}^{\rho}(\mathcal{M}_i)$ exactly as for algorithms above. We typically use π to denote a deterministic policy and ρ for a possibly-randomized one. When a policy depends only on the current state s, we write $\rho(s)$ rather than $\rho(h)$.

The following observation is crucial to our techniques throughout: the ex-ante benchmark is in a sense "non-adaptive", as our online algorithms will be.

Observation 5.2.1. *Any ex-ante feasible algorithm for* CMS*, without loss of generality, consists of n policies $\rho_1, \ldots, \rho_n$ that are executed independently on $\mathcal{M}_1, \ldots, \mathcal{M}_n$, respectively.*

Proof. Let ALG be any ex-ante feasible algorithm. For each i, define ρ_i to be the following policy on $\mathcal{M}_i$: run ALG on $\mathcal{M}_i$ together with simulated independent copies of $(\mathcal{M}_{i'} : i' \neq i)$. The algorithm that executes each ρ_i independently on $\mathcal{M}_i$ has the same probability of claiming each $\mathcal{M}_i$ as ALG does, so it is ex-ante feasible, and it also has the same expected welfare as ALG. □

We will use the above observation to analyze, in particular, the behavior of any ex-ante optimal algorithm for CMS.

Ex-Ante Relaxation. Recall that the set of MSPs we may select is constrained, typically according to a matroid constraint. We compare our *ex-post* feasible algorithm to the optimal *ex-ante* feasible algorithm (Definition 5.2.5) as a benchmark. The ex-ante benchmark may interact with each MSP independently, subject to some fixed probability of claiming. As any ex-post feasible selection is also ex-ante feasible, this provides an upper bound on the value of any optimal algorithm for ex-post CMS.

5.2.3 MSP Amortization

Our costly information results thus far have been obtained by directly leveraging generalizations of Lemma 2.2.1 to amortize and analyze algorithms or strategies. It is not clear how to amortize a policy on an MSP. However, we will adjust the definitions of Section 2.3.2 to match our new MSP notation for bandits.

Definition 5.2.6 (MSP Weitzman Index, Capped Value (following Section 2.3))**.** Given a bandit process $B = (S, A, P, C, V)$, the *Weitzman index* σ_s and the *capped value* κ_s of each state s are defined by backwards induction as follows. For any terminal state s, $\sigma_s = \kappa_s = V(s)$. For any non-terminal state s, let the random variable $n(s)$ be the next state reached when advancing from s. Define σ_s to be the unique value satisfying

$$\mathop{\mathbb{E}}_{n(s)}[(\kappa_{n(s)} - \sigma_s)^+] = C(s)$$

where $(\,\cdot\,)^+ = \max\{\cdot, 0\}$ and $C(s)$ is the cost of taking the unique "advance" action from s. We define $\kappa_s := \min\{\sigma_s, \kappa_{n(s)}\}$. We also write $\kappa = \kappa_{s^*}$ where s^* is the initial state of the process.

We emphasize that κ_s, $n(s)$ are random variables, while σ_s is a constant which can be computed prior to the realization of any randomness.

We rephrase the concepts of exposure and the key Lemma 2.2.1 here in the language of MSPs, for convenience.

Definition 5.2.7 (Exposure, Rephrasing of Definition 2.3.3)**.** A policy ρ is *exposed* on a bandit B if, with nonzero probability, ρ advances from some state s, eventually visits some state s' with $\sigma_{s'} > \sigma_s$, and does not advance from s'. Otherwise, ρ is *non-exposed*.

Lemma 5.2.1 (Rephrasing of Lemma 2.2.1)**.** *For any bandit process B and policy ρ, $\mathbb{E}[\mathrm{Perf}^\rho(B)] \leq \mathbb{E}[\mathbb{A}^\rho\kappa]$, with equality if and only if ρ is non-exposed on B.*

5.2.4 Online Selection & Prophet Inequalities

The core technique of this chapter is leveraging constant-factor approximations given by prophet inequalities to obtain the same approximation factor for CMS. In the classic prophet inequality setting, a decisionmaker must select items from a set of many alternatives of uncertain value. Simple thresholding techniques allow us to approximate the optimal solution when items must be considered one at a time [Samuel-Cahn, 1984]. These techniques are robust to replacing the random variables that the decisionmaker chooses between with Pandora's Boxes [Kleinberg et al., 2016]. We show in Section 5.3 that these techniques extend even when we replace these alternatives with MSPs, which are much more complex objects. We do this via a chain of reductions from CMS to two other novel settings, which we call *choice-over-bandits* and *choice-over-rewards*, and finally to the original prophet inequality setting.

In Section 5.4 we show how to overcome inefficiencies in these reductions, and arrive at an efficient constant-factor approximation (Theorem 5.0.2).

5.3 Reducing Combinatorial Markov Search to Prophet Inequalities

In this section, we prove Theorem 5.0.1: whenever the constraint system $\mathcal{F}$ admits an ex-ante prophet inequality, the CMS($\mathcal{F}$) problem admits an online algorithm with the same approximation factor, albeit inefficiently. A prophet inequality bounds the advantage of being able to select a subset of random rewards "offline" versus needing to select them online. Theorem 5.0.1 states that even when rewards are replaced by arbitrary Markov Search Processes, and the offline algorithm may explore the MSPs concurrently, its advantage is no greater.

Our approach consists of a chain of three reductions to simpler versions of the CMS problem where the Markov Search Processes $\mathcal{M}_1, \ldots, \mathcal{M}_n$ are replaced with progressively simpler structures. Our first reduction replaces MSPs with *Choice-Over-Bandits* processes, in which each alternative i consists of picking one bandit process, then sticking with that choice. We then reduce Choice-Over-Bandits processes to *Choice-Over-Rewards* in which each alternative i consists of picking one random reward to reveal out of several hidden options. Finally, we reduce Choice-Over-Rewards to random rewards, the setting of prophet inequalities.

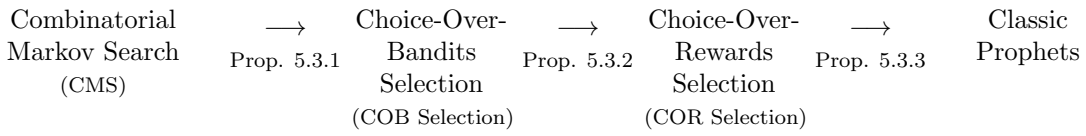


Figure 5.2: Reduction Roadmap for CMS to Prophet Inequality

Given any downward-closed feasibility constraint and ex-ante α prophet inequality, we obtain an online α-approximation algorithm for CMS via a series of reductions.

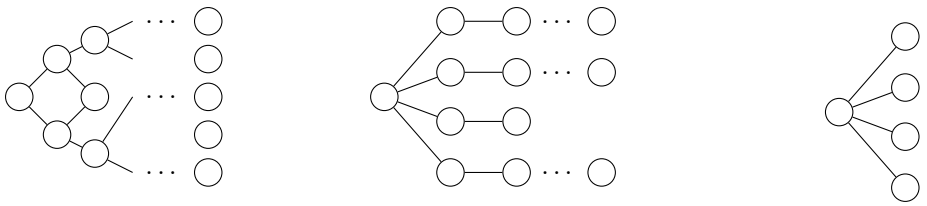
5.3.1 Reduction 1: Markov Search Process to Choice-Over-Bandits

We begin with the key and most complicated reduction, replacing MSPs with Choice-Over-Bandits processes. First, we define a *Choice-Over-Bandits* process and the *Choice-Over-Bandits Selection* problem.

Definition 5.3.1 (Choice-Over-Bandits)**.** A *Choice-Over-Bandits* (COB) process $\mathcal{C}$ consists of m bandit processes $B^1, \ldots, B^m$ for some $m \geq 1$. An algorithm initially takes a costless action corresponding to some $j \in [m]$. From there, the algorithm interacts with the bandit process B^j, incurring all transition costs, eventually halting, and possibly claiming a reward. We allow the random transitions of $B^1, \ldots, B^m$ to be arbitrarily correlated, i.e. to access shared randomness.[5]

The COB Selection($\mathcal{F}$) problem is defined as the Combinatorial Markov Search($\mathcal{F}$) problem where each MSP $\mathcal{M}_i$ is replaced by a COB $\mathcal{C}_i$. All other definitions, such as welfare and performance,

[5] Allowing shared randomness will simplify our reduction later. We note that a COB process is not technically a special case of an MSP which, for simplicity, was defined with independent randomness on all transitions. However, the definition of MSP could be broadened to allow shared randomness without affecting any of our results, at the cost of a more complex definition.



(a) MSP (b) COB (c) COR

Figure 5.3: CMS Reduction Structures

MSP**,** COB**, and** COR**.** A simplified representation of the difference between the action structures used in our reductions, ommitting the randomness of transitions (shown in Figure 5.1 as dashed lines). Actions advance left to right, with vertices representing states and edges representing actions. (5.3a) A full MSP, where costly actions may form any DAG. (5.3b) A Choice-Over-Bandits structure, where after an inital action with cost 0, each state has only one legal action. (5.3c) A Choice-Over-Rewards structure, where each initial costless action leads to a random reward.

are unchanged.

The reduction from CMS to COB comprises the core of the proof of Theorem 5.0.1, and we elaborate here on the COB Selection problem individually.

The COB Selection problem restricts the number of choices available in any MSP to only one initial choice: which bandit to follow? This problem alone is an interesting generalization of of the Pandora's Box model, capturing nonobligatory inspection settings (e.g. Doval [2018]) and matching settings, with appropriate selection of feasibility structures [Bowers and Waggoner, 2026]. The COB Selection setting introduces some level of choice over the evolution of each process for the decisionmaker, without allowing arbitrary sets of choices over each MSP. This change is sufficient to immediately moves simple single-selection feasibility structures from computationally feasible to NP-hard [Fu et al., 2023].

Intuitively, our reduction from an MSP to a COB "detangles" the paths through the MSP, converting any sequence of choices over actions to a single choice of which bandit to pursue (see Figure 5.3). The key to this detangling process is the idea of *induced bandits* (Definition 5.3.2), which allows us to produce a separate bandit for every deterministic policy on a give MSP, and then choose over those bandits to produce a COB instance.

Definition 5.3.2 (Induced bandit)**.** Given an MSP $\mathcal{M}_i$ and a deterministic policy π on $\mathcal{M}_i$, define the *induced bandit process* B_i^π to be the bandit MSP that mimics following π on $\mathcal{M}_i$. More formally, states in B_i^π consist of histories h in $\mathcal{M}_i$ reachable by π; the only legal action at each non-terminal h is "continue"; and continuing at a non-terminal h has cost and transition probabilities equal to taking the action $\pi(h)$ in $\mathcal{M}_i$. If $\pi(h)$ halts or claims, then h is a terminal state in B with reward zero or $V(s)$ respectively, where s is the final state in history h.

Reduction 1 (MSPtoCOB)**.** Given an instance $\mathcal{I} = (\mathcal{M}_1, \ldots, \mathcal{M}_n)$ of CMS($\mathcal{F}$), the MSPtoCOB($\mathcal{I}$) reduction constructs an instance $\mathcal{I}'$ of COB Selection($\mathcal{F}$) as follows. For each MSP $\mathcal{M}_i$ in $\mathcal{I}$, create a Choice-Over-Bandits process $\mathcal{C}_i$ in $\mathcal{I}'$. Its choices consist of the bandit processes B_i^π induced by every deterministic policy π on $\mathcal{M}_i$ (of which there are finitely many).

We note that, for this reduction, it is necessary in the definition of COB that the bandits may have correlated transitions.

Lemma 5.3.1. *Let $\mathcal{I}$ be an instance of* CMS *and $\mathcal{I}'$ =*MSPtoCOB($\mathcal{I}$)*. For any policy ρ' on $\mathcal{C}_i$, there exists a policy ρ on $\mathcal{M}_i$ such that* $\text{Perf}^\rho(\mathcal{M}_i) = \text{Perf}^{\rho'}(\mathcal{C}_i')$ *and the probabilities that ρ and ρ' claim are the same. Furthermore, given any policy ρ on $\mathcal{M}_i$, there exists a policy ρ' on $\mathcal{C}_i$ with the same guarantee.*

Sketch (proof in Section F). ρ' first selects one of the bandits B_i^π in $\mathcal{C}_i$. At each subsequent step, ρ' either advances B_i^π, claims $\mathcal{C}_i$, or halts. We define ρ to always make the same choice in $\mathcal{M}_i$, i.e. $\pi(h)$ at the current history h, claim, or halt. By construction, the choice is always legal in $\mathcal{M}_i$ at the current history h. The costs incurred and reward claimed, if any, are the same.

To prove the converse, we first observe that a randomized policy ρ for $\mathcal{M}_i$, WLOG, first draws a deterministic policy π from some probability distribution and then follows it. Define ρ' to draw π from this distribution and choose the bandit B_i^π, always advancing unless at a terminal state with reward zero, in which case ρ' halts. Then ρ' simulates the behavior of ρ on $\mathcal{M}_i$. □

Proposition 5.3.1 (CMS to COB Selection)**.** *Let $\mathcal{F}$ be downward-closed. If there exists an online ex-ante α-approximation algorithm* ALG$'$ *for* COB Selection($\mathcal{F}$)*, then there exists an online ex-ante α-approximation algorithm* ALG*, not necessarily efficient, for* CMS($\mathcal{F}$)*.*

Sketch (proof in Section F). Lemma 5.3.1 implies the following claim: any algorithm for COB Selection that consists of executing a policy on $\mathcal{C}_1$, then a policy on $\mathcal{C}_2$, and so on, can be converted to an algorithm for CMS with the same approximation factor. The reverse is also true. In particular, this claim applies to the online algorithm ALG$'$. Also, by Observation 5.2.1, the optimal ex-ante feasible algorithms OPT$'$ for COB Selection and OPT for CMS consist WLOG of collections of independent policies for each alternative, so the above claim applies to OPT and OPT$'$ as well. □

Note that the number of deterministic policies on an MSP $\mathcal{M}$ is at least exponential in the size of the state graph of $\mathcal{M}$, so the above reduction is not efficient. In Section 5.4, we will use the same reduction for analysis, but give an algorithm that efficiently selects a policy without constructing the reduction explicitly.

5.3.2 Reduction 2: Choice-Over-Bandits to Choice-Over-Rewards

We next reduce to the simpler case where each alternative i is a *Choice-Over-Rewards*.

Definition 5.3.3 (Choice-Over-Rewards)**.** A *Choice-Over-Rewards* (COR) process $\mathcal{C}_i$ consists of m random variables $X_i^1, \ldots, X_i^m$ for some $m \geq 1$. An algorithm initially chooses $j \in [m]$ and observes X_i^j. The algorithm then may decide either to discard i; or to claim i, obtaining reward X_i^j. We allow $X_i^1, \ldots, X_i^m$ to be arbitrarily correlated with each other (but independent of any $X_{i'}^j$ for $i' \neq i$).

The COR Selection($\mathcal{F}$) problem is defined as the Combinatorial Markov Search($\mathcal{F}$) problem where each MSP $\mathcal{M}_i$ is replaced by a COR $\mathcal{C}_i$. The performance on $\mathcal{C}_i$ is simply the reward X_i^j claimed at i if any; and the welfare is the sum of claimed rewards.

Here, we may picture each i as a "cabinet" consisting of m "drawers", inside each of which

is a random prize.[6] An algorithm may only open one of the drawers on each cabinet.[7]

Our next reduction is from Choice-Over-Bandits to Choice-Over-Rewards. Our reduction amortizes the transition costs of the bandit processes using techniques from the Pandora's Box literature [Kleinberg et al., 2016, Bowers and Waggoner, 2026]. Essentially, we condense each bandit process into a representative index representing an expected overall reward for continuing to advance the bandit. This indexing allows us to make an initial decision over "rewards" summarizing the expected benefit of each bandit using a Choice-Over-Rewards algorithm, and combining this with known bandit approaches we obtain an algorithm for Choice-Over-Bandits.

Reduction 2 (COBtoCOR)**.** Given a Choice-Over-Bandits $\mathcal{C}_i$ consisting of bandits $B_i^1, \ldots, B_i^m$, we let the Choice-Over-Rewards instance $\mathcal{C}_i'$ consist of random rewards $\kappa_i^1, \ldots, \kappa_i^m$ where κ_i^j is the capped value of B_i^j (Definition 5.2.6). Given an instance $\mathcal{I} = (\mathcal{C}_1, \ldots, \mathcal{C}_n)$ of COB Selection($\mathcal{F}$), the COBtoCOR($\mathcal{I}$) reduction constructs an instance $\mathcal{I}'$ of COR Selection($\mathcal{F}$) consisting of $(\mathcal{C}_1', \ldots, \mathcal{C}_n')$.

We use the fact that online algorithms for COR Selection are, WLOG, "quantile-based". This is related to the the "monotone" property of prophet-inequality algorithms Kleinberg and Weinberg [2012].

Definition 5.3.4 (Quantile-based)**.** An algorithm for COR Selection is *quantile-based* if, for each alternative i and drawer j, there exists $q_i^j \in [0, 1]$ such that, conditioned on the algorithm opening drawer j on alternative i, the algorithm claims i if and only if X_i^j is in its top q_i^j quantile of realizations.[8]

Observation 5.3.1. *For any online algorithm for* COR Selection*, there exists a quantile-based online algorithm with at least as large expected performance and the same feasibility guarantee.*

Proof. For any online algorithm which claims X_i^j with probability q_i^j, we can maintain feasibility while moving all probability of claiming to the top q_i^j quantile of reward realizations. Because

[6] Thanks to Bobby Kleinberg for giving this metaphor for the problem.

[7] Thus, the COR Selection problem is a type of stochastic probing problem. We have not found a reference to it specifically in the literature.

[8] For a discrete random variable, this may involve a randomized decision; for example, given a Bernoulli($\frac{1}{2}$) reward and $q_i^j = 0.25$, the reward should be taken half of the time when its value is 1 and never when its value is 0.

the algorithm is online, the criteria for selection does not affect our selection of any preceding alternatives. And as the probability of selecting i remains the same, following alternatives can be claimed with the same probability. □

We now complete this portion of the reduction exactly as the previous portion: showing that policies for COR can be converted to policies for COB, then "lifting" this guarantee to algorithms that consist of running policies on the alternatives one at a time.

Lemma 5.3.2. *Let $\mathcal{I}$ be an instance of* COB Selection *and $\mathcal{I}'$ =*COBtoCOR($\mathcal{I}$)*. For any policy ρ' on $\mathcal{C}_i'$, there exists a policy ρ on $\mathcal{C}_i$ such that* $\text{Perf}^{\rho}(\mathcal{C}_i) \geq \text{Perf}^{\rho'}(\mathcal{C}_i')$ *and the probabilities that ρ and ρ' claim are the same. Given any policy ρ on $\mathcal{C}_i$, the reverse is also true.*

Proof. Given policy ρ' on the Choice-Over-Rewards $\mathcal{C}_i'$, by Observation 5.3.1, WLOG it is quantile-based with parameters $\{q_i^j\}$. Define ρ on the Choice-Over-Bandits $\mathcal{C}_i$ to initially make the same choice j as ρ'. Here, treat j as a random variable, the choice. Let ρ be the non-exposed policy on bandit B_i^j that claims if κ_i^j is in its top q_i^j quantile of realizations; it is constructed in Lemma F.0.1. By non-exposure and Lemma 5.2.1, $\mathbb{E}[\text{Perf}^{\rho}(\mathcal{C}_i)] = \mathbb{E}[\mathbb{A}_i^{\rho}\kappa_i^j] = \mathbb{E}[\mathbb{A}_i^{\rho'}X_i^j] = \mathbb{E}[\text{Perf}^{\rho'}(C_i')]$.

For the reverse direction, given policy ρ on the COB $\mathcal{C}_i$, let q_i^j be the probability with which it claims conditioned on making choice j. Define ρ' to make the same choice j as ρ, then claim if X_i^j is in its upper q_i^j quantile. Because $X_i^j = \kappa_i^j$ in distribution and by definition of q_i^j, we have $\mathbb{E}[\mathbb{A}_i^{\rho'}X_i^j] \geq \mathbb{E}[\mathbb{A}_i^{\rho}\kappa_i^j]$. Then, using Lemma 5.2.1, $\mathbb{E}[\text{Perf}^{\rho'}(\mathcal{C}_i')] = \mathbb{E}[\mathbb{A}_i^{\rho'}X_i^j] \geq \mathbb{E}[\mathbb{A}_i^{\rho}\kappa_i^j] \geq \mathbb{E}[\text{Perf}^{\rho}(\mathcal{C}_i)]$. □

Proposition 5.3.2 (COB Selection to COR Selection)**.** *Let $\mathcal{F}$ be downward-closed. If there exists an online, ex-ante α-approximation for* COR Selection($\mathcal{F}$)*, there exists an online, ex-ante α-approximation for* COB Selection($\mathcal{F}$)*.*

Proof. Exactly analogous to Proposition 5.3.1, substituting Lemma 5.3.2 for Lemma 5.3.1. □

5.3.3 Reduction 3: Choice-Over-Rewards to Prophets

Finally, we reduce from COR Selection($\mathcal{F}$) to prophet inequalities, which address the online version of the following problem.

Definition 5.3.5 (Variable Selection)**.** An instance of the Variable Selection($\mathcal{F}$) problem consists of a set $X = (X_1, \ldots, X_n)$ of independent random variables. The algorithm's goal is to select the set $F \in \mathcal{F}$ maximizing $\sum_{i \in F} X_i$.

An online ex-ante α-approximation for Variable Selection($\mathcal{F}$) is called an *ex-ante α prophet inequality.*

This result relies on observations of the structure of the optimal ex-ante algorithm for the COR Selection problem. We observe that the optimal ex-ante COR Selection algorithm induces a distribution over rewards chosen for each Choice-Over-Rewards structure, independently and uncorrelated to the realization of each reward. Furthermore, the ex-ante algorithm's probability of selecting each reward can be computed and combined with the realizations of each reward to simulate a Prophets instance with a single reward.

Proposition 5.3.3 (COR to Variable Selection)**.** *Let $\mathcal{F}$ be a non-empty, downward-closed set system. If there exists an ex-ante α prophet inequality for $\mathcal{F}$, there exists an online ex-ante α-approximation for* COR Selection($\mathcal{F}$)*.*

Proof. The optimal ex-ante feasible algorithm OPT for COR Selection($\mathcal{F}$) induces some $Q \in \mathcal{P}_{\mathcal{F}}$ where Q_i is the probability of claiming $\mathcal{M}_i$. Given Q_i, WLOG by Observation 5.2.1, the selection decision on $\mathcal{C}_i$ is statistically independent of all actions on other alternatives $\mathcal{C}_{i'}$. Therefore, also WLOG, OPT decides which drawer $j^*(i)$ to open on $\mathcal{C}_i$ independently according to some distribution $\lambda_i \in \Delta_{[m_i]}$.

Let Y_i be a random variable distributed as $Y_i = X_i^{j^*(i)}$, or in other words, $Y_i = X_i^j$ with probability $\lambda_i(j)$. The Y_is are independent, and $\text{Welf}^{\text{OPT}}(X) = \text{Welf}^{\text{OPT}'}(Y)$, where OPT' is the optimal ex-ante feasible algorithm for Variable Selection($\mathcal{F}$).

Given a prophet inequality, i.e. given an online ex-ante α-approximation ALG$'$ for Variable Selection($\mathcal{F}$), and given $\Lambda = (\lambda_1, \ldots, \lambda_n)$, we construct an online algorithm ALG for COR Selection($\mathcal{F}$). We instantiate ALG$'$ on the distributions of $Y_1, \ldots, Y_n$. On arrival i, we open drawer j with probability $\lambda_i(j)$. We set $Y_i = X_i^j$ and claim alternative i if and only if ALG$'$ claims on Y_i. Our algorithm ALG is ex-post feasible because ALG$'$ is, and its performance on each arrival equals that of ALG$'$. So

$$\mathbb{E}[\text{Welf}^{\text{ALG}}(X)] = \mathbb{E}[\text{Welf}^{\text{ALG}'}(Y)] \geq \alpha \cdot \mathbb{E}[\text{Welf}^{\text{OPT}'}(Y)] = \alpha \cdot \mathbb{E}[\text{Welf}^{\text{OPT}}(X)].$$

□

Proof of Theorem 5.0.1. Let $\mathcal{F}$ be downward-closed. By Proposition 5.3.3, if there is an ex-ante α prophet inequality for $\mathcal{F}$, then there is an online ex-ante α-approximation algorithm for COR Selection($\mathcal{F}$). By Proposition 5.3.2, there is therefore one for COB Selection($\mathcal{F}$), and by Proposition 5.3.1, therefore one for CMS($\mathcal{F}$). □

Lee and Singla [2018] provide a $\frac{1}{2}$-approximation ex-ante prophet inequality for matroids: it follows that there exists a $\frac{1}{2}$ approximate online algorithm for CMS($\mathcal{F}$). They additionally provide a $\left(1 - \frac{1}{e}\right)$-approximation ex-ante prophet inequality for matroids when arrival order is stochastic, which translates to the same guarantee for CMS($\mathcal{F}$) as well.

5.4 Efficient Approximation Algorithm

In this section, we prove Theorem 5.0.2: existence of an **efficient** $(\frac{1}{2} - \epsilon)$-approximation algorithm for CMS($\mathcal{F}$), for every $\epsilon > 0$, whenever $\mathcal{F}$ forms a matroid. To do so, we will define algorithms of a particular form. They turn out to be *incentive-compatible* in the mechanism design sense (see Subsection 5.4.3), and incentive-compatibility will be crucial to achieving an efficient approximation algorithm.

5.4.1 Threshold-Based Algorithms

Traditional prophet inequality algorithms often set a fixed threshold for each arriving random reward, accepting the reward if and only if it exceeds the threshold. In mechanism-design settings, we can view the threshold as a posted price. The arrival is an agent with a random value for "winning" in the auction, i.e., being included in the final solution $F \subseteq [n]$. The agent chooses to "buy", i.e. be included in F, if and only if her value exceeds the posted price.

Our reductions in Section 5.3 converts such prophet inequalities into algorithms that are not incentive-compatible[9] nor computationally efficient. We will address both problems by defining *threshold-based* algorithms in the Combinatorial Markov Search setting. An alternative $\mathcal{M}_i$ is viewed as an agent who interacts with a Markov Search Process to discover their value for "winning" in the auction. Given a threshold τ, the agent optimally explores $\mathcal{M}_i$ and decides whether or not to "buy" the revealed reward for price τ. This exploration may involve abandoning the search at any point if the expected utility for continuing becomes negative given that the final reward will cost τ to claim. We emphasize that, for now, the agent is merely a thought experiment used to define our algorithm.

Definition 5.4.1 (SAUP, threshold-based)**.** Given a single Markov Search Process $\mathcal{M}$ and a threshold τ, the *Single-Agent Utility Problem (SAUP)* is to compute a policy π on a given MSP $\mathcal{M}$ that maximizes $\mathbb{E}[\text{Perf}^{\pi}(\mathcal{M}) - \mathbb{A}^{\pi}\tau]$. An algorithm for Combinatorial Markov Search is *threshold-based* if, for each arrival $\mathcal{M}_i$, it computes a threshold τ_i based on the problem instance and on previous decisions, then solves $\text{SAUP}(\mathcal{M}_i, \tau_i)$.

For example, suppose $\mathcal{M}_i$ consists of a Pandora's box, i.e. there is a single action with cost c_i that leads to a stochastic reward. In this case, $\text{SAUP}(\mathcal{M}_i, \tau_i)$ is the following policy: if τ_i is greater than the Weitzman index σ_i of the box, then discard the box immediately without paying to open it; otherwise, open the box, discard the reward if it is smaller than τ_i, and otherwise claim

[9] The reduction in Section 5.3 takes a threshold-based prophet inequality, computes the probability q_i with which it claims an arrival i, then selects a policy π for $\mathcal{M}_i$ randomly from a certain distribution chosen to maximize "utility" subject to claiming with probability q_i.

the reward. We now show that SAUP can be computed efficiently for a generic Markov Search Process. A similar result was derived independently by Chawla et al. [2026].

Proposition 5.4.1. *There is a polynomial-time algorithm for SAUP$(\mathcal{M}, \tau)$.*

The algorithm computes a policy π^* by backward induction as follows.

- At a terminal state s: π^* claims if $V(s) \geq \tau$, else halts. Let $\text{Val}^{\pi^*,s} := (V(s) - \tau)^+$.
- At a non-terminal state s: Let $p_{a,s} := P(\,\cdot\,; a, s)$ be the probability distribution over the next state when taking action a. Define $v^{\pi^*}(a; s) := \mathbb{E}_{s' \sim p_{a,s}}[\text{Val}^{\pi^*,s'}] - C(a, s)$. Let $a^* = \arg\max_a v^{\pi^*}(a; s)$ and set $\pi^*(s) = a^*$ unless $v^{\pi^*}(a^*; s) \leq 0$, in which case $\pi^*(s) =$ halt. Let $\text{Val}^{\pi^*,s} = 0$ if $\pi^*(s) =$ halt, else $\text{Val}^{\pi^*,s} = v^{\pi^*}(a^*, s)$.

Intuitively, at state s, the algorithm inductively calculates a policy for each possible successor of s, as well as its expected reward, and selects the action a^* with the highest expected reward, then takes action a^* or halts if its expected reward is nonpositive.

Proof of Proposition 5.4.1. For any deterministic policy π, consider running π on $\mathcal{M}$ starting from state s. Let $\mathbb{A}^{\pi,s}$ be the indicator that π claims and let $\text{Perf}^{\pi,s}$ be the performance. The objective for SAUP$(\mathcal{M}, \tau)$ is $\mathbb{E}[\text{Perf}^{\pi,s} - \mathbb{A}^{\pi,s}\tau]$. Observe that for $\pi = \pi^*$, this is equal to $\text{Val}^{\pi^*,s}$. We show $\text{Val}^{\pi,s} \leq \text{Val}^{\pi^*,s}$ for all s by backward induction. At terminal states s, $\text{Val}^{\pi,s} \leq (V(s) - \tau)^+ = \text{Val}^{\pi^*,s}$. At non-terminal states s, if $\text{Val}^{\pi,s} \leq 0$ then $\text{Val}^{\pi,s} \leq \text{Val}^{\pi^*,s}$ immediately. Otherwise:

$$\begin{aligned}
\text{Val}^{\pi,s} &= \mathop{\mathbb{E}}_{s' \sim p_{\pi(s),s}}[\text{Val}^{\pi,s'}] - C(\pi(s), s) \\
&\leq \mathop{\mathbb{E}}_{s' \sim p_{\pi(s),s}}[\text{Val}^{\pi^*,s'}] - C(\pi(s), s) && \text{inductive hypothesis} \\
&\leq \mathop{\mathbb{E}}_{s' \sim p_{\pi^*(s),s}}[\text{Val}^{\pi^*,s'}] - C(\pi^*(s), s) && \text{construction of } \pi^* \\
&= \text{Val}^{\pi^*,s}.
\end{aligned}$$

In particular, when s is the start state, this proves that π^* is optimal among deterministic policies. Any randomized algorithm can be written as a distribution over deterministic policies, so π^* is optimal among all policies.

For each state s of $\mathcal{M}$ and for each action $a \in A^s$, the algorithm performs a polynomial-time computation to determine $v^{\pi^*}(a, s)$. Therefore, the algorithm runs in polynomial time in the size of the state graph of $\mathcal{M}$. □

5.4.2 A Matroid "Prophet" Algorithm

We design an online threshold-based algorithm. It has two components: computing the threshold τ_i for each arrival $\mathcal{M}_i$, and running SAUP($\mathcal{M}_i, \tau_i$) to interact with $\mathcal{M}_i$ and determine whether to claim it. We have already given a polynomial-time algorithm for SAUP($\mathcal{M}_i, \tau_i$) in Proposition 5.4.1, so we now turn to computing the thresholds.

Computing thresholds and approximating ex-ante OPT.. Matroid prophet algorithms such as in Kleinberg and Weinberg [2012], Lee and Singla [2018] require knowledge of the optimal fully-adaptive solution in order to compute thresholds. While we do not know how to compute or even approximate OPT in our setting, we will approximate the ex-ante feasible OPT, which is an upper bound on the ex-post feasible OPT.

Approximation, however, presents an additional challenge because existing algorithms of Kleinberg and Weinberg [2012], Lee and Singla [2018] cannot obviously be adapted to work when OPT is only approximated.[10] Our algorithm addresses this in part by requiring only a single call to an approximation algorithm for the ex-ante feasible OPT. We then base all of the algorithm's thresholds on this solution. More precisely, we require the approximation algorithm to output the following.

Definition 5.4.2 (ϵ-approximate collection)**.** Given an input $\mathcal{M}_1, \ldots, \mathcal{M}_n$ to the ex-ante CMS($\mathcal{F}$) problem, an *ϵ-approximate collection* is a pair of vectors $(Q, \hat{U})$ satisfying the following:

(1) $Q \in \mathcal{P}_{\mathcal{F}}$, the feasible polytope.

(2) there exists an ex-ante feasible algorithm $\widehat{\text{OPT}}$ that is a $1 - \epsilon$ approximation and satisfies $Q_i \geq \Pr[i \in \widehat{\text{OPT}}]$ for all i.

[10] The thresholds in these algorithms are based on differences in OPT across different inputs, so an approximation guarantee for OPT does not immediately translate to any approximation guarantee for those algorithms.

(3) letting U_i be the expected utility $\widehat{\text{OPT}}$ obtains on $\mathcal{M}_i$, $\hat{U}$ approximates U from below, i.e. $\hat{U}_i \leq U_i$ for all i and $\sum_i \hat{U}_i \geq (1-\epsilon)\sum_i U_i$.

We can think of $(Q, \hat{U})$ as a witness for the existence of an approximately-optimal ex-ante feasible algorithm. We allow $\hat{U}$ to be approximate because it may be computationally intractable to compute U exactly.

Proposition 5.4.2. *When $\mathcal{F}$ is a matroid, there is an algorithm for ex-ante* CMS($\mathcal{F}$) *that ouputs an ϵ-approximate collection in time polynomial in the input size and $\frac{1}{\epsilon}$.*

The full proof is contained in Subsection G.1. The algorithm uses backward induction on each MSP to approximate, at each state, the concave nondecreasing function $f(q)$ mapping a probability q to the maximum expected utility that can be generated, starting from that state, subject to claiming with probability at most q. Convex programming is used at each step, with the accumulation in error carefully bounded. We initially give an additive-multiplicative approximation guarantee, then round small enough values down to zero, obtaining the ϵ-approximate collection. Using the same technique, we are also able to give a Fully-Polynomial Time Approximation Scheme (FPTAS) for ex-ante CMS($\mathcal{F}$) whenever $\mathcal{F}$ is a matroid, presented in Proposition G.1.2.

The matroid prophet algorithm.. Our online algorithm is based on the matroid prophet approach of Kleinberg and Weinberg [2012], Lee and Singla [2018], in which thresholds τ_i are based on the expected marginal contribution of arrival $\mathcal{M}_i$ to the optimal solution that is constrained to coincide with the algorithm's choices so far (i.e. on arrivals $1, \ldots, i-1$). The outline of the algorithm is as follows.

(1) Compute an ϵ-approximate collection $(Q, \hat{U})$ to the ex-ante optimum.

(2) Decompose Q into a probability distribution D_Q over feasible subsets $F \in \mathcal{F}$. This turns out to be achievable in polynomial time because matroid polytopes have efficient separation oracles.

(3) On each arrival $\mathcal{M}_i$, use D_G and $\hat{U}$ to compute a threshold τ_i. The formula for τ_i represents

an "expected contribution" of $\hat{U}_i$ to the subset of $\{1, \ldots, i-1\}$ claimed by our algorithm so far, intersected with a random feasible subset drawn from D_Q. Run SAUP$(\mathcal{M}_i, \tau_i)$ to interact with $\mathcal{M}_i$ and determine whether to claim it.

See Algorithm G.2.1 for full details.

Proposition 5.4.3. *Let $\mathcal{F}$ be a matroid. Given $\mathcal{M}_1, \ldots, \mathcal{M}_n$, let $(Q, \hat{U})$ be an ϵ-approximate collection for any $\epsilon \geq 0$. There is a polynomial-time online algorithm* ALG *for* CMS$(\mathcal{F})$ *that takes as input $(Q, \hat{U})$, interacts with each arriving $\mathcal{M}_i$ only through calls to* SAUP$(\mathcal{M}_i, \tau)$*, and guarantees* $\mathbb{E}[\text{ALG}] \geq \frac{1}{2} \sum_{i=1}^{n} \hat{U}_i$.

The full proof is contained in Subsection G.2. We note that the algorithm takes any feasible $(Q, \hat{U})$ and provides a guarantee relative to $\hat{U}$. The better the algorithm $\widehat{\text{OPT}}$ associated with Q, and the better $\hat{U}$ approximates the true performances U of that algorithm, the better the approximation. In the case $\epsilon = 0$, we obtain a $\frac{1}{2}$ approximation.

To prove Proposition 5.4.3, we reduce to the Choice-Over-Reward Selection setting. There, we consider a standard prophet inequality decomposition of performance into "revenue" (interpreted as payments of agents for "winning") and "utility" of the agents. Although Choice-Over-Reward Selection appears more complex than traditional prophets settings, it remains true that, by solving SAUP, the "utility" of our algorithm on a given arrival exceeds the "utility" of OPT against the same "posted price" threshold. We can then continue with a somewhat standard analysis, although care is needed to adapt known techniques. For example, the algorithm utilizes an expectation over "Bernoullified" rewards for efficiency reasons, but the proof needs to consider "re-randomized" rewards that are converted to Bernoullified rewards at just the right point in the analysis.

We now have all the components to prove our second main result, an efficient $(\frac{1}{2} - \epsilon)$-approximation for CMS$(\mathcal{F})$ for any matroid $\mathcal{F}$.

Proof of Theorem 5.0.2. Let OPT be the optimal ex-ante feasible solution, which upper-bounds the optimal ex-post feasible solution. Given $\epsilon > 0$ and $\mathcal{M}_1, \ldots, \mathcal{M}_n$, using Proposition 5.4.2, we compute in time poly$(\mathcal{M}_1, \ldots, \mathcal{M}_n, \frac{1}{\epsilon})$ an ϵ-approximate collection $(Q, \hat{U})$. Then, we execute the

online algorithm of Proposition 5.4.3 with $\mathcal{M}_1, \ldots, \mathcal{M}_n$ and $(Q, \hat{U})$ as input. The online algorithm runs in polynomial time, making n calls to our SAUP algorithm of Proposition 5.4.1, each of which runs in polynomial time.

The algorithm's performance satisfies

$$\begin{aligned}
\mathbb{E}[\text{ALG}] &\geq \frac{1}{2}\sum_{i=1}^{n}\hat{U}_i && \text{Proposition 5.4.3}\\
&\geq \frac{1-\epsilon}{2}\sum_{i=1}^{n}U_i && \text{Definition 5.4.2}\\
&\geq \frac{(1-\epsilon)^2}{2}\text{OPT} && \text{Definition 5.4.2}\\
&\geq \left(\tfrac{1}{2}-\epsilon\right)\text{OPT}.
\end{aligned}$$

□

5.4.3 Application to Mechanism Design

Consider a setting with n strategic agents and a resource to be allocated. There is a constraint $\mathcal{F} \subseteq 2^{[n]}$ specifying which subsets S of agents may be feasibly allocated the resource at the same time. Each agent i interacts with an independent, private Markov Search Process $\mathcal{M}_i$ representing investment, research, and development regarding how to utilize the resource. The costs of the MSP are incurred by the agent, and the available reward of the MSP is the value generated for the agent if they are allocated, i.e. included in S. An agent's utility is their reward if allocated (else zero), minus the sum of all costs incurred in their MSP, minus any payment the agent makes. We call the version of the CMS problem where agents control each MSP strategic the CMS-Auction problem.

A *mechanism* is a game played by the n agents where an outcome consists of an allocation $S \in \mathcal{F}$ along with payments charged to each agent. We assume the agents must reach a terminal state of the MSP and reveal a reward to be allocated. A mechanism has a *Price of Anarchy (PoA)* of α if, in every Nash equilibrium, the expected sum of agent utilities and the payments is at least α times the expected performance of the optimal algorithm for CMS($\mathcal{F}$) on $(\mathcal{M}_1, \ldots, \mathcal{M}_n)$.

We now prove Corollary 5.0.1, yielding a Price of Anarchy of 1/2 for this problem (or $1/2 - \epsilon$

for a mechanism that runs in polynomial time). We note that in all cases, the agents' equilibrium strategies can be computed in polynomial time.

Proof of Corollary 5.0.1. The mechanism is a sequential posted-price mechanism. We approach the agents in order $1, \ldots, n$ one at a time and offer each i a price τ_i for being included in the allocation. We compute τ_i using Theorem 5.0.2. It is a unique best-response for i to solve SAUP$(\mathcal{M}_i, \tau_i)$, i.e. to implement the current stage of our algorithm. The performance of the algorithm is equal to the sum of agent utilities and payments, which implies the result. We note that, to inefficiently obtain a factor exactly $\frac{1}{2}$, we can compute the ex-ante optimal policy exactly and use it with the matroid prophet algorithm constructed in Proposition 5.4.3 instead of using the approximation to OPT. □

Bibliography

Saeed Alaei, Ali Makhdoumi, Azarakhsh Malekian, and Rad Niazadeh. Descending price auctions with bounded number of price levels and batched prophet inequality. arXiv preprint arXiv:2203.01384, 2022.

Ali Aouad, Jingwei Ji, and Yaron Shaposhnik. The pandora's box problem with sequential inspections. Available at SSRN 3726167, 2020.

Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. Management Science, 62(8):2374–2391, 2016.

Itai Ashlagi, Mark Braverman, Yash Kanoria, and Peng Shi. Clearing matching markets efficiently: informative signals and match recommendations. Management Science, 66(5):2163–2193, 2020.

Hedyeh Beyhaghi and Linda Cai. Pandora's problem with nonobligatory inspection: Optimal structure and a ptas. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC. ACM, 2023a.

Hedyeh Beyhaghi and Linda Cai. Recent developments in the pandora's box problem: Variants and applications. SIGecom Exchanges, 21(1), 2023b.

Hedyeh Beyhaghi and Linda Cai. Recent developments in pandora's box problem: Variants and applications. ACM SIGecom Exchanges, 21(1):20–34, 2024.

Hedyeh Beyhaghi and Robert Kleinberg. Pandora's problem with nonobligatory inspection. In Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, pages 131–132, 2019.

Shant Boodaghians, Federico Fusco, Philip Lazos, and Stefano Leonardi. Pandora's box problem with order constraints. In Proceedings of the 21st ACM Conference on Economics and Computation, pages 439–458, 2020a.

Shant Boodaghians, Federico Fusco, Philip Lazos, and Stefano Leonardi. Pandora's box problem with order constraints. EC 2020, page 439–458. Association for Computing Machinery, 2020b.

Robin Bowers and Bo Waggoner. High-welfare matching markets via descending price. In Jugal Garg, Max Klimm, and Yuqing Kong, editors, Web and Internet Economics. Springer Nature Switzerland, 2024.

Robin Bowers and Bo Waggoner. Matching with nested and bundled pandora boxes. In Marios Mavronicolas, Qi Qi, and Grant Schoenebeck, editors, Web and Internet Economics. Springer Nature Switzerland, 2026.

David B. Brown and James E. Smith. Optimal sequential exploration: Bandits, clairvoyants, and wildcats. Operations Research, 61:644–665, 2013.

Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. An algorithmic characterization of multi-dimensional mechanisms. CoRR, abs/1112.4572, 2017. URL http://arxiv.org/abs/1112.4572.

Hector Chade, Jan Eeckhout, and Lones Smith. Sorting through search and matching models in economics. Journal of Economic Literature, 55(2):493–544, 2017.

Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In Proceedings of the forty-second ACM symposium on Theory of computing, pages 311–320, 2010.

Shuchi Chawla, Evangelia Gergatsouli, Yifeng Teng, Christos Tzamos, and Ruimin Zhang. Pandora's box with correlations: Learning and approximation. In IEEE 61st Annual Symposium on Foundations of Computer Science, FOCS, pages 1214–1225. IEEE, 2020a.

Shuchi Chawla, Evangelia Gergatsouli, Yifeng Teng, Christos Tzamos, and Ruimin Zhang. Pandora's box with correlations: Learning and approximation. In 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS, pages 1214–1225. IEEE, 2020b. doi: 10.1109/FOCS46700.2020.00116.

Shuchi Chawla, Kira Goldner, Anna R Karlin, and J Benjamin Miller. Non-adaptive matroid prophet inequalities. In International Symposium on Algorithmic Game Theory, pages 389–404. Springer, 2024.

Shuchi Chawla, Dimitris Christou, and Trung Dang. Commitment gap via correlation gap. 2025.

Shuchi Chawla, Dimitris Christou, Amit Harlev, and Ziv Scully. Combinatorial selection with costly information. SODA. SIAM, 2026. URL https://arxiv.org/abs/2412.03860.

Yeon-Koo Che and Olivier Tercieux. Efficiency and stability in large matching markets. Journal of Political Economy, 127(5):2301–2342, 2019.

Yan Chen and YingHua He. Information acquisition and provision in school choice: a theoretical investigation. Economic Theory, pages 1–35, 2021.

William H Cunningham. Testing membership in matroid polyhedra. Journal of Combinatorial Theory, Series B, 36(2):161–188, 1984. ISSN 0095-8956. doi: https://doi.org/10.1016/0095-8956(84)90023-6.

Laura Doval. Whether or not to open pandora's box. Journal of Economic Theory, 175:127–158, 2018. ISSN 0022-0531. doi: https://doi.org/10.1016/j.jet.2018.01.005.

Ioana Dumitriu, Prasad Tetali, and Peter Winkler. On playing golf with two balls. SIAM Journal of Discrete Mathematics, 16(4):604–615, 2003. ISSN 0895-4801. doi: 10.1137/S0895480102408341.

Paul Dutting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. SIAM Journal on Computing, 49(3):540–582, 2020.

Hossein Esfandiari, MohammadTaghi HajiAghayi, Brendan Lucier, and Michael Mitzenmacher. Online pandora's boxes and bandits. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01):1885–1892, Jul. 2019. doi: 10.1609/aaai.v33i01.33011885. URL https://ojs.aaai.org/index.php/AAAI/article/view/4014.

Tomer Ezra, Stefano Leonardi, Rebecca Reiffenhäuser, Matteo Russo, and Alexandros Tsigonias-Dimitriadis. Prophet inequalities via the expected competitive ratio. In International Conference on Web and Internet Economics, pages 272–289. Springer, 2023.

Michal Feldman, Nicole Immorlica, Brendan Lucier, Tim Roughgarden, and Vasilis Syrgkanis. The price of anarchy in large games. In Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, STOC '16, page 963–976, New York, NY, USA, 2016a. Association for Computing Machinery. ISBN 9781450341325. doi: 10.1145/2897518.2897580.

Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms, pages 1014–1033. SIAM, 2016b.

Marcelo A Fernandez, Kirill Rudov, and Leeat Yariv. Centralized matching with incomplete information. Working Paper 29043, National Bureau of Economic Research, July 2021. URL http://www.nber.org/papers/w29043.

Hu Fu, Jiawei Li, and Daogao Liu. Pandora box problem with nonobligatory inspection: Hardness and approximation scheme. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC. ACM, 2023.

David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. The American Mathematical Monthly, 69(1):9–15, 1962.

John C Gittins. Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society Series B: Statistical Methodology, 41(2):148–164, 1979.

K. D. Glazebrook. Indices for Families of Competing Markov Decision Processes with Influence. The Annals of Applied Probability, 3(4):1013 – 1032, 1993. doi: 10.1214/aoap/1177005270.

Kevin D. Glazebrook. A profitability index for alternative research projects. Omega, 4(1):79–83, 1976. ISSN 0305-0483. doi: https://doi.org/10.1016/0305-0483(76)90041-4.

Kevin D Glazebrook. On a sufficient condition for superprocesses due to whittle. Journal of Applied Probability, 19(1):99–110, 1982.

Daniel Granot and Dror Zuckerman. Optimal sequencing and resource allocation in research and development projects. Management Science, 37:140–156, 1991.

Sudipto Guha and Kamesh Munagala. Approximation algorithms for bayesian multi-armed bandit problems. CoRR, abs/1306.3525, 2013. URL http://arxiv.org/abs/1306.3525.

Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and Adaptivity Gaps for Stochastic Probing, pages 1731–1747. 2016. doi: 10.1137/1.9781611974331.ch120. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611974331.ch120.

Anupam Gupta, Haotian Jiang, Ziv Scully, and Sahil Singla. The markovian price of information. In Integer Programming and Combinatorial Optimization: 20th International Conference, IPCO 2019, Ann Arbor, MI, USA, May 22-24, 2019, Proceedings 20, pages 233–246. Springer, 2019.

Dylan Hadfield-Menell. Multitasking: efficient optimal planning for bandit superprocesses. In Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI, page 345–354. AUAI Press, 2015. ISBN 9780996643108.

Rustamdjan Hakimov, Dorothea Kübler, and Siqi Pan. Costly information acquisition in centralized matching markets. Rationality and Competition Discussion Paper Series 280, CRC TRR 190 Rationality and Competition, Feb 2021. URL https://ideas.repec.org/p/rco/dpaper/280.html.

Rustamdjan Hakimov, Dorothea Kübler, and Siqi Pan. Costly information acquisition in centralized matching markets. Quantitative Economics, 14(4):1447–1490, 2023.

John William Hatfield and Paul R. Milgrom. Matching with contracts. American Economic Review, 95(4):913–935, 2005.

Yinghua He and Thierry Magnac. Application costs and congestion in matching markets. 2020.

Nicole Immorlica, Jacob Leshno, Irene Lo, and Brendan Lucier. Information acquisition in matching markets: The role of price discovery. Available at SSRN 3705049, 2020a.

Nicole Immorlica, Sahil Singla, and Bo Waggoner. Prophet inequalities with linear correlations and augmentations. In Proceedings of the 21st ACM Conference on Economics and Computation, EC, page 159–185, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450379755. doi: 10.1145/3391403.3399452.

Nicole Immorlica, Brendan Lucier, Vahideh Manshadi, and Alexander Wei. Designing Approximately Optimal Search on Matching Platforms, pages 632–633. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450385541. URL https://doi.org/10.1145/3465456.3467530.

T Tony Ke and J Miguel Villas-Boas. Optimal learning before choice. Journal of Economic Theory, 180:383–437, 2019.

Godfrey Keller and Alison Oldale. Branching bandits: a sequential search process with correlated pay-offs. Journal of Economic Theory, 113(2):302–315, 2003. ISSN 0022-0531. doi: https://doi.org/10.1016/S0022-0531(03)00092-9.

Alexander S. Kelso and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. Econometrica, 50(6):1483–1504, 1982. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913392.

Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, STOC, pages 123–136, 2012.

Robert Kleinberg, Bo Waggoner, and E. Glen Weyl. Descending price optimally coordinates search. In Proceedings of the 2016 ACM Conference on Economics and Computation, EC, page 23–24. Association for Computing Machinery, 2016. ISBN 9781450339360. doi: 10.1145/2940716.2940760.

Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. Probability on Banach spaces, 4(197-266):1–2, 1978.

Euiwoong Lee and Sahil Singla. Optimal online contention resolution schemes via ex-ante prophet inequalities. arXiv preprint arXiv:1806.09251, 2018.

Brendan Lucier. An economic view of prophet inequalities. SIGecom Exchanges, 16(1):24–47, 2017. doi: 10.1145/3144722.3144725.

Brendan Lucier and Vasilis Syrgkanis. Greedy algorithms make efficient mechanisms. In Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC, pages 221–238. ACM, 2015. doi: 10.1145/2764468.2764506.

Will Ma. Improvements and generalizations of stochastic knapsack and markovian bandits approximation algorithms. Mathematics of Operations Research, 43(3):789–812, 2018.

Peter Nash. Optimal allocation of Resources Between Research Projects. PhD thesis, University of Cambridge, 1973.

Peter Nash. A generalized bandit problem. Journal of the Royal Statistical Society, 42:165–169, 1980.

Tim Roughgarden, Vasilis Syrgkanis, and Eva Tardos. The price of anarchy in auctions. Journal of Artificial Intelligence Research, 59:59–101, 2017.

Aviad Rubinstein and Sahil Singla. Combinatorial prophet inequalities. In Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1671–1687. SIAM, 2017.

Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent random variables. Annals of Probability, 12(4):1212–1216, 1984.

Ziv Scully and Laura Doval. Local hedging approximately solves pandora's box problems with nonobligatory inspection, 2025. URL `https://arxiv.org/abs/2410.19011`.

Ziv Scully and Alexander Terenin. The gittins index: A design principle for decision making under uncertainty. In Tutorials in Operations Research: Advances in Analytics and Operations Research: Improving Decisions to Secure the Future, pages 28–70. INFORMS, 2025.

Sahil Singla. The price of information in combinatorial optimization. In Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pages 2523–2532. SIAM, 2018.

Bo Waggoner and E Glen Weyl. Matching markets via descending price. Available at SSRN 3373934, 2019.

Martin L. Weitzman. Optimal search for the best alternative. Econometrica, 47(3):641–654, 1979.

Peter Whittle. Multi-armed bandits and the gittins index. Journal of the Royal Statistical Society: Series B (Methodological), 42(2):143–149, 1980.

Qiqi Yan. Mechanism design via correlation gap. In Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms, pages 710–719. SIAM, 2011.

Appendix A

Omitted Proofs for Bandit Processes

This section provides the proofs omitted from Section 2.3.2.

First, we prove a fact that will be helpul in the proof of the key lemma, Lemma 2.3.1.

Lemma A.0.1. *The expected welfare contribution of bandit i under any algorithm satisfies*

$$\mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} c_i^{(\ell)}\right] = \mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} \left(\kappa_i^{(\ell+1)} - \kappa_i^{(\ell)}\right)\right].$$

Intuitively, this proof is relatively direct from the definition of nested Weitzman index,

$$\mathbb{E}\left[\left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+ \,\middle|\, s_i^{(1)}, \ldots, s_i^{(\ell-1)}\right] = c_i^{\ell},$$

along with the observation that $\mathbb{I}_i^{(\ell)}$ is independent of $(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)})^+$ conditioned on $s_i^{(1)}, \ldots, s_i^{(\ell-1)}$. (This follows because the decision to advance to stage ℓ comes prior to observing any of the signals $s_i^{(\ell)}, \ldots, s_i^{(d-1)}$ and value v_i that define $\kappa_i^{(\ell+1)}$.) However, the precise argument is subtle, so we go somewhat slowly and carefully.

Proof. To simplify notation for this proof, let $s_{i,\ell} := (s_i^{(1)}, \ldots, s_i^{(\ell-1)})$.

We first expand the second term to explicitly represent the dependence on received signals. We use $\mathbb{E}_{s_{i,\ell}}[\mathbb{E}[\cdot \mid s_{i,\ell}]]$ to denote an outer expectation taken over the random draws of $s_i^{(1)}, \ldots, s_i^{(\ell-1)}$, and an inner expectation of $(\cdot)$ conditioned on $s_i^{(1)}, \ldots, s_i^{(\ell-1)}$ and taken over all other randomness.

$$\mathbb{E}\left[\sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} c_i^{(\ell)}\right] = \sum_{\ell=1}^{d} \mathop{\mathbb{E}}_{s_{i,\ell}}\left[\mathbb{E}\left[\mathbb{I}_i^{(\ell)} \,\middle|\, s_{i,\ell}\right] c_i^{(\ell)}\right].$$

Now, we substitute in for $c_i^{(\ell)}$ using the definition of the bandit Weitzman index and capped value. To be precise, it is important to note that the randomness in the definition of $c_i^{(\ell)}$ is over a different, independent space, whose first $\ell - 1$ signals we will denote analogously by $s'_{i,\ell}$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{s_{i,\ell}}\left[\mathbb{E}\left[\mathbb{I}_i^{(\ell)} \,\middle|\, s_{i,\ell}\right] c_i^{(\ell)}\right] &= \mathop{\mathbb{E}}_{s_{i,\ell}}\left[\mathbb{E}\left[\mathbb{I}_i^{(\ell)} \,\middle|\, s_{i,\ell}\right] \mathop{\mathbb{E}}_{s'_{i,\ell}}\left[\mathbb{E}\left[\left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+ \,\middle|\, s'_{i,\ell}\right]\right]\right] \\
&= \mathop{\mathbb{E}}_{s_{i,\ell}}\mathop{\mathbb{E}}_{s'_{i,\ell}}\left[\mathbb{E}\left[\mathbb{I}_i^{(\ell)} \,\middle|\, s_{i,\ell}\right] \mathbb{E}\left[\left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+ \,\middle|\, s'_{i,\ell}\right]\right] \\
&= \mathop{\mathbb{E}}_{s_{i,\ell}}\left[\mathbb{E}\left[\mathbb{I}_i^{(\ell)} \,\middle|\, s_{i,\ell}\right] \mathbb{E}\left[\left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+ \,\middle|\, s_{i,\ell}\right]\right] \qquad \text{(A.1)} \\
&= \mathop{\mathbb{E}}_{s_{i,\ell}}\left[\mathbb{E}\left[\mathbb{I}_i^{(\ell)} \left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+ \,\middle|\, s_{i,\ell}\right]\right] \qquad \text{(A.2)} \\
&= \mathbb{E}\left[\mathbb{I}_i^{(\ell)} \left(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)}\right)^+\right].
\end{aligned}$$

Equality A.1 follows because the worlds of $s_{i,\ell}$ and $s'_{i,\ell}$ have the same distribution. Equality A.2 follows by conditional independence, as discussed above. It only remains to observe that $(\kappa_i^{(\ell+1)} - \sigma_i^{(\ell)})^+ = \kappa_i^{(\ell+1)} - \min\{\kappa_i^{(\ell+1)}, \sigma_i^{(\ell)}\} = \kappa_i^{(\ell+1)} - \kappa_i^{(\ell)}$. □

We use this to prove the key lemma 2.3.1 relating actual utility to an amortized version, which is used throughout the paper.

Proof of Lemma 2.3.1. First, Lemma A.0.1 gives

$$\mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} c_i^{(\ell)}\right] = \mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} \left(\kappa_i^{(\ell+1)} - \kappa_i^{(\ell)}\right)\right].$$

The proof is direct, but depends crucially on the fact that $\mathbb{I}_i^{(\ell)}$ and $\kappa_i^{(\ell+1)}$ are independent conditioned on $s_i^{(1)}, \ldots, s_i^{(\ell-1)}$. Let ℓ^* be a random variable denoting the last inspection stage the algorithm performs for bandit i. If the algorithm never advances i, then $\ell^* = 0$, and if it advances

to the end (including the case where $\mathbb{A}_i = 1$), then $\ell^* = d$. We have

$$\begin{aligned}
\mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{d} \mathbb{I}_i^{(\ell)} c_i^{(\ell)}\right] &= \mathbb{E}\left[\mathbb{A}_i v_i - \sum_{\ell=1}^{\ell^*} \left(\kappa_i^{(\ell+1)} - \kappa_i^{(\ell)}\right)\right] \\
&= \mathbb{E}\left[\mathbb{A}_i \kappa_i^{(d+1)} - \left(\kappa_i^{(\ell^*+1)} - \kappa_i^{(1)}\right)\right] \\
&\leq \mathbb{E}\left[\mathbb{A}_i \left(\kappa_i^{(d+1)} - \kappa_i^{(\ell^*+1)} + \kappa_i^{(1)}\right)\right] \\
&= \mathbb{E}\left[\mathbb{A}_i \kappa_i^{(1)}\right],
\end{aligned}$$

where the inequality holds because $\mathbb{A}_i \geq 0$ and $\kappa_i^{(\ell)}$ is nondecreasing in ℓ; and the final equality holds in both the case $\mathbb{A}_i = 0$ and the case $\mathbb{A}_i = 1$ (in which $\ell^* = d$). Furthermore, the inequality is strict if and only if, with nonzero probability, $\mathbb{A}_i = 0$ and $\kappa_i^{(\ell^*+1)} > \kappa_i^{(1)}$ for some $\ell^* \in \{1, \ldots, d\}$. Observe that $\kappa_i^{(1)} = \min\{\gamma_i^{(\ell^*)}, \kappa_i^{(\ell^*+1)}\}$, so this is equivalent to $\kappa_i^{(\ell^*+1)} > \gamma_i^{(\ell^*)}$ and $\mathbb{I}_i^{(\ell^*+1)} > \mathbb{I}_i^{(\ell^*)}$: the algorithm being exposed. □

Appendix B

Price of Anarchy for the No-Rebate Setting

In addition to our 1/4-rebate mechanism, we find that the mechanism without a rebate originally proposed by Waggoner and Weyl [2019] also obtains a Bayes-Nash price of anarchy of 1/8 in the inspection setting.

Theorem B.0.1. *In the nonnegative values setting, the no-rebate Marshallian Match has a Bayes-Nash price of anarchy of at least 1/8.*

Our proof here closely mirrors the proofs of Section 3.4, that the 1/4-rebate MM achieves a constant price of anarchy. We begin by introducing a deviation strategy and show that it is non-exposed, allowing us to apply Lemma 3.4.1. We then make a similar smoothness argument to complete the proof.

We first introduce a deviation strategy under which i bids 0 or $\kappa_{ij}/2$ on all possible matches j. The deviation strategy b'_i for agent i is as follows. While $p(t) > \sigma_{ij}/2$, $b'_{ij}(t) = 0$. When $p(t) = \sigma_{ij}/2$, inspect j to learn $\kappa_{ij} = \min(v_{ij}, \sigma_{ij})$. When $p(t) = \kappa_{ij}/2$, set bid to $b'_{ij}(t) = \kappa_{ij}/2$.

In other words, i sets her bid on all possible matches to 0 until she performs an inspection, at half her Weitzman index. After inspecting, if $v_{ij} \geq \sigma_{ij}$, she immediately forces the match. Otherwise, she waits until v_{ij} is twice the current price and then forces the match if both she and j are still available.

We observe that this strategy has several helpful properties considered in other sections.

Lemma B.0.1. *The deviation strategy b' exercises in the money and has nonnegative capped value $\bar{u}_i(b_{-i}, b')$ for all strategy profiles b.*

Recall the definition of covered call utility, $\bar{u}_i(b) = \kappa_i(b) - \pi_i(b)$. In the no-rebate setting the payment term $\pi_i(b) = b_{ij}(t)$, the bid i had on j when they matched at time t.

Proof. **Non-exposure.** Inspection occurs in two cases: either i is forced to inspect agent j because they have matched, or i chooses to inspect j. In the first case, $A_{ij} = 1$ and i must claim j whether or not $v_{ij} \geq \sigma_{ij}$. In the second case, i inspects j at time t such that $p(t) = \sigma_{ij}/2$. If $\kappa_{ij} \geq \sigma_{ij}$, then $b'_{ij}(t) = \kappa_{ij}/2 \geq \sigma_{ij}/2 = p(t)$, and i immediately raises her bid and matches to j.

Nonnegative capped value. If i does not match, then i has zero capped value. Otherwise, i matches to some j in two cases: either before or after she raises her bid on j. In the first case, $\pi_i(b) = 0$ and $\bar{u}_i(b_{-i}, b') = \kappa_{ij} \geq 0$. In the second, we know i matches to j exactly when she raises her bid to $b'_{ij}(t) = \kappa_{ij}/2 = \pi_{ij}(b_{-i}, b')$. Thus, $\bar{u}_i(b_{-i}, b') = \kappa_{ij}/2 \geq 0$. □

Lemma B.0.2 (No-Rebate Capped Value Smoothness)**.** *For any pair i and j and strategy profile b,*

$$\bar{u}_i(b_{-i}, b') + \frac{p_i(b)}{4} + \frac{p_j(b)}{4} \geq \frac{\kappa_{ij}}{8}.$$

Proof. Since payments are nonnegative, and by Lemma B.0.1 i has nonnegative capped value utility under the deviation, it suffices to show that at least one term on the left hand side is at least $\kappa_{ij}/8$. If $p_i(b)/2 \geq \kappa_{ij}/2$ or $p_i(b)/2 \geq \kappa_{ij}/2$, the lemma follows immediately.

Otherwise, in deviation i matches to some agent k at time t. Note $p(t) \geq \kappa_{ij}/2$, as i must have inpsected j by that point ($\kappa_{ij}/2 \leq \sigma_{ij}/2$ by definition), and as j is still available when the price is $\kappa_{ij}/2$, i would match j when $p(t) \geq \kappa_{ij}/2$ if she were still unmatched in deviation.

In deviation i matches to k at time t, so k must have been available to match at time t prior to deviation. Furthermore, $b'_{ik}(t) + b_{ki}(t) \geq p(t)$ as the match occurs in deviation. Since $p_{ij} < \kappa_{ij}/2$, the total bid on the $\{i, k\}$ edge prior to deviation was $b_{ik}(t) + b_{ki}(t) < \kappa_{ij}/2 \leq p(t)$, so $0 \leq b_{ik}(t) < b'_{ik}(t)$, and i must have raised her bid on k in deviation by time t. Thus, i obtains capped utility $\bar{u}_i(b_{-i}, b') = \kappa_{ik}/2 \geq \kappa_{ij}/2$ in deviation. □

Theorem B.0.2. *In the inspection setting with nonnegative values, the no-rebate Marshallian Match guarantees a Bayes-Nash price of anarchy of at least* $1/8$.

The proof follows exactly the proof of Theorem 3.4.1 given in Section 3.4, substituting in the smoothness lemma B.0.2 and the fact that the no-rebate deviation strategy given in this section is non-exposed for the equivalent facts used for the deviation strategy presented in Section 3.4.

A corollary of this fact is that the no-rebate MM also achieves a constant price of anarchy in the deterministic values setting, without inspection.

Corollary B.0.1. *In the nonnegative values setting, the no-rebate Marshallian Match has a Bayes-Nash price of anarchy of at least 1/8.*

Proof. We model the setting without inspection costs as having $c_{ij} = 0$ for all inspections, and a deterministic distribution of the value v_{ij}. When $c_{ij} = 0$ for all i, j pairs, the Weitzman indices cease to be unique: any value of σ_{ij} above the upper limit of the support of v_{ij} satisfies the identity $\mathbb{E}\left[(v_{ij} - \sigma_{ij})^+\right] = 0$. We define $\sigma_{ij} = v_{ij}$, the deterministic value of the match for i. Thus, $\kappa_{ij} = v_{ij}$ deterministically, and the deviation strategy b' reduces to bidding 0 on every edge until the price drops to $v_{ij}/2$ for some j, and then immediately inspecting and matching the edge. □

Appendix C

Full-Rebate Marshallian Match has Price of Anarchy 0

We provide a setting in which the 1/2-rebate Marshallian match has unboundedly bad social welfare in certain equilibria, relative to the optimal matching. Here, note that we use interchangeably the notation of a 1/2-rebate and full-rebate mechanism, as the 1/2-rebate denotes the fraction of payment returned to one bidder in a match, meaning that in the 1/2-rebate setting the full payment is redistributed.

Lemma C.0.1. *There exists a setting for which the* 1/2*-rebate Marshallian Match with inspection has price of anarchy 0.*

Proof. We show this using a specific setting with four agents, illustrated in Figure C.1. Agents a, d, and c are all able to match to one another, with symmetric identical deterministic costs and values:

$$v_{ad} = v_{da} = v_{ac} = v_{ca} = v_{dc} = v_{cd} = 1,$$

$$c_{ad} = c_{da} = c_{ac} = c_{ca} = c_{dc} = c_{cd} = 1.$$

An additional agent, agent x, is only able to match with agent c:

$$v_{cx} = 1, c_{cx} = 1,$$

$$v_{xc} = 2, c_{xc} = 1$$

for some arbitrary k.

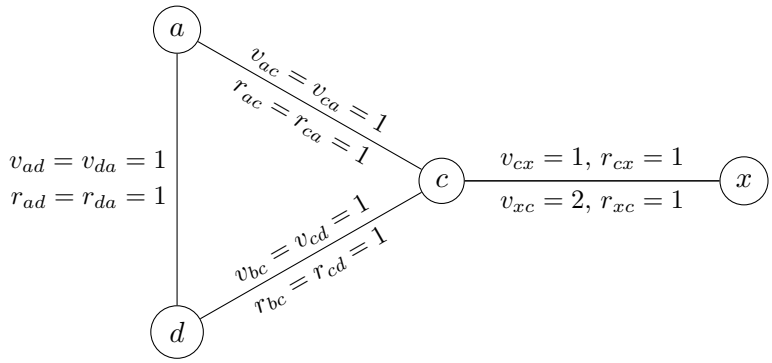


Figure C.1: Graph Instance with Low-Welfare Full-Rebate Marshallian Match Equilibrium

In the 1/2-rebate Marshallian Match when the full payment is distributed back amongst bidders, there exist equilibria with unboundedly low welfare relative to the social welfare-maximizing matching.

The highest social welfare allocation is for c and x inspect and match while a and d either do nothing and remain unmatched or inspect and match each other, for overall welfare of $(1-1)+(2-1)=1$.

We will now construct an equilibrium in the full-rebate MM where a and c match, leaving d and x without any available matches, for total welfare $(1-1)+(1-1)=0$, giving PoA $=0$.

Our equilibrium relies on careful tiebreaking between edges. If two actions are attempted at the same price $p(t)$, we break ties according to the edge on which the actions occur: $(a,c) \succ (a,d) \succ (d,c) \succ (c,x)$.

The three agents forming a 3-clique behave identically within the clique: each attempts to inspect both its neighbors when $p(t)=2$ and maintains $b_{ij}(t)=2$ for all t. By tiebreaking, the first edge for which the total bid is equal to the current price, (a,c), is matched and forced to inspect at both endpoints, before any other inspections or matches occur. Agent c and x will attempt to inspect one another when $p(t)=0$ and both maintain $b_{cx}(t)=b_{xc}(t)=0$ throughout.

As all strategies are deterministic and constant and all information is known, any deviation bidding strategy $b(t)$ for agent i becomes equivalent to placing a constant bid of $b_{ij}(t^*)$ on each possible edge (i,j), where t^* is the smallest time at which $p(t)=b_{ij}(t)$ or zero if $p(t)>b_{ij(t)}$ for all t. We thus drop the time dependence, and simply write a deviation bidding strategy as b_{ij}. Additionally, any deviation strategy can only have higher utility if inspection of j is delayed until $p(t)=b_{ij}$, or until an inspection is forced by a match. Combining these observations, then, any deviation strategy for i is weakly dominated by a strategy fully characterized by a constant bid b_{ij} for each possible match j, where i initiates inspection when $p(t)=b_{ij}$.

We consider deviations strategies of this form, and show that this is an equilibrium for each agent:

- x: Under the equilibrium profile, x is unatched and performs no inspections for total utility 0. To match, x would have to place a bid $b_{xc} \geq 4$ to match with c. Since c never bids above 0 on x, x's utility for this deviation would be $2-1-b_{xc}/2 \leq 1-2<0$ and x cannot

profitably deviate.

- a, c: since c has value 0 for inspecting and matching to their "outside option" x, we treat c as identical to a in being able to match to one of the other trio of bidders or go unmatched for value 0. WLOG we analyze a here. Under the equilibirum profile, a gets total utility 0 for matching to c. Since a is already matching, increasing their bid only lowers their utility. If a deviates to lowering their bid, then (d, c) now has the highest bid and will be matched, leaving a unmatched with utility 0, and not increasing their utility.

- d: in the equilibrium, d is unmatched and performs no inspections, for utility 0. By lowering either of their bids on a or c, d remains unmatched. If d increases their bid to $b_{di} > 2$ for $i \in \{a, c\}$, then d will be immediately matched for utility $1 - 1 - (b_{di} - 2)/2 < 0$, making the deviation unprofitable.

Thus, the given strategy profile is indeed an equilibrium for all agents, and achieves social welfare 0, as opposed to the optimal allocation which achieves welfare 1. □

The proof of Lemma C.0.1 can be easily modified to obtain a similar negative result for the fulll-rebate Marshallian match without inspection, by simply setting agents' values deterministically to $v_{ij} - c_{ij}$ for each edge (i, j), and leaving bids as stated in the equilibrium presented in the proof.

Corollary C.0.1. *The* $1/2$*-rebate Marshallian Match without inspection has price of anarchy 0.*

Bipartite Matching. An interesting feature of this counuterexample is its reliance on a 3-cycle to threaten each high-bidding agent in the clique into maintaining their over-bid. The unmatched agent d gains nothing by lowering their bid, while either of the matched agents a and c would immediately lose their match to d if they lowered their bid at all. If d were not a threat, a could lower their bid on c so that the total bid between a and c was asymmetrical, but still high enough to win. In this case, $(b_{ac} + b_{ca})/2 > b_{ac}$ and a would receive a positive payment from c for the match.

This dynamic extends to higher-order odd cycles: as long as there is one agent available to enforce over-bids, equilibria such as the one in the proof of Lemma C.0.1 can be constructed. However, this technique cannot be applied in even cycles. While unlikely, it is an open question if the full-rebate Marshallian Match has nonzero price of anarchy in the bipartite setting usually considered in matchings.

If this is the case, it seems as though the smoothness techniques used in this paper would be insufficient to prove the price of anarchy result directly. Our earlier smoothness results are agnostic to the graph structure, while any such result here would have to rely on the bipartiteness of the graph.

Appendix D

Group Matchings with Inspections

In this appendix, we extend our price of anarchy with inspection results from Section 3.4 to the group matching setting considered in Section 3.5.

In this appendix, we prove that the 1/4-rebate Marshallian Match achieves a constant Bayes-Nash price of anarchy in the group matching setting with nonnegative values. Our approach is similar to the proof of the bipartite matching case in Section 3.3.

Group matching with inspection. Recall that in the group matching setting, we are given a hypergraph with agents as vertices and each hyperedge S representing a feasible group, i.e. subset of agents of size at most k. Agent i has value $v_{iS} \geq 0$ for being assigned to group S. We augment this for the inspection setting by having a cost r_{iS} and distribution $v_{iS} \sim D_{iS}$ for every i, S such that $i \in S$. The distributions D_{iS} are common knowledge amongst agents, but agents must pay the cost r_{iS} to reveal their value for matching S.

A "matching" or assignment M consists of a subset of the sets such that no agent is in two different groups $S, S' \in M$. The surplus of a group S is $s_S = \sum_{i \in S} v_{iS}$. The social welfare of an assignment M is $\sum_{S \in M} s_S$.

All machinery for strike prices, covered call values, and exercising in the money transfer over, replacing the agent j to whom i could match with a set S to which she belongs. The proof of Lemma 3.4.1 remains unchanged replacing j with S, as well.

Marshallian Match Mechanism. The 1/4-rebate Marshallian Match for this setting is modified into a "$\frac{1}{2k}$-rebate MM", where k is the size of the largest possible group. Participant i

maintains bids b_{iS} for all of her feasible groups S. At any time, an agent i may choose to inspect any group S at cost r_{iS} to learn her value, v_{iS}, as long as no members of S are already matched and i is a member of S. We call S available if these criteria are met, and no other agents observe i's choice to inspect. When the descending price matches the total sum of bids on a group S whose members are, i.e. $p(t) = \sum_{i \in S} b_{iS}$, that group is matched and drops out of the mechanism. All members $j \in S$ must pay their inspection cost r_{jS} if they have not yet inspected the matched group, and each pay their current bid $b_{jS}(t)$. Each then receives a rebate of $\frac{p(t)}{2|S|}$, with the mechanism keeping $\frac{p(t)}{2}$.

Smoothness deviation and proof. We consider a deviation strategy b' for agent i in the inspection setting. Each agent computes σ_{iS} such that $\mathbb{E}[(v_{iS} - \sigma_{iS})^+] = c_{iS}$, and maintains $b'(t) = 0$ on S until $p(t) = \sigma_{iS}$, at which point i attempts to inspect group S. Let t^* be the time at which this happens. If S is available, i pays c_{iS} and learns v_{iS} or otherwise learns that S is unavailable. i then updates her bid to $b'_{iS}(t) = \kappa_{iS} = \min(v_{iS}, \sigma_{iS})$ for all $t \geq t^*$ and maintains that bid on S until she is matched.

We note that 3.4.2 extends to the group inspection setting, since the strategy b' is the same as in Section 3.4 but on sets instead of neighbors.

We now prove the equivalent of Lemma 3.4.3 in the group setting.

Lemma D.0.1 (Group covered call smoothness)**.** *For any i and S such that $i \in S$ and strategy profile b, for all realizations of types and values,*

$$\bar{u}_i(b_{-i}, b'_i) + \frac{1}{k^2} \sum_{j \in S} \frac{p_j(b)}{\geq} \frac{\kappa_{iS}}{2k^2}.$$

Proof. From extending Lemma 3.4.2 to the group setting, $\bar{u}_i(b_{-i}, b'_i)$ is nonnegative along with all mechanism payment terms $p_j(b)$. Thus, we need only one of the terms to be at least $\frac{\kappa_{iS}}{2k^2}$.

If any of the payment terms $p_j(b) \geq \frac{\kappa_{iS}}{2}$, we are done. We assume all payment terms are smaller than $\frac{\kappa_{iS}}{2}$, and show that i's utility in deviation is at least $\frac{\kappa_{iS}}{2k^2}$.

If $p_j(b) < \kappa_{iS}/2$, then S is available prior to deviation at least until $p(t) = \kappa_{iS}$, call this time t^*. i then matches in deviation to some S' at least by time t^*, since if i is still unmatched she

matches with S at that time.

Since $p_i(b) < \kappa_{iS}/2$ by assumption, $b'_{iS'} > b_{iS'}$ since no other agents' bids can change on S', but i matches in deviation. Specifically, $b_{iS'} > 0$ at some time $t' \geq t^*$, which means $\kappa_{iS'} = b_{iS'} > 0$ at t', and $\kappa_{iS'} \geq p(t') \geq p(t^*) = \kappa_{iS'}$.

Since i recoups at least a $1/(2|S'|)$-factor of her bid from the rebate mechanism, $\bar{u}_i(b_{-i}, b'_i) \geq \frac{\kappa_{iS'}}{2|S'|} \geq \frac{\kappa_{iS}}{2k} \geq \frac{\kappa_{iS}}{2k^2}$, proving our claim. □

We now prove a price of anarchy guarantee for the group inspection setting.

Theorem D.0.1. *In the group inspection setting with nonnegative values, the $\frac{1}{2k}$-rebate Marshallian Match guarantees a Bayes-Nash price of anarchy of at least $1/(2k^2)$, where $k = \max_S |S|$.*

Recall our notation from Section 3.4, and denote the event that i matches with group S by A_{iS} and the event that i inspects S by I_{iS}. We require that $A_{iS} = A_{jS}$ for all $i, j \in S$ (a group cannot partially match) and $I_{iS} \geq A_{iS}$ (inspection by all members is required for a group to match). If $i \notin S$, for convenience we define $A_{iS} = I_{iS} = 0$.

For a realization of play, an agent's utility is then $u_i(b) = \sum_S A_{iS} v_{iS} - I_{iS} c_{iS} - \pi_i(b)$. The final term is the total payment to the mechanism i makes.

Proof. Let b be a Bayes-Nash equilibrium and M^* the optimal assignment (a random variable). We have the following:

$$\text{Welf}(b) \geq \mathbb{E} \sum_i \left(u_i(b) + \frac{p_i(b)}{k} \right),$$

with the final term being added as the payment the mechanism receives for a match involving i is at least $1/k$ times the number of agents involved in the match.

Applying the deviation strategy defined above for each agent and using that it exercises in

the money, we obtain

$$\begin{aligned}\text{Welf}(b) &\geq \mathbb{E}\sum_i \left(u_i(b_{-i}, b'_i) + \frac{p_i(b)}{k}\right) \\ &= \mathbb{E}\sum_i \left(A_{iS}\kappa_{iS} - \pi_i(b_{-i}, b'_i) + \frac{p_i(b)}{k}\right) \\ &= \mathbb{E}\sum_i \left(\bar{u}(b_{-i}, b'_i) + \frac{p_i(b)}{k}\right).\end{aligned}$$

We introduce the random variable M^* for the optimal matching, and while potentially dropping some agents, re-index the sum as

$$\begin{aligned}\text{Welf}(b) &\geq \mathbb{E}\sum_{S^*\in M^*}\sum_{i\in S^*} \left(\bar{u}_i(b_{-i}, b'_i) + \frac{p_i(b)}{k}\right) \\ &\geq \mathbb{E}\sum_{S^*\in M^*}\sum_{i\in S^*} \left(\bar{u}_i(b_{-i}, b'_i) + \frac{1}{k^2}\sum_{j\in S^*} p_j(b)\right) \\ &\geq \mathbb{E}\sum_{S^*\in M^*}\sum_{i\in S^*} \left(\bar{u}_i(b_{-i}, b'_i) + \frac{1}{k^2}\sum_{j\in S^*} p_j(b)\right) \\ &\geq \mathbb{E}\sum_{S^*\in M^*}\sum_{i\in S^*} \frac{\kappa_{iS^*}}{2k^2} \\ &= \frac{1}{2k^2}\text{Welf}(\text{OPT}).\end{aligned}$$

The lower bound of the first line holds as all agents make nonnegative payments and in deviation all agents have nonnegative covered call utility, while the second to last line follows from Lemma D.0.1. □

Appendix E

Proofs for Section 4.4

This appendix contains the proofs and examples used in Section 4.4, "no dessert". We detail the calculation of indices used in the proof of our "no dessert" theorem, Theorem 4.4.1.

Lemma E.0.1. *For the bandit (i, j) resulting by orienting the edge so that i is inspected prior to j, the initial Weitzman index and the capped values are the following:*

- $\sigma_{ij}^{(1)} = \frac{1}{\alpha} - 1$.
- $\kappa_{ij}^{(1)} = \frac{1}{\alpha} - 1$ *with probability α, otherwise negative.*
- $\mathbb{E}[(\kappa_{ij}^{(1)})^+] = 1 - \alpha$.

Proof. We utilize Figure E.1. We just need to prove that each $\sigma_{ij}^{(k)}$ and $\kappa_{ij}^{(k)}$ given in Figure E.1 is correct. They are computed by backward induction: $\sigma_{ij}^{(2)}$, then $\kappa_{ij}^{(2)}$, then $\sigma_{ij}^{(1)}$, and finally $\kappa_{ij}^{(1)}$. First, we show that in the case $v_{ij} = \frac{1}{\alpha^3}$, it is correct that $\sigma_{ij}^{(2)} = \frac{1}{\alpha^3} - \frac{1}{\alpha^2} + \frac{1}{\alpha}$. Specializing Definition 2.3.2 to this setting:

$$\begin{aligned}
\mathbb{E}\left[(\bar{v} - \sigma_{ij}^{(2)})^+ \;\middle|\; v_{ij} = \tfrac{1}{\alpha^3}\right] &= \mathbb{E}\left[(\tfrac{1}{\alpha^3} + v_{ji} - (\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha^2} + \tfrac{1}{\alpha}))^+\right] \\
&= \mathbb{E}\left[(v_{ji} + \tfrac{1}{\alpha^2} - \tfrac{1}{\alpha})^+\right] \\
&= \alpha^2\left(\tfrac{1}{\alpha^2} - \tfrac{1}{\alpha}\right) + (1 - \alpha^2)(0) \\
&= 1 - \alpha \\
&= c_{ji},
\end{aligned}$$

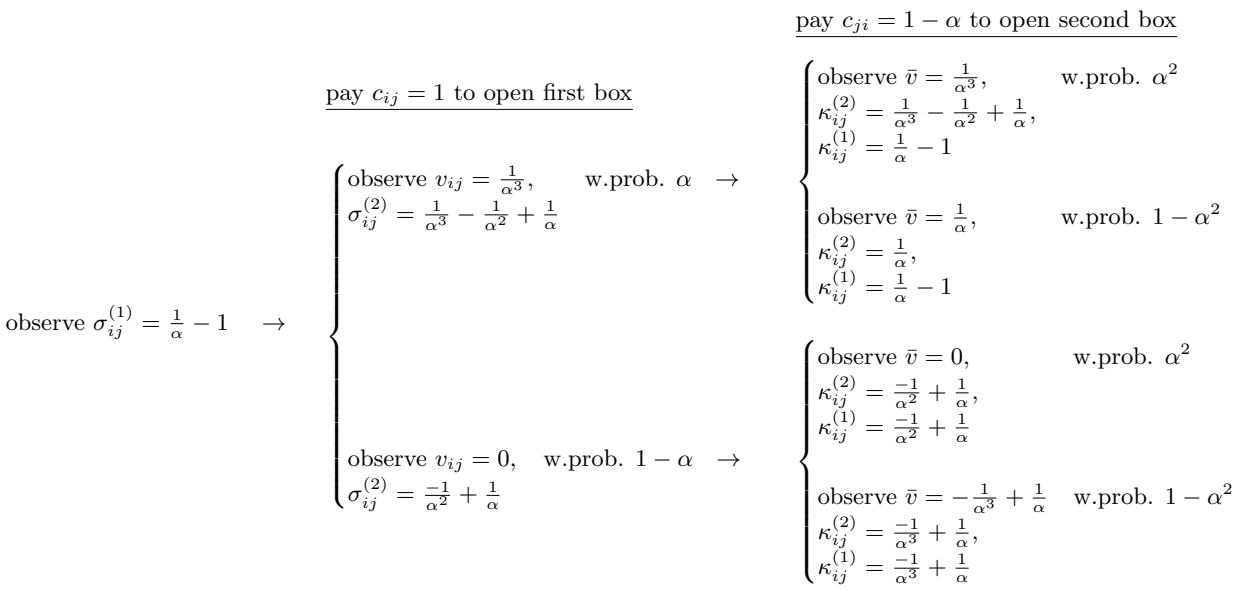


Figure E.1: No Dessert Bandit (i, j)

The bandit (i, j). Here $\bar{v} = v_{ij} + v_{ji}$. After opening the first box, $\sigma_{ij}^{(2)}$ is determined. After opening the second box, $\bar{v}$ is determined and we can calculate $\kappa_{ij}^{(2)} = \min\{\bar{v}, \sigma_{ij}^{(2)}\}$ as well as $\kappa_{ij}^{(1)} = \min\{\bar{v}, \sigma_{ij}^{(2)}, \sigma_{ij}^{(1)}$.

as required. For the case $v_{ij} = 0$, we claim $\sigma_{ij}^{(2)} = \frac{-1}{\alpha^2} + \frac{1}{\alpha}$:

$$\begin{aligned}
\mathbb{E}\left[(\bar{v} - \sigma_{ij}^{(2)})^+ \;\middle|\; v_{ij} = 0\right] &= \mathbb{E}\left[(v_{ji} - (\tfrac{-1}{\alpha^2} + \tfrac{1}{\alpha}))^+\right] \\
&= \alpha^2\left(0 + \tfrac{1}{\alpha^2} - \tfrac{1}{\alpha}\right) + (1-\alpha^2)(0) \\
&= 1 - \alpha \\
&= c_{ji}.
\end{aligned}$$

Now we can define $\kappa_{ij}^{(2)} = \min\{\bar{v}, \sigma_{ij}^{(2)}\}$ and check that Figure E.1 is correct in all four states. Then, we just need to check that $\sigma_{ij}^{(1)} = \frac{1}{\alpha} - 1$ as claimed. With Definition 2.3.2:

$$\begin{aligned}
\mathbb{E}\left[(\kappa_{ij}^{(2)} - \sigma_{ij}^{(1)})^+\right] &= \mathbb{E}\left[(\kappa_{ij}^{(2)} - \tfrac{1}{\alpha} + 1)^+\right] \\
&= \alpha^3\left(\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha^2} + \tfrac{1}{\alpha} - \tfrac{1}{\alpha} + 1\right) \\
&\quad + \alpha(1-\alpha^2)\left(\tfrac{1}{\alpha} - \tfrac{1}{\alpha} + 1\right) \\
&\quad + (1-\alpha)\alpha^2(0) + (1-\alpha)(1-\alpha^2)(0) \\
&= \alpha^3\left(\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha^2} + 1\right) + \alpha(1-\alpha^2)(1) \\
&= 1 - \alpha + \alpha^3 + \alpha - \alpha^3 \\
&= 1 \\
&= c_{ij}.
\end{aligned}$$

We immediately calculate $\kappa_{ij}^{(1)} := \min\{\bar{v}, \sigma_{ij}^{(2)}, \sigma_{ij}^{(1)}\}$ in each state of the world, as given in Figure E.1. Observing that $\kappa_{ij}^{(1)} = \frac{1}{\alpha} - 1$ with probability α and is negative otherwise, we obtain $\mathbb{E}[(\kappa_{ij}^{(1)})^+] = 1 - \alpha$. □

Lemma E.0.2. *For the bandit (j, i) resulting by orienting the edge so that j is inspected prior to i, the initial Weitzman indices and expected capped values are the following:*

- $\sigma_{ji}^{(1)} = \frac{1}{\alpha^2} - \frac{1}{\alpha}$.
- $\kappa_{ji}^{(1)} = \frac{1}{\alpha^2} - \frac{1}{\alpha}$ *with probability α^3, otherwise nonpositive.*

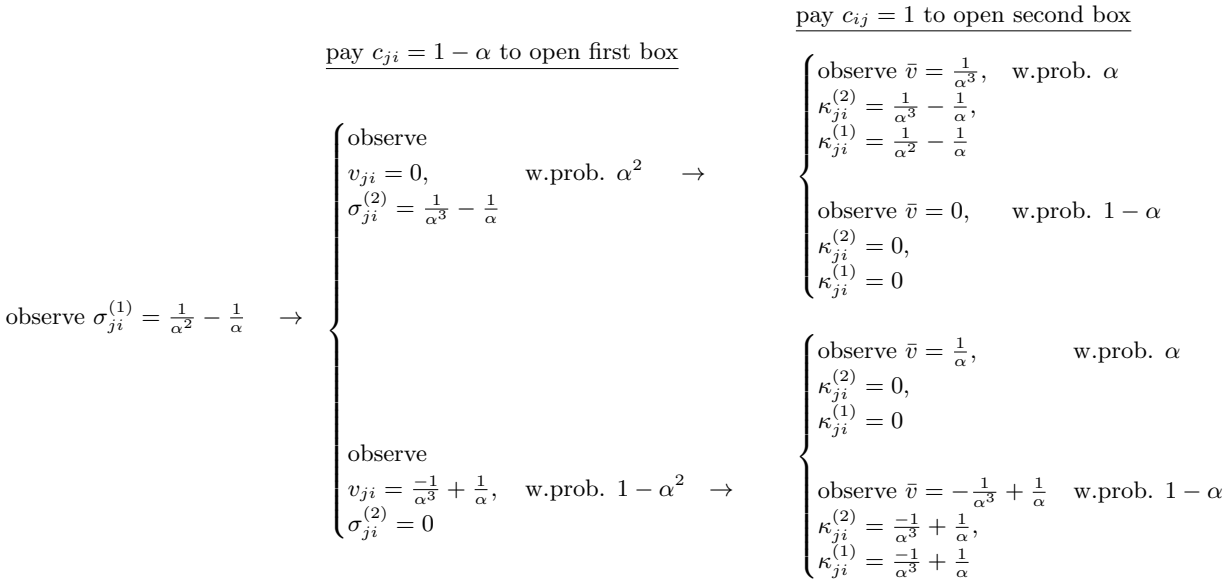


Figure E.2: Reversed No Dessert Bandit (j, i)

The bandit (j, i). Again, $\bar{v} = v_{ij} + v_{ji}$.

- $\mathbb{E}[(\kappa_{ji}^{(1)})^+] = \alpha - \alpha^2$.

Proof. We first show that, conditioned on $v_{ji} = 0$, we have $\sigma_{ji}^{(2)} = \frac{1}{\alpha^3} - \frac{1}{\alpha}$.

$$\begin{aligned}\mathbb{E}\left[(\bar{v} - \sigma_{ji}^{(2)})^+ \;\middle|\; v_{ji} = 0\right] &= \mathbb{E}\left[(v_{ij} - \tfrac{1}{\alpha^3} + \tfrac{1}{\alpha})^+\right] \\ &= \alpha\left(\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha^3} + \tfrac{1}{\alpha}\right) \\ &= 1 \\ &= c_{ij}.\end{aligned}$$

Next, conditioned on $v_{ji} = \frac{-1}{\alpha^3} + \frac{1}{\alpha}$, we have $\sigma_{ji}^{(2)} = 0$.

$$\begin{aligned}\mathbb{E}\left[(\bar{v} - \sigma_{ji}^{(2)})^+ \;\middle|\; v_{ji} = \tfrac{-1}{\alpha^3} + \tfrac{1}{\alpha}\right] &= \mathbb{E}\left[(v_{ij} - \tfrac{1}{\alpha^3} + \tfrac{1}{\alpha})^+\right] \\ &= \alpha\left(\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha^3} + \tfrac{1}{\alpha}\right) \\ &= 1 \\ &= c_{ij}.\end{aligned}$$

Now, we check that Figure E.2 correctly computes $\kappa_{ji}^{(2)} := \min\{\bar{v}, \sigma_{ji}^{(2)}\}$ in each state of the world. Then, to show $\sigma_{ji}^{(1)} = \frac{1}{\alpha^2} - \frac{1}{\alpha}$:

$$\begin{aligned}\mathbb{E}\left[(\kappa_{ji}^{(2)} - \sigma_{ji}^{(1)})^+\right] &= \mathbb{E}\left[(\kappa_{ji}^{(2)} - \tfrac{1}{\alpha^2} + \tfrac{1}{\alpha})^+\right] \\ &= \alpha^3\left(\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha} - \tfrac{1}{\alpha^2} + \tfrac{1}{\alpha}\right) \\ &= \alpha^3\left(\tfrac{1}{\alpha^3} - \tfrac{1}{\alpha^2}\right) \\ &= 1 - \alpha \\ &= c_{ji}.\end{aligned}$$

We now check that Figure E.2 correctly computes $\kappa_{ji}^{(1)} := \min\{\bar{v}, \sigma_{ji}^{(2)}, \sigma_{ji}^{(1)}\}$ in each state of the world. We obtain $\kappa_{ji}^{(1)} = \frac{1}{\alpha^2} - \frac{1}{\alpha}$ with probability α^3, and nonpositive otherwise, so $\mathbb{E}[(\kappa_{ji}^{(1)})^+] = \alpha - \alpha^2$. □

E.1 No Dessert Alternate Proof

For a reader who would prefer to double-check without relying on the Nested Box machinery, we give an alternate proof of the "No-Dessert" Theorem: no deterministic edge-based fixed-orientation algorithm has nontrivial welfare.

Proof of Theorem 4.4.1. The proof relies on a particular edge, shown in Figure 4.2. Every edge-based fixed orientation algorithm either orients that edge in one direction or the other. We will show that depending on the outside option available, either choice can be a fatal mistake.

Instance 1. We consider a graph consisting of just the edge in Figure 4.2.

There are three nontrivial algorithms to consider:

(1) First inspect (i, j), then if $v_{ij} > 0$, continue to inspecting (j, i), then match the edge if $v_{ij} + v_{ji} > 0$. Denote its welfare by $\text{Welf}(i, j)$.

(2) First inspect (j, i), then if $v_{ji} = 0$, continue to (i, j), then match the edge if $v_{ij} + v_{ji} > 0$. Denote its welfare by $\text{Welf}(j, i)$.

(3) Inspect both endpoints, then match the edge if $v_{ij} + v_{ji} > 0$. Denote its welfare by $\text{Welf}(\{i, j\})$. Observe this strategy is available to an algorithm that has oriented the edge in either direction.

We can calculate the $\text{Welf}(i, j)$ as follows. It matches the edge whenever $v_{ij} + v_{ji} > 0$. The possible positive values of $v_{ij} + v_{ji}$ are $\frac{1}{\alpha^3}$, which occurs with probability α^3, and $\frac{1}{\alpha}$, which occurs with probability $\alpha(1 - \alpha^2)$. The probability of paying c_{ij} is 1, and the probability of paying c_{ji} is α. Overall,

$$\begin{aligned}
\text{Welf}(i, j) &= \alpha^3 \left(\frac{1}{\alpha^3}\right) + \alpha(1 - \alpha^2) \left(\frac{1}{\alpha}\right) - 1 - \alpha(1 - \alpha) \\
&= 1 + 1 - \alpha^2 - 1 - \alpha + \alpha^2 \\
&= 1 - \alpha.
\end{aligned}$$

We calculate $\text{Welf}(j,i)$ in the same way. The total value $\frac{1}{\alpha^3}$ is obtained with probability α^3. In all other cases, the edge is not matched. The probability of paying c_{ij} is α^2 and the probability of paying c_{ji} is 1.

$$\begin{aligned}\text{Welf}(j,i) &= \alpha^3\left(\frac{1}{\alpha^3}\right) - \alpha^2(1) - (1-\alpha)\\ &= 1 - \alpha^2 - 1 + \alpha\\ &= \alpha - \alpha^2.\end{aligned}$$

Finally, for $\text{Welf}(\{i,j\})$, whenever $v_{ij} + v_{ji} > 0$ the edge is matched, and both costs are always paid.

$$\begin{aligned}\text{Welf}(\{i,j\}) &= \alpha^3\left(\frac{1}{\alpha^3}\right) + \alpha(1-\alpha^2)\left(\frac{1}{\alpha}\right) - 1 - (1-\alpha)\\ &= 1 + 1 - \alpha^2 - 1 - 1 + \alpha\\ &= \alpha - \alpha^2.\end{aligned}$$

Because $\inf_{\alpha>0} \frac{\text{Welf}(j,i)}{\text{Welf}(i,j)} = \inf_{\alpha>0} \frac{\text{Welf}(\{i,j\})}{\text{Welf}(i,j)} = 0$, we conclude that **no edge-based fixed-orientation algorithm that orients $\{i,j\}$ as (j,i) has a nonzero approximation guarantee on Instance 1.**

Instance 2. We will consider a star graph with i at the center. There is a special edge $\{i,k\}$ with $v_{ik} = \frac{1}{\alpha}$, $c_{ik} = 0$, $v_{ki} = 0$, $c_{ki} = 0$. In other words, i has an "outside option" $\frac{1}{\alpha}$ it can match to at any time. Every other edge in the graph $\{i,j_1\},\ldots,\{i,j_m\}$ is an independent copy of the edge $\{i,j\}$ in Figure 4.2. The number of copies m will be chosen later.

Welfare of (j,i) orientation.

Note that one of the copies improves on the outside option if and only if $v_{ij} = \frac{1}{\alpha^3}$ and $v_{ji} = 0$ (a "success"). To lower-bound the welfare achievable on this instance, we consider the algorithm that orients all copies as (j,i), then iterates through until it finds a success. On each edge, it inspects (j,i), then continues to (i,j) if $v_{ji} = 0$, then stops and matches if $v_{ij} = \frac{1}{\alpha^3}$. If no edge is a success, the algorithm takes the outside option $\frac{1}{\alpha}$.

Define the algorithm's initial welfare as $\frac{1}{\alpha}$, and for each edge, we can calculate the expected **gain** from an edge as the expected increase in the welfare of the algorithm after processing this edge. The algorithm will increase its matched value by $\frac{1}{\alpha^3} - \frac{1}{\alpha}$ on success, will lose the cost $c_{ji} = 1 - \alpha$ with probability 1, and will lose the cost $c_{ij} = 1$ with probability α^2. So for each copy, we have

$$\begin{aligned}\mathbb{E}[\text{gain}] &= \alpha^3 \left(\frac{1}{\alpha^3} - \frac{1}{\alpha}\right) - (1-\alpha) - \alpha^2(1) \\ &= 1 - \alpha^2 - 1 + \alpha - \alpha^2 \\ &= \alpha - 2\alpha^2.\end{aligned}$$

The algorithm obtains this expected gain every time it inspects a copy, and it inspects copies until it runs out or finds a success. If the number of copies is $m = \frac{1}{\alpha^{3-\epsilon}}$ for any $\epsilon > 0$, then the probability of finding a success is $o(1)$, which means the algorithm inspects all of the copies with high probability. So its total expected gain is $(1 - o(1))m(\alpha - 2\alpha^2) = \Omega\left(\frac{1}{\alpha^{2-\epsilon}}\right)$ for $\alpha \to 0$.

Welfare of (i, j) orientation. On the other hand, consider any edge-based fixed-orientation algorithm that orients $\{i, j\}$ as (i, j). An analogous calculation shows that its expected gain is negative:[1]

$$\begin{aligned}\text{gain}(i,j) &= \alpha^3 \left(\frac{1}{\alpha^3} - \frac{1}{\alpha}\right) - 1 - \alpha(1-\alpha) \\ &= 1 - \alpha^2 - 1 - \alpha + \alpha^2 \\ &= -\alpha.\end{aligned}$$

Therefore, the optimal such algorithm does no inspections and takes the outside option, yielding welfare $\frac{1}{\alpha}$. Because the welfare ratio tends to zero as $\alpha \to 0$, we conclude **no edge-based fixed-orientation algorithm that orients $\{i, j\}$ as (i, j) has a nonzero approximation guarantee on Instance 2.** □

[1] The intuition is that both endpoints must "succeed" for this edge to exceed the outside option, so either endpoint has a veto. It is cheaper to inspect (j, i) first because it has a much higher chance of failure.

Appendix F

Proofs for Section 5.3

We prove Lemma 5.3.1, restated here for convenience.

Lemma 5.3.1. *Let $\mathcal{I}$ be an instance of* CMS *and $\mathcal{I}'$ =*MSPtoCOB($\mathcal{I}$)*. For any policy ρ' on $\mathcal{C}_i$, there exists a policy ρ on $\mathcal{M}_i$ such that* $\text{Perf}^{\rho}(\mathcal{M}_i) = \text{Perf}^{\rho'}(\mathcal{C}_i')$ *and the probabilities that ρ and ρ' claim are the same. Furthermore, given any policy ρ on $\mathcal{M}_i$, there exists a policy ρ' on $\mathcal{C}_i$ with the same guarantee.*

Proof. Given a policy ρ' on $\mathcal{C}_i$, construct a policy ρ on $\mathcal{M}_i$ as follows. For every deterministic policy π, let ρ select π with the same probability that ρ' selects bandit B_i^{π}. Whenever ρ' advances from state s, take action $\pi(h)$ from the corresponding history h. If ρ' halts or claims on a state s, halt or claim, respectively, on the corresponding history h. It follows from Definition 5.2.2 that $\mathbb{E}[\text{Perf}^{\pi}(\mathcal{M}_i)] = \mathbb{E}[\text{Perf}^{\rho'}(B_i^{\pi})]$. Note that ρ claims exactly when ρ' respectively claims, so ρ and ρ' claim with the same probability.

Now, given a policy ρ on $\mathcal{M}_i$, construct a policy ρ' on $\mathcal{C}_i$ as follows. Note that ρ WLOG first draws a deterministic policy π from some distribution and then follows π. Construct ρ' by drawing a deterministic policy π according to ρ, selecting bandit B_i^{π}, advancing to a terminal state s, and claiming if $V(s) > 0$. If $V(s) = 0$, claim if and only if π claims on the corresponding history h. By exactly the argument given above, $\mathbb{E}[\text{Perf}^{\rho}(\mathcal{M}_i)] = \mathbb{E}[\text{Perf}^{\rho'}(B_i^{\pi})]$ and ρ' claims with the same probability as ρ. □

We now prove Proposition 5.3.1, restated here for convenience.

Proposition 5.3.1 (CMS to COB Selection)**.** *Let $\mathcal{F}$ be downward-closed. If there exists an online ex-ante α-approximation algorithm* ALG$'$ *for* COB Selection($\mathcal{F}$)*, then there exists an online ex-ante α-approximation algorithm* ALG*, not necessarily efficient, for* CMS($\mathcal{F}$)*.*

Proof. Given an instance $\mathcal{I}$ of CMS($\mathcal{F}$), we construct $\mathcal{I}' = \text{MSPtoCOB}(\mathcal{I})$. Because ALG$'$ is online, it executes an independent policy ρ_i' on each $\mathcal{C}_i$ before either claiming $\mathcal{C}_i$ or proceeding to C_{i+1}. By Lemma 5.3.1, for each ρ_i', there exists a policy ρ_i on $\mathcal{M}_i$ such that $\mathbb{E}[\text{Perf}^{\rho_i'}(\mathcal{C}_i)] = \mathbb{E}[\text{Perf}^{\rho_i}(\mathcal{M}_i)]$. Let ALG be the online algorithm for CMS($\mathcal{F}$) that executes policy ρ_i on MSP $\mathcal{M}_i$ for all $i \in [n]$. We have that

$$\mathbb{E}[\text{Welf}^{\text{ALG}}(\mathcal{I})] = \sum_{i=1}^{n} \mathbb{E}[\text{Perf}^{\rho_i'}(\mathcal{C}_i)] = \sum_{i=1}^{n} \mathbb{E}[\text{Perf}^{\rho_i}(\mathcal{M}_i)] = \mathbb{E}[\text{Welf}^{\text{ALG}'}(\mathcal{I}')].$$

By Lemma 5.3.1, because ALG claims $\mathcal{M}_i$ exactly when ALG$'$ claims $\mathcal{C}_i$, and ALG$'$ is ex-post feasible, ALG is ex-post feasible.

Recall that Lemma 5.3.1 also tells us that, given a policy ρ for $\mathcal{M}_i$, there exists a policy ρ for $\mathcal{C}_i$ with the same performance and probability of claiming. By an identical argument to the above, given an algorithm ALG for CMS($\mathcal{F}$), there is a feasible algorithm ALG$'$ for COB Selection($\mathcal{F}$) with the same expected welfare.

Now, let OPT and OPT$'$ be the optimal ex-ante feasible algorithms for CMS and COB Selection respectively. By Observation 5.2.1, each consists of independently executing policies $\rho_1, \ldots, \rho_n$ on $\mathcal{M}_1, \ldots, \mathcal{M}_n$ (in the case of OPT) or $\rho_1', \ldots, \rho_n'$ on $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (in the case of OPT$'$.) Then, applying the above two arguments, there exists an ex-ante algorithm for CMS($\mathcal{F}$) with the same expected welfare as OPT$'$, and there exists an ex-ante algorithm for COB Selection($\mathcal{F}$) with the same expected welfare as OPT. It follows that $\text{Welf}^{OPT}(\mathcal{I}) = \text{Welf}^{\text{OPT}'}(\mathcal{I}')$.

Given an online ex-ante α-approximation algorithm ALG$'$ for COB Selection($\mathcal{F}$), we can construct an online algorithm ALG for CMS($\mathcal{F}$) with the same expected welfare. We have that

$$\mathbb{E}[\text{Welf}^{\text{ALG}}(\mathcal{I})] = \mathbb{E}[\text{Welf}^{\text{ALG}'}(\mathcal{I}')] \geq \alpha \cdot \mathbb{E}[\text{Welf}^{\text{OPT}'}(\mathcal{I}')] = \alpha \cdot \mathbb{E}[\text{Welf}^{\text{OPT}}(\mathcal{I})],$$

so ALG is an ex-ante α-approximation algorithm for CMS($\mathcal{F}$). □

We construct a policy for COB selection that is used in 5.3.2.

Lemma F.0.1 (non-exposed policy)**.** *For any bandit B with capped value κ, given probability q, there exists a non-exposed policy ρ on B which claims with probability exactly q, and also claims exactly when the value of κ is the top q quantile of its distribution.*

We construct ρ to advance from state s whenever σ_s is in the top q quantile of the distribution of κ. Because our probability distributions are discrete, describing ρ precisely requires a tie-breaking procedure. We set a threshold τ based on the distribution of κ, and then choose, possibly randomly, between the following two policies: either advance from the current state s if and only if $\sigma_s > \tau$, or if and only if $\sigma_s \geq \tau$. We call policies that, for some τ, either advance if and only if $\sigma_s > \tau$ or advance if and only if $\sigma_s \geq \tau$ **threshold-based**. The concept of threshold-based algorithms is explored in detail in Section 5.4. We first prove that all threshold-based policies are non-exposed.

Lemma F.0.2. *All threshold-based policies are non-exposed.*

Proof. Suppose for contradiction that ρ is exposed. Then there is positive probability that ρ advances from a state s and later halts in a state s' such that $\sigma_s < \sigma_{s'}$. But since ρ advanced from state s, we either have that $\tau \leq \sigma_s < \sigma_{s'}$, or $\tau < \sigma_s < \sigma_{s'}$. In either case, $\tau < \sigma_{s'}$ and ρ should have advanced from s', a contradiction. □

We now prove Lemma F.0.1.

Proof of Lemma F.0.1. If there exists $v \in \text{Supp}(\kappa)$ such that $\Pr[\kappa \geq v] = q$, set a threshold $\tau = v$ and advance from state s whenever $\sigma_s \geq \tau$. With probability exactly q, $\kappa = \min_s\{\sigma_s\} \geq \tau$, so $\sigma_s \geq \tau$ for all σ_s, and therefore we claim with probability exactly q.

Otherwise, there is some $v \in \text{Supp}(\kappa)$ such that $\Pr[\kappa \geq v] > q \geq \Pr[\kappa > v]$[1] Set $\tau = v$. With probability $\frac{q - \Pr[\kappa > \tau]}{\Pr[\kappa = \tau]}$, follow the threshold policy of advancing whenever $\sigma_s \geq \tau$. Otherwise, follow the threshold policy of advancing whenever $\sigma_s > \tau$. ρ claims whenever $\kappa > \tau$, and when $\kappa = \tau$,

[1] the second inequality is strict except when $q = 0$, in which case v is the maximum element of $\text{Supp}(\kappa)$ and $\Pr[\kappa > v] = 0$.

claims with probability $\frac{q-\Pr[\kappa>\tau]}{\Pr[\kappa=\tau]}$. Therefore, the probability that ρ claims is

$$\Pr[\kappa > \tau] + \Pr[\kappa = \tau] \cdot \frac{q - \Pr[\kappa > \tau]}{\Pr[\kappa = \tau]} = \Pr[\kappa > \tau] + (q - \Pr[\kappa > \tau]) = q.$$

ρ claims with probability q, and moreover claims exactly when κ is in its top q quantile. By Lemma F.0.2, ρ is non-exposed, as it randomizes between threshold-based policies.

□

Appendix G

Proofs for Section 5.4

G.1 Proof of Proposition 5.4.2 (Ex-ante FPTAS)

In this section, we give an algorithm to compute an ϵ-approximate collection. We recall the definition and the statement of Proposition 5.4.2.

Definition 5.4.2 (ϵ-approximate collection)**.** Given an input $\mathcal{M}_1, \ldots, \mathcal{M}_n$ to the ex-ante CMS($\mathcal{F}$) problem, an *ϵ-approximate collection* is a pair of vectors $(Q, \hat{U})$ satisfying the following:

(1) $Q \in \mathcal{P}_{\mathcal{F}}$, the feasible polytope.

(2) there exists an ex-ante feasible algorithm $\widehat{\mathrm{OPT}}$ that is a $1-\epsilon$ approximation and satisfies $Q_i \geq \Pr[i \in \widehat{\mathrm{OPT}}]$ for all i.

(3) letting U_i be the expected utility $\widehat{\mathrm{OPT}}$ obtains on $\mathcal{M}_i$, $\hat{U}$ approximates U from below, i.e. $\hat{U}_i \leq U_i$ for all i and $\sum_i \hat{U}_i \geq (1-\epsilon)\sum_i U_i$.

Proposition 5.4.2. *When $\mathcal{F}$ is a matroid, there is an algorithm for ex-ante* CMS($\mathcal{F}$) *that ouputs an ϵ-approximate collection in time polynomial in the input size and $\frac{1}{\epsilon}$.*

We first recall the background and define some notation. Recall that $\mathcal{F}$ is a matroid and the input to the problem are MSPs $\mathcal{M}_1, \ldots, \mathcal{M}_n$. We assume WLOG that $\{i\} \in \mathcal{F}$ for all i, as otherwise we can discard i from the input. Recall from Observation 5.2.1 that the optimal ex-ante feasible policy, without loss of generality, commits to a statistically independent policy ρ_i on each $\mathcal{M}_i$.

Given an ex-ante feasible algorithm whose output is a set S, we define the following functions for each $i \in [n]$: letting ρ range over all possibly-randomized policies for $\mathcal{M}_i$,

$$f_i(q) := \max_{\rho} \ \mathbb{E}[\text{Perf}^{\rho}(i)]$$
$$\text{s.t.} \quad \Pr_{\rho}[i \in S] \leq q.$$

The optimal ex-ante feasible algorithm solves $\max_{Q \in \mathcal{P}_{\mathcal{F}}} \sum_{i=1}^{n} f_i(Q_i)$.

Our main ingredient is the following algorithm.

Proposition G.1.1. *There is an algorithm that, on input $c, \epsilon > 0$ and an MSP $\mathcal{M}_i$, runs in time polynomial in the description length of $\mathcal{M}$, in $\log \frac{1}{c}$, and in $\frac{1}{\epsilon}$, and produces an explicitly-represented, monotone, concave function $\hat{f}_i : [0,1] \to \mathbb{R}$ such that, for all $q \in [0,1]$,*

$$\frac{f_i(q) - c}{1+\epsilon} \leq \hat{f}_i(q) \leq (f_i(q) + c)(1+\epsilon).$$

We now take Proposition G.1.1 as given and prove Proposition 5.4.2.

Algorithm G.1.1 (APPROX-COLLECTION)**.**

APPROX-COLLECTION$(\mathcal{M}_1, \ldots, \mathcal{M}_n, \epsilon)$:

(1) Set $\epsilon' := \frac{1}{3} \min\left\{\epsilon, \frac{1}{2}\right\}$.

(2) Compute the expected performance of SAUP$(\mathcal{M}_i, 0)$ for each i, and let W be the largest.

(3) Set the parameter $c := \frac{\epsilon' W}{2n}$.

(4) Run the algorithm of Proposition G.1.1 on each M_i to obtain $\hat{f}_i$.

(5) Solve the convex programming problem $Q := \arg\max_{Q \in \mathcal{P}_{\mathcal{F}}} \sum_{i=1}^{n} \hat{f}_i(Q_i)$.

(6) For each i, let $\hat{U}_i := \max\left\{0, \ \frac{\hat{f}_i(Q_i)}{1+\epsilon} - c\right\}$.

(7) Return $(Q, \hat{U})$.

Proof of Proposition 5.4.2. **(Running time.)** Computing the expected performance of $\mathrm{SAUP}(\mathcal{M}_i, 0)$ takes polynomial time, as the expected performance is computed in the course of constructing the SAUP oracle in Proposition 5.4.1. The algorithm of Proposition G.1.1 is called n times and runs in time polynomial in the length of the given MSP, in $\frac{1}{\epsilon}$, and in $\log \frac{1}{c} = \log \frac{2n}{\epsilon' W}$. The convex programming problem is solvable in polynomial time because it is a maximization of a concave function over a convex polytope, that, because it is a matroid polytope, has an efficient separation oracle [Cunningham, 1984]. We note that, in addition to explicit representations of $\hat{f}_i$, we also have access to the gradients and supergradients of $\hat{f}_i$, a piecewise linear function (see proof of Proposition G.1.1).

(ϵ-approximate collection.) Let OPT^Q be the optimal ex-ante feasible algorithm that claims each i with probability Q_i. First, we note that for any feasible $Q' \in \mathcal{P}_{\mathcal{F}}$, we have

$$\begin{aligned}
\mathrm{OPT}^Q &= \sum_{i=1}^{n} f_i(Q) \\
&\geq \left(\frac{\sum_{i=1}^{n} \hat{f}_i(Q)}{1+\epsilon'} \right) - cn && \text{Proposition G.1.1} \\
&\geq \left(\frac{\sum_{i=1}^{n} \hat{f}_i(Q')}{1+\epsilon'} \right) - cn && \text{definition of } Q \\
&\geq \left(\frac{\sum_{i=1}^{n} f_i(Q')}{(1+\epsilon')^2} \right) - \frac{cn}{1+\epsilon'} - cn && \text{Proposition G.1.1} \\
&\geq (1-2\epsilon')\mathrm{OPT}^{Q'} - cn\frac{2+\epsilon'}{1+\epsilon'} \\
&\geq (1-2\epsilon')\mathrm{OPT}^{Q'} - 2cn \\
&\geq (1-2\epsilon')\mathrm{OPT}^{Q'} - \epsilon' W. && \text{(G.1)}
\end{aligned}$$

We first use (G.1) to derive that $\mathrm{OPT}^Q \geq W/2$. In particular, letting $\bar{Q}$ be the distribution putting probability 1 on a particular i, we have $\mathrm{OPT}^{\bar{Q}} = \mathrm{SAUP}(\mathcal{M}_i, 0) = W$. Therefore,

$$\begin{aligned}
\mathrm{OPT}^Q &\geq (1-2\epsilon')W - \epsilon' W \\
&= (1-3\epsilon')W \\
&\geq \frac{W}{2},
\end{aligned}$$

using that $\epsilon' \leq \frac{1}{6}$. We also have $\mathrm{OPT} \geq W$, as W is the welfare of one feasible solution.

We now show that OPT^Q is a $1-\epsilon$ approximation. Let Q^* be the probabilities for the optimal algorithm, i.e. $\mathrm{OPT} = \mathrm{OPT}^{Q^*}$. By (G.1),

$$\begin{aligned}\mathrm{OPT}^Q &\geq (1-2\epsilon')\mathrm{OPT} - \epsilon' W \\ &\geq (1-3\epsilon')\mathrm{OPT} \\ &\geq (1-\epsilon)\mathrm{OPT}\end{aligned}$$

using that $\epsilon' \leq \epsilon/3$. Finally, we show that $\hat{U}$ approximates U from below. First, by the guarantee of Proposition G.1.1 and $U_i \geq 0$, we obtain $\hat{U}_i \leq f_i(Q_i) = U_i$ for all i. Second, we have

$$\begin{aligned}\sum_{i=1}^n \hat{U}_i &\geq \frac{\sum_{i=1}^n \hat{f}_i(Q_i)}{1+\epsilon'} - cn && \text{definition of } \hat{U} \\ &\geq \frac{\sum_{i=1}^n f_i(Q_i)}{(1+\epsilon')^2} - \frac{cn}{1+\epsilon'} - cn && \text{Proposition G.1.1} \\ &\geq (1-2\epsilon')\mathrm{OPT}^Q - 2cn \\ &= (1-2\epsilon')\mathrm{OPT}^Q - \epsilon' W \\ &\geq (1-2\epsilon')\mathrm{OPT}^Q - \frac{\epsilon'\mathrm{OPT}^Q}{2} \\ &\geq (1-3\epsilon')\mathrm{OPT}^Q \\ &\geq (1-\epsilon)\mathrm{OPT}.\end{aligned}$$

We have shown the output is an ϵ-approximate collection. □

G.1.1 Proof of Proposition G.1.1

Algorithm G.1.1 will compute the following functions:

Definition G.1.1. Given an MSP $\mathcal{M} = (S, A, P, C, V)$, define $g_s(q)$ and $g_{s,a}(q')$ as follows. At a terminal state s, let $g_s(q) = qV(s)$. At a non-terminal state s, let:

$$g_{s,a}(q') = \max_{\{q_{s'}\}} \sum_{s'} P(s'; a, s) g_{s'}(q_{s'}) \qquad \text{s.t. } \sum_{s'} P(s'; a, s) q_{s'} = q'$$

$$g_s(q) = \max_{\lambda, \{q_{s,a}\}} \sum_{a \in A^s} \lambda(a) \left[g_{s,a}(q_{s,a}) - C(a, s) \right] \qquad \text{s.t. } \sum_{a \in A^s} \lambda(a) q_{s,a} = q.$$

For comparison, given any state s, let

$$f_s(q) = \max_{\bar{\pi}} \mathbb{E}[\text{Perf}^{\bar{\pi}}(s)] \text{ s.t. } \Pr[\bar{\pi} \text{ claims}] \leq q.$$

Lemma G.1.1. *$f_s = g_s$.*

Proof. By backward induction. At a terminal state s, the only two possibilities are to claim reward $V(s)$ or not claim. The policy that claims with probability q has expected reward $q \cdot V(s)$.

At a non-terminal state s, we first argue that $g_{s,a}(q')$ is the optimal expected performance achievable by any policy that begins after action a has been chosen in state s, i.e. begins by drawing $s' \sim P(\cdot; a, s)$. Suppose by backward induction that at all possible transition states s', i.e. where $P(s'; a, s) > 0$, we have $g_s(q) = f_s(q)$. The optimal policy $\bar{\pi}$ at s that is required to take action a in state s and must claim with probability at most q' has, for each possible s', some probability $q_{s'}$ of claiming conditioned on transitioning to state s'. It must satisfy $\sum_{s'} P(s'; a, s) q_{s'} = q'$. Furthermore, for each state s', it runs the optimal policy $\bar{\pi}_{s'}$ starting at state s' that claims with probability $q_{s'}$. Therefore, $g_{s,a}(q')$ achieves that optimum, i.e. maximizes expected performance over all policies.

Now, we argue $g_s(q) = f_s(q)$, i.e., g_s optimizes expected performance subject to claiming with probability q. The optimal such policy must first choose to either halt (action $\perp$) or take an action $a \in A^s$, according to some probability distribution λ. In doing so, it incurs a cost $C(a, s)$. For each action a, this optimal policy has some probability $q_{s,a}$ of claiming conditioned on taking action a. It must satisfy $\sum_{a \in A^s \cup \{\perp\}} \lambda(a) q_{s,a} = q$. And conditioned on choosing a, this optimal policy continues by running the policy that maximizes expected performance among all policies that begin after action a has been chosen in state s and that claim with probability $q_{s,a}$. That is, it conditioned on choosing a and paying $C(a, s)$, the optimal policy obtains $g_{s,a}(q_{s,a})$. Therefore, $g_s(q)$ achieves that optimum. □

Lemma G.1.2. *$g_{s,a}$ and g_s are concave and monotone nondecreasing. Furthermore, $g_{s,a}(0) = g_s(0)$ while $g_{s,a}(1)$ and $g_s(1)$ are computable in polynomial time.*

Proof. To show that $g_{s,a}(1)$ and $g_s(1)$ are computable in polynomial time, observe that the performance of an algorithm that never claims cannot be above zero, while an algorithm that always claims can be implemented by SAUP$(\mathcal{M}, 0)$. (see Proposition 5.4.1)

Now, we again use backward induction. At a terminal state s, $g_s(q)$ is linear in q and nondecreasing because $V(s) \geq 0$. Now consider a non-terminal state s. Suppose for each s' reachable from s, g_s is concave and monotone nondecreasing. Then $g_{s,a}$ is monotone nondecreasing because, if $q < q'$ and $\{q_{s'}\}$ is a valid solution to the maximization for $g_{s,a}(q)$, then there is a solution $\{q'_{s'}\}$ for $g_{s,a}(q')$ that satisfies $q'_{s'} \geq q_{s'}$ for all s'. The claim then follows by monotonicity of each $g_{s'}$.

Next, we directly prove that $g_{s,a}(q)$ is concave for all a. Let $\alpha \in [0,1]$ and $q, q' \in [0,1]$. Let $\{q_{s'}\}$ and $\{q'_{s'}\}$ respectively solve the maximization objectives for $g_{s,a}(q)$ and $g_{s,a}(q')$. Then

$$\begin{aligned}\alpha g_{s,a}(q) + (1-\alpha) g_{s,a}(q') &= \sum_{s'} P(s'; a, s) \left[\alpha g_{s'}(q_{s'}) + (1-\alpha) g_{s'}(q'_{s'})\right] \\ &\leq \sum_{s'} P(s'; a, s) g_{s'}(\alpha q_{s'} + (1-\alpha) q'_{s'}). \qquad \text{(G.2)}\end{aligned}$$

And by taking a convex combination of the constraints, they satisfy

$$\begin{aligned}\alpha q + (1-\alpha) q' &= \alpha \sum_{s'} P(s'; a, s) q_{s'} + (1-\alpha) \sum_{s'} P(s'; a, s) q'_{s'} \\ &= \sum_{s'} P(s'; a, s) \left(\alpha q_{s'} + (1-\alpha) q'_{s'}\right).\end{aligned}$$

So the collection $\{\alpha q_{s'} + (1-\alpha) q'_{s'}\}$ is a feasible solution for $g_{s,a}(\alpha q + (1-\alpha) q')$. By (G.2), that collection gives a weakly higher value than $\alpha g_{s,a}(q) + (1-\alpha) g_{s,a}(q')$. As $g_{s,a}(\alpha q + (1-\alpha) q')$ is the maximum over all feasible collections, we have $g_{s,a}(\alpha q + (1-\alpha) q') \geq \alpha g_{s,a}(q) + (1-\alpha) g_{s,a}(q')$.

Next, we consider $g_s(q)$. For each a, the function $q_{s,a} \mapsto g_{s,a}(q_{s,a}) - C(a, s)$ is a concave, monotone increasing function minus a constant, so it is still concave, monotone increasing. Now, g_s is the concave closure of $\{g_{s,a} - C(a, s) : a \in A^s \cup \{\perp\}\}$. That is, it is the pointwise smallest concave function that lies weakly above every $g_{s,a}(q) - C(a, s)$ function. This implies that $g_s(q)$ is monotone increasing and concave. $\square$

We have already shown that f_i is concave:

Lemma G.1.3. *Given an MSP $\mathcal{M}_i$, the function f_i is concave.*

Proof. We have that $f_i = f_{s^*}$, where s^* is the start state of the MSP. By Lemma G.1.1, $f_{s^*} = g_{s^*}$, and by Lemma G.1.2, g_{s^*} is concave. □

Next, we give an efficient approximation guarantee for $f_{s^*} = g_{s^*}$, proving Proposition G.1.1.

Algorithm G.1.2 (APPROX-g)**.**

- Given: $\mathcal{M} = (S, A, P, C, V)$, $\epsilon > 0$. Let $m = |S|$. Define $\alpha = \epsilon/(4m)$.
- Parameter: $c < 1$.
- Output: for each s and each $a \in A^s$, concave functions $\hat{g}_s, \hat{g}_{s,a} : [0, 1] \to \mathbb{R}$, each implemented as an efficient oracle for values and gradients.
- Let $b = \frac{c}{\max_s V(s)}$.
- Let $\Theta = (0, b, b(1+\alpha), b(1+\alpha)^2, \ldots, 1)$. Alternatively, notate the set $\Theta = (\theta_0, \ldots, \theta_T)$.
- By backward induction on the DAG structure:
  - At a terminal state s, we set $\hat{g}_s(q) = qV(s)$.
  - At a non-terminal state s, for each action a, we first compute $\hat{g}_{s,a}(q)$ for each $q \in P$ by solving its concave optimization problem of Definition G.1.1, using as oracles $\hat{g}_{s'}$. We then "iron" $\hat{g}_{s,a}$ as follows. We set $\hat{g}_{s,a}(b) = \min\{\hat{g}_{s,a}(b), b \max_s V(s)\}$. We then enforce monotonicity by setting $\hat{g}_{s,a}(\theta_t) = \max\{\hat{g}_{s,a}(\theta_t), \hat{g}_{s,a}(\theta_{t-1})\}$ for $t = 2, \ldots, T$. We then define $\hat{g}_{s,a}(q)$ to be the concave envelope of $\hat{g}$, which we can compute and represent in closed form efficiently by taking the convex hull of the values computed above.

* Then, at the same non-terminal state s, we first compute $\hat{g}_s(q)$ for each $q \in P$ by solving its own concave optimization problem of Definition G.1.1, using as oracles $\hat{g}_{s,a}$, and ironing in the same way.

Lemma G.1.4. *For all s and a, for all $q \in [0,1]$, we have $\frac{g_{s,a}(q)-c}{1+\epsilon} \leq \hat{g}_{s,a}(q) \leq (g_{s,a}(q)+c)(1+\epsilon)$ and $\frac{g_s(q)-c}{1+\epsilon} \leq \hat{g}_s(q) \leq (g_s(q)+c)(1+\epsilon)$.*

Proof. We use backward induction. If s is a terminal state, g_s is at level 0 of the induction. Any $g_{s,a}$ where the maximum level is $k-1$ over all s' with $P(s';a,s) > 0$, is at level k. Similarly, g_s where the maximum level is $k-1$ over all $g_{s,a}$ is at level k. We use backward induction to prove that, at the kth level of induction, the following hold:

$$q \in \Theta \implies \frac{g_{s,a}(q)-c}{(1+\alpha)^{k-1}} \leq \hat{g}_{s,a}(q) \leq (\hat{g}_{s,a}(q)+c)(1+\alpha)^{k-1} \tag{G.3}$$

$$q \notin \Theta \implies \frac{g_{s,a}(q)-c}{(1+\alpha)^{k}} \leq \hat{g}_{s,a}(q) \leq (\hat{g}_{s,a}(q)+c)(1+\alpha)^{k} \tag{G.4}$$

$$q \in \Theta \implies \frac{g_{s}(q)-c}{(1+\alpha)^{k-1}} \leq \hat{g}_{s}(q) \leq (\hat{g}_{s}(q)+c)(1+\alpha)^{k-1} \tag{G.5}$$

$$q \notin \Theta \implies \frac{g_{s}(q)-c}{(1+\alpha)^{k}} \leq \hat{g}_{s}(q) \leq (\hat{g}_{s}(q)+c)(1+\alpha)^{k}. \tag{G.6}$$

Base case: at a terminal state s, $\hat{g}_s = g_s$, i.e. zero error.

Now consider a non-terminal state at level k of the induction. Consider $g_{s,a}(q)$ (Definition G.1.1). First fix any $q \in \Theta$ and consider the values initially computed by the algorithm before ironing. Let $h_{s,a}(\{q_{s'}\}) = \sum_{s'} P(s';a,s) g_{s'}(q_{s'})$. Consider $\hat{h}_{s,a}(\{q_{s'}\}) = \sum_{s'} P(s';a,s) \hat{g}_{s'}(q_{s'})$. Because it is a convex combination, we have

$$\frac{h_{s,a}(\{q_{s'}\})+c}{(1+\alpha)^{k-1}} \leq \hat{h}_{s,a}(\{q_{s'}\}) \leq (h_{s,a}(\{q_{s'}\})+c)(1+\alpha)^{k-1}.$$

This implies the same for $\hat{g}_{s,a}(q)$, i.e. (G.3).

Now, consider values computed at $q \in \Theta$ after enforcing monotonicity and taking the concave envelope. If $q = 0$, then $\hat{g}_{s,a}(0) = g_{s,a}(0) = 0$. If $q = b$, then we either do not change $\hat{g}_{s,a}(b)$ or decrease it to equal $b \max_s V(s)$. As this is the maximum possible value of $g_{s,a}(b)$, we have not worsened the approximation guarantee of (G.3).

For other $q \in \Theta$, we maintain the guarantee (G.3), as follows. In case 1, $\hat{g}_{s,a}(q)$ has not changed and (G.3) holds. In case 2, $\hat{g}_{s,a}(q)$ has been set equal to $\hat{g}_{s,a}(q/(1+\alpha))$ to maintain monotonicity. In this case, since the value has been raised, it still lies above $g_{s,a}(q)(1-\alpha)$, and we still have $\hat{g}_{s,a}(q) \leq \hat{g}_{s,a}(q/(1+\alpha)) \leq g_{s,a}(q/(1+\alpha))(1+\alpha)^{k-1} \leq g_{s,a}(q)(1+\alpha)^{k-1}$ by induction on $q \in \Theta$. In case 3, $\hat{g}_{s,a}(q)$ has changed to be larger so that it equals the convex combination of $\hat{g}_{s,a}$ at some $q', q'' \in \Theta$. Then the approximation applies to $\hat{g}_{s,a}(q'), \hat{g}_{s,a}(q'')$ and we obtain (G.3) as well.

Now fix any $q \in [b, 1], q \notin \Theta$. Let $x < q < y$ for $x, y \in \Theta$ with $y = x(1+\alpha)$. Note that because $g_{s,a}$ is concave and monotone, and because $g_{s,a}(0) = 0$, we have $dg_{s,a}(q) \geq g_{s,a}(q)/q$, where $dg_{s,a}$ is any supergradient. This implies $g_{s,a}(y) \leq g_{s,a}(x)(y/x) = g_{s,a}(x)(1+\alpha)$. By monotonicity, $g_{s,a}(q) \leq g_{s,a}(y) \leq g_{s,a}(x)(1+\alpha) \leq \hat{g}_{s,a}(x)(1+\alpha)^k \leq \hat{g}_{s,a}(q)(1+\alpha)^k$. Similarly, $g_{s,a}(q) \geq g_{s,a}(x) \geq g_{s,a}(y)/(1+\alpha) \geq \hat{g}_{s,a}(y)/(1+\alpha)^k \geq \hat{g}_{s,a}(q)/(1+\alpha)^k$.

Finally, consider any $q \in (0, b)$. We claim that $g_{s,a}$ and g_s are $\max_s V(s)$ Lipschitz. The result follows because each $g_{s,a}$ cannot have a steeper slope than one of the $g_{s'}$ than it depends on, and so forth, and the steepest slope at any terminal state is $\max_s V(s)$. It therefore holds that $g_{s,a}(q) \in [0, b \max_s V(s)]$, and we also have $[\hat{g}_{s,a}(q) \in 0, b \max_s V(s)]$. So (G.4) holds, even if we remove the $(1+\alpha)^k$ term.

We now repeat the argument at $\hat{g}_s$. Although the optimization problem is slightly different, the arguments are all exactly the same. In particular, we just utilize $h_s(\lambda, \{q_{s,a}\}) = \sum_a \lambda(a) g_{s,a}(q_{s,a})$ and $\hat{h}_s$ in the exactly analogous way, and then the entire argument is the same. This completes the induction.

Finally, as $|S| = m$, there can be at most $2m$ levels of induction, so using that $(1+\alpha)^{2m} \leq e^{2\alpha m} = e^{\epsilon/2} \leq (1+\epsilon)$ completes the proof. □

Proof of Proposition G.1.1. Lemma G.1.4 implies the accuracy result. Considering the running time of the algorithm, for each state and action, we build a piecewise linear concave function $\hat{g}_{s,a}$ with $T+2$ pieces, where T is the floor of the solution to $b(1+\alpha)^T = 1$, i.e. $T = O(\frac{1}{\alpha} \ln(\frac{1}{b})) =$

$O(\frac{|S|}{\epsilon}\ln(\frac{\max_s V(s)}{b}))$. For each piece of the piecewise function, we solve a concave optimization problem subject to a linear constraint, using constant-time oracles for concave functions (including access to supergradients) that have already been constructed, and do polynomial-time post-processing. The construction of $\hat{g}_s$ is efficient by a similar argument. We thus build a polynomial number of functions, each taking polynomial time. □

G.1.2 The FPTAS

We now show that the analysis above yields a Fully-Polynomial Time Approximation Scheme (FPTAS) for the optimal ex-ante feasible algorithm for CMS($\mathcal{F}$), for any matroid $\mathcal{F}$. (In fact, the result extends to any downward-closed constraint for whose feasibility polytope there exists an efficient separation oracle.)

Proposition G.1.2. *There is an algorithm running in time polynomial in the lengths of $\mathcal{M}_1, \ldots, \mathcal{M}_n$ and in $\frac{1}{\epsilon}$ that, for any $\epsilon > 0$, guarantees a $1-\epsilon$ approximation to the optimal ex-ante feasible algorithm for CMS($\mathcal{F}$).*

Sketch. Our algorithms compute a vector $Q \in \mathcal{P}_{\mathcal{F}}$ such that, if we optimally interact with each $\mathcal{M}_i$ subject to claiming with probability Q_i, then we achieve a $1-\epsilon$ approximation to the problem $\max_Q \sum_{i=1}^n \hat{f}_i(Q)$. Within each $\mathcal{M}_i$, at the source state s, we compute a probability distribution λ_s over available actions and a probability $q_{s,a}$ for each action such that **(a)** $\sum_a \lambda_s(a) q_{s,a} = Q_i$, and **(b)** it is optimal to draw an action a from λ_s, then proceed optimally subject to claiming with probability $q_{s,a}$. Our algorithm recursively computes such a distribution at every intermediate state, so it implements the optimal algorithm for the given Q and approximations $\hat{f}_i$. We lose another factor of $1-\epsilon$ because $\hat{f}_i$ only approximates f_i. □

G.2 Proof of Proposition 5.4.3

In this section, we prove:

Proposition 5.4.3. *Let $\mathcal{F}$ be a matroid. Given $\mathcal{M}_1, \ldots, \mathcal{M}_n$, let $(Q, \hat{U})$ be an ϵ-approximate collection for any $\epsilon \geq 0$. There is a polynomial-time online algorithm* ALG *for* CMS($\mathcal{F}$) *that takes as input $(Q, \hat{U})$, interacts with each arriving $\mathcal{M}_i$ only through calls to* SAUP($\mathcal{M}_i, \tau$)*, and guarantees* $\mathbb{E}[\text{ALG}] \geq \frac{1}{2}\sum_{i=1}^{n} \hat{U}_i$.

We will reduce to the simpler Choice-Over-Rewards Selection setting. Recall that in this setting, each alternative i is in the form of a "cabinet" with some number of closed "drawers". Each drawer j contains a reward X_i^j, and only one drawer may be opened. The rewards in the drawers may be correlated. We summarize the input as the collection (X_i^j), representing a list of random variables ranging over all $i \in [n]$ and each "drawer" j of each i.

Our algorithm will interact with the instance through two oracles. The first is a SAUP oracle, as we will construct threshold-based algorithm. The second is an ϵ-approximate collection $(Q, \hat{U})$ that describes an approximate solution to the ex-ante optimum. We now note how these definitions apply in the special case of the COR Selection setting.

Definition G.2.1 (Threshold-based, COR Selection)**.** In the COR Selection setting, an online algorithm is threshold-based (Definition 5.4.1) if it follows the following format for each arrival i:

- Set a threshold τ_i.
- Solve the Single-Agent Utility Problem on i with τ_i:
  - Pick the drawer j maximizing $\mathbb{E}[(X_i^j - \tau)^+]$, open it, and claim i if and only if $X_i^j \geq \tau$.

The definition of an ϵ-approximate collection is also the same, but we state it in the COR Selection setting for convenience.

Definition G.2.2 (ϵ-approximate collection, COR Selection)**.** Given an input (X_i^j) to the ex-ante COR Selection($\mathcal{F}$) problem, an ϵ-approximate collection (Definition 5.4.2) is a pair of vectors $(Q, \hat{U})$ satisfying the following. First, $Q \in \mathcal{P}_{\mathcal{F}}$, the feasible polytope. Second, there exists an ex-ante feasible algorithm $\widehat{\text{OPT}}$ that is a $1 - \epsilon$ approximation and satisfies $Q_i \geq \Pr[i \in \widehat{\text{OPT}}]$ for

all i. Third, letting U_i be the expected utility $\widehat{\text{OPT}}$ obtains on i, $\hat{U}$ approximates U from below, i.e. $\hat{U}_i \leq U_i$ and $\sum_{i=1}^n \hat{U}_i \geq (1-\epsilon)\sum_{i=1}^n U_i$.

We now give our "matroid prophet" algorithm for the COR Selection setting. The input is the instance (X_i^j), an ϵ-approximate collection $(Q, \hat{U})$ along with an oracle for SAUP. The following algorithm and analysis contains many ideas from existing matroid prophet algorithms such as those contained in Kleinberg and Weinberg [2012], Lee and Singla [2018], but with some subtle changes as well.

First, some notation. We will define a set of random variables $Z = (Z_1, \ldots, Z_n)$. For that set, define for any $S \in \mathcal{F}$:

$$R(S) := \max_{S'} \left\{ \sum_{i \in S'} Z_i \;\middle|\; S' \cap S = \emptyset, S' \cup S \in \mathcal{F} \right\}.$$

$R(S)$ may be interpreted as the optimal total value that can be added to S while remaining feasible, with value measured by Z.

Algorithm G.2.1 (matroid − prophet)**.** The matroid − prophet algorithm takes an ϵ-approximate collection $(Q, \hat{U})$ for some $\epsilon \geq 0$ and an oracle for the SAUP problem. The algorithm is defined in both the CMS and COR Selection settings because it only interacts with the arrivals $\mathcal{M}_1, \ldots, \mathcal{M}_n$ through the given SAUP oracle.

matroid − prophet$((X_i^j), (Q, \hat{U}), \text{SAUP}(\cdot,\cdot))$:

- Define $\hat{z}_i := \frac{\hat{U}_i}{Q_i}$.
- Compute a distribution D_Q on $2^{[n]}$ satisfying $Q = \mathbb{E}_{F \sim D_Q} 1_F$, where 1_F is the indicator vector for $i \in F$.
- Define Z as follows: sample $F \sim D_Q$, then set $Z_i = \hat{z}_i \mathbb{1}[i \in F]$.
- Initialize $A_0 = \emptyset$, and for each arrival $i = 1, \ldots, n$:
  - If $A_{i-1} \cup \{i\} \notin \mathcal{F}$, the set $\tau_i = \infty$, discard i, set $A_i = A_{i-1}$, and continue to $i+1$.

* Otherwise, set $\tau_i = \frac{1}{2}\,\mathbb{E}_Z\left[R(A_{i-1}) - R(A_{i-1} - A_i)\right]$.

* Run SAUP(i, τ_i). If it claims, set $A_i = A_{i-1} \cup \{i\}$, else set $A_i = A_{i-1}$.

Lemma G.2.1. *If $\mathcal{F}$ is a matroid, then the algorithm runs in polynomial time.*

Proof. We can Computed D_Q in polynomial time with convex programming, obtaining an explicit representation of D_Q (i.e. a probability distribution as a vector of probabilities, necessarily of at most polynomial size). This is because there is an efficient separation oracle for any matroid polytope [Cunningham, 1984], and by using such an oracle, one can obtain $\mathcal{D}_Q$ explicitly in polynomial time for any polytope (e.g. Theorem 4 of Cai et al. [2017], citing older work).

Other than solving the SAUP problem, for which we have an oracle, the only other nontrivial step is computing τ_i. Because D_Q is explicitly represented, we simply sum over the support of D_Q and enumerate all realizations of $F \sim D_Q$, which determines all possible realizations of the vector Z, in order to compute the sum. □

We now turn to analyzing the approximation guarantee. For this, we first introduce more notation. Given the instance (X_i^j) and the vector $Q \in \mathcal{P}_{\mathcal{F}}$, define U_i to be the utility obtained on arrival i by OPT^Q, the optimal algorithm that claims each i with probability at most Q_i. Define $z_i = \frac{U_i}{Q_i}$. Recall that by definition of an ϵ-approximate collection, we have $\hat{U}_i \leq U_i$ for all i.

The algorithm OPT^Q, WLOG, opens a drawer $j^*(i) \sim \lambda$ for each arrival i according to some fixed probability distribution λ, independently of all other arrivals. Let $X_i := X_i^{j^*(i)}$, a random variable representing the reward[1] observed by OPT^Q on arrival i.

Reconsider the random variables F and Z defined in the algorithm. For the purpose of analysis, we additional define the "re-randomized" random variables $Y = (Y_1, \ldots, Y_n)$ as follows: If $i \in F$, then draw Y_i from the distribution of X_i conditioned on being in its top Q_i quantile; otherwise, draw Y_i from the distribution of X_i conditioned on not being in the top Q_i quantile.

We note that Y_i has the same marginal distribution as X_i, and that $\mathbb{E}[Y_i \mid i \in F] = \frac{U_i}{Q_i} = z_i$.

[1] We note that the remainder of the analysis from this point would be much more cumbersome in the CMS setting as opposed to the COR Selection setting that we are considering.

For the purpose of analysis, we also consider an independent copy (F', Z') of (F, Z) drawn from the same distribution. Define $R'(S)$ analogously to $R(S)$, but for Z'.

Lemma G.2.2. *For any matroid $\mathcal{F}$,* matroid – prophet *run on an ϵ-approximate collection $(Q, \hat{U})$ with a* SAUP *oracle has an expected performance of at least $\frac{1}{2}\sum_{i=1}^{n} \hat{U}_i$.*

Proof. For shorthand, write $X := (X_i^j)$. Let $j(i)$ denote the drawer of arrival i opened by our algorithm. Let $\text{Util}_i(X) = (X_i^{j(i)} - \tau_i)^+$, interpreted as the utility of agent i who opens drawer $j(i)$ and pays τ_i for the reward iff it exceeds τ_i. Let $\text{Util}(X) = \sum_{i=1}^{n} \text{Util}_i(X)$. Note that because $i \in A_n \iff X_i^{j(i)} \geq \tau_i$, where A_n is the algorithm's claimed set, we also have $\text{Util}(X) = \sum_{i \in A_n} \text{Util}_i(X)$. Therefore, for any fixed realization of X,

$$\begin{aligned}
\text{Welf}^{\text{ALG}} &= \left(\sum_{i \in A_n} \tau_i\right) \;+\; \text{Util}(X) \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{Z} \sum_{i \in A_n} \left(R(A_{i-1}) - R(A_{i-1} \cup \{i\})\right) \;+\; \text{Util}(X) \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{Z} \sum_{i \in A_n} \left(R(A_{i-1}) - R(A_i)\right) \;+\; \text{Util}(X) \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{Z} \left[R(A_0) - R(A_n)\right] \;+\; \text{Util}(X). \qquad \text{(G.7)}
\end{aligned}$$

We first lower-bound the second term, $\text{Util}(X)$. First, we use the definition of SAUP to observe that $\text{Util}_i(X) \geq \mathbb{E}\,(X_i - \tau_i)^+$. This uses independence across arrivals and holds for all fixed realizations of all prior arrivals $i' = 1, \ldots, i-1$ and all choices of τ_i, so it holds in expectation.

Next, we use the fact that X_i is independent of τ_i and that Y_i has the same marginal distribution to conclude that $\mathbb{E}(X_i - \tau_i)^+ = \mathbb{E}(Y_i - \tau_i)^+$. Therefore, we have $\text{Util}_i(X) \geq \mathbb{E}(Y_i - \tau_i)^+$.

Now, given realizations of X, F, Z, define $V = \arg\max_{S'} \left\{\sum_{i \in S'} Z_i \middle| S' \cap A_n = \emptyset, S' \cup A_n \in \mathcal{F}\right\}$. In other words, V is the set whose total weight is $R(A_n)$. We will use that, WLOG, $i \in V$ iff $Z_i = \hat{z}_i$,

as otherwise $Z_i = 0$.

$$\begin{aligned}
\mathbb{E}\,\mathrm{Util}(X) &\geq \mathbb{E}\sum_{i=1}^{n}(Y_i - \tau_i)^+ \\
&\geq \mathbb{E}\sum_{i\in V}(Y_i - \tau_i)^+ \\
&\geq \mathbb{E}\sum_{i\in V}(Y_i - \tau_i) \\
&= \mathbb{E}\sum_{i\in V} Y_i \;-\; \mathbb{E}\sum_{i\in V}\tau_i \\
&= \mathbb{E}\sum_{i\in V}\mathbb{E}[Y_i \mid i \in F] \;-\; \mathbb{E}\sum_{i\in V}\tau_i \\
&= \mathbb{E}\sum_{i\in V} z_i \;-\; \mathbb{E}\sum_{i\in V}\tau_i \\
&\geq \mathbb{E}\sum_{i\in V} \hat{z}_i \;-\; \mathbb{E}\sum_{i\in V}\tau_i \\
&= \mathbb{E}\sum_{i\in V} Z_i \;-\; \mathbb{E}\sum_{i\in V}\tau_i \\
&= \mathbb{E}\,R(A_n) \;-\; \mathbb{E}\sum_{i\in V}\tau_i. \qquad \text{(G.8)}
\end{aligned}$$

(G.9)

Plugging (G.8) into (G.7), we obtain

$$\mathrm{Welf}^{\mathrm{ALG}} \geq \frac{1}{2}\,\mathbb{E}\,R(\emptyset) + \frac{1}{2}\,\mathbb{E}\,R(A_n) - \mathbb{E}\sum_{i\in V}\tau_i. \qquad \text{(G.10)}$$

We now claim the following.

Lemma G.2.3. $\mathbb{E}\sum_{i\in V}\tau_i \leq \frac{1}{2}\,\mathbb{E}\,R(A_n)$.

Proof. Let (F', Z') be drawn as an independent copy of (F, Z), and let and R' be the corresponding marginal gain function. Then

$$\mathbb{E}\sum_{i\in V}\tau_i = \mathop{\mathbb{E}}_{X,F}\sum_{i\in V}\frac{1}{2}\mathop{\mathbb{E}}_{F'}\left[R'(A_{i-1}) - R'(A_{i-1}\cup\{i\})\right].$$

Fix any realizations of X, F, F'. Proposition 2 of Kleinberg and Weinberg [2012] gives the following: for every matroid $\mathcal{F}$, for every list of n numbers $(z_1, \ldots, z_n)$, and for every pair of

disjoint sets A, V with $A \cup V \in \mathcal{F}$, letting $A_i = A \cap [i]$, defining $F^* = \max_{S \in \mathcal{F}} \sum_{i \in S} z_i$ and $R'(S) = \max_{S' \subseteq F^*} \left\{ \sum_{i \in S'} z_i \ : \ S' \cap A = \emptyset, S' \cup A \in \mathcal{F} \right\}$, we have

$$\sum_{i \in V} \left[R'(A_{i-1}) - R'(A_{i-1} \cup \{i\}) \right] \leq R(A_n). \quad \text{Proposition 2 of Kleinberg and Weinberg [2012]}$$

Applying this to our V, Z', R', A, and taking expectations,

$$\frac{1}{2} \mathbb{E} \sum_{i \in V} \left[R'(A_{i-1}) - R'(A_{i-1} \cup \{i\}) \right] \leq \frac{1}{2} \mathbb{E}\, R'(A_n).$$

□

Plugging Lemma G.2.3 into (G.10), we obtain:

$$\begin{aligned}
\text{Welf}^{\text{ALG}} &\geq \frac{1}{2} \mathbb{E}\, R(\emptyset) + \frac{1}{2} \mathbb{E}\, R(A_n) - \frac{1}{2} \mathbb{E}\, R(A_n) \\
&= \frac{1}{2} \mathbb{E}\, R(\emptyset) \\
&= \frac{1}{2} \mathbb{E} \sum_{i \in F} Z_i \\
&= \frac{1}{2} \sum_{i=1}^{n} Q_i \hat{z}_i \\
&= \frac{1}{2} \sum_{i=1}^{n} \hat{U}_i.
\end{aligned}$$

□

We can now prove Proposition 5.4.3, that Algorithm G.2.1 (matroid – prophet), run on a CMS instance and $(Q, \hat{U})$, is efficient and provides an approximation guarantee relative to $\hat{U}$.

Proof of Proposition 5.4.3. Efficiency is Lemma G.2.1.

Let $\mathcal{I}$ be an instance of CMS, let $\mathcal{I}' = \text{MSPtoCOB}(\mathcal{I})$, and let $\mathcal{I}'' = \text{COBtoCOR}(\mathcal{I}')$. Let $(Q, \hat{U})$ be an ϵ-approximate collection for $\mathcal{I}$. Then it is also an ϵ-approximate collection for $\mathcal{I}''$, by the reductions of Lemmas 5.3.1 and 5.3.2.

Given $(Q, \hat{U})$, let ALG be the CMS version of matroid – prophet run on $\mathcal{I}$, while ALG$''$ is the COR Selection version of matroid – prophet run on $\mathcal{I}''$. Then $\mathbb{E}\, \text{Welf}^{\text{ALG}}(\mathcal{I}) = \mathbb{E}\, \text{Welf}^{\text{ALG}''}(\mathcal{I}'')$, because for each arrival i and threshold τ_i, SAUP(i, τ_i) has the same expected performance and

probability of claiming. In fact, both obtain performance $\max_\pi \mathbb{E}[(\kappa_i^\pi - \tau_i)^+]$, where κ_i^π is the capped value of the bandit that mimics following policy π on $\mathcal{M}_i$, and the maximum is over all deterministic policies; see Proposition J.0.1 for details.

Lemma G.2.2 proves $\mathbb{E}\,\mathrm{Welf}^{\mathrm{ALG}''}(\mathcal{I}'') \geq \frac{1}{2}\sum_{i=1}^n \hat{U}_i$, proving the claim. □

Appendix H

Hardness of CMS

The problem of Pandora's Non-Obligatory Inspection, introduced by Doval [2018], can be naturally expressed as a special case of CMS. In the **Pandora's Non-Obligatory Inspection** (PNOI) problem, an algorithm is presented with n boxes $\{1, \ldots, n\}$. Box i contains a value v_i distributed according to a distribution D_i, and has an associated inspection cost c_i to open. At each step, the algorithm may open some box i, paying its cost c_i and revealing its value v_i, drawn independently from D_i, claim the value of a box that has previously been opened and halt, or claim the value in an unopened box i without paying the inspection cost c_i and halt. We denote the welfare of an algorithm ALG for an instance $\mathcal{I}$ of PNOI by $\text{Welf}^{\text{ALG}}(\mathcal{I})$, in keeping with our other notation. Any PNOI instance in which every D_i has finite support can be represented as a special case of CMS, with one small subtlety regarding claiming an unopened box.

Let $\mathcal{F}$ be the rank-one matroid constraint $\mathcal{F} = \{\emptyset, \{1\}, \{2\}, \ldots, \{n\}\}$. That is, we may only select a single item. Given an instance $\mathcal{I}$ of PNOI, we construct an instance $\mathcal{I}'$ of CMS($\mathcal{F}$) as follows.

For each box i, construct an MSP $\mathcal{M}_i$ with two legal actions from the start state s_i^*, "inspect" and "take". Create terminal states t_i and $s_i^1, \ldots, s_i^{d_i}$ for each of the d_i possible values in $\text{Supp}(D_i)$. The "inspect" action has cost c_i and transitions to a terminal state s_i^k with probability $\Pr[D_i = v_i^k]$, for every $v_i^k \in \text{Supp}(D_i)$. The "take" action deterministically transitions to state t_i with value $\mathbb{E}[D_i]$ and incurs no cost. Note that because the "take" action is free and its transition function is deterministic, WLOG, any algorithm that chooses action "take" on some $\mathcal{M}_i$ immediately claims

$\mathcal{M}_i$.

Fu et al. [2023] showed that computing the optimal policy for PNOI is NP-hard, and specifically that computing optimal policies on a subclass of "low-cost-low-return-support-3" (LCLRS3) problems is NP-hard. Crucially, this set of instances only uses boxes with three possible realizations, allowing construction of a CMS instance of size polynomial in the number of boxes of the PNOI instance. It follows that CMS is NP-hard.

Appendix I

Efficiency of the Inefficient Reduction

In Section 5.3, we showed that any ex-ante prophet inequality translates to an online approximation algorithm for CMS($\mathcal{F}$). However, the latter is not generally computationally efficient. In this section, we provide more detail on which steps of the reduction can be made efficient generically.

As summarized in Figure 5.2, we use a sequence of reductions:

- In Proposition 5.3.1, we reduce CMS($\mathcal{F}$) to COB Selection($\mathcal{F}$) (Choice-Over-Bandits Selection). In general, this reduction is not efficient because the size of the instance is increased exponentially: for each MSP with a DAG state graph, we create a choice-over-bandits process with one choice for each path in the original DAG.

- In Proposition 5.3.2, we reduce COB Selection($\mathcal{F}$) to COR Selection($\mathcal{F}$) (Choice-Over-Rewards Selection). It is almost immediate that this reduction is efficient (as shown below).

- In Proposition 5.3.3, we reduce COR Selection($\mathcal{F}$) to a classic ex-ante prophet inequality with constraint $\mathcal{F}$. Nontrivially, this reduction turns out to be efficient for certain classes of $\mathcal{F}$.

Therefore, we will show the following result. Here a prophet inequality algorithm is **monotone**[1] if, when it observes a value and then claims it, it would also have claimed any strictly larger value; see e.g. Kleinberg and Weinberg [2012].

[1] For any non-monotone prophet inequality, there exists a monotone one with the same guarantee. We do not know of any cases where a non-monotone but computationally efficient algorithm is known, yet a monotone computationally efficient algorithm with the same performance is unknown. It seems unlikely, but we have not ruled out the possibility for all feasibility structures.

Proposition I.0.1. *If the polytope $\mathcal{P}_{\mathcal{F}}$ admits a polynomial-time separation oracle, and there is a polynomial-time monotone ex-ante α prophet inequality for constraint $\mathcal{F}$, then there is a polynomial-time online α-approximation algorithm for Choice-Over-Bandits Selection (*COB Selection($\mathcal{F}$)*).*

Proof. Suppose we have a monotone polynomial-time ex-ante α prophet inequality.

First, in Proposition I.0.2 below, we show that there is a polynomial-time algorithm for computing the ex-ante optimal algorithm for COR Selection($\mathcal{F}$) and in particular the associated probability distribution λ_i from which to pick a drawer on each arriving "cabinet" i. We now observe that, given $(\lambda_1, \ldots, \lambda_n)$, our reduction in Proposition 5.3.3 is immediately computationally efficient, as it simply opens a drawer according to λ_i and feeds the observed value to the prophet-inequality algorithm. This completes the efficient alogrithm for COR Selection($\mathcal{F}$).

We now produce an algorithm for COB Selection($\mathcal{F}$). First, following Observation 5.3.1, we can make the monotone algorithm for COR Selection($\mathcal{F}$) quantile-based. We can furthermore do so efficiently: we simply compute the probability q_i^j with which i is claimed conditioned on drawer j being opened, and modify the algorithm to claim iff X_i^j is in its top q_i^j quantile (see Observation 5.3.1).

Now, for COB Selection($\mathcal{F}$), we simply check that the reduction of Proposition 5.3.2 is already efficient.[2] When i arrives, we draw $j \sim \lambda_i$ computed by the COR Selection($\mathcal{F}$) algorithm above. Then, we simply claim bandit process B_i^j if its capped Weitzman index κ_i^j is in its top q_i^j quantile. As discussed in Proposition 5.3.2, this is achievable by simply advancing the bandit until the Weitzman index drops below a fixed threshold (possibly with random tiebreaking), claiming iff the index remains above. □

We now complete this section by stating and proving Proposition I.0.2. In particular, we compute $\Lambda = (\lambda_1, \ldots, \lambda_n)$, the set of distributions from which ex-ante OPT select a "drawer" from each "cabinet". This turns out to be a convex programming problem over the polytope $\mathcal{P}_{\mathcal{F}}$. Therefore, we can efficiently calculate Λ and the ex-ante OPT policy for COR Selection for

[2] The reduction requires the COR Selection($\mathcal{F}$) algorithm to be quantile-based, which we have ensured above.

downward-closed constraints such as matroids where there exists an efficient separation oracle.

Proposition I.0.2. *Let $\mathcal{F}$ be downward-closed such that there exists a polynomial-time separation oracle for $\mathcal{P}_{\mathcal{F}}$. Given an instance $\mathcal{I}$, let λ_i be the distribution from which the ex-ante optimal algorithm* OPT *for $\mathcal{I}$ selects a drawer $j^*(i)$ on* COR $\mathcal{C}_i$*. There is a polynomial-time algorithm that, given $\mathcal{I}$, computes the ex-ante* OPT *algorithm and in particular computes $(\lambda_1, \dots, \lambda_n)$.*

Proof. By Observation 5.2.1, ex-ante OPT WLOG executes an independent policy ρ_i on each $\mathcal{C}_i$. By Observation 5.3.1, each ρ_i is WLOG quantile-based, selecting X_i^j if and only if its realization is in the top q_i^j quantile of its distribution for some $q_i^j \in [0,1]$. Define $g_i^j(q) := q\,\mathbb{E}[X_i^j \mid X_i^j \text{ in top } q \text{ quantile}]$, the "upper expectation" of X_i^j for quantile q. We first claim that g_i^j is a concave function (this fact is also asserted in Feldman et al. [2016b], Section 4.1). To see this, let $G(q) = x$ such that[3] $\Pr[X_i^j \le x] = 1 - q$. In other words, $G(q) = F^{-1}(1-q)$ where F is the CDF. We have $g_i^j(q) = \int_{y=q}^1 dG(y)$; this is a version of the fact $\mathbb{E}[X_i^j] = \int_0^\infty (1 - F(x))dx$. And G is a positive, monotone decreasing function, so (noting the bounds on the integral) g_i^j is concave.

Now, given a fixed Q_i, the optimal algorithm on cabinet i that claims it independently with probability Q_i solves

$$f_i(Q_i) = \max_{\lambda_i \in \Delta_{[m_i]}, (q_i^j : j \in [m_i])} \sum_{j \in [m_i]} \lambda_i(j) g_i^j(q_i^j),$$

subject to $\sum_{j \in [m_i]} \lambda_i(j) q_i^j = Q_i$. We can rephrase. Let $\beta_i \in \Delta_{[0,1]}$ be a distribution over points $q \in [0,1]$ satisfying that $\mathbb{E}_{q \sim \beta_i} q = Q_i$. Define

$$\hat{f}_i(Q_i) = \max_{\beta_i \in \Delta_{[0,1]}} \mathop{\mathbb{E}}_{q \sim \beta_i} \max_{j \in [m_i]} g_i^j(q).$$

We claim that $\hat{f}_i = f_i$. We have $\hat{f}_i(q) \ge f_i(q)$ because any valid solution $(\lambda_i, \{q_i^j\})$ can be rephrased as a distribution β_i. But because each g_i^j is concave, WLOG, β_i is never supported on multiple points where the same g_i^j achieves the maximum; it is better by Jensen's inequality to put all their weight on the average of the points. So the solution β_i is WLOG a distribution over m_i points $\{q_i^j\}$ where $j = \text{argmax}_{j'} g_i^{j'}(q_i^j)$, so f_i can achieve the same value as $\hat{f}_i$. In fact, $\hat{f}_i$ is just the concave

[3] If there are multiple x where this is true, they lie in a region where X_i^j has zero density and any choice of x in this region is valid for the argument.

envelope of $(g_i^j : j \in [m_i])$, i.e. the pointwise smallest concave function that lies weakly above every g_i^j.

Given Q_i, $\hat{f}_i(Q_i)$ is efficiently computable: it is a concave maximization problem over the linear constraint that the expectation of β_i is Q_i. The solution β_i produces $\lambda_i, \{q_i^j\}$. We also note that the gradient of f_i at Q_i is computable efficiently as well.

$\hat{f}_i(Q_i)$ is also decreasing, as it represents the maximum value obtainable by selecting with probability Q_i. This implies that the function $h(Q_i) := Q_i f_i(Q_i)$ is concave, as we can make the same argument as with g_i^j. The ex-ante relaxation problem is

$$\max_{Q \in \mathcal{P}_\mathcal{F}} \sum_i Q_i f_i(Q_i),$$

a concave maximization problem subject to a polytope constraint. If $\mathcal{P}_\mathcal{F}$ has an efficient separation oracle, it is efficiently solvable to find Q. Given Q, we have already shown that λ_i is efficiently computable. □

Appendix J

Weitzman Index Interpretation of SAUP

The efficient SAUP algorithm described in Proposition 5.4.1 can be nicely characterized in the language of the Weitzman indices we use for bandit processes. We first develop some notation. Given an MSP $\mathcal{M}$ and a state s, consider the MSP $\mathcal{M}'$ that we get by taking s to be our start state. Given a deterministic policy π on $\mathcal{M}'$, let B^π be the bandit process induced by π. Define σ_s^π to be the Weitzman index σ_{s^*} of the start state s^* of B^π, and κ_s^π to be κ_{s^*}.

We prove that from any starting state s in $\mathcal{M}$, the value achieved by the algorithm in Proposition 5.4.1, is exactly equal to a Weitzman amortization of the costs.

Proposition J.0.1. *For any $\mathcal{M}'$ rooted at state s, $\mathbb{E}[\text{Perf}^{\pi^*}(\mathcal{M}') - \mathbb{A}^{\pi^*}\tau] = \mathbb{E}[(\kappa_s^{\pi^*} - \tau)^+] = \max_\pi \mathbb{E}[(\kappa_s^\pi - \tau)^+]$.*

Proof. We will prove via induction that $\text{Val}^{\pi^*,s} \max_\pi \mathbb{E}[(\kappa_s^\pi - \tau)^+]$. For B^{π^*}, the bandit induced by π^*, it immediately follows that $\text{Val}^{\pi^*,s} = \mathbb{E}[(\kappa_s^{\pi^*} - \tau)^+]$. We will finally argue that $\mathbb{E}[\text{Perf}^{\pi^*}(\mathcal{M}') - \mathbb{A}^{\pi^*}\tau] = \mathbb{E}[(\kappa_s^{\pi^*} - \tau)^+]$, completing the proof.

We proceed with the inductive portion. For a terminal state s, $\text{Val}^{\pi^*,s} = (V(s) - \tau)^+ = \max_\pi \mathbb{E}[(\kappa_s^\pi - \tau)^+]$. For a nonterminal state s, assume as our inductive hypothesis that $\text{Val}^{\pi^*,n(s)} = \max_\pi \mathbb{E}[(\kappa_{n(s)}^\pi - \tau)^+]$ for all possible successors $n(s)$. By definition of our algorithm,

$$\begin{aligned}
\text{Val}^{\pi^*,s} &= \max_a v^{\pi^*}(a; s) \\
&= \max_a \mathop{\mathbb{E}}_{s' \sim p_{a,s}} [\text{Val}^{\pi^*,s'}] - C(a, s) \\
&= \max_a \max_\pi \mathop{\mathbb{E}}_{s' \sim p_{a,s}} [(\kappa_{n(s)}^\pi - \tau)^+] - C(a, s)
\end{aligned}$$

by the inductive hypothesis.

We define $\sigma^{\pi}_{a;s}$ as the Weitzman index for the bandit process starting at s and taking action a and then following policy π, referring to the Bandit process notation of Definition 5.2.6. Recall by the definition of the index σ that $\mathbb{E}[(\kappa^{\pi}_{n(s)} - \sigma^{\pi}_{a;s})^+] = C(a,s)$ for any policy π. Noting that $C(a,s)$ is a constant for any action a, we can rewrite $\mathrm{Val}^{\pi^*,s}$ replacing the cost term with a constant function of the policy π:

$$\mathrm{Val}^{\pi^*,s} = \max_a \max_\pi \left(\mathop{\mathbb{E}}_{s' \sim p_{a,s}} [(\kappa^{\pi}_{s'} - \tau)^+] - \mathbb{E}[(\kappa^{\pi}_{s'} - \sigma^{\pi}_{a;s})^+] \right),$$

with the maximum over π now being taken over both terms in the expression. We rewrite with π^* and a^* the maximizing policy and action chosen as s, respectively, to get

$$\mathrm{Val}^{\pi^*,s} = \mathop{\mathbb{E}}_{s' \sim p_{a,s}} [(\kappa^{\pi^*}_{s'} - \tau)^+ - (\kappa^{\pi^*}_{s'} - \sigma^{\pi^*}_{a^*;s})^+]. \tag{J.1}$$

We consider two cases here: $\tau \geq \sigma^{\pi^*}_{a^*;s}$ and $\tau < \sigma^{\pi^*}_{a^*;s}$. In the first case, in the expectation of equation (J.1) the term $(\kappa^{\pi^*}_{s'} - \tau)^+$ will always be at most $(\kappa^{\pi^*}_{s'} - \sigma^{\pi^*}_{a^*;s})^+$, giving $\mathrm{Val}^{\pi^*,s} \leq 0$. By construction of the algorithm, if the best action would produce $\mathrm{Val}^{\pi^*,s} < 0$ it simply halts, giving $\mathrm{Val}^{\pi^*,s} = 0 = \mathrm{Val}^{\pi^*,s} = \mathbb{E}_{s' \sim p_{a,s}}[(\min(\sigma^{\pi^*}_{a^*;s}, \kappa^{\pi^*}_{s'}) - \tau)^+] = \mathbb{E}_{s' \sim p_{a,s}}[(\kappa^{\pi^*}_{s} - \tau)^+]$.

When $\tau < \sigma^{\pi^*}_{a;s}$, we analyze the interior of the expectation pointwise. Consider the realization of $\kappa^{\pi^*}_{s'}$ relative to $\sigma^{\pi^*}_{a^*;s}$. If $\kappa^{\pi^*}_{s'} < \sigma^{\pi^*}_{a^*;s}$, then

$$\begin{aligned}(\kappa^{\pi^*}_{s'} - \tau)^+ - (\kappa^{\pi^*}_{s'} - \sigma^{\pi^*}_{a^*;s})^+ &= (\kappa^{\pi^*}_{s'} - \tau)^+ \\ &= (\kappa^{\pi^*}_{s} - \tau)^+,\end{aligned}$$

with the last equality following the definition of $\kappa^{\pi^*}_{s} = \min(\sigma^{\pi^*}_{a^*;s}, \kappa^{\pi^*}_{s'})$. Otherwise, $\tau < \sigma^{\pi^*}_{a^*;s} \leq \kappa^{\pi^*}_{s'}$ and

$$\begin{aligned}(\kappa^{\pi^*}_{s'} - \tau)^+ - (\kappa^{\pi^*}_{s'} - \sigma^{\pi^*}_{a^*;s})^+ &= \kappa^{\pi^*}_{s'} - \tau - \kappa^{\pi^*}_{s'} + \sigma^{\pi^*}_{a^*;s} \\ &= \sigma^{\pi^*}_{a^*;s} - \tau \\ &= (\sigma^{\pi^*}_{a^*;s} - \tau)^+ \\ &= (\kappa^{\pi^*}_{s} - \tau)^+,\end{aligned}$$

again following the definition of $\kappa_s^{\pi^*} = \min(\sigma_{a^*;s}^{\pi^*}, \kappa_{s'}^{\pi^*})$. Therefore, when $\tau < \sigma_{a^*;s}^{\pi^*}$ we have $\text{Val}^{\pi^*,s} = \mathbb{E}_{s' \sim p_{a,s}}[(\kappa_s^{\pi^*} - \tau)^+] = \max_\pi \mathbb{E}[(\kappa_s^\pi - \tau)^+]$ pointwise for all realizations. We now argue that $\mathbb{E}[\text{Perf}^{\pi^*}(\mathcal{M}') - \tau \mathbb{A}^{\pi^*}] = \mathbb{E}[(\kappa_s^{\pi^*} - \tau)^+]$ by claiming that π^* is **non-exposed** (Definition 5.2.7). In the structure of the proof above, we consider two cases for τ vs. $\sigma_{a^*;s}^{\pi^*}$ at any state s. When $\sigma_s^{\pi^*} \leq \tau$, the policy halts. Otherwise, the policy continues. This means that π^* stops in state s the first time $\sigma_s^{\pi^*} \leq \tau$, and in all previous states s', $\sigma_{s'}^{\pi^*} > \sigma_s^{\pi^*}$. The policy is therefore non-exposed, and by Lemma 5.2.1, $\mathbb{E}[\text{Perf}^{\pi^*}(\mathcal{M}')] = \mathbb{E}[\text{Perf}^{\pi^*}(B^{\pi^*})] = \mathbb{E}[\mathbb{A}^{\pi^*} \kappa_s^{\pi^*}]$. Thus, $\mathbb{E}[\text{Perf}^{\pi^*}(\mathcal{M}') - \tau \mathbb{A}^{\pi^*}] = \mathbb{E}[\mathbb{A}^{\pi^*} \kappa_s^{\pi^*} - \mathbb{A}^{\pi^*} \tau] = \mathbb{E}[\mathbb{A}^{\pi^*}(\kappa_s^{\pi^*} - \tau)] = \mathbb{E}[(\kappa_s^{\pi^*} - \tau)^+]$, as π^* only claims if every intermediate $\sigma_{s'}^{\pi^*}$ is at least τ. □

We highlight one fact used within the above proof: the policy π^* is non-exposed.