Fakultät für Mathematik
Technische Universität München

Lotteries, Prophets, and Pandora's Box: A New Take on Classic Problems in Mechanism Design and Online Selection

Alexandros Tsigonias-Dimitriadis

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:
Prof. Dr. Silke Rolles

Prüfende der Dissertation:
1. Prof. Dr. Andreas S. Schulz
2. Prof. Dr. Martin Bichler
3. Prof. Dr. Thomas Lidbetter

Die Dissertation wurde am 20.10.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 01.03.2022 angenommen.

Abstract

Online markets have transformed many economic and social activities and have brought forward new computational challenges. In particular, dealing with large-scale user data has become an essential feature of modern algorithm design. This new reality creates the need to revisit classic theoretical results and reshape and extend standard models to better reflect real-world applications. In this thesis, we design simple and robust mechanisms and algorithms for fundamental problems in auction design, pricing, and online decision-making. We rethink central assumptions in three well-studied theoretical models and frameworks: Bayesian revenue-optimal auctions, the secretary problem, and the Pandora's box problem.

Specifically, we first study a multi-dimensional revenue maximization problem. Unlike traditional Bayesian auction design, the seller knows only some moments of a buyer's prior distribution on the item valuations. Our goal is to design mechanisms that achieve good revenue against an ideal optimal auction that has full knowledge of the distribution in advance. We characterize the optimal selling mechanism for this limited information model, which is randomized but very simple. Next, we explore two new models in online decision-making. We start with a data-driven version of the secretary problem. In particular, we develop a simple combinatorial model that increases the overall solution quality by leveraging past data. We obtain the best possible algorithms, both in the case of adversarial and random order inputs and for any amount of past data available. Finally, we study online variations of the Pandora's box problem, where the decision-maker cannot control the order of exploration. We show that in some cases, simple threshold policies that origin from prophet inequalities perform surprisingly well.

Zusammenfassung

Online-Märkte haben viele wirtschaftliche und soziale Aktivitäten verändert und neue rechnerische Herausforderungen mit sich gebracht. Insbesondere der Umgang mit großen Mengen an Nutzerdaten ist zu einem wesentlichen Merkmal moderner Algorithmen geworden. Diese neue Realität macht es notwendig, klassische theoretische Ergebnisse zu überdenken und Standardmodelle umzugestalten und zu erweitern, um realen Anwendungen besser gerecht zu werden. In dieser Arbeit entwerfen wir einfache und robuste Mechanismen und Algorithmen für grundlegende Probleme bei der Gestaltung von Auktionen, der Preisbildung und der Online-Entscheidungsfindung. Wir überdenken zentrale Annahmen in drei gut untersuchten theoretischen Modellen und Rahmenwerken: Bayes'sche ertragsoptimale Auktionen, das Sekretärproblem und das Problem der Büchse der Pandora.

Konkret untersuchen wir zunächst ein mehrdimensionales Erlösmaximierungsproblem. Im Gegensatz zum traditionellen Bayes'schen Auktionsdesign kennt der Verkäufer nur einige Momente der Vorabverteilung der Artikelbewertungen eines Käufers. Unser Ziel ist es, Mechanismen zu entwickeln, die im Vergleich zu einer idealen, optimalen Auktion mit vollständiger Kenntnis der Verteilung im Voraus kennt. Wir charakterisieren den optimalen Verkaufsmechanismus für dieses Modell mit begrenzten Informationen, das zwar randomisiert, aber sehr einfach ist.

Als nächstes untersuchen wir zwei neue Modelle für die Online-Entscheidungsfindung. Wir beginnen mit einer datengesteuerten Version des Sekretärproblems. Insbesondere entwickeln wir ein einfaches kombinatorisches Modell, das die Gesamtqualität der Lösung durch die Nutzung vergangener Daten erhöht. Wir erhalten die bestmöglichen Algorithmen, sowohl im Fall von gegnerischen und zufälligen Eingaben als auch für jede verfügbare Menge an Vergangenheitsdaten. Schließlich untersuchen wir Online-Varianten des Problems der Büchse der Pandora, bei denen der Entscheidungsträger die Reihenfolge der Exploration nicht kontrollieren kann. Wir zeigen, dass in einigen Fällen einfache Schwellenwertstrategien, die aus Propheten-Ungleichungen stammen, überraschend gut funktionieren.

Acknowledgements

Writing the final words of this dissertation, I would like to thank all the people without whom reaching this point would not have been possible. This short acknowledgments section can only express up to a certain extent the immense gratitude I feel towards all of them.

First and foremost, I would like to thank my supervisor, Prof. Andreas S. Schulz, for his wise guidance and support throughout this process and for providing a very welcoming research environment at the Operations Research group. A big thank you goes to my mentor, Prof. Yiannis Giannakopoulos. Yiannis initiated me into the world of research, taught me some of his beautiful writing style, and had the patience to discuss with me and offer his advice on all sorts of topics, within the academic world and beyond. I hope that one day I will be able to approach topics and questions in the multifaceted and rigorous way that he does. I owe a big thanks to Prof. Martin Bichler, who always took time aside in his busy schedule to provide me with invaluable advice and for being part of my committee. I would also like to thank Prof. Thomas Libdetter and Prof. Silke Rolles for serving in my committee, and particularly Tom for his suggestions that led to several improvements. Moreover, I was very fortunate to be introduced to the world of algorithms and theoretical computer science by one of the best teachers I ever encountered, my undegraduate advisor Prof. Dimitris Fotakis.

During my time as a PhD student, I was very fortunate to meet and collaborate with a lot of great people. In particular, I was extremely lucky to have Yiannis, Diogo, Jose, Andres, Laurent, Tim, Evangelia, Rojin, Yifeng, Stefano, Tomer, Rebecca, and Matteo as co-authors. Without them, the research outcome, most part of which constitutes this thesis, would not have been possible. One of the most valuable experiences I had was working in various research environments and learning from people who tackle research questions in distinct ways. For that, I would like to thank Jose Correa and Stefano Leonardi, for giving me the opportunity to visit them and their groups at the University of Chile and the Sapienza University of Rome, respectively. As a PhD student, I was lucky enough to be part of the Operations Research group and the research training group “Advanced Optimization in a Networked Economy”. My colleagues created a very stimulating environment, and I will fondly remember the many hours that we spent together and the different activities in which we participated. Thank you for that Felix, Diogo, Katie, Ulf, Yiannis, Simon, Marinus, Marcus, Marilena, Themis, Nicola, Benedikt, Daniel, Clara, Anne, Alexander, Matthias, Alexandre, Alexander, Richard, Layla, Ramin, Eleni, Carolin, Baturhan, Giacomo, Soroush, Anja, Max, Stefan, Gregor, Breno, Caroline, and Donghao. Finally, I deeply thank Themis, Laurent, Tim, and Anna for proofreading parts of this thesis and providing extremely valuable feedback that led to many improvements in the presentation.

My friends in both places that I now call home have been an indispenable part of this journey. In Munich, I want to thank my friends for all their support and the great time we had together. Adrian, Alexandra, Anna M., Anna P., Basti, Clara, Eleni, Eleonora, Giannis, Isidora, Khanh, Lida, Markus, Maria C., Maria G., Sarah, Sebas, Serkan, Tomas, and Zozo: it would not have been the same without you. The amazing people I was very fortunate to meet throughout my life in Greece made me always looking forward to traveling back. I want to thank my friends that make Athens a great place: Alexandros, Danai, Dimitris, Foivos A., Foivos V., Giannis P., Giannis S., Giorgos B., Giorgos T., Iasonas, Katerina, Konstantinos B., Konstantinos B., Marilia, Marios, Nikos, Nikolas, Vasilis M., and Vasilis P.. And my friends that I meet in Serifos and, thus, will always be linked to the beautiful summers on the island: Evangelia, Ioanna, Maria, Natasa, Katerina, Tzo, and Vanta.

My family has played an important role in how I was shaped. I want to thank my parents, Anna and Michalis, for instilling in me the passion for learning and their work ethics. I also want to thank the other close members of my family: my grandma Chryssoula and my cat Stivo. A big thank you goes to my uncle Yiannis, for being the most relatable (and only) example of an academic I had in my family circle and for inspiring me from a young age to pursue this path as well. Finally, I want to express my deepest gratitude to my partner Anna for her love and companionship and for always offering a firm, supporting hand in the most difficult of times. I am very happy that I shared this experience with you and am looking forward to the next common steps, whatever and wherever those may be.

Contents

List of Figures

List of Tables

1 Introduction

Over the last decades, breakthrough technologies have allowed online marketplaces and crowdsourcing platforms to grow and evolve at an unprecedented rate. As more and more economic and social activities take place on online platforms, the tasks that these platforms face become increasingly complex, and new informational and computational challenges arise. The question of how to design algorithms and mechanisms to tackle them lies at the heart of machine learning, algorithmic game theory, and optimization.

Take as an example the ride-hailing company Uber and its surge pricing algorithm. In order to design the dynamic pricing model, Uber has to take into account several factors. First, it should accurately predict supply and demand, both short-term and long-term, by efficiently analyzing large-scale data from customers and drivers (machine learning). Moreover, it should model customers' behavior and understand under which conditions they are willing to book trips and choose their services over competitors. At the same time, if drivers expect prices to rise in a specific area in the near future, they might wait to get one of these profitable rides. Thus, Uber should also account for the potential strategic behavior of the drivers (algorithmic game theory). Finally, Uber should frequently update prices in real-time by considering a vast amount of parameters, including the ones mentioned above (optimization). Apart from ride-sharing, companies face similar challenges in online retail, airline pricing, freelancing platforms, advertisement auctions (a.k.a. ad-auctions), and more.

From the previous example, several questions arise of both theoretical and practical importance. As designers of an online platform, can we model such a complex, uncertain environment and uncover some of its structural properties? How can we utilize historical data efficiently to inform operational decisions, and how much can we rely on their accuracy? How does limited information or lack of data affect the quality of the algorithms' outcomes? In which ways does the stochastic nature of the input and the deployed mechanisms change our design choices and algorithms? This dissertation takes a step towards answering some of these questions.

Broadly speaking, we study problems related to two ubiquitous challenges in online markets: pricing of goods and sequential decision-making under uncertainty. Due to the interdisciplinary nature of these challenges, our work lies at the intersection of theoretical computer science, economics, and operations research. We revisit three fundamental models in mechanism design and online decision-making; we rethink their assumptions or use them as building blocks. Our goal is to identify natural conditions common in real-world problems and incorporate them into those models. By rethinking classic theory and its results, we hope to provide new design insights and better guide practice.

We should always keep two principles in mind when designing algorithms and mechanisms for the problems in this thesis: First, they should be simple; simplicity will make their implementation easier and also help us overcome some of the computational hurdles. Second, they should be robust; if they can provide good guarantees in a wide range of environments, then we can possibly deal with informational challenges (e.g., the potential scarcity of data).

1.1 Contributions of this thesis

In the first part of the thesis (Chapter 3), we design algorithms in the presence of strategic agents. These are algorithms that need to work well when their input comes from selfish agents who want to maximize their happiness. We call such algorithms mechanisms and the field of research algorithmic mechanism design. Our contribution is in auction design, which focuses on mechanisms related to selling goods and services to interested buyers. We take the perspective of a seller who wants to maximize her revenue by selling goods in an environment for which she has little prior information. In the second part of the thesis (Chapter 4 and Chapter 5), we design optimization algorithms for online decision-making. The input now is not strategic, but it is not available to the algorithm from the start. Instead, it arrives over time; the main challenge is that we need to make decisions in a sequential manner based on the current input and without good knowledge of the future.

Our work builds upon three fundamental theoretical problems and frameworks: Myerson's optimal auction (and more generally, Bayesian auction design), the secretary problem, and the Pandora's box problem. More concretely, we take a robust approach to various parameters of these problems and their extensions in different ways. The common goal is to move towards more realistic assumptions and design solutions that can fit different scenarios.

In Chapter 3, we consider a revenue maximization problem where the seller has limited information on how much the buyers value the goods for sale. We design mechanisms that rely on minimal statistical information of the probability distribution of a buyer's valuation. Modeling the auction environment in such a way allows the seller to be robust to misspecification of the buyers' valuations. In Chapter 4 we develop a data-driven version of the secretary problem. This models very simple decision processes in which we want to leverage historical data. Following the principles of the secretary problem, we assume no underlying distribution for the samples nor impose any other structural property. This enables us to design algorithms that take this data as input and are robust even when an adversary chooses them. Finally, in Chapter 5 we explore how varying the order of exploration changes our algorithms and performance guarantees in the Pandora's box problem. In search theory, it is usually assumed that all the options are available from the start and that we can explore them in the order we want. Our main goal is to understand how we should adopt her policies when the input, or part of it, is presented in an online fashion. Identifying simple and provably good policies when we cannot (fully) control the order of exploration makes the problem robust to exogenously defined orderings.

Following, we present the topics of this thesis in more detail and state the main results we obtain in each chapter.

Beyond Bayesian auction design: Revenue maximization with statistical information
Designing revenue-maximizing auctions, known as *optimal auction design*, is a fundamental and well-studied problem in mechanism design. Auctions are used to buy and sell goods and services and therefore have broad applicability. Characteristic examples include spectrum auctions, the ad auctions run by companies such as Google and Facebook, the auctions for emission allowances, and more.

The standard Bayesian approach assumes that each bidder has a valuation function, which represents the amount of money they are willing to spend on each subset of items. The seller does not know the valuation functions of the bidders but has full knowledge of a prior distribution F over them. The goal of the seller is to design a revenue-maximizing auction using this prior knowledge. The single-item case was fully resolved 40 years ago in a seminal paper by Roger Myerson [126]. This result was one of the main contributions that led to him winning the 2007 Nobel prize in Economics.

However, the seller having full knowledge of a prior joint distribution of the bidders' valuations is arguably a strong assumption. We study the problem of multi-dimensional revenue maximization when selling m items to a buyer that has additive valuations for them, drawn from a (possibly correlated) prior distribution. Unlike traditional Bayesian auction design, we assume that the seller has a very restricted knowledge of this prior: she only knows the mean μ_j and an upper bound σ_j on the standard deviation of each item's marginal distribution.

Modeling the limited knowledge of the prior in such a way serves two main purposes: First, by using statistical information, we try to capture different types of scenarios. One such scenario is when we do not have access to individual past data because of privacy restrictions, and we cannot learn the prior from them. Moreover, representing a high-dimensional distribution can be computationally very challenging. In such cases, we can assume that we have access to some approximate values of moment conditions of the aggregate data (such as the mean and the standard deviation). We also believe that moment conditions are quite natural, as we often use them to describe a distribution. Second, as mentioned above, it is very likely that the seller will have inaccurate prior beliefs of the bidders' valuations. The true underlying distribution might not be very different, but nevertheless, we do not want to design a mechanism that overfits to a possibly misspecified prior. Instead, we would prefer to optimize over a set of distributions and ensure that our mechanisms are robust enough. Using moment conditions helps us define a natural *uncertainty set*; we then design mechanisms that optimize simultaneously over all distributions in the set.

Our main question is the following: Can we design mechanisms that achieve good revenue against an ideal optimal auction that has *full* knowledge of the distribution in advance? Informally, our main contribution is a tight quantification of the interplay between the dispersity of the priors and the aforementioned robust approximation ratio. Furthermore, this can be achieved by very simple selling mechanisms. More precisely, we show that selling the items via separate price lotteries achieves an $O(\log r)$ approximation ratio where $r = \max_j(\sigma_j/\mu_j)$ is the maximum coefficient of variation across the items. If forced to restrict ourselves to deterministic mechanisms, this guarantee degrades to $O(r^2)$. Assuming independence of the item valuations, these ratios can be further improved by pricing the full bundle. For the case of identical means and variances, in particular, we get a guarantee of $O(\log(r/m))$ which converges to optimality as the number of items grows large. We demonstrate the optimality of the above mechanisms by providing matching lower bounds. Our contributions improve and extend prior work from both the economics and computer science literature.

Let us return for a moment to online markets and their complex environments. It is often the case that, as designers, the settings for which we need to find good solutions are not static. In this sense, the previous auction design model is static since we implicitly assume that the bidders and the goods are all present *simultaneously*. However, the nature of those markets has a good deal of stochasticity; it is likely that in the previous auction, the bidders and the goods arrive over time. In this case, we need to make decisions when only a subset of them are present on the platform and without being sure who and when will arrive in the near future. Therefore, it is also crucial to understand how we can make effective decisions in a sequential way. Our following two contributions study simple algorithms for models of sequential-decision making in uncertain environments.

A data-driven secretary problem The secretary problem is probably the most well-studied optimal stopping problem with many applications in economics and management. In the secretary problem, a decision-maker faces an unknown sequence of values, revealed one after the other, and has to make irrevocable take-it-or-leave-it decisions. Her goal is to select the maximum value in the sequence.

In the modern world, online platforms and marketplaces rely on gathering and analyzing vast amounts of data to optimize their decisions. As a result, any optimization algorithm and mechanism designed for a task in these ecosystems leverages the available historical data. From a theoretical point of view, this motivates us to better understand how algorithms for online decision-making can account for the presence of past data. Due to its simplicity and generality, the secretary problem can serve as the foundation to start exploring this research direction. While in the classic secretary problem, the values of upcoming elements are entirely unknown, in many realistic situations, the decision-maker still has access to some information, for example, in the form of past data.

The previous description might hint at machine learning methods that use predictors to learn the distribution from available samples. However, for problems that can be modeled as data-driven versions of the secretary problem, these learning procedures can be very complicated. Thus, providing rigorous theoretical guarantees for such procedures is also a challenging task. Furthermore, as mentioned before, one of our goals in this thesis is to work on models that entail some notion of robustness. More specifically, in the current setting, we want to make minimal assumptions on the input data of our algorithms. Our main questions can be formulated as follows: Is there a simple combinatorial model that incorporates historical data? Can we design simple algorithms that achieve improved performance guarantees by utilizing this additional information?

To answer these questions, we take a sampling approach to the secretary problem and assume that before starting the sequence, each element is independently sampled with the same fixed probability p. This leads to what we call the random order and adversarial order secretary problems with p-sampling. In the former, the sequence is presented in random order, while in the latter, the order is adversarial. Our main result is to obtain the best possible algorithms for both problems and all values of p. As p grows to 1, the obtained guarantees converge to the optimal guarantees in the full information case, that is, when the values are i.i.d. random variables from a known distribution. Notably, we establish that the best possible algorithm in the adversarial order setting is a simple fixed threshold algorithm. In the random order setting, we characterize the best possible algorithm by a sequence of thresholds, dictating at which point in time we should start accepting a value. Surprisingly, this sequence is independent of p.

From a practical perspective, our approach and threshold algorithms seem to have connections with behavioral phenomena arising in Goldstein et al. [82]. The authors conduct field experiments where people repeatedly play a secretary problem with a fixed number of boxes (with hidden values). They want to understand how people adapt their strategies as they repeat the game and learn more values. After some rounds, people have acquired enough samples to be able to design a near-optimal strategy (the optimal thresholds, in this case, are given in Gilbert and Mosteller [80]). Our model and algorithms could help explain how people play along the way and if they use the information they gain after each round in a near-optimal way.

Online decision-making with search costs In the Pandora's box problem, a foundational mathematical model in search theory, a decision-maker faces n boxes with known, independent distributions of their hidden rewards. To learn the reward of a box, she has to pay an inspection cost, and she can choose the order of inspection. The objective is to maximize the collected reward from an inspected box minus the costs paid. The decision-maker has to design a possibly adaptive policy that dictates the order of inspection and when to stop. From the seminal work of Weitzman [150] we know the optimal policy for this problem, which turns out to be surprisingly simple to describe and easy to compute.

Now consider a slightly different simple scenario: We are selling a single item and want to give it to the buyer who values it the most. As is often the case, the interested buyers are not present simultaneously but arrive over time. We assume that the buyer nor we know her exact value for the item, but we may know some prior distribution over it. For both of us to learn the exact value, we have to invest some costs. Our goal is to try and find the buyer with the highest value while avoiding paying very high costs. A similar situation might arise when we want to hire a skilled worker on a crowdsourcing/hiring platform. The workers appear over time and apply for the job. Their curriculum vitae gives us some prior belief over their skills, but we have to go through the costly process of interviewing them to learn more precisely how suitable they are for the job. We can also decide to reject them right away, without interviewing. Let us also assume for simplicity that the period is short enough so that if we offer the job to an interviewed person, they will accept.

More generally, we think of scenarios where we want to allocate resources to agents who arrive according to some non-stationary process in the platform. We might have some knowledge of their types due to past data, but to completely determine their value for the resources, we have to suffer some search costs. Note that the agents likely do not know their exact value, and they need to inspect the resources first to find out. We blend ideas from the Pandora's box problem and the prophet inequality to design a simple model which captures the scenarios mentioned above.

Some of the questions that arise compared to the classic Pandora's box problem are the following: When we cannot (at least partially) control the search order, how should we design our strategies? Can we come up with simple optimization procedures when agents' requests arrive in an online fashion? Can we quantify the impact of various order constraints on our performance guarantees? We take a first step towards answering some of these questions. We start from the most robust way of modeling the arrival order; an adversary orders the requests, and they are presented to us one by one. Then we move on to study the random arrival order, which is also assumed in many well-known problems in online decision-making, such as in the secretary problem. Our broader goal is to examine different types of stochastic arrivals, ranging from close to free order to close to adversarial and combinations thereof.

More specifically, in this thesis, we study a variation of the Pandora's box problem, where the decision-maker cannot freely choose the inspection order. Instead, the boxes are presented one by one either in an order fixed by an adversary or in random order. We explore different modeling assumptions for the constrained-order Pandora's box problem. For example, the decision-maker might be able to skip a box without inspecting it. For this variant, we show that we can adapt threshold-based policies used in prophet inequalities and apply them in this model. Although designed for making immediate and irrevocable decisions on whether to collect the reward of an inspected box, they still provide good performance guarantees. In particular, they are near-optimal when using Weitzman's optimal policy (in the free order) as our benchmark. For the models we consider, we compare the performance of simple policies either to the optimal adaptive one in the same order or to Weitzman's in the free order.

1.2 Bibliographic notes

Chapter 3 is based on joint work with Yiannis Giannakopoulos and Diogo Poças. A preliminary version of this work appeared in the proceedings of the 16th Conference on Web and Internet Economics (WINE'20) (Giannakopoulos et al. [78]). Parts of Chapter 4 are based on joint work with José Correa, Andrés Cristi, Laurent Feuilloley and Tim Oosterwijk. A preliminary version of this work appeared in the proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA'21) (Correa et al. [55]).

Chapter 5 is based on unpublished work with Evangelia Gergatsouli, Rojin Rezvan, and Yifeng Teng. The author has also other related work (see Ezra et al. [72]) that is not part of this thesis.

2 Preliminaries

In this chapter, we formalize the notions mentioned in the introduction. We also provide most of the definitions and notation needed throughout this thesis. Some models/problems might require some more specialized notation that will be stated in the corresponding chapters. In Section 2.2 we provide the definitions and most of the necessary notation for Chapter 3. In Section 2.3 we do the same mainly for Chapter 4 and Chapter 5.

2.1 General notation

We first introduce some notation that we will use throughout the thesis. We denote $[n] = \{1, 2, \dots, n\}$ for any positive integer n. We denote by $\mathbb{R}_{\geq 0}$ the set of nonnegative reals. For two sets X and Y we use the standard notation $X \times Y$ for the Cartesian product. For an n-dimensional vector $\vec{x}$, we denote by $\vec{x}_{-i}$ a $(n-1)$-dimensional vector with the i-th coordinate of $\vec{x}$ removed. Moreover, we denote by $(x', \vec{x}_{-i})$ an n-dimensional vector where we replaced the value of the i-th coordinate of $\vec{x}$ with x'.

2.2 Concepts in mechanism design

As mentioned in the introduction, mechanism design is a broad field with many different types of problems and applications. Since parts of this thesis deal with problems in auction design and pricing, we introduce the key concepts through the lens of auction theory. A setting in auction design is of the following form in its outmost generality: There is an *auctioneer* offering m items for sale and n *bidders* interested in purchasing a subset of the items. At the end of the process (which might involve e.g. eliciting bids, several rounds of interaction between the auctioneer and the bidders, and more) the auctioneer chooses an outcome w among a set of feasible outcomes W. In an auction environment, a feasible outcome corresponds to an allocation of the items to the bidders, subject to potential constraints on the form of the allocation.

Types and valuations Each bidder i has a type θ_i which is an element of her type space Θ_i. We can think of a type θ_i as the private information that bidder i maintains for her preferences. We will denote the type of bidder i by θ_i and $\vec{v_i}$ interchangeably, since, in some contexts, the latter serves better the presentation purposes. To exemplify the notion of type, consider an auction where a house is being sold. In this case, a bidder's type is how much she values the house, and the type space is $\Theta_i \subset \mathbb{R}_{\geq 0}$. Note that there can be more than one bidder with the same type space. Each bidder also has a set of available actions $\mathcal{A}_i$. We denote by $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$ the set of action profiles. An *allocation rule* is a function x that maps an action profile a to an allocation $w \in W$.

For each bidder i we also define her *valuation function* $V_i : \Theta_i \times W \to \mathbb{R}_{\geq 0}$. The valuation function depends on the realized type and the allocation, and it can be thought as the maximum amount of money a bidder is willing to spend to buy the items she receives under an allocation w. In the literature, depending on the auction setting at hand, different assumptions about the form of the bidders' valuation functions are made. For example, coming back to the property auction, bidder i's valuation function is just her valuation θ_i for the house if she wins it, and 0 otherwise. However, in multi-bidder, multi-item combinatorial auctions, the form of the valuation functions might be very complex. Because of that, we are often interested in some restricted classes of valuation functions that capture important practical scenarios and can also be handled from a theoretical perspective.

In Chapter 3 we study *additive* valuation functions: For this particular class, the type space for each bidder i is $\Theta_i \subseteq \mathbb{R}^m_{\geq 0}$, and her type is an m-dimensional vector $\vec{v_i} = (v_{i1}, v_{i2}, \cdots, v_{im}) \in \Theta_i$ where v_{ij} is bidder's i valuation for item j. The valuation v_{ij} again expresses how much bidder i is willing to pay for item j. Note that an allocation w induces disjoint subsets of items $(B_1, B_2, ..., B_n)$, one for each bidder, such that $\bigcup_{i=1}^n B_i = [m]$. An additive valuation function is then simply the sum of the valuations of the subset of items $B_i \subseteq [m]$ that the bidder receives, i.e., $V_i(\vec{v_i}, w) = \sum_{j \in B_i} v_{ij}$.

Moreover, the auctioneer charges bidder i an amount of money for the allocated subset of items according to a payment rule $\pi_i : \mathcal{A} \to \mathbb{R}_{\geq 0}$. The bidder has a *utility* u_i which intuitively expresses the amount of happiness she gets from the items allocated to her and what she paid for them. We assume that the utilities are *quasilinear*, that is, for each bidder i, her utility can be written as $u_i = V_i(\theta_i, x(a)) - \pi_i(a)$, where $a \in \mathcal{A}$. Finally, we assume that the bidders are *rational*, which means that they always pick an action that maximizes their utility.

Bayesian auction design Recall that the type θ_i is private infomation of the respective bidder; otherwise the auctioneer would know exactly how each bidder encodes her preferences and as a result could always optimize the desired objective. On the other hand, it is very difficult for the auctioneer to design a good pair of allocation and payment rules in the dark, having no information on the participating bidders. To go beyond the two extremes, a standard assumption in auction design is to assume that there is a prior distribution F over the space of type profiles $\Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_n$. When this F is fully known to the auctioneer and all the n bidders, then we are in the *Bayesian* setting.

Environments We define the two main general types of auction environments: *single-dimensional* and *multi-dimensional*.

In a *single-dimensional* environment each bidder's type θ_i is just a scalar. A single-item environment is single-dimensional as well: There are n interested bidders, and a bidder's type is her valuation v_i for the only item being sold. A *single-parameter* environment is also single-dimensional. There are types of auctions in which there are multiple items for sale, but each bidder's type θ_i can still be expressed by a single parameter. Take as an example simple multi-unit auctions: There are k identical items for sale and each bidder wants to buy only one unit or, in a slightly more general scenario, wants to buy up to ℓ units and her type is just her valuation for one unit multiplied with the total number of units she gets (up to ℓ). In any case, the type is indeed a scalar, and thus we are in a single-dimensional environment.

In a *multi-dimensional* environment each bidder's type is made up of several parameters. For example, as we mentioned earlier, when the valuation functions are additive, the types are made up by m different parameters (the various v_{ij}, one for each item j). The same holds for a unit-demand bidder i, whose valuation function is $V_i = \max_{j \in B_i} v_{ij}$ for a subset of items $B_i \subseteq [m]$ and her type θ_i is again an m-dimensional vector $\vec{v_i}$.

Direct revelation mechanisms In the general case of n bidders and m items, a selling *mechanism* $\mathcal{M}$ is defined by an allocation rule x, and a payment rule π_i for each bidder i. A mechanism might be implemented in several rounds, include different types of interaction between the auctioneer and the bidders, and, in general, according to the definition we gave it can be very complicated.

Now consider the following family of mechanisms: The bidders report a type b_i to the auctioneer, who collects all the reported types and then directly chooses an allocation and charges payments. These are called *direct revelation* mechanisms. More formally, when $\mathcal{A}_i = \Theta_i$ for all bidders, we have a direct revelation mechanism.

Next we provide the formal definition of a direct revelation mechanism for the special case of a single additive bidder and m items, which is the setting that we study in the next chapter. A (possibly randomized) selling mechanism for a single additive bidder and m items is defined by a pair (x, π) where $x : \mathbb{R}^m_{\geq 0} \to [0,1]^m$ is the allocation rule and $\pi : \mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ is the payment rule. If the buyer submits as bid a valuation vector of $\vec{v}$, then they receive each item i with probability $x_i(\vec{v})$, and are charged (a total of) $\pi(\vec{v})$. When the mechanism A is *deterministic*, the allocation rule satisfies $x(\vec{v}) \in \{0,1\}^m$, for all $\vec{v}$. When the mechanism is *randomized* the allocation rule satisfies $x(\vec{v}) \in [0,1]^m$, for all $\vec{v}$.

Truthfulness We can think of auctions as games of incomplete information, and the bidders as strategic agents. Given that, some type of *equilibrium*, which is the solution concept in these games, can also equivalently be defined in the context of an auction. Informally, in a direct revelation mechanism the reported types $(b_1, b_2, \dots, b_n)$ form an equilibrium if no bidder i can increase her utility by unilaterally reporting a different type b'_i, often given some information for the other bidders' types (such as the joint prior F over the type space). For more on games with incomplete information the reader can refer to one of the standard textbooks on Game Theory or Algorithmic game theory (e.g. [128, 131]). For a properly defined notion of equilibrium, we can study properties and the quality of our mechanism at the equilibrium state. The two most common equilibrium notions are the dominant-strategy equilibrium (DSE) and the Bayes-Nash equilibrium

(BNE). Since we deal mostly with a single bidder in this thesis, we need not define the concepts formally here.

At this point, note that each bidder reports a type b_i to the auctioneer which might not necessarily be her true private type θ_i. In some scenarios though, it might be the case that the bidders maximize their (expected) ulitity by reporting their true types. It is easy to see why such a property would be desirable for both parties: The bidders would not have to spend a possibly large computational effort to determine which is the best type to report, and the auctioneer would also know in advance that the mechanism they designed induces a specific, well-defined behavior for the bidders. We call a mechanism *truthful* when in an equilibrium state bidders truthfully report their privately-known types. Formally, a direct revelation mechanism is truthful or *dominant strategy incentive compatible (DSIC)* if it holds that

$$V_i\left(\theta_i, x\left(\theta_i, \vec{b}_{-i}\right)\right) - \pi_i\left(\left(\theta_i, \vec{b}_{-i}\right)\right) \geq V_i\left(\theta_i, x\left(\theta_i', \vec{b}_{-i}\right)\right) - \pi_i\left(\left(\theta_i', \vec{b}_{-i}\right)\right) \quad \forall i, \theta_i', \vec{b}_{-i}.$$

Note that there exist also other, weaker notions of truthfulness; the one we defined here and use in this thesis is the strongest and is usually called dominant strategy truthfulness.

Since we restrict our study in this thesis mainly to a single additive bidder and multi-item, direct revelation, truthful mechanisms, we state here the conditions which characterize them:

$$x(\vec{v}) \cdot \vec{v} - \pi(\vec{v}) \geq x(\vec{w}) \cdot \vec{v} - \pi(\vec{w}) \qquad \text{for all } \vec{v}, \vec{w}, \tag{2.1}$$

$$x(\vec{v}) \cdot \vec{v} - \pi(\vec{v}) \geq 0 \qquad \text{for all } \vec{v}, \tag{2.2}$$

where $\vec{v}$ is the true valuation vector of the bidder and $\vec{w} \in \mathbb{R}^m_{\geq 0}$. The first condition states that the bidder can not be "better off" by misreporting their true valuation; this is the truthfulness condition for the single bidder case. The second condition, known as *individual rationality* (IR), ensures that the bidder cannot harm herself by truthfully participating in the mechanism. Recall that $\vec{v}$ Note that when stating the two constraints here, we implicitly assumed that the bidder is risk neutral, that is, if she receives an item j with probability x_j and pays π_j for it, then her utility from the item is $x_j v_j - \pi_j$.

Revelation principle The revelation principle is a very important result from the seminal work of Myerson [126]. It is the reason why in many scenarios we can restrict our attention to direct revelation, truthful mechanisms without loss in our objective. It states that any mechanism $\mathcal{M}$ with some outcome in equilibrium can be converted into a direct revelation, truthful mechanism with the same outcome in the truth-telling equilibrium.

Note that the revelation principle fails to hold in some environments. For example, when the agents learn their values over time, or when the auctioneer does not know the joint prior distribution F, the revelation principle does not hold [91, Section 2.10].

2.2.1 Welfare maximization and the VCG auction

The *social welfare* is a common and natural objective when studying the performance of auctions under various criteria. The social welfare expresses the collective good of the participating bidders (or the society more generally, hence the name) and, in a Bayesian setting, given a direct revelation and truthful mechanism $\mathcal{M}$ with an allocation x, and a joint prior F over the type space Θ, we formally define it as

$$\mathrm{WEL}(\mathcal{M}; F) = \underset{\vec{\theta} \sim F}{\mathbb{E}} \left[\sum_{i=1}^{n} V_i \left(\theta_i, x \left(\vec{\theta} \right) \right) \right].$$

The *VCG mechanism* [50, 85, 149] is a celebrated truthful mechanism that chooses the allocation that maximizes the social welfare, i.e., $x^* \left(\vec{\theta} \right) = \arg\max_{w \in W} \sum_{i=1}^{n} V_i(\theta_i, w)$. In order to establish the truthfulness property, VCG charges each bidder i the "harm" that she causes to the other bidders by participating in the mechanism. In the economics literature, this is called the *externality* imposed on other bidders by her participation. Formally, the payment rule amounts to

$$\pi_i \left(\vec{\theta} \right) = \max_{w \in W} \sum_{j \in [n] \setminus i} V_j \left(\theta_j, w \right) - \sum_{j \in [n] \setminus i} V_j \left(\theta_j, x^* \left(\vec{\theta} \right) \right) .$$

Recall that the payment rule takes the vector of types $\vec{\theta}$ as an argument, because VCG is indeed a direct revelation, truthful mechanism. In a single-item environment, the corresponding welfare-maximizing, truthful mechanism is called the Vickrey auction [149]. It is simply a second-price auction, meaning that the bidder with the highest reported valuation gets the item and pays the second highest reported valuation.

2.2.2 Revenue maximization and Myerson's optimal auction

We state the necessary definitions and results from the celebrated paper of Myerson [126], which characterized the revenue-optimal mechanism for selling a single item in the Bayesian setting. We start with some technical conditions on the distributions that appear often in the optimal auction design literature.

For what follows, we consider a continuous distribution F supported over an interval D_F of nonnegative reals. Let f denote the density function of F. Moreover, we use $F(\cdot)$ for the cumulative function (cdf) of distribution F and $F(p-) = \Pr[X < p] = \lim_{x \to p^-} F(x)$, where $X \sim F$.

Regularity We will say that F is *regular* if its *virtual valuation* function, defined by

$$\phi(x) \equiv x - \frac{1 - F(x)}{f(x)}$$

is monotonically nondecreasing in D_F.

Monotone Hazard Date (MHR) condition The hazard rate of F is defined by

$$h(x) = \frac{f(x)}{1 - F(x)} .$$

The distibution has *monotone hazard rate* if $h(x)$ is monotonically nondecreasing in D_F.

Intuitively, MHR distributions have exponentially decreasing tails. It is easy to check that every MHR distribution is also regular. Although they are a subset of the regular distributions, they are still general enough to give rise to a wide family of natural distributions, such as the uniform, exponential, (truncated) normal and gamma.

λ-regularity Finally, we consider the notion of λ-regularity, which has already been studied in the context of revenue maximization, e.g. by Schweizer and Szech [139] and Cole and Rao [51]. To be precise, Cole and Rao [51] use the notion of α-strong regularity, originally introduced by Cole and Roughgarden [52]; this corresponds exactly to the notion of λ-regularity used in [139], for $\alpha = 1 - \lambda$.

Consider a real parameter $\lambda \in [0, 1]$. We will say that F is λ-*regular* if its λ-*generalized virtual valuation* function

$$\phi_\lambda(x) \equiv \lambda \cdot x - \frac{1 - F(x)}{f(x)}$$

is monotonically nondecreasing in D_F.

It is not difficult to see that, for any $0 \leq \lambda \leq \lambda' \leq 1$, any λ-regular distribution is also λ'-regular. For the special case of $\lambda = 1$, the above definition recovers exactly the notion of regularity. On the other extreme of the range, for $\lambda = 0$ we get the definition of MHR distributions.

Myerson's revenue-optimal mechanism Myerson showed that the revenue-maximizing mechanism for a single item and n bidders whose values for the item follow independent, regular distibutions has a very simple form. A crucial step for proving this was a powerful lemma, which informally states that maximizing the expected revenue reduces to maximizing the expected social welfare *on the virtual values* (termed the expected virtual welfare). Thus, the optimal auction gives the item to the bidder with the highest non-negative *virtual* value by running a Vickrey auction on the virtual values.

From the previous result we can make the following observation: If the bidders are moreover identical, then the revenue is maximized by running a second price auction with a reserce price $p = \phi^{-1}(0)$. This means that the item is allocated to the highest bidder if she beats the reserve price, in which case she pays the maximum of the second highest bid and the reserve price. If there is no bid above the reserve price then the item is not allocated.

Note that the independence assumption is crucial; if the distributions are correlated the aforementioned results collapse. On the other hand, the regularity assumption is not necessary; through an operation on the virtual values called ironing (which appears already in Myerson's work), the results continue to hold for non-regular distributions.

Finally, from the above we can also infer that for the single-item, single-bidder setting the optimal revenue can always be achieved by a deterministic mechanism which offers a take-it-or-leave-it price p. For the optimal such price p, the revenue becomes

$$\mathrm{OPT}_m(F) = \sup_{p \geq 0} p \cdot (1 - F(p-)) \ . \tag{2.3}$$

In Chapter 3, we shall call $\mathrm{OPT}_m(\cdot)$ the *Myerson operator* and for now we simply observe that this is a functional mapping distributions to real nonnegative numbers.

Robust Auction Design Achieving provably good guarantees across a wide range of environments is a very desirable for our mechanisms. In Chapter 3, our goal is to maximize the revenue of a seller, but a different objective (e.g., the social welfare) can be considered instead. In principle, we want to know in which type of situations our mechanism will be used, so that we design it in a way that it takes into account all such situations. This procedure entails a notion of *robustness*; the mechanism should simultaneously achieve good revenue in all the environments, and thus its design should not depend on the details of a specific environment.

In auction design there are different criteria for which one might want to be robust; we are mentioning here some of the most natural ones that have already appeared in the literature. Recall that Myerson's optimal mechanism crucially relies on the assumption that the bidders draw their valuation for the item independently. In many practical applications though, there might be some type of correlation among bidders or items, and the auctioneer wants the mechanism to work well when such correlations are present. This criterion of robustness is often called correlation robustness (e.g., Bei et al. [18]). Other mechanisms crucially rely on assuming a specific model for the bidders' beliefs. If this model is (sometimes slightly) misspecified by the auctioneer, the mechanism might fail. In this case, the auctioneer might want to be robust across a number of such models, and this is often called informational robustness (e.g., Brooks and Du [26]).

Finally, another natural type of robustness is with respect to the prior distribution F over the bidders' type space Θ. Always assuming that the auctioneer fully knows F might be quite strong, which means that we want to go beyond the Bayesian setting and try to be robust against a family of distributions. These families of distributions are often called *ambiguity sets* (e.g., Koçyiğit et al. [107]) in the economics literature and the goal is for the mechanism to achieve good revenue against *any* distribution F in the defined ambiguity set. This is the type of distributional robustness that we focus on in Chapter 3. In our setting with a single additive bidder and m items, the auctioneer has some *moment information* on the marginal distribution F_j of each item j. In particular she knows the mean μ_j and an upper bound on the standard deviation σ_j. Then the induced ambiguity set $\mathcal{F}$ can be stated as follows:

$$\mathcal{F} = \left\{ F \in \Delta\left(\mathbb{R}^m_{\geq 0}\right) : \mu_j(F) = \mu_j \ \text{ and } \ \sigma_j(F) \leq \sigma_j, \ \forall j \in [m] \right\},$$

where by $\Delta\left(\mathbb{R}^m_{\geq 0}\right)$ we denote the space of all distributions over the nonnegative m-dimensional vectors $\vec{v}$, $\mu_j(F) = \mathbb{E}_{\vec{v} \sim F}[v_j]$ is the mean of the marginal distribution F_j of item j and $\sigma_j(F) = \mathbb{E}_{\vec{v} \sim F}\left[(v_j - \mathbb{E}_{\vec{v} \sim F}[v_j])^2\right]$ is the corresponding standard deviation.

There are two main approaches that we can take for characterizing a good robust mechanism and proving its revenue guarantees: Either try to solve an optimization problem exactly (e.g., [34, 35, 107, 146]) or try to approximate a stronger benchmark (our work). In the first case, we have a maximin optimization problem for the revenue; we have to find the mechanism M^* which maximizes the revenue against the worst-case distribution $F \in \mathcal{F}$, which is chosen by an adversary (often also called Nature). A different approach is to view the problem through the lens of approximation and choose a meaningful benchmark. We then design a mechanism which provably approximates the chosen benchmark for any distribution $F \in \mathcal{F}$. One such strong and natural benchmark is the optimal revenue with *full knowledge* of the distribution F, i.e., when we are in the Bayesian setting. We can then say that a mechanism $\mathcal{M}$ with uncertainty about the true prior is c-robust with respect to the ambiguity set $\mathcal{F}$, if $\mathrm{REV}(\mathcal{M}, F) \geq c \cdot \mathrm{OPT}(F)$ for all $F \in \mathcal{F}$ and for some $c \in (0, 1)$. Both ways of approaching the problem have their merits and might lead to different quantitative and qualitative insights, depending on the application domain.

2.3 Sequential decision-making under uncertainty

In this section, we state some of the notions that we mainly need in Chapter 4 and Chapter 5 (and to a lesser extent in Chapter 3). We start with some basic definitions and continue with three fundamental problems in online decision-making. Note that these and related problems have been studied from different angles; sequential decision-making under uncertainty is a truly interdisciplinary research topic. Online algorithms, Markov decision processes, and optimal stopping theory are examples of research areas that fall under this umbrella. It spans theoretical computer science, economics, operations research, mathematics, control engineering, and more.

First, we define the metric which is often used in computer science to measure the quality of the solution of an online algorithm. In online learning, such as in multi-armed bandit problems, a different metric called the *regret* of an algorithm is often used. For more details on the notion of regret, see e.g. [37, 143].

Competitive ratio In online algorithms, we deal with settings where the input arrives piece by piece, and the algorithm tries to optimize an objective by making decisions on the fly without knowledge of the future input. This is in contrast to offline algorithms, where the whole input is given to the algorithm beforehand. The most common metric for the performance of an online algorithm is the competitive ratio, which we state for the case of maximization problems. Suppose we want to solve a maximization problem P on a family of instances $\mathcal{I}$. We fix an online algorithm ALG and the offline optimum OPT.

The online algorithm ALG has a (strictly) *competitive ratio* $c < 1$, if it holds that

$$\mathrm{ALG}(I) \geq c \cdot \mathrm{OPT}(I), \quad \forall I \in \mathcal{I} \, .$$

For completeness, we also define the analogous metric in approximation algorithms.

Approximation ratio The approximation ratio is to approximation algorithms what the competitive ratio is to online algorithms. In approximation algorithms both the algorithm and the optimal solution are given the whole input in advance. We want to solve again a maximization problem and we fix an algorithm ALG that we benchmark against the optimum solution OPT.

The approximation algorithm ALG has an *approximation ratio* $\alpha < 1$, if it holds that

$$\text{ALG}(I) \geq \alpha \cdot \text{OPT}(I), \ \ \forall I \in \mathcal{I} \ .$$

Fundamental problems in sequential decision-making

In this section, we consider three important problems in sequential decision-making: the *secretary problem*, the *prophet inequality*, and the *Pandora's box problem.* All three of them were defined several decades ago (and in the case of the secretary problem probably even longer, see Ferguson [73]) and numerous variants of them have been studied. Since the last decade, there has been a surge of interest in the computer science community in this type of problems. The main reason is their connections to various applications and problems in areas of computer science, particularly in algorithmic game theory and mechanism design. In the following, we define them and informally describe the most important algorithms and results for them. For complete proofs of the statements in this section the reader can refer to excellent surveys and articles [54, 73, 86, 115].

Secretary problem The secretary problem is one of the most well-known problems in sequential decision-making. The first official solution to the problem that appears in a journal is usually attributed to Lindley [113] and a bit later to Dynkin [67], but, as we already mentioned, it has been formulated and appeared in different versions since much earlier.

We describe the classical secretary problem in the following (artificial) scenario: Suppose we want to hire an applicant for a job from a pool of n applicants. We know the exact number n beforehand, but we have no information for the quality of the applicants, so we need to interview them. The way we conduct the interviews is by creating a pile of the applications in a *uniform random order* (i.e., each permutation $\pi : [n] \to [n]$ is equally likely to occur) and inviting the applicants one by one. Upon interviewing an applicant, we assign them a relative rank in the ranking of applicants interviewed so far. Let us also assume that there are no ties between the applicants. Here comes the critical part: As soon as we assign a rank to an applicant, we have to make an *immediate* and *irrevocable* decision whether to hire this applicant or not. If we do, then we never get to interview the remaining ones; if we do not, then we cannot hire this applicant at a later point. One can consider different objectives, but in the classical setting our goal is to maximize the probability of picking the best applicant.

The random order assumption is crucial for the secretary problem; it is not difficult to show that for large n no non-trivial guarantee can be achieved when the order is chosen by an adversary (see e.g. Gupta and Singla [86]). Now let us see how the following simple algorithm would perform: Reject the first half of the applicants (regardless of their relative rank), and then pick the first applicant from the second half that ranks above all previous applicants. This algorithm gives a $\frac{1}{4}$ probability of picking the best applicant. This is because the algorithm succeeds when the second best applicant is in the first half of the sequence, and the best applicant in the second half. Both these events happen with probability roughly $\frac{1}{2}$.

It turns out that we can do better: The optimal algorithm observes the first $\frac{n}{e}$ (instead of $\frac{n}{2}$) and rejects them, and then picks the first applicant that is assigned the best relative rank. The result can be stated as follows: The $\frac{n}{e}$-algorithm chooses the best applicant with probability at least $\frac{1}{e}$ as $n \to \infty$. Moreover, no algorithm can perform better than $\frac{1}{e}$.

Different ways of proving the guarantee of $\frac{1}{e}$ have been discovered. One of the proofs for the result proceeds in the following two steps: We first show that the optimal algorithm belongs to the family of algorithms that first reject a fraction of the sequence, and then pick the first applicant that is the best seen so far. We can then calculate the probability of choosing the best for this family of algorithms, and then show that as $n \to \infty$ the expression is maximized at $\frac{1}{e}$ for a cutoff value of $\frac{n}{e}$.

We deal with the secretary problem in Chapter 4.

The prophet inequality We will now describe the setting of the classic prophet inequality [109, 110] through a different scenario. Suppose now that we are a gambler and want to play the following card game: There are n cards with hidden nonnegative values facing down. Each card i has a distribution F_i (over nonnegative values) written on the side facing up, so that we can see all the distributions. We are also told that each value v_i is drawn independently from F_i. Then an *adversary* orders the cards and the game starts. We start flipping the cards one by one in this fixed order, and when the hidden value of a card is revealed, we have again to make an immediate and irrevocable decision whether to keep the card. Note that we are allowed to pick only one card. If we decide to pick a card, we stop and we cannot flip any of the remaining cards; if we decide to pass on a flipped card, we cannot recall it later. We will say that the chosen value is our reward, and the main question is which strategy maximizes our expected reward.

One observation is that the optimal strategy can be calculated via a simple backwards induction: Calculate the optimal expected reward conditioned on starting the game at the last card, then do the same for the card with index $n-1$ and then proceed accordingly backwards until the first card of the sequence. In their classic result, Krengel and Sucheston [110] showed that we can obtain expected reward that is at least half of the reward of a *prophet*, who knows all the hidden values and always picks the maximum. Remarkably, Samuel-Cahn [136] proved that in fact we can achieve the same guarantee with a very simple policy; a single-threshold strategy suffices to achieve at least half of the prophet's reward. The single-threshold strategy simply fixes a value τ according to some rule and then accepts the first value v_i above the threshold. What is surprising is that the guarantee of $\frac{1}{2}$ is tight, and the threshold algorithm is optimal. The example proving that $\frac{1}{2}$ is the best we can hope for, consists of only two boxes.

Due to recently discovered analogies to pricing problems, the prophet inequality and its variants have received a lot of attention, particularly from the fields of economics, operations research and computer science.

The Pandora's box problem Let us conclude with the Pandora's box problem, which was introduced by Weitzman [150]. The setting has a slightly different flavor from the previous two "purely online" problems. The Pandora's box problem is considered a fundamental model in search theory, a field in economics where, informally, an agent is searching among several alternatives for the one with the best quality.

As we did with the previous two, we can describe this problem using a slightly artificial, real-life scenario: Suppose that we want to rent an apartment, and after a first screening we have narrowed down the search to n of them. Now it is time to start inspecting them and try to choose the best one. An important difference between the Pandora's box problem and the secretary problem or the prophet inequality is that we assume that an inspected apartment will be always available at a later point in time if we decide to rent it. For the sake of the example, let us also make a couple of extra assumptions that simplify the problem. First, we assume that after inspecting an apartment, and in case we decide to rent it, the owner will always give it to us. Second, we assume that the rent is the same for all the apartments. This means that we do not have to calculate and compare the price to quality ratio, but we just want to find the best apartment for us.

The model is as follows: There are again different distributions $F_1, F_2, \dots, F_n$ and the true values of the apartments $v_1, v_2, \dots, v_n$ are independently drawn from their respective distributions. We again fully know each distribution F_i of apartment i. In order to learn the true realization v_i, we now have to pay an inspection cost c_i. The cost captures various parameters; for example, the time we need to invest for the inspection, or the cost of consulting a real-estate agent. We get to choose one apartment to rent and our goal is to maximize the value of the apartment we choose minus the sum of the inspection costs. There is an extra decision that we have to make in this problem: Apart from choosing a stopping time, we also get to inspect the houses in the order we want.

One might naturally expect that the optimal algorithm is very difficult to characterize; we might need to choose the order adaptively depending on the realized values so far, and decide on the stopping time based on the history and the uninspected apartments. Perhaps defying a bit the intuition, Weitzman [150] gave a surprisingly simple answer to this problem. The policy he defined is an "index-based" policy that briefly works as follows: For each apartment, we calculate a value σ_i (usually called the reservation value), whose value depends on the inspection cost c_i and the distribution F_i. Then we inspect the apartments in decreasing order of reservation values, and we stop when we find a realized value that is larger than the reservation values of all the remaining apartments. In his seminal work, Weitzman showed that this simple policy is indeed optimal.

We deal with the Pandora's box problem in Chapter 5.

3 Robust Revenue Maximization Under Minimal Statistical Information

3.1 Introduction

Optimal auction design is one of the most well-studied and fundamental problems in (algorithmic) mechanism design. In the traditional Myersonian [126] setting, an auctioneer has a single item for sale and there are n interested bidders. Each bidder has a (private) valuation for the item which, intuitively, represents the amount of money they are willing to spend to buy it. The standard Bayesian approach is to assume that the seller has only an incomplete knowledge of these valuations, in the form of a prior joint distribution F. A selling mechanism receives bids from the buyers and then decides to whom the item should be allocated (which, in general, can be a randomized rule) and for what price. The goal is to design a truthful selling mechanism that maximizes the auctioneer's revenue, in expectation over F.

Myerson [126] provided a complete and very elegant solution for this problem when bidder valuations are independent, that is, F is a product distribution. In particular, when the distributions are identical and further satisfy a regularity assumption, the optimal mechanism takes the very satisfying form of a second-price (Vickrey) auction with a reserve price. Unfortunately, in general these characterizations collapse when we move to multi-dimensional environments where there are $m > 1$ items for sale. Multi-item optimal auction design is one of the most challenging and currently active research areas of mechanism design. Given that the exact description of the revenue maximizing auctions in such settings is a notoriously hard task, there is an impressive stream of recent papers, predominantly from the algorithmic game theory community, that try to provide good approximation guarantees to the optimal revenue.

The critical common underlying assumption throughout the aforementioned optimal auction design settings is that the seller has *full knowledge* of the prior joint distribution F of the bidders' valuations. In many applications though, this might arguably be an unrealistic assumption to make: usually an auctioneer can derive some distributional properties about the bidder population, but to completely determine the actual distribution would require enormous resources. Thus, inspired by the parametric auctions of Azar and Micali [8] for the single-dimensional case, we would like to be able to design robust auctions that (1) make only use of *minimal statistical information* about the valuation distribution, namely its mean and variance; and (2) still provide good revenue guarantees even in the worst case against an adversarial selection of the actual distribution F; in particular, *no further assumptions* (e.g., independence of item valuations or regularity) should in general be made about F. This is our main goal in this chapter.

3.1.1 Related work

As mentioned in the introduction, there has been an impressive stream of recent work on optimal [32, 60, 77, 88, 119] and approximately-optimal [13, 31, 39, 90, 111, 134, 152] multi-dimensional auction design, which tries to extend the traditional, single-dimensional auction setting studied in the seminal paper of Myerson [126]. A prominent characteristic that can often be seen in these papers is the "simplicity vs optimality" approach: knowing the computational hardness [43, 44, 59] and structural complexity [60, 89] of describing exact optimality, emphasis is placed on designing both simple and practical mechanisms that can still provide good revenue guarantees. Of course, this idea can be traced back to the work of Hartline and Roughgarden [92] and Bulow and Klemperer [29] for the single-dimensional setting. For a more thorough overview we refer to the recent review article of Roughgarden and Talgam-Cohen [133] and the textbook of Hartline [91].

Related to this, and placed under the general theme of what has come to be known as "Wilson's doctrine" [151] (see also [120, Section 5.2]), there has also been significant effort towards the direction of *robust* revenue maximization: designing auctions that make as few assumptions as possible on the seller's prior knowledge about the bidders' valuations for the items. Examples include models where the auctioneer can perform quantile queries [42] or knows some estimate of the actual prior [19, 30, 112]. Another line of work studies robustness with respect to the correlation of valuations across bidders or items [18, 35, 84]. Other approaches regarding the parameterization of partial distributional knowledge were considered by Dütting et al. [66] and Bandi and Bertsimas [14]. See also the recent survey by Carroll [36].

Most relevant to our work is the model of *parametric auctions*, introduced by Azar and Micali [8]. More specifically, they study single-dimensional (digital goods and single-item) auction settings with independent item valuations, under the assumption that the seller has only access to the mean μ_i and the variance σ_i^2 of each buyer's i prior distribution. Using Chebyshev-like tail bounds, they show that for the special single-bidder, single-item case, deterministically pricing at a multiple of the standard deviation below the mean, i.e. offering a take-it-or-leave-it price of $\mu - k \cdot \sigma$, guarantees an approximation ratio of $\tilde{\rho}(r)$, where $\tilde{\rho}$ is an increasing function taking values in $[1, \infty)$ and $r = \sigma/\mu$. In Appendix A.2, we actually quantify this bound and show that it grows quadratically. Under an extra assumption of Monotone Hazard Rate (MHR), they show how the even simpler selling mechanism that just prices at μ achieves an approximation ratio of e.

It is interesting to notice here that Azar and Micali [8] provide an *exact* solution, for deterministic mechanisms, to the robust optimization problem of maximizing the expected revenue. Then, they use this maximin revenue-optimal mechanism and compare it to the optimal social welfare (which is trivially also an upper bound on the optimal revenue), to finally derive their upper bound guarantee on the approximation ratio of revenue. As such, their results are not tailored to be tight for the *ratio* benchmark. As a matter of fact, in [10] the authors also provide an explicit lower bound that can be written as $1 + r^2$. This is an important motivating factor for our work, since one of our main goals is to close these gaps and provide tight approximation ratio bounds.

Azar et al. [9] use a clever reduction (see also the work of Chawla et al. [40]) to show how these results can be paired with the work of Dhangwatnotai et al. [63] regarding the VCG mechanism with reserves, in order to design parametric auctions for very general single-dimensional settings. In particular, they show how in matroid-constrained environments with the extra assumption of regularity on the prior distributions (or MHR for more general downward-closed environments), using the aforementioned parametric prices as lazy reserves guarantees a $2\tilde{\rho}(r)$-approximation to the optimal (Myersonian) revenue and a $\tilde{\rho}(r)$-approximation to the optimal social welfare. Here $r = \max_i \sigma_i/\mu_i$.

Another work which is close to ours is that of Carrasco et al. [34]. The authors essentially extend the model of Azar and Micali [8] to randomized mechanisms, solving the maximin robust optimization problem with respect to revenue. Again, in principle their results cannot be immediately translated to tight bounds for the approximation ratio; however, unlike the deterministic case for which in this work we have to design a new mechanism in order to achieve ratio optimality, we will show that the maximin optimal lottery of Carrasco et al. [34] is actually also optimal for the ratio benchmark.

Sample access vs knowledge of moments Another stream of research studies models where the auctioneer has sample access to the distribution [52, 63, 74, 83, 94, 123, 147]. It is not hard to imagine scenarios where such access to individual past data might be infeasible or impractical, e.g. due to data protections and privacy restrictions. Furthermore, there might also exist computational limitations in representing a distribution, or storing and reasoning with a large number of samples. In such settings, it is more natural to assume access to only some statistical aggregates of the underlying data, such as the mean and the standard deviation.

From a theoretical perspective, the sample access model is incomparable with the moment-based model that we consider here, as they rely on different distributional assumptions. In particular, independence, regularity and/or upper bounds on the support are standard assumptions in the aforementioned sample complexity papers. As a matter of fact, these are necessary to derive non-trivial results (see e.g. the counterexample of Cole and Roughgarden [52, Footnote 3]). Furthermore, if independence is dropped, Dughmi et al. [65] demonstrate that an exponential number of samples is required in order to achieve a constant-factor approximation to the optimal revenue. In our setting, on the other hand, we require none of the above. However, we do assume (as a design principle) exact knowledge of the mean and an upper bound on the standard deviation. This information cannot be retrieved exactly via any finite amount of samples, although intervals of confidence can be used to estimate it; we leave as future work the study of the revenue maximization problem when having only approximate knowledge of the distribution moments.

3.1.2 Results and techniques

The main focus of this chapter is a multi-dimensional auction setting where a single bidder has additive valuations for m items, drawn from a joint probability distribution F. We make no further assumptions on F; in particular, we do not require F to be a product distribution nor do we enforce any kind of regularity. The seller knows only the mean μ_j and (an upper bound on) the standard deviation σ_j of each item's j marginal distribution. Based on this limited statistical information, they are asked to fix a truthful (possibly randomized) mechanism to sell the items. Then, an adversary chooses the actual distribution F (respecting, of course, the statistical (μ_j, σ_j)-information) and the seller realizes the expected revenue of the auction, in the standard Bayesian way, in expectation with respect to F. The main quantity of interest, which we call the *robust approximation ratio* is the ratio of the optimal revenue (which has full knowledge of F in advance) to this revenue.

Our worst-case, min-max approach is similar in spirit to the previous work of Azar et al. [9], Azar and Micali [10] and Carrasco et al. [34]. However, the critical difference that our main goal is to optimize the *ratio* against the optimal revenue and not just the expected revenue of the selling mechanism on its own. It turns out that, similarly to the aforementioned previous work, our bounds can be stated with respect to the ratio $r_j = \sigma_j/\mu_j$ of each item's marginal distribution. This is an important statistical quantity called the *coefficient of variation (CV)*; it is essentially a "unit-independent" measure of the dispersion of the distribution (see, e.g., [124] or [100, Sec. 2.21]).

In Section 3.2 we formally introduce our model and necessary notation. In the following two sections we focus on the single-item case, since this will be the building block for all our results. In particular, in Section 3.3 we show that the robust approximation ratio of deterministic mechanisms is *exactly* $\rho_D(r) \approx 1 + 4r^2$ (see Definition 1), closing a gap open from the work of Azar and Micali [10]. Similarly to previous work, in order to achieve this we solve exactly the corresponding min-max problem (see Lemma 2); however, the method and the solution itself have to be different, since we are dealing with the ratio, which is a more "sensitive" quantity than the revenue on its own. By "sensitive" we mean that its value changes in a less smooth and more unpredictable way for small perturbations of the distribution and the mechanism.

Next, in Section 3.4 we deal with general randomized auctions and we show that a lottery proposed by Carrasco et al. [34], which we term *log-lottery*, although designed for a different objective achieves an approximation ratio of $\rho(r) \approx 1 + \ln(1 + r^2)$ (see Definition 1) in our setting, which is asymptotically optimal. We start with a quantitative analysis of the log-lottery mechanism (Theorem 2). In particular, we show an upper bound to the robust approximation ratio that grows logarithmically in r. This bound already establishes a strong separation between the power of deterministic and randomized mechanisms. The question then becomes if a different randomized selling mechanism can achieve a sublogarithmic or even constant upper bound. We answer this in the negative by showing that the logarithmic upper bound is asymptotically tight. The construction of the lower bound instance (Theorem 3) is arguably the most technically challenging part of this chapter, and is based on a novel utilization of Yao's minimax principle that might be of independent interest for deriving robust approximation lower bounds in other Bayesian mechanism design settings as well. Informally, the adversary offers a distribution over two-point mass distributions, finely-tuned such that the resulting mixture becomes a truncated "equal-revenue style" distribution (see Fig. 3.2c). The main difference to other settings in the literature where Yao's principle is applied is that the adversary has to randomize over probability distributions, which form an infinite-dimensional space. We can imagine this as a space of "distributions over distributions". This introduces new technical challenges since the adversary's model of randomization needs to be properly defined, and more importantly, Yao's principle does not hold anymore. Thus, our goal is twofold: we need to carefully describe how the adversary constructs a space of distributions over distributions and then show that we can extend Yao's principle to such spaces.

It is important to restate that we work under the assumption that we know an *upper bound* on the standard-deviation σ and not its exact value. Although this makes our upper bounds more powerful, it is not a source of "artificial" additional power for the adversary when designing our lower bounds. We formalize this in Lemma 6. Furthermore, this helps us to formally demonstrate (see Proposition 2) that our aforementioned, Yao-based, lower bound construction lies at the "border of simplicity" of any non-trivial lower bound.

In Section 3.5 we demonstrate how the $O(\log r)$-approximate mechanism of the single-item case can be utilized to provide optimal approximation ratios for the multi-dimensional case of m items as well. More specifically, we show that selling each item j separately using the log-lottery guarantees an approximation ratio of $\rho(r_{\max})$ where $r_{\max} = \max_j r_j$ is the maximum CV across the items. If the seller has extra information that item valuations are independent (that is, F is a product distribution), then switching to a lottery that offers all items in a single full bundle can give an improved approximation ratio of $\rho(\bar{r})$, where $\bar{r} = \sqrt{\sum_j \sigma_j^2} / \sum_j \mu_j$ is the CV of the average valuation. We complement these upper bounds by tight lower bounds in Theorem 5; these constructions have at their core the single-item lower bound, but they take care of delicately assigning valuations to the remaining items so that they respect independence and the common prior statistical information. We want to highlight that the lower bound of Theorem 5 is strong enough to hold for *any* number of items and *any* choice of coefficients of variation $r_1, r_2, \dots, r_m$. An interesting corollary of our upper bounds (Corollary 1) is that for the special case of independent valuations with the same mean and variance, the approximation ratio is at most $\rho\left(\frac{\sigma}{\mu\sqrt{m}}\right)$, converging to optimality as the number of items grows large.

In Section 3.6.1 we diverge from our main model to discuss some additional "peripheral" results that can be deduced as direct corollaries of previous work combined with our upper bounds, in a "black-box" way. First, we study the single-dimensional, multi-bidder setting of *parametric auctions* introduced by Azar and Micali [10]. More specifically, we show how the positive results derived in Azar et al. [9, Theorem 4.3] can be further improved: running VCG with lazy reserve prices drawn from the log-lottery guarantees a $2\rho(r)$ approximation to the optimal Myersonian revenue (Corollary 2).

Secondly, in Section 3.6.2 we discuss how a relaxation of our model that only assumes knowledge of the mean (that is, without any information about the variance σ^2) can still produce good robust approximation ratios under an extra regularity assumption. More precisely, in Proposition 3 we give an upper bound on the approximation ratio of the mechanism that just offers the mean μ as a take-it-or-leave-it price, under the extra assumption that the item's valuation distribution is λ-regular (see Fig. 3.3a); we remind the reader that this is a general notion of regularity that interpolates smoothly between regularity à la Myerson ($\lambda = 1$) and the Monotone Hazard Rate (MHR) condition ($\lambda = 0$). Distributions that are λ-regular have been considered in recent papers in the area of mechanism design (e.g., [79, 139]). Our result extends the e-approximation for MHR distributions of Azar and Micali [8, Theorem 3]. Finally, we provide a more detailed characterization of the relationship between the knowledge of λ-regularity and knowledge of σ, with respect to the resulting robust approximation ratio upper bound (see Fig. 3.3b).

Size of the coefficient of variation It is worth discussing briefly the implications of the size of the CV, our main quantity of interest, for our results. We can observe that our upper bounds do *not* increase with the number of items m; as a matter of fact, for the case of independently distributed items with the same mean and variance, the upper bound even decreases with respect to m. Although the CV of a distribution could be arbitrarily large in general, one could argue that, for many practical scenarios, it is unlikely to encounter data with very large dispersion. From a theoretical perspective, note that the CV is actually bounded for important special classes of distributions, like MHR (which include, e.g., the truncated normal, uniform, exponential and gamma [15]) and, more generally, λ-regular for a fixed $\lambda < 1/2$ (see (3.17)). Furthermore, for general distributions, if one assumes that the CV of the item marginals are bounded by a universal constant, then our bounds yield a constant robust approximation ratio to the optimal pricing, even for correlated distributions (and regardless of the number of items).

3.2 Preliminaries

3.2.1 Model and notation

A real nonnegative random variable will be called (μ, σ)-*distributed* if its expectation is μ and its standard deviation is *at most* σ. We let $\mathbb{F}_{\mu,\sigma}$ denote the class of (μ, σ) distributions. We shall also briefly (see Lemma 6) discuss the restriction to distributions with standard deviation of *exactly* σ; this subclass will be denoted by $\mathbb{F}^{=}_{\mu,\sigma}$.

As mentioned in the introduction, for the most part of this chapter we study auctions with m items and a single additive bidder, whose valuations $(v_1, \dots, v_m)$ for the items are drawn from a joint distribution F over $\mathbb{R}^m_{\geq 0}$. We denote the marginal distribution of v_j by F_j, and assume that it has finite mean and variance. In general, we make no further assumptions for F; in particular, we do not assume independence of the random variables $v_1, \dots, v_m$ nor do we enforce any regularity or continuity assumption. For vectors $\vec{\mu} = (\mu_1, \dots, \mu_m) \in \mathbb{R}^m_{>0}$, $\vec{\sigma} = (\sigma_1, \dots, \sigma_m) \in \mathbb{R}^m_{\geq 0}$ we denote by $\mathbb{F}_{\vec{\mu},\vec{\sigma}}$ the class of all m-dimensional distributions whose j-th marginal is (μ_j, σ_j)-distributed, for all $j = 1, \dots, m$.

Let $\mathbb{A}_m$ denote the space of all truthful selling mechanisms. Then, given an m-dimensional distribution F, we denote by

- $\text{REV}(A;F) = \mathbb{E}_{\vec{v}\sim F}[\pi(\vec{v})]$, the expected *revenue* of A (the expectation is taken w.r.t. F);
- $\text{WEL}(A;F) = \mathbb{E}_{\vec{v}\sim F}[x(\vec{v}) \cdot \vec{v}]$, the expected *welfare of* A;
- $\text{OPT}(F) = \sup_{A\in\mathbb{A}_m} \text{REV}(A;F)$, the optimum revenue;
- $\text{VAL}(F) = \sup_{A\in\mathbb{A}_m} \text{WEL}(A;F)$, the optimum welfare. By definition, in this is also the welfare of a VCG auction; moreover, for a single additive bidder with a joint distribution in $\mathbb{F}_{\vec{\mu},\vec{\sigma}}$, this is just the sum of the marginal expectations, $\text{VAL}(F) = \sum_{j=1}^{m} \mu_j$.

Note that, due to (2.2), we immediately have the so-called *welfare bounds* for the above quantities: for any mechanism and distribution,

$$\text{REV}(A;F) \leq \text{WEL}(A;F) \qquad \text{and} \qquad \text{OPT}(F) \leq \text{VAL}(F).$$

Our goal in this chapter is to quantify the following benchmark

$$\text{APX}(\vec{\mu},\vec{\sigma}) = \inf_{A\in\mathbb{A}_m} \sup_{F\in\mathbb{F}_{\vec{\mu},\vec{\sigma}}} \frac{\text{OPT}(F)}{\text{REV}(A;F)}, \tag{3.1}$$

which we call the *robust approximation ratio.* The semantics are the following: a seller chooses the best (revenue-maximizing) selling mechanism A, given only knowledge of the means $\vec{\mu}$ and standard deviations $\vec{\sigma}$ and then an adversary ("nature") responds by choosing a worst-case "valid" distribution that respects the statistical information $\vec{\mu}$ and $\vec{\sigma}$. Sometimes we restrict our attention to *deterministic* mechanisms A; that is, mechanisms whose allocation rule satisfies $x(\vec{v}) \in \{0,1\}^m$, for all $\vec{v}$. Under this additional constraint, the quantity in (3.1) will be denoted by $\text{DAPX}(\vec{\mu},\vec{\sigma})$.

For the special case of a single item ($m = 1$), we know from the seminal work of Myerson [126] that every deterministic mechanism $A \in \mathbb{A}_1$ is completely determined by a single take-it-or-leave-it price $p \geq 0$; thus, we will feel free to sometimes abuse notation and write $\text{REV}(p;F)$ instead of $\text{REV}(A;F)$ if A is the deterministic auction that sells at price p.

Most importantly for our work, every randomized auction for a single item can be seen as a nonnegative random variable over prices (see Carrasco et al. [34, Footnote 10]). In particular, since the allocation rule is monotone and takes values in $[0,1]$, it can be interpreted as the cumulative distribution of a certain randomization over prices, which assigns the item with the same probability as the original mechanism.[1] In this way, for a randomized single-item auction we can abuse notation and write $p \sim A$ to denote that a price p is sampled according to A. In this way, $\text{REV}(A;F) = \mathbb{E}_{p\sim A}[\text{REV}(p;F)]$.

[1] There are only two subtle technical issues that need to be taken into account; x need not be right-continuous, and $\lim_{v\to\infty} x(v)$ need not equal 1; we can assume these without loss of generality. Otherwise, one could take the right-continuous closure of x, and either assign the remainder probability to high prices, or apply a suitable scaling, which would only increase expected revenue.

3.2.2 Determinism vs randomization

We would like to give some basic intuition on how randomization helps to hedge uncertainty. To this end, we present a simple example where a randomized strategy beats every price.

Example 1. Assume that we are facing a very restricted adversary who can choose between two distributions. Distribution A has just a point mass at 1. Distribution B is a two-point mass distribution, which returns either 0 or 2 with probability 1/2 each.

If the seller is restricted to deterministic pricing rules, it is not hard to see that their best strategy is to post a price equal to 1 (and for the adversary to choose distribution B), for a worst-case expected revenue of $\frac{1}{2}$. If the seller posts anything above 1, then the adversary will always respond with distribution A, resulting in zero revenue. Consider now the following randomization over prices: The seller posts a price of 1 with probability 2/3, and a price of 2 with probability 1/3. If the adversary chooses Distribution A, then the expected revenue will be $1 \cdot \frac{2}{3} = \frac{2}{3}$. Similarly if Distribution B is chosen, then the expected revenue becomes $1 \cdot \frac{2}{3} \cdot \frac{1}{2} + 2 \cdot \frac{1}{3} \cdot \frac{1}{2} = \frac{2}{3}$.

Regardless of the adversarial response, a randomization over two prices strictly outperforms the best deterministic pricing. In subsequent sections we formalize this intuition, by showing a significant separation between the power of deterministic and randomized mechanisms. A separation between determinism and randomization in single-dimensional settings, but under a sample access model, has been demonstrated by Fu et al. [74].

3.2.3 Auxiliary functions and distributions

To state our bounds, it will be convenient to define the following auxiliary functions.

Definition 1 (Functions ρ_D, ρ)**.** For any $r \geq 0$, let $\rho_D(r) = \rho$, resp. $\rho(r) = \rho$, be the unique positive solution of equation

$$\frac{(\rho-1)^3}{(2\rho-1)^2} = r^2, \qquad \text{resp.} \qquad \frac{1}{\rho^2}\left(2e^{\rho-1}-1\right) = r^2+1.$$

Plots of these functions, for small values of r, can be seen in Fig. 3.1. Their asymptotic behaviour is given in the following lemma, whose proof is deferred to Appendix A.1 (Lemmas 38 and 39).

Lemma 1. *For the functions ρ_D, ρ defined in Definition 1, we have the bounds and asymptotics,*

$$1+4r^2 \leq \rho_D(r) \leq 2+4r^2 \qquad \textit{for all } r \geq 0; \qquad\qquad \rho(r) = 1+(1+o(1))\ln(1+r^2).$$

We now define a specific *randomized* selling mechanism, which essentially corresponds to the lottery proposed by Carrasco et al. [34, Prop. 4]:

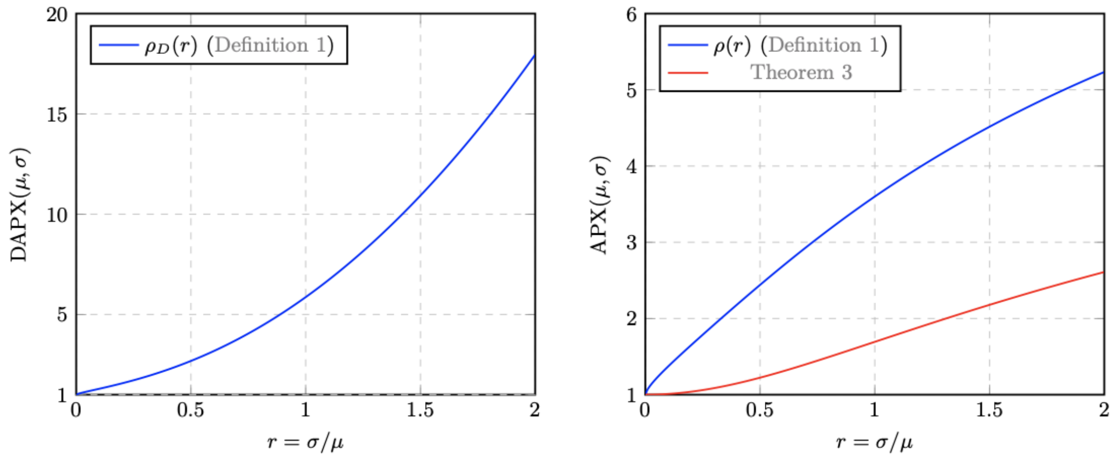


Figure 3.1: The robust approximation ratio for deterministic (left) and randomized (right, blue) selling mechanisms for a single (μ, σ)-distributed item, for small values of the coefficient of variation $r = \sigma/\mu$. The former is tight and given in Theorem 1. The latter is the upper bound given by Theorem 2; it is asymptotically matching the lower bound (red) of Theorem 3.

Definition 2 (Log-Lottery)**.** Fix any $\mu > 0$ and $\sigma \geq 0$. A *log-lottery* is a randomized mechanism that sells at a price $P^{\log}_{\mu,\sigma}$, which is distributed over the nonnegative interval support $[\pi_1, \pi_2]$ according to the cdf

$$F^{\log}_{\mu,\sigma}(x) = \frac{\pi_2 \ln \frac{x}{\pi_1} - (x - \pi_1)}{\pi_2 \ln \frac{\pi_2}{\pi_1} - (\pi_2 - \pi_1)},$$

where parameters π_1, π_2 are the (unique) solutions of the system

$$\begin{cases} \pi_1 \left(1 + \ln \dfrac{\pi_2}{\pi_1}\right) = \mu & \text{(3.2a)} \\ \pi_1(2\pi_2 - \pi_1) = \mu^2 + \sigma^2. & \text{(3.2b)} \end{cases}$$

We will sometimes slightly abuse notation and use $P^{\log}_{\mu,\sigma}$ to refer both to the log-lottery mechanism and the corresponding random variable of the prices.

3.3 Single item: Deterministic pricing

In this section, we begin our study of robust revenue maximization by looking at the simplest case: one item and deterministic pricing rules. Note that Azar and Micali [10] already established a lower bound of $1+r^2$ for this setting, together with an upper bound which can be shown to be $1+\left(\frac{27}{4} + o(1)\right) r^2$ (they actually characterized the upper bound via the solution of a cubic equation; we provide the exact asymptotics of that solution in Appendix A.2). Our result (Theorem 1) is a refined analysis that captures the exact robustness ratio (and in particular the "correct" constant in the quadratic term).

Our first observation (Lemma 2) will be that the worst-case adversarial response (for a specific selling price) can be characterized in terms of a two-point mass distribution, which allows the problem to be solved exactly. These types of distributions have appeared already in the results of Azar and Micali [8] and Carrasco et al. [34], and we will start by introducing some notation to reason about them.

A two-point mass distribution F takes some value x with probability α and some value y with probability $1-\alpha$, where without loss $x < y$. When the distribution is constrained to have mean μ and variance exactly equal to σ^2, only one free parameter remains, i.e. F can be characterized by the position x of its first point mass. The other two parameters can be obtained as

$$y(x) = \mu + \frac{\sigma^2}{\mu - x} \quad \text{and} \quad \alpha(x) = \frac{\sigma^2}{\sigma^2 + (\mu - x)^2},$$

by solving the first and second moment conditions $\mu = \alpha x + (1-\alpha)y$ and $\mu^2 + \sigma^2 = \alpha x^2 + (1-\alpha)y^2$. For the remainder, we let F_x, $x \in [0, \mu)$, denote this distribution. Note that the limiting case $x \to \mu$ corresponds to $\alpha(x) \to 1$ and $y(x) \to \infty$, meaning that F_x weakly converges to μ.

By first solving the innermost optimization problem in (3.1), i.e. by characterizing the worst-case adversarial response against a specific *deterministic* pricing, we can derive the robustness ratio for deterministic mechanisms:

Lemma 2. *For any choice of mean μ and variance σ^2, and any deterministic pricing scheme, the worst-case robust approximation ratio is achieved over a limiting two-point mass distribution. Formally, for any μ, σ, and any price p,*

1. *if $p \geq \mu$, then the worst-case response corresponds to playing F_x with $x \to \mu^-$, and*
$$\sup_{F \in \mathbb{F}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(p; F)} = \infty;$$
2. *if $0 < p < \mu$, then the worst-case response corresponds to playing F_x with $x \to p^-$, and*
$$\sup_{F \in \mathbb{F}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(p; F)} = \max\left\{1 + \frac{\sigma^2}{(\mu - p)^2}, \frac{\mu}{p} + \frac{\sigma^2}{p(\mu - p)}\right\}.$$

Proof. If $p \geq \mu$, then the worst-case robust approximation ratio can become arbitrarily large by taking $x \to \mu^-$, that is, x arbitrarily close to μ, so that $\alpha(x) \to 1$. Indeed, we have that $\mathrm{REV}(p; F_x) \leq p(1 - \alpha(x)) \to 0$, whereas $\mathrm{OPT}(F_x) \geq x \to \mu$, so that the supremum of the ratio is unbounded.

Next, let us suppose that $0 < p < \mu$. First, we compute the limit of the approximation ratio for distribution F_x, as $x \to p^-$. Observe that $\mathrm{OPT}(F_x) = \max\{x, (1 - \alpha(x))y(x)\}$; and since $x < p$, we sell the item with probability $1 - \alpha(x)$, to obtain $\mathrm{REV}(p; F_x) = p(1 - \alpha(x))$. Therefore,

$$\lim_{x \to p^-} \frac{\mathrm{OPT}(F_x)}{\mathrm{REV}(p, F_x)} = \lim_{x \to p^-} \frac{\max\{x, (1 - \alpha(x))y(x)\}}{p(1 - \alpha(x))}$$

$$= \max\left\{\frac{1}{1-\alpha(p)}, \frac{y(p)}{p}\right\}$$
$$= \max\left\{1 + \frac{\sigma^2}{(\mu-p)^2}, \frac{\mu}{p} + \frac{\sigma^2}{p(\mu-p)}\right\}.$$

Thus, it only remains to show that for any random variable X drawn from a (μ, σ) distribution F, we have that

$$\frac{\mathrm{OPT}(F)}{\mathrm{REV}(p; F)} \le \max\left\{\frac{1}{1-\alpha(p)}, \frac{y(p)}{p}\right\}.$$

We first derive a lower bound on the probability of selling the item at price p via a one-sided version of Chebyshev's inequality, also called Cantelli's inequality[2] (see, e.g., [24, p. 46]),

$$\Pr[X \ge p] = \Pr[X - \mu \ge -(\mu - p)] \ge 1 - \frac{\sigma^2}{\sigma^2 + (\mu-p)^2} = 1 - \alpha(p). \tag{3.3}$$

Let p^* denote the optimal take-it-or-leave-it price for distribution F, so that $\mathrm{OPT}(F) = p^* \Pr[x \ge p^*]$. Again, we consider two cases: if $p^* \le p$, then we have

$$\frac{\mathrm{OPT}(F)}{\mathrm{REV}(p, F)} = \frac{p^* \Pr[X \ge p^*]}{p \Pr[X \ge p]} \le \frac{1}{1-\alpha(p)} \le \max\left\{\frac{1}{1-\alpha(p)}, \frac{y(p)}{p}\right\}$$

where in the first inequality we used (3.3) and the bounds $p^* \le p$, $\Pr[X \ge p^*] \le 1$.

Next, consider the case $p^* > p$. By looking at the conditional random variable $(X|X \ge p)$, we observe that

$$\frac{p^* \Pr[X \ge p^*]}{\Pr[X \ge p]} = p^* \Pr\left[X \ge p^* | X \ge p\right] = \mathrm{REV}(p^*; F | X \ge p) \le \mathbb{E}\left[X | X \ge p\right]; \tag{3.4}$$

the inequality holds because the social welfare is always an upper bound to the revenue.

In order to bound the conditional expectation, we use a result in Mallows and Richter [118, Eq. (1.2)]. It states that if X is a real-valued random variable with mean μ and variance σ^2 and E is a non-zero probability event, then

$$\mathbb{E}[X \mid E] - \mu \le \sigma\sqrt{\frac{1-\Pr[E]}{\Pr[E]}}.$$

In our case, we use $E = (X \ge p)$, together with the lower bound in (3.3), to get

$$\mathbb{E}\left[X | X \ge p\right] \le \mu + \sigma\sqrt{\frac{1}{\Pr[X \ge p]} - 1} \le \mu + \sigma\sqrt{\frac{1}{1-\alpha(p)} - 1} = \mu + \frac{\sigma^2}{\mu - p} = y(p).$$

[2] Although the original statement of Cantelli's inequality is for a random variable with variance *equal to* σ^2, by monotonicity the same holds if σ^2 is instead an upper bound on the variance.

Finally, combining the above with (3.4) yields

$$\frac{\mathrm{OPT}(F)}{\mathrm{REV}(p,F)} = \frac{p^* \Pr[X \geq p^*]}{p \Pr[X \geq p]} \leq \frac{\mathbb{E}[X|X \geq p]}{p} \leq \frac{y(p)}{p} \leq \max\left\{\frac{1}{1-\alpha(p)}, \frac{y(p)}{p}\right\},$$

which concludes the proof. □

Theorem 1. *The* deterministic *robust approximation ratio of selling a single* (μ, σ)-*distributed item is* exactly *equal to*

$$\mathrm{DAPX}(\mu, \sigma) = \rho_D(r) \approx 1 + 4 \cdot r^2,$$

where $r = \sigma/\mu$ *and function* $\rho_D(\cdot)$ *is given in Definition 1. In particular, this is achieved by offering a take-it-or-leave-it price of* $p = \frac{\rho_D(r)}{2\rho_D(r)-1} \cdot \mu$.

Proof. For fixed μ and σ, Lemma 2 gives the worst-case approximation ratio for any choice of p. Thus, from the seller's perspective, it is clear that one should offer a price below the mean, and furthermore the outermost optimization problem reduces to finding

$$\rho = \inf_{0<p<\mu} \max\left\{1 + \frac{\sigma^2}{(\mu-p)^2}, \frac{\mu}{p} + \frac{\sigma^2}{p(\mu-p)}\right\}. \tag{3.5}$$

We begin by analysing when the first branch of the maximum is higher than the second branch. Some algebraic manipulation yields

$$1 + \frac{\sigma^2}{(\mu-p)^2} \geq \frac{\mu}{p} + \frac{\sigma^2}{p(\mu-p)} \iff (\mu-p)^3 \leq \sigma^2(2p-\mu). \tag{3.6}$$

When $p \leq \mu/2$, the right expression is nonpositive and hence the second branch of the maximum is highest. Next, observe that $\frac{(\mu-p)^3}{2p-\mu}$ is decreasing over $p \in (\mu/2, \mu)$, with a positive pole at $p = \mu/2$, and vanishing at $p = \mu$. Hence, for any choice of μ, σ, there is a unique point p^* at which (3.6) holds with equality. It follows that for $p \geq p^*$ the maximum is achieved on the first branch and for $p \leq p^*$ the maximum is achieved on the second branch.

Next, observe that $1 + \frac{\sigma^2}{(\mu-p)^2}$ is increasing on $p \in (p^*, \mu)$. To see that the second branch of the maximum in (3.5) is decreasing on $p \in (0, p^*)$, we take its derivative

$$\frac{d}{dp}\left(\frac{\mu}{p} + \frac{\sigma^2}{p(\mu-p)}\right) = \frac{-\mu(\mu-p)^2 + \sigma^2(2p-\mu)}{p^2(\mu-p)^2}.$$

When $p \leq p^*$ we have (by definition of p^*) that $\sigma^2(2p-\mu) \leq (\mu-p)^3$ and hence the above quantity is at most $-1/p$, which is negative. We conclude that the minimum occurs when both branches intersect, i.e. at $p = p^*$; using the fact that $(\mu - p^*)^3 = \sigma^2(2p^* - \mu)$, we can further express the value of the minimum as

$$\rho = 1 + \frac{\sigma^2}{(\mu-p^*)^2} = 1 + \frac{\mu - p^*}{2p^* - \mu} = \frac{p^*}{2p^* - \mu}.$$

We can now use this to express p^* in terms of ρ,

$$p^* = \frac{\rho}{2\rho - 1} \cdot \mu; \qquad \mu - p^* = \frac{\mu(\rho-1)}{2\rho-1}; \qquad 2p^* - \mu = \frac{\mu}{2\rho - 1}.$$

Putting these together, we get

$$\sigma^2 = \frac{(\mu - p^*)^3}{2p^* - \mu} = \mu^2 \frac{(\rho-1)^3}{(2\rho-1)^2} \quad \Longleftrightarrow \quad \frac{(\rho-1)^3}{(2\rho-1)^2} = \left(\frac{\sigma}{\mu}\right)^2 \equiv r^2;$$

and the desired asymptotics follow from Definition 1 and Lemma 1.

For the current proof to be self contained, we repeat here the arguments that show the bounds and asymptotics for ρ and can be found also in Lemma 38. One can directly check that the expression $\frac{(\rho-1)^3}{(2\rho-1)^2}$ is increasing and goes from 0 at $\rho = 1$ to ∞ at $\rho \to \infty$, so that for any nonnegative r there is a unique solution $\rho \in [1, \infty)$ to the above equation. Moreover, we can write

$$r^2 = \frac{(\rho-1)^3}{(2\rho-1)^2} = \frac{1}{4}\rho - \frac{1}{4} - \frac{(\rho - \frac{3}{4})(\rho-1)}{4(\rho-\frac{1}{2})^2} \quad \Longleftrightarrow \quad \rho = 1 + 4r^2 + \frac{(\rho-\frac{3}{4})(\rho-1)}{(\rho-\frac{1}{2})^2};$$

since the fraction appearing on the right-hand side takes values between 0 and 1 (for $\rho \in [1, \infty)$), this gives us the desired global bounds. □

3.4 Single item: Lotteries

In this section, we continue to focus on a single-item setting, but now we study the robust approximation ratio that can be achieved by a randomized mechanism, i.e., by randomizing over posted prices. Carrasco et al. [34] have given the explicit solution to the robust *absolute* revenue problem,

$$\sup_{A \in \mathbb{A}_1} \inf_{F \in \mathbb{F}_{\mu,\sigma}} \mathrm{REV}(A; F). \tag{3.7}$$

We state below a proposition that can be directly derived from their work and which would be very useful for our setting.

Proposition 1. *For $\mu > 0$, $\sigma \geq 0$, the value of the maximin problem* (3.7) *is given by*

$$\sup_{A \in \mathbb{A}_1} \inf_{F \in \mathbb{F}_{\mu,\sigma}} \mathrm{REV}(A; F) = \pi_1,$$

where π_1 is derived by the unique solution of the system (3.2a)-(3.2b). *Moreover, this value is achieved by the log-lottery $P^{\log}_{\mu,\sigma}$ described in Definition 2.*

Proof. In this proof we refer to multiple points in the paper from Carrasco et al. [34]. The optimal mechanism for (3.7) is given by the allocation rule (see their Proposition 4)

$$x(v) = \begin{cases} 0, & \text{for} \quad v \leq \pi_1, \\ \lambda_1 \ln \frac{v}{\pi_1} + 2\lambda_2(v - \pi_1), & \text{for} \ \ \pi_1 \leq v \leq \pi_2, \\ 1, & \text{for} \ \ \pi_2 \leq v, \end{cases} \tag{3.8}$$

and the value of the maximin problem (3.7) is (see end of page 274[3])

$$\sup_{A\in\mathbb{A}_1}\ \inf_{F\in\mathbb{F}_{\mu,\sigma}}\ \mathrm{REV}(A;F)=\lambda_0+\lambda_1\mu+\lambda_2(\mu^2+\sigma^2), \tag{3.9}$$

where the values of $\lambda_0,\lambda_1,\lambda_2$ are given by (see (B.4-B.6))

$$\lambda_0=-\frac{\pi_1(2\pi_2-\pi_1)}{2\left(\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)\right)};\lambda_1=\frac{\pi_2}{\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)};\lambda_2=-\frac{1}{2\left(\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)\right)}. \tag{3.10}$$

Note that, as we explained at the end of Section 3.2.1, the allocation rule $x(v)$ from (3.8) can be interpreted as the cdf of a randomization over prices which forms an equivalent mechanism. Moreover, by replacing the values of $\lambda_0,\lambda_1,\lambda_2$ as in (3.10) we get

$$x(v)=\frac{\pi_2\ln\frac{v}{\pi_1}-(v-\pi_1)}{\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)},$$

which is exactly the log-lottery of Definition 2.

Finally, by replacing the values of $\lambda_0,\lambda_1,\lambda_2$ from (3.10), and the values of μ and σ from (3.2a),(3.2b), into (3.9), the value of the maximin problem can be greatly simplified to

$$\begin{aligned}&\lambda_0+\lambda_1\mu+\lambda_2(\mu^2+\sigma^2)\\&=-\frac{\pi_1(2\pi_2-\pi_1)}{2\left(\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)\right)}+\frac{\pi_2\pi_1\left(1+\ln\frac{\pi_2}{\pi_1}\right)}{\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)}-\frac{\pi_1(2\pi_2-\pi_1)}{2\left(\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)\right)}\\&=\frac{\pi_1\left(\pi_2+\pi_2\ln\frac{\pi_2}{\pi_1}-2\pi_2+\pi_1\right)}{\pi_2\ln\frac{\pi_2}{\pi_1}-(\pi_2-\pi_1)}=\pi_1,\end{aligned}$$

as we wanted to prove. □

The above characterization can be directly used to derive a logarithmic upper bound on the robust approximation ratio:

Theorem 2. *The robust approximation ratio of selling a single (μ,σ)-distributed item is at most*

$$\mathrm{APX}(\mu,\sigma)\le\rho(r)\approx 1+\ln(1+r^2),$$

where $r=\sigma/\mu$ and function ρ is given in Definition 1. In particular, this is achieved by the log-lottery described in Definition 2.

[3] Carrasco et al. [34] define their solutions in terms of the moments $k_1\equiv\mu$ and $k_2\equiv\mu^2+\sigma^2$.

Proof. By Proposition 1, if A is the log-lottery from Definition 2, then for any (μ, σ) distribution F we have that $\mathrm{REV}(A; F) \geq \pi_1$. Thus, using the trivial upper bound of $\mathrm{OPT}(F) \leq \mu$ for the optimal revenue, we can derive an upper bound of $\frac{\mu}{\pi_1}$ on the approximation ratio. For convenience, let us denote this by $\rho \equiv \mu/\pi_1$.

Manipulating (3.2a) we get

$$\pi_1 \left(1 + \ln \frac{\pi_2}{\pi_1}\right) = \mu \quad \Longleftrightarrow \quad \ln \frac{\pi_2}{\pi_1} = \frac{\mu}{\pi_1} - 1 \quad \Longleftrightarrow \quad \frac{\pi_2}{\pi_1} = e^{\rho - 1}$$

and so from (3.2b) we can derive

$$\pi_1(2\pi_2 - \pi_1) = \mu^2 + \sigma^2 \quad \Longleftrightarrow \quad \frac{\pi_1^2}{\mu^2}\left(2\frac{\pi_2}{\pi_1} - 1\right) = \frac{\sigma^2}{\mu^2} + 1 \quad \Longleftrightarrow \quad \frac{1}{\rho^2}\left(2e^{\rho-1} - 1\right) = r^2 + 1,$$

which is exactly the equation in Definition 1. The asymptotic behaviour follows from Lemma 1. □

By looking at the proof of the previous theorem, it is not difficult to see that our upper bound is also an upper bound with respect to welfare (which for a single (μ, σ) distribution is simply given by μ). If we were interested in comparing the revenue of our auction to the maximum welfare, then it immediately follows from Proposition 1 that the bound is exact and tight. However, our main goal in the current chapter is to provide tight bounds with respect to the optimal *revenue*, and achieving this requires some extra work. The rest of our section is devoted to proving and discussing the following lower bound, which asymptotically matches that of Theorem 2.

Theorem 3. *For a single (μ, σ)-distributed item, the robust approximation ratio is at least*

$$\mathrm{APX}(\mu, \sigma) \geq 1 + \ln(1 + r^2),$$

where $r = \sigma/\mu$.

Before we go into the actual construction of our lower bound instances, we need some technical preliminaries and to recall Yao's principle (see, e.g., [23, Sec. 8.3] or [125, Sec. 2.2.2]). As we already mentioned (see Section 3.2.1), a randomized mechanism $A \in \mathbb{A}_1$ can be interpreted as a randomization over prices $p \sim A$. From (3.1), we are interested in the value of a game in which the mechanism designer plays first, randomizing over posted prices, and the adversary plays second, choosing a worst-case distribution. Intuitively, Yao's principle states that this is at least the value of another game in which the adversary plays first, randomizing over their choices, and the mechanism designer plays second, choosing a deterministic response, i.e., a single posted price.

However, to define this second game formally, we would have to first explain what it means for the adversary to randomize over probability distributions, which form an infinite-dimensional space. In order to avoid technical or measure-theoretical issues, we focus on a specific model of randomization, which in the literature gives rise to the concept of *mixture* or *contagious* distribution (see, e.g., Mood et al. [121, Ch. III.4]).

Definition 3. Let $\mathfrak{F}$ be a class of cumulative distribution functions over the nonnegative reals, and consider any measure space over a ground set T. By an $\mathfrak{F}$*-mixture with parameter space* T, we mean a pair (Θ, F), where Θ is a probability measure in T, and F is a measurable function of type $F : \mathbb{R}_{\geq 0} \times T \to \mathbb{R}$, whose sections are in $\mathfrak{F}$; i.e. for any parameter $\theta \in T$, the function

$$F_\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}, \quad F_\theta(x) = F(x; \theta),$$

is a cumulative distribution in $\mathfrak{F}$.

Given an $\mathfrak{F}$-mixture (Θ, F), we denote its *posterior* distribution by $\mathbb{E}_{\theta\sim\Theta}[F_\theta]$; this is specified by the cdf

$$\mathop{\mathbb{E}}_{\theta\sim\Theta}[F_\theta](z) = \int F(z;\theta)d\Theta(\theta) = \mathop{\mathbb{E}}_{\theta\sim\Theta}[F_\theta(z)].$$

When $\mathfrak{F} = \mathbb{F}_{\mu,\sigma}$ is the class of (μ, σ) distributions, we shall let $\Delta_{\mu,\sigma}$ denote the class of (μ, σ) mixtures, that is, the class of mixtures over $\mathbb{F}_{\mu,\sigma}$ (with arbitrary, unspecified parameter space). We can interpret (Θ, F) as a convex combination of distributions, so that the cdf of $\mathbb{E}_{\theta\sim\Theta}[F_\theta]$ is the convex combination of the corresponding cdfs; alternatively, $\mathbb{E}_{\theta\sim\Theta}[F]$ can be seen as the cdf of a random variable that first samples a distribution F_θ according to $\theta \sim \Theta$, and then samples a value z according to F_θ.

Now that we have carefully described the adversarial model, we can formally state a version of Yao's principle (Lemma 4 below) that will help us prove lower bounds. Since this applies on "non-standard" continuous spaces, for completeness we need to formally derive it "from scratch"; this is what we do in the next lemma.

Lemma 3. *Let $(X, \Sigma_X, \mathcal{F})$ and $(Y, \Sigma_Y, \mathcal{G})$ be arbitrary probability spaces,*[4] *i.e.*

- *Σ_X and Σ_Y are σ-algebras over X and Y respectively;*
- *$\mathcal{F}$ and $\mathcal{G}$ are probability measures over (X, Σ_X) and (Y, Σ_Y) respectively.*

Let also $h : X \times Y \to \mathbb{R}_{\geq 0}$, $g : Y \to \mathbb{R}_{>0}$ be measurable functions. Then[5]

$$\sup_{y\in Y} \frac{g(y)}{\mathbb{E}_{x\sim\mathcal{F}}[h(x,y)]} \geq \inf_{x\in X} \frac{\mathbb{E}_{y\sim\mathcal{G}}[g(y)]}{\mathbb{E}_{y\sim\mathcal{G}}[h(x,y)]}.$$

Proof. Let α be an arbitrary nonnegative real number, and suppose that

$$\inf_{x\in X} \frac{\mathbb{E}_{y\sim\mathcal{G}}[g(y)]}{\mathbb{E}_{y\sim\mathcal{G}}[h(x,y)]} \geq \alpha, \tag{3.11}$$

[4] For formal definitions of the measure-theoretic notions used in this lemma see, e.g., Tao [148].

[5] Throughout this lemma, we handle ratios of the form $\frac{g}{h}$ where $g > 0$ and $h \geq 0$. For convenience, if $h = 0$ we interpret the ratio as being equal to ∞. This means that, for any nonnegative real number α, we have the following relation, even when $h = 0$:

$$\frac{g}{h} \geq \alpha \quad \iff \quad g \geq \alpha \cdot h.$$

that is, $\mathbb{E}_{y\sim\mathcal{G}}[g(y)] \geq \alpha \sup_{x\in X} \mathbb{E}_{y\in\mathcal{G}}[h(x,y)]$. This implies that, for every $x \in X$, we have $\mathbb{E}_{y\sim\mathcal{G}}[g(y)] \geq \alpha\mathbb{E}_{y\in\mathcal{G}}[h(x,y)]$. Hence, by sampling x according to $\mathcal{F}$, we also have

$$\mathbb{E}_{y\sim\mathcal{G}}[g(y)] \geq \alpha\mathbb{E}_{x\sim\mathcal{F}}[\mathbb{E}_{y\sim\mathcal{G}}[h(x,y)]] = \alpha\mathbb{E}_{y\sim\mathcal{G}}[\mathbb{E}_{x\sim\mathcal{F}}[h(x,y)]];$$

the equality holds due to Tonelli's theorem (see, e.g., Tao [148, Theorem 1.7.15]), since h is measurable and nonnegative, and $\mathcal{F}$, $\mathcal{G}$ are finite measures. By the previous inequality between expectations, we must conclude that it holds for some realization of $\mathcal{G}$, that is, there must exist $y \in \operatorname{supp}(\mathcal{G})$ such that $g(y) \geq \alpha\mathbb{E}_{x\sim\mathcal{F}}[h(x,y)]$. This implies that

$$\frac{g(y)}{\mathbb{E}_{x\sim\mathcal{F}}[h(x,y)]} \geq \alpha, \quad \text{and hence} \quad \sup_{y\in Y} \frac{g(y)}{\mathbb{E}_{x\sim\mathcal{F}}[h(x,y)]} \geq \alpha.$$

As α was any real number that satisfies (3.11), the desired inequality follows. □

Now we are ready to apply this novel version of Yao's principle for the robust approximation ratio in our model.

Lemma 4. *For any μ, σ, we have the following lower bound on the robust approximation ratio,*

$$\inf_{A\in\mathbb{A}_1} \sup_{F\in\mathbb{F}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(A;F)} \geq \sup_{(\Theta,F)\in\Delta_{\mu,\sigma}} \inf_{p\geq 0} \frac{\mathbb{E}_{\theta\sim\Theta}[\mathrm{OPT}(F_\theta)]}{\mathbb{E}_{\theta\sim\Theta}[\mathrm{REV}(p;F_\theta)]}.$$

Proof. Start by fixing an arbitrary truthful mechanism $A \in \mathbb{A}_1$ and an arbitrary (μ,σ) mixture (Θ, F) over parameter space T. Since A can be interpreted as a randomization over prices, $(\mathbb{R}_{\geq 0}, \mathcal{L}, A)$ is a well-posed probability space.

Next, define the functions

$$h : \mathbb{R}_{\geq 0} \times T \to \mathbb{R}, \quad g : T \to \mathbb{R};$$

$$h(p,\theta) = \mathrm{REV}(p;F_\theta); \quad g(\theta) = \mathrm{OPT}(F_\theta).$$

Clearly, h is nonnegative and g is positive since F_θ is (μ,σ)-distributed. We just need to argue that both are measurable. Note that

$$h(p,\theta) = \mathrm{REV}(p;F_\theta) = p(1 - F_\theta(p-)) = \inf_{y<p} p(1 - F(y;\theta)).$$

As F is measurable and taking extrema preserves measurability, so is h. In a similar way, g is measurable as it can be expressed as the supremum

$$g(\theta) = \mathrm{OPT}(F_\theta) = \sup_{p\geq 0} \mathrm{REV}(p;F_\theta).$$

Hence, we can directly apply Lemma 3 and conclude that

$$\sup_{F\in\mathbb{F}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathbb{E}_{p\sim A}[\mathrm{REV}(p;F)]} \geq \sup_{\theta\in T} \frac{\mathrm{OPT}(F_\theta)}{\mathbb{E}_{p\sim A}[\mathrm{REV}(p;F_\theta)]} \geq \inf_{p\geq 0} \frac{\mathbb{E}_{\theta\sim\Theta}[\mathrm{OPT}(F_\theta)]}{\mathbb{E}_{\theta\sim\Theta}[\mathrm{REV}(p;F_\theta)]}.$$

As A and (Θ, F) were arbitrary, we can take the supremum on the right-hand side over (μ,σ) mixtures, and the infimum on the left-hand side over truthful mechanisms; the result follows. □

Note that, by using (2.3), we can rewrite the denominator of the previous quantity as follows:

$$
\begin{aligned}
\sup_{p\geq 0} \mathop{\mathbb{E}}_{\theta\sim\Theta}[\mathrm{REV}(p;F_\theta)] &= \sup_{p\geq 0} \mathop{\mathbb{E}}_{\theta\sim\Theta}[p(1-F_\theta(p-))] \\
&= \sup_{p\geq 0} p\left(1-\mathop{\mathbb{E}}_{\theta\sim\Theta}[F_\theta(p-)]\right) \\
&= \sup_{p\geq 0} p\left(1-\mathop{\mathbb{E}}_{\theta\sim\Theta}[F_\theta](p-)\right) \\
&= \sup_{p\geq 0} \mathrm{REV}\left(p;\mathop{\mathbb{E}}_{\theta\sim\Theta}[F_\theta]\right) \\
&= \mathrm{OPT}\left(\mathop{\mathbb{E}}_{\theta\sim\Theta}[F_\theta]\right).
\end{aligned}
$$

The second equality comes from linearity of expectation and the third one follows from the definition of a mixture distribution. Putting all these together, we arrive at the following key technical result:

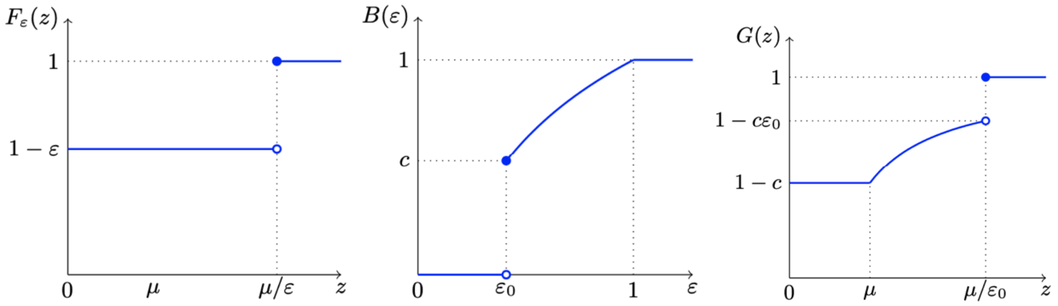
Lemma 5. *For any μ,σ, the robust approximation ratio is lower bounded by*

$$
\mathrm{APX}(\mu,\sigma) \geq \sup_{(\Theta,F)\in\Delta_{\mu,\sigma}} \frac{\mathbb{E}_{\theta\sim\Theta}[\mathrm{OPT}(F_\theta)]}{\mathrm{OPT}(\mathbb{E}_{\theta\sim\Theta}[F_\theta])}\,. \tag{3.12}
$$

From a practical perspective, the above result has a positive consequence. It allows us to obtain lower bounds by constructing a single (μ,σ) mixture, (Θ,F), and calculating the expected optimal revenue before and after the realization of $\theta\sim\Theta$. Our goal is to make this ratio as high as possible and, ideally, match the competitive ratio of the log-lottery pricing. From this, we can gain some insight into how to construct a "good" mixture. By looking at the right-hand side of the inequality in Lemma 4, we would intuitively expect that different posted prices p yield similar revenues of $\mathbb{E}_{\theta\sim\Theta}[\mathrm{REV}(p;F_\theta)] = \mathrm{REV}(p;\mathbb{E}_{\theta\sim\Theta}[F_\theta])$. Thus, we would aim for a mixture (Θ,F) for which the posterior distribution is equal-revenue for at least some subset of its support.

From a theoretical perspective, the quantity in (3.12) is interesting by itself. One can check that the Myerson operator is convex, that is, the revenue achieved by a convex combination of distributions can only be smaller than the convex combinations of the corresponding revenues. Thus, by Jensen's inequality, the ratio in (3.12) is always at least 1. On the other hand, for a linear functional $\mathcal{L}$, we have that $\mathbb{E}_{\theta\sim\Theta}[\mathcal{L}(F_\theta)] = \mathcal{L}(\mathbb{E}_{\theta\sim\Theta}[F_\theta])$. Thus, (3.12) somehow attempts to quantify the extent to which OPT is nonlinear, or in other words, it can be understood as a *measure of convexity of the Myerson operator.* In any case, we can use this result to construct lower bound instances and prove the main result of this section:

Proof of Theorem 3. We shall construct a (μ,σ) mixture over two-point mass distributions. Each two-point mass distribution F_ε is given by a unique choice of parameter



(a) Two-point mass distribution with one mass at 0 and another at μ/ε.

(b) Mixing in parameter space; each ε corresponds to a two-point mass distribution as in Fig. 3.2a. Note that this distribution has a mass at ε_0.

(c) Posterior distribution from the mixture obtained via Figs. 3.2a and 3.2b.

Figure 3.2: The cdfs of the various distributions used in the lower bound construction of Theorem 3.

$\varepsilon \in (0,1]$; F_ε returns 0 with probability $1-\varepsilon$ and μ/ε with probability ε. Note that F_ε has mean μ and variance $\mu^2(1/\varepsilon - 1)$. The upper bound of σ^2 on the variance implies that we can only take values of $\varepsilon \geq \varepsilon_0 \equiv \frac{1}{1+r^2}$, where r is the coefficient of variation (our quantity of interest).

Our next step is to describe the convex mixture of these distributions. Define a random variable with support $[\varepsilon_0, 1]$ and distributed according to B as follows:

- B has a point mass at ε_0 of size c;
- B is continuous over $(\varepsilon_0, 1]$, with density $\beta(\varepsilon) = c/\varepsilon$.

The value of c is given by $c = \frac{1}{1+\ln(1+r^2)}$ and is chosen as a normalizing constant; indeed,

$$1 = \mathop{\mathbb{E}}_{\varepsilon \sim B}[1] = c + c \ln \frac{1}{\varepsilon_0} = c\left(1 + \ln\left(1 + r^2\right)\right).$$

Our (μ, σ) mixture distribution thus corresponds to sampling F_ε where $\varepsilon \sim B$. Next, we describe the posterior distribution $G = \mathbb{E}_{\varepsilon \sim B}[F_\varepsilon]$. Its cumulative function can be seen in Fig. 3.2c.

- Mass at 0: as each F_ε has a point mass at 0, so does G. The value of this mass is given by

$$\mathop{\mathbb{E}}_{\varepsilon \sim B}[\text{mass of } F_\varepsilon \text{ at } 0] = \int_{\varepsilon_0}^{1} (1-\varepsilon)\beta(\varepsilon) d\varepsilon + (1-\varepsilon_0)c = c \ln \frac{1}{\varepsilon_0} = 1 - c;$$

- Mass at μ/ε_0: as B has a point mass at ε_0 and F_{ε_0} has a point mass at μ/ε_0, this implies that G has a point mass at μ/ε_0 of size $c\varepsilon_0$;

- cdf in $[\mu, \mu/\varepsilon_0)$: for each $z \in [\mu, \mu/\varepsilon_0)$, $F_\varepsilon(z)$ is $(1-\varepsilon)$ for $\varepsilon < \mu/z$ and 1 for $\varepsilon \geq \mu/z$; thus the cdf of G can be computed as

$$G(z) = \int_{\varepsilon_0}^{\mu/z} (1-\varepsilon)\beta(\varepsilon)d\varepsilon + \int_{\mu/z}^{1} \beta(\varepsilon)d\varepsilon + (1-\varepsilon_0)c = 1 - \frac{c\mu}{z}.$$

We can interpret $G(z)$ as a *truncated equal-revenue* distribution over the interval $[\mu, \mu/\varepsilon_0)$, *with additional point masses* at 0 and μ/ε_0. In particular, every posted price in $[\mu, \mu/\varepsilon]$ yields the same (optimal) revenue, and $\mathrm{OPT}(G) = c\mu = \frac{\mu}{1+\ln(1+r^2)}$. On the other hand, note that for every $\varepsilon > 0$ we have $\mathrm{OPT}(F_\varepsilon) = \mu$, so $\mathbb{E}_{\varepsilon\sim B}[\mathrm{OPT}(F_\varepsilon)] = \mu$. Plugging these into (3.12) yields a lower bound of $1/c = 1 + \ln(1+r^2)$ as desired. □

From the previous proof, some further discussion and remarks are in order. Note that our mixture uses distributions F_ε, which for $\varepsilon > \varepsilon_0$ have a variance *strictly smaller* than σ^2. Since we have defined our adversarial model to play (μ, σ) distributions, such instances are allowed. However, one may wish to ensure that the adversary only picks distributions in $\mathbb{F}^{=}_{\mu,\sigma}$ (i.e. with *exact* equality on the variance); this might be relevant, for example, if the seller had extra information about the exact value of σ; or, from a theoretical perspective, such a restriction of the adversary would only make our lower bound more "clear" and powerful. We shall now argue that indeed our assumption on having just a bound on the standard deviation, is not only a technical convenience (and, arguably, more realistic), but also is without loss of generality for our bounds. Intuitively, for any mechanism A and any (μ, σ) distribution F, one can "perturb" F into a distribution in $\mathbb{F}^{=}_{\mu,\sigma}$ having nearly the same approximation ratio. Below we formalize this intuition for single-item settings, although it is not hard to see how to generalize it to higher dimensions.

Lemma 6. *For single-item settings, the restriction of the robust approximation problem from (μ, σ) distributions to distributions in $\mathbb{F}^{=}_{\mu,\sigma}$ does not change its value. Formally, for any $\mu > 0$, $\sigma \geq 0$, and any mechanism A, we have*

$$\sup_{F\in\mathbb{F}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(A;F)} = \sup_{F\in\mathbb{F}^{=}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(A;F)}\ ;$$

and hence

$$\inf_{A\in\mathbb{A}_1} \sup_{F\in\mathbb{F}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(A;F)} = \inf_{A\in\mathbb{A}_1} \sup_{F\in\mathbb{F}^{=}_{\mu,\sigma}} \frac{\mathrm{OPT}(F)}{\mathrm{REV}(A;F)}\ .$$

Proof. Let μ and σ be given, and let A be any mechanism and F_0 any (μ, σ) distribution. Suppose that the variance of F_0 is $\tilde{\sigma}^2 < \sigma^2$. For each $\delta \in (0, 1]$, let us define the perturbed distribution F_δ as the following convex combination of distributions:

- with probability $1-\delta$, sample a value according to F_0;
- with probability δ, sample a value according to the rare event distribution that is 0 with probability $1-\varepsilon$ and μ/ε with probability ε;

- the value of ε is chosen so that F_δ has variance *exactly equal to* σ^2; in other words, it is obtained by solving the system

$$(1-\delta)(\mu^2+\tilde{\sigma}^2)+\delta\mu^2/\varepsilon=\mu^2+\sigma^2 \quad\Longrightarrow\quad \varepsilon=\frac{\delta\mu^2}{\delta\mu^2+\sigma^2-(1-\delta)\tilde{\sigma}^2}.$$

Note that, for each δ, F_δ has the desired mean of μ as it is the convex combination of two distributions of mean μ. Moreover, as $\delta \to 0$, also $\varepsilon \to 0$, so that F_δ weakly converges to F_0. Finally, we have the trivial bounds

$$\mathrm{REV}(A;F_\delta)\le(1-\delta)\mathrm{REV}(A;F_0)+\delta\mu; \qquad \mathrm{OPT}(F_\delta)\ge(1-\delta)\mathrm{OPT}(F_0),$$

which can be combined to yield

$$\frac{\mathrm{OPT}(F_\delta)}{\mathrm{REV}(A;F_\delta)}\ge\frac{(1-\delta)\mathrm{OPT}(F_0)}{(1-\delta)\mathrm{REV}(A;F_0)+\delta\mu}\;.$$

By letting δ go to 0, we have

$$\sup_{F\in\mathbb{F}^{=}_{\mu,\sigma}}\frac{\mathrm{OPT}(F)}{\mathrm{REV}(A;F)}\ge\lim_{\delta\to0}\frac{(1-\delta)\mathrm{OPT}(F_0)}{(1-\delta)\mathrm{REV}(A;F_0)+\delta\mu}=\frac{\mathrm{OPT}(F_0)}{\mathrm{REV}(A;F_0)}\;.$$

Taking suprema over F_0 on the right-hand side yields the first statement of our lemma; and taking infima over A on both sides yields the last statement. □

It should also be mentioned that, in principle, we could accommodate the proof of Theorem 3 to handle distributions with exact equality with respect to σ, with minor technical modifications. More precisely, one would define $F_{\varepsilon,\delta}$ as a perturbation of F_ε as in the proof of Lemma 6. This would yield an approximation ratio that depends on δ, which would then be taken in the limit $\delta \to 0$.

Another observation is that the "bad instances" that we used for Theorem 3 were two-point mass distributions, with one of the points being 0. Note that these differ from the instances we used in the deterministic lower bounds (Lemma 2, Theorem 1), which were two-point mass distributions with exact variance of σ^2. These latter instances were actually shown in [34] to be worst-case distributions for their objective function, and they were also used in Azar and Micali [10] to prove maximin optimality in their model. Thus, it would be natural to wonder whether such instances could have been actually enough to prove a matching lower bound in the randomized setting. Below we answer this question in the negative; in other words, we prove a *constant* upper bound when the adversary is forced to pick one of these distributions.

Proposition 2. *For every choice of μ,σ, there is a randomized mechanism A that achieves (at least) a $\frac{1}{4}$-fraction of the optimal revenue on* any *distribution F that is a two-point mass with mean μ and variance σ^2. In particular, A is the mechanism that offers price $\frac{1}{2}\mu$ with probability $\frac{1}{2}$ and $\mu+\frac{\sigma^2}{\mu}$ with probability $\frac{1}{2}$.*

Proof. Let us analyse the performance of A on a two-point mass distribution F_x, say with a point mass at x of size $\alpha(x)$ and another at $y(x)$ of size $1-\alpha(x)$, with $x<\mu<y(x)$. If $\frac{1}{2}\mu \leq x$ then the mechanism chooses with probability $1/2$ a price that always sells, guaranteeing revenue of $\frac{\mu}{4}$, which is also a $1/4$-fraction of $\mathrm{OPT}(F)$. Next, suppose that $x \leq \frac{1}{2}\mu$. This implies

$$1-\alpha(x)=\frac{(\mu-x)^2}{\sigma^2+(\mu-x)^2}, \quad y(x)=\mu+\frac{\sigma^2}{\mu-x}\leq \mu+2\frac{\sigma^2}{\mu},$$

since $y(x)$ is a nondecreasing function. Moreover, we have that

$$(1-\alpha(x))y(x)=\frac{\sigma^2(\mu-x)+(\mu-x)^2\mu}{\sigma^2+(\mu-x)^2}\geq \frac{\mu}{2}\frac{\sigma^2+2(\mu-x)^2}{\sigma^2+(\mu-x)^2}\geq\frac{\mu}{2}\geq x,$$

so that $\mathrm{OPT}(F)$ is achieved by pricing at $y(x)$. Our mechanism A chooses with probability $1/2$ a price of $\mu+\frac{\sigma^2}{\mu}$, which sells with probability $1-\alpha(x)$. Thus the approximation ratio is at least

$$\frac{\frac{1}{2}(1-\alpha(x))\left(\mu+\frac{\sigma^2}{\mu}\right)}{(1-\alpha(x))y(x)}\geq\frac{1}{2}\frac{\mu+\frac{\sigma^2}{\mu}}{\mu+2\frac{\sigma^2}{\mu}}=\frac{1}{4}\frac{\sigma^2+\mu^2}{\sigma^2+\frac{1}{2}\mu^2}>\frac{1}{4};$$

so that the mechanism achieves a $1/4$-fraction of $\mathrm{OPT}(F)$ in this case as well. □

The proposition above implies that the lower bound from Theorem 3 would break down, if the adversary is restricted to the family of two-point mass distributions with exact variance of σ^2.

3.5 Multiple items

In this section we finally consider the more general setting of a single additive buyer with valuations for m items. As it turns out, the main tools developed in Section 3.4 can be leveraged very naturally to produce similar upper and lower bounds. We begin by proving upper bounds for both correlated and independent item valuations.

Theorem 4. *The robust approximation ratio of selling m (possibly correlated) $(\vec{\mu},\vec{\sigma})$-distributed items is at most*

$$\mathrm{APX}(\vec{\mu},\vec{\sigma})\leq\rho(r_{\max}), \qquad \textit{where} \quad r_{\max}=\max_{j=1,\dots,m} r_j, \quad r_j=\frac{\sigma_j}{\mu_j}$$

and function ρ is given in Definition 1. This is achieved by selling each item j separately *using the log-lottery $P^{\log}_{\mu_j,\sigma_j}$ from Definition 2.*

Furthermore, if the items are independently *distributed, the above bound improves to*

$$\mathrm{APX}(\vec{\mu},\vec{\sigma})\leq\rho(\bar{r}), \qquad \textit{where} \quad \bar{r}=\frac{\bar{\sigma}}{\bar{\mu}}, \quad \bar{\mu}=\sum_{j=1}^{m}\mu_j, \quad \bar{\sigma}=\sqrt{\sum_{j=1}^{m}\sigma_j^2},$$

achieved by selling the items in a single full-bundle *using the log-lottery $P^{\log}_{\bar{\mu},\bar{\sigma}}$ from Definition 2.*

Proof. Let X_j, $j = 1, \dots, m$, be (μ_j, σ_j)-distributed random variables corresponding to the marginals of the joint m-dimensional valuation distribution F. Their sum $Y = \sum_{i=1}^{m} X_i$ has an expected value of $\mathbb{E}[Y] = \sum_{j=1}^{m} \mu_j = \bar{\mu} = \text{VAL}(F)$. Furthermore, if $X_1, \dots, X_j$ are independent, its variance is $\text{Var}[Y] = \sum_{j=1}^{m} \text{Var}[X_j] \leq \sum_{j=1}^{m} \sigma_j^2 = \bar{\sigma}^2$. Denote the distribution of Y by F_Y. Also, recall that the optimal revenue of F cannot exceed the expected welfare, thus we have the trivial upper bound of

$$\text{OPT}(F) \leq \text{VAL}(F) = \sum_{j=1}^{m} \mu_j,$$

no matter if the distributions are independent or not.

For our general upper bound first, observe that selling item j using a lottery A_j, where $A_j = P^{\log}_{\mu_j,\sigma_j}$ is the log-lottery of Definition 2, guarantees (Theorem 2) a revenue of at least

$$\text{REV}(A_j; F_j) \geq \frac{\mu_j}{\rho(r_j)}. \tag{3.13}$$

Thus, if A is the mechanism that sells independently each item j using A_j, we can get the following approximation ratio upper bound for our total revenue

$$\frac{OPT(F)}{\text{REV}(A;F)} = \frac{OPT(F)}{\sum_{j=1}^{m} \text{REV}(A_j;F_j)} \leq \frac{\sum_{j=1}^{m} \mu_j}{\sum_{j=1}^{m} \frac{\mu_j}{\rho(r_j)}} \leq \rho(r_{\max}),$$

where the last inequality holds due to the monotonicity of $\rho(\cdot)$: $\rho(r_j) \leq \rho(r_{\max})$ for all j.

For the case of independent valuations, observe that a feasible selling mechanism for our items is to bundle them all together and treat them as a single item, i.e., price their sum of valuations Y. Since Y is $(\bar{\mu}, \bar{\sigma})$-distributed, offering a log-lottery $A = P^{\log}_{\bar{\mu},\bar{\sigma}}$ for Y results in an approximation ratio guarantee of

$$\text{APX}(\vec{\mu}, \vec{\sigma}) \leq \frac{OPT(F)}{\text{REV}(A;F_Y)} \leq \frac{\mathbb{E}[Y]}{\frac{1}{\rho(\bar{r})}\mathbb{E}[Y]} = \rho(\bar{r}),$$

for $\bar{r} = \bar{\sigma}/\bar{\mu}$.

Finally, to verify that $\rho(\bar{r}) \leq \rho(r_{\max})$, due to the monotonicity of $\rho(\cdot)$ it is enough to see that

$$\bar{r} = \frac{\bar{\sigma}}{\bar{\mu}} = \frac{\left(\sum_{j=1}^{m} \sigma_j^2\right)^{1/2}}{\bar{\mu}} \leq \frac{\sum_{j=1}^{m} \sigma_j}{\bar{\mu}} = \frac{\sum_{j=1}^{m} \mu_j r_j}{\sum_{j=1}^{m} \mu_j}$$

is a weighted average of $r_1, r_2, \dots, r_m$, and thus at most $r_{\max}$. □

Corollary 1. *The robust approximation ratio of selling m independently (μ, σ)-distributed items is at most*

$$\text{APX}(\vec{\mu}, \vec{\sigma}) \leq \rho\left(\frac{r}{\sqrt{m}}\right),$$

where $r = \sigma/\mu$, achieved by selling the items in a single full-bundle *using the mechanism given in Theorem 2.*

Proof. In the proof of Theorem 4, if $X_1, \dots, X_m$ are independent random variables with mean μ and standard deviation at most σ, then for their sum Y we have $\bar{\mu} = m \cdot \mu$ and $\bar{\sigma} \leq \sqrt{m\sigma^2} = \sqrt{m}\sigma$. □

Remark. For deterministic mechanisms, it is not difficult to see that the robust approximation ratio of selling m (possibly correlated) $(\vec{\mu}, \vec{\sigma})$-distributed items is at most $\mathrm{DAPX}(\vec{\mu}, \vec{\sigma}) \leq \tilde{\rho}(r_{\max})$ (where $\tilde{\rho}$ is given in Appendix A.2); just replace ρ by $\tilde{\rho}$ in the proof of Theorem 4. In particular, the validity of (3.13) is implied by (A.1).

We make a few observations at this point. Notice that when moving from a single item to many items, our approximation guarantees do not degrade; in particular, the robust approximation ratio is at most that of the "worst" item (i.e. the item with the highest coefficient of variation). In fact, for m independently (μ, σ)-distributed items the approximation ratio even converges to optimality (Corollary 1); this can be seen as a reinterpretation of the known result that full-bundling is asymptotically optimal for an additive bidder and many i.i.d. items (see Hart and Nisan [90, A.5.]), but in our framework of minimal statistical information.

Although the mechanisms presented in Theorem 4 are extremely simple (lotteries over separate pricing or bundle pricing), we can actually show asymptotically matching lower bounds for *any* choice of the coefficients of variation:

Theorem 5. *Fix any positive integer m and positive real numbers $r_1, \dots, r_m$, and let $r = \max_j r_j$. Then, for any $\varepsilon > 0$, there exist $\vec{\mu} = (\mu_1, \dots, \mu_m) \in \mathbb{R}^m_{>0}$, $\vec{\sigma} = (\sigma_1, \dots, \sigma_m) \in \mathbb{R}^m_{\geq 0}$ with $r_j = \sigma_j/\mu_j$, such that*

$$\mathrm{APX}(\vec{\mu}, \vec{\sigma}) \geq 1 - \varepsilon + \ln(1 + r^2).$$

Furthermore, this lower bound is achieved by independent *(μ_j, σ_j)-distributions.*

Proof. Let $m, r_1, \dots, r_m, \varepsilon$ be as in the statement of the theorem, and without loss assume $\max_j r_j = r_1$. Let $\delta > 0$ be chosen such that $\delta \ln(1 + r^2)(1 + \ln(1 + r^2))^2 < \varepsilon$. We shall choose the values for the mean and variance as

$$\mu_1 = 1, \quad \sigma_1 = r_1,$$
$$\mu_j = \frac{\delta}{m-1}, \quad \sigma_j = r_j \frac{\delta}{m-1} \quad \text{for } j \geq 2.$$

The idea is that we create a "bad" instance in which items $2, \dots, m$ are rare event distributions with very little welfare and so their contribution to the revenue will be negligible. To that end, we must first introduce some notation. For every item $j \geq 2$, denote

$$p_j = \frac{1}{1 + r_j^2}, \quad \alpha_j = (1 + r_j^2) \frac{\delta}{m-1},$$

and for every $S \subseteq \{2, \dots, n\}$, i.e. for every subset of the "low" items,

$$p_S = \prod_{j \in S} p_j \cdot \prod_{j \notin S \cup \{1\}} (1 - p_j).$$

Also, define the event

$$E_S = \left[\bigwedge_{j \in S} (v_j = \alpha_j)\right] \wedge \left[\bigwedge_{j \notin S \cup \{1\}} (v_j = 0)\right].$$

Next, let A be any m-dimensional truthful mechanism, i.e.. a mechanism for selling m items to a single bidder. For each $S \subseteq \{2, \dots, n\}$, let A_S be the 1-dimensional mechanism induced by event E_S; intuitively, this mechanism allocates according to A with the values v_j set as in E_S, but discounting the payment by the welfare from items in S. Formally, if A is defined by allocation and payment rules, $A = (\vec{x}, \pi)$, then $A_S = (x_S, \pi_S)$ can be defined as

$$x_S(v_1) = x_1(v_1, \vec{v}_{-1}), \qquad \pi_S(v_1) = \pi(v_1, \vec{v}_{-1}) - \vec{v}_{-1} \cdot \vec{x}_{-1}(0, \vec{v}_{-1}),$$

where $\vec{v}_{-1} = (v_2, \dots, v_m)$ and, for $j \geq 2$, we have $v_j = \alpha_j$ if $j \in S$; and $v_j = 0$ if $j \notin S$. One can directly check that A_S defines a truthful mechanism.

Now define $\bar{A} = \sum_S p_S A_S$ to be the convex combination of mechanisms A_S. This can be interpreted as the one-dimensional mechanism that samples a subset $S \subseteq \{2, \dots, n\}$ with probability p_S and then runs mechanism A_S. Finally, we apply Theorem 3 that ensures the existence of a "bad" single-item distribution for mechanism A_S, i.e. a distribution F_1 with mean μ_1 and standard deviation σ_1 such that

$$\text{REV}(\bar{A}; F_1) \leq \frac{\text{OPT}(F_1)}{1 + \ln(1 + r^2)}. \tag{3.14}$$

Each of the remaining distributions, F_j for $j = 2, \dots, m$, is a rare event distribution that assigns a mass of p_j on value α_j, and a mass of $1 - p_j$ on value 0. It is not hard to see that F_j has the desired mean of μ_j and variance of σ_j^2. To conclude the proof, let $F = F_1 \times \cdots \times F_m$ be the product distribution corresponding to item-independent valuations; it only remains to show that

$$\frac{\text{OPT}(F)}{\text{REV}(A; F)} \geq 1 - \varepsilon + \ln(1 + r_1^2).$$

We first recall a standard revenue-decomposition inequality (see the proof of Hart and Nisan [90, Lemma 8]). For any $S \subseteq \{2, \dots, n\}$, we know that

$$\text{REV}(A; F_1 \times \cdots \times F_m | E_S) \leq \text{REV}(A_S; F_1) + \text{VAL}(F_2 \times \cdots \times F_m | E_S).$$

By the construction of our two-point mass distributions F_j, $j \geq 2$, we know that E_S form a partition of all possible valuation profiles, each event occurring with probability p_S; in this way, we can sum over the conditional expected revenues,

$$\begin{aligned}\text{REV}(A; F) &= \sum_S p_S \text{REV}(A; F_1 \times \cdots \times F_m | E_S) \\ &\leq \sum_S p_S \left(\text{REV}(A_S; F_1) + \text{VAL}(F_2 \times \cdots \times F_m | E_S)\right)\end{aligned}$$

$$= \text{REV}\left(\sum_S p_S A_S; F_1\right) + \sum_S p_S \text{VAL}(F_2 \times \cdots \times F_m | E_S)$$
$$\leq \frac{\text{OPT}(F_1)}{1+\ln(1+r^2)} + \text{VAL}(F_2 \times \cdots \times F_m). \tag{3.15}$$

Next, we consider two cases. If $\text{REV}(A;F) \leq \frac{1}{(1+\ln(1+r^2))^2}$, then recall that by the mechanism presented in Theorem 2 one can extract revenue of at least $\frac{1}{1+\ln(1+r^2)}$ from F_1, hence

$$\frac{\text{OPT}(F)}{\text{REV}(A;F)} \geq \frac{1/(1+\ln(1+r^2))}{1/(1+\ln(1+r^2))^2} = 1+\ln(1+r^2).$$

Hence we can assume that $\text{REV}(A;F) \geq \frac{1}{(1+\ln(1+r^2))^2}$. Note that by selling the items separately, and in particular using a price of α_j for items $j = 2, \ldots, m$ we can lower bound the optimal revenue by

$$\text{OPT}(F_1, F_2, \ldots, F_m) \geq \text{OPT}(F_1) + \sum_{j=2}^{m} \text{OPT}(F_j) = \text{OPT}(F_1) + \text{VAL}(F_2 \times \cdots \times F_m). \tag{3.16}$$

Using this bound, together with the derivation in (3.15) and the fact that $\text{VAL}(F_2 \times \cdots \times F_m) = \delta$, yields

$$\frac{\text{OPT}(F)}{\text{REV}(A;F)} \geq \frac{\text{OPT}(F_1)+\delta}{\text{REV}(A;F)}$$
$$\geq \frac{(1+\ln(1+r^2))(\text{REV}(A;F)-\delta)+\delta}{\text{REV}(A;F)}$$
$$= 1+\ln(1+r^2) - \delta\frac{\ln(1+r^2)}{\text{REV}(A;F)}$$
$$\geq 1+\ln(1+r^2) - \delta\ln(1+r^2)(1+\ln(1+r^2))^2$$
$$\geq 1+\ln(1+r^2) - \varepsilon,$$

as we wanted to prove. □

One observation at this point is that our result for multiple items is in line with the main result of Carroll [35], but for the robust approximation ratio objective and in our framework of minimal statistical information. Carroll also considers a multi-dimensional setting with m items and a single additive buyer. In contrast to ours, the seller has full knowledge of the marginal distributions (but again does not know the joint distribution) and wants to optimize the maximin expected revenue. A crucial common point with our model is that the seller knows nothing about the correlation between the items. Similar to our main result, he proves that selling the items separately is maximin optimal. In other words, with no information regarding correlations, the seller chooses to *never* bundle items. A possible interpretation of this result is the following: We know that for some correlation structures, bundling works fine, while for others, it can be very bad.

Thus, the seller, who wants to be robust against an unknown, possibly correlated joint distribution, might hesitate to sell as a single unit items with no information about their correlation. At the same time, the seller can calculate the optimal revenue from selling each item separately in Carroll's model. Combining these two facts intuitively makes selling separately a natural candidate for maximin optimality of the expected revenue. Our result supports this interpretation for the ratio objective and partial distributional knowledge of the marginals. Even when facing uncertainty for the revenue from a single item, the seller still chooses not to bundle items when the correlation structure is entirely unknown.

3.6 Further results

3.6.1 Parametric auctions with lazy reserves

In this section, we present (Corollary 2) an additional immediate consequence of our results to the setting of Azar et al. [9]. Since this is not the main focus of our work, we refer to the above papers, as well as Hartline [91, Ch. 4] for formal definitions. The key components are that we consider a single-dimensional, matroid-constrained environment with n bidders, meaning that the set of feasible allocations forms a matroid over $\{1, \dots, n\}$. A class of mechanisms of particular interest are called Lazy-VCG with reserve prices $(P_1, \dots, P_n)$, where $P_1, \dots, P_n$ are nonnegative random variables. This auction works by first selecting a welfare-maximizing set W of candidate winners (i.e., running a VCG auction) and then offering to an agent $i \in W$ a take-it-or-leave-it price sampled according to P_i. An important result in this setting is the following black-box reduction from many bidders to one bidder with good performance guarantees (see also Chawla et al. [40, Thm. A.3]):

Theorem 6 (Azar et al. [9]). *Assume a single-dimensional, matroid-constrained environment with n bidders having valuations drawn independently from regular distributions $F_1, F_2, \dots, F_n$. If $P_1, \dots, P_n$ are nonnegative random variables such that for all players i*

$$\mathbb{E}_{p \sim P_i}[\mathrm{REV}(p; F_i)] \geq c_1 \cdot \mathrm{OPT}(F_i) \qquad \textit{and} \qquad \mathbb{E}_{p \sim P_i}[\mathrm{WEL}(p; F_i)] \geq c_2 \cdot \mathrm{VAL}(F_i)$$

for constants $c_1, c_2 \in [0, 1]$, then Lazy-VCG with random reserves $(P_1, \dots, P_n)$ guarantees (in expectation) a $\frac{1}{2}c_1$-fraction of the optimal revenue and a c_2-fraction of the optimal welfare.

As an immediate consequence, since our log-lotteries from Section 3.4 satisfy the conditions of Theorem 6 with a suitable choice of c_1, c_2, we get the following:

Corollary 2. *Assume a single-dimensional, matroid-constrained environment with n bidders having independent regular valuations with mean μ_i and standard deviation σ_i. Then Lazy-VCG with a reserve for player i drawn from the log-lottery $P^{\log}_{\mu_i,\sigma_i}$ (see Definition 2) guarantees a $2\rho(r)$-approximation to the optimal revenue and a $\rho(r)$-approximation to the optimal welfare, where $r = \max_i \frac{\sigma_i}{\mu_i}$ and function $\rho(\cdot)$ is defined in Definition 1.*

Proof. Take $c_1 = c_2 = \frac{1}{\rho(r)} \leq \frac{1}{\rho(\sigma_i/\mu_i)}$ for all i. Note that the welfare bounds come "for free" since for any mechanism $A \in \mathbb{A}_1$ we have $\text{WEL}(A; F_i) \geq \text{REV}(A; F_i)$ and the upper bound in Theorem 2 was derived with respect to $\text{VAL}(F_i) = \mu_i$. □

3.6.2 Regularity vs dispersion

Note that regularity plays an important role in the previous Corollary 2, as it enables the black-box reduction of Azar et al. [9] to achieve meaningful upper bounds on the robust approximation ratio for a class of multi-bidder auctions. Given this observation, an obvious question would be whether additional knowledge of regularity can help us design better auctions, even for the single-item, single-bidder setting of Sections 3.3 and 3.4. In this section, we consider the notion of λ-regularity, prove some basic results (Corollary 3) and discuss some interesting implications.

We will need the following auxiliary results for λ-regular distributions, which follow from Propositions 2 and 4, and their corresponding proofs, of [139].

Proposition 3 (Schweizer and Szech [139])**.**

1. *Let F be λ-regular for some $\lambda \in [0, 1)$. Then F has a finite mean, say μ, and we have the inequality*

$$P(X > \mu) \geq (1-\lambda)^{\frac{1}{\lambda}} \quad \text{for } \lambda \neq 0, \quad P(X > \mu) \geq \frac{1}{e} \quad \text{for } \lambda = 0.$$

2. *Let F be λ-regular for some $\lambda \in [0, 1/2)$. Then F has a finite variance, say σ^2, and we have the inequality*

$$\sigma^2 \leq \frac{\mu^2}{1-2\lambda}.$$

Now we can state our main result in this section:

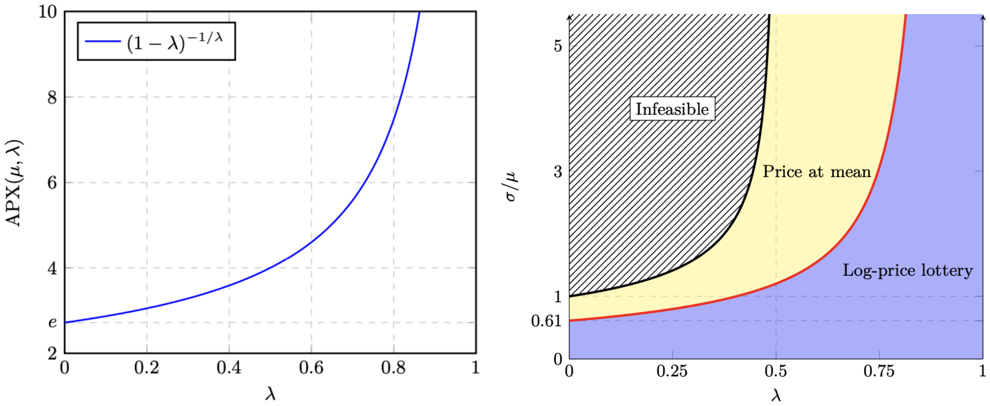
Corollary 3. *Consider a single-item, single-bidder setting in which the seller has knowledge of the mean μ and an upper bound on the regularity $\lambda \in (0, 1]$ of distribution F. Then we can achieve a robust approximation ratio of $(1-\lambda)^{-1/\lambda}$ by offering the mean as a selling price.*

Proof. Using an upper bound of μ on the revenue of an optimal auction, and the lower bound on the selling probability given by Proposition 3, the result immediately follows as

$$\frac{\text{OPT}(F)}{\text{REV}(\mu; F)} \leq \frac{\mu}{\mu(1-\lambda)^{1/\lambda}} = (1-\lambda)^{-1/\lambda}.$$

□

Note that Corollary 3 gives an upper bound that degrades from e at $\lambda = 0$ (MHR), to ∞ at $\lambda = 1$ (regular); see Fig. 3.3 for a plot of this quantity. Next, we compare this ratio against the logarithmic ratio from Theorem 2. In other words, consider a model in which the bidder has information about three quantities of the distribution F: its mean



(a) Approximation ratio for λ-regular distributions. **(b)** Knowledge of λ-regularity vs variance σ^2.

Figure 3.3: (a) The robust approximation ratio upper bound when pricing at the mean μ of a λ-regular distribution. (b) Description of our proposed single-item, single-bidder mechanism under knowledge of (μ, σ, λ). Note that $\lambda < 1/2$ already implies an upper bound on the coefficient of variation σ/μ (black curve). Moreover, if σ/μ is sufficiently small (in particular, smaller than the function of λ given by the red curve), then offering a lottery over prices (blue area) guarantees a better approximation ratio than simply pricing at the mean (yellow area).

μ, an upper bound of σ^2 on its variance, and an upper bound of λ on its "regularity". Combining our results so far, we can postulate a selling strategy, summarized in Fig. 3.3. The first observation is that some triples (μ, σ, λ) are infeasible in the following sense: if the seller knows an upper bound on λ, and furthermore $\lambda < 1/2$, then this immediately implies an upper bound on the coefficient of variation by Proposition 3; in particular, the seller would know that

$$\sigma/\mu \leq \sqrt{1/(1-2\lambda)} \tag{3.17}$$

Thus, we can assume without loss that triple (μ, σ, λ) obeys this inequality.

Next, we compare the robust approximation ratios of our two candidate strategies, to determine when the log-lottery of Definition 2 outperforms the pricing-at-the-mean from Corollary 3. This amounts to solving the inequality

$$\rho\left(\frac{\sigma}{\mu}\right) \leq \frac{1}{(1-\lambda)^{1/\lambda}}.$$

Since ρ is strictly increasing, this is equivalent to

$$\frac{\sigma}{\mu} \leq \rho^{-1}\left(\frac{1}{(1-\lambda)^{1/\lambda}}\right) = \sqrt{\frac{1}{(1-\lambda)^{2/\lambda}}\left(2e^{(1-\lambda^{1/\lambda}-1} - 1\right) - 1},$$

where for the last equality we simply rewrote the equation in Definition 1 in terms of r. The conclusion is that the upper bound for the log-lottery is better than the upper bound for pricing-at-the-mean iff σ/μ is below a certain cutoff point (which depends on λ). Note that Fig. 3.3 does not show the actual approximation ratio, but rather it partitions the $(\lambda, \sigma/\mu)$-space into regions where (the approximation guarantee of) each mechanism is better.

Some additional observations about Fig. 3.3 are in order. First, in the limit $\lambda \to 1$, our best guarantee comes from offering the log-lottery mechanism (i.e. knowledge of 1-regularity does not improve the currently best known approximation guarantees for a single-item and a single-bidder); secondly, there is a value of σ/μ, approximately equal to 0.61, below which offering the log-lottery mechanism achieves a better guarantee than that provided by pricing-at-the-mean, *regardless of* the regularity parameter $\lambda \in [0, 1]$. Intuitively, one could say that *knowing that the standard deviation of F is at most 61% its mean gives better revenue guarantees than knowing that F is MHR*, at least for single-item, single-bidder settings.

3.7 Discussion and future directions

In this chapter, we studied the robust approximation ratio of revenue maximization under minimal statistical information of the bidders' prior distribution on the item valuations. The fundamental quantities of interest turn out to be the coefficients of variation (CV), $r_j = \sigma_j/\mu_j$, of the marginal distributions. For the single-item, single-bidder case, we completely characterized this ratio for deterministic mechanisms (quadratic in r) and gave asymptotically tight bounds for randomized mechanisms (logarithmic in r). This yields natural upper bounds for the multi-item, single-additive-bidder setting. The tight lower bound is particularly powerful as it works for any choice of the r_j. Moreover, the results hold for a possibly correlated prior distribution F over the items, with only knowledge of the mean and an upper bound on the standard deviation of each marginal. The optimal mechanism turns out to be very simple: sell the items separately using the optimal randomized mechanism for the single-item case. It is also worth mentioning that although the upper bounds for the single item generalize straightforwardly to multiple items via the welfare bounds (which are trivial upper bounds to the optimal revenue), proving that these are the "correct" bounds requires careful technical work. At the heart of our analysis lies a new version of Yao's principle, which applies to the "non-standard" continuous spaces that arise in the single-item setting and might be of independent interest. As an interesting consequence, we have observed how our results can be immediately applied to the single-dimensional, multi-bidder setting proposed by Azar et al. [9], and also made a short digression into a setting in which additional information on the regularity is assumed.

We believe that the general topic of "robust revenue with minimal statistical information" gives rise to many interesting questions and variants; below we propose directions for possible future work.

Approximation ratio vs absolute revenue As we already mentioned in this chapter, besides the robust approximation *ratio* in (3.1), another quantity of independent interest is that given in (3.7):

$$\sup_{A \in \mathbb{A}_1} \inf_{F \in \mathbb{F}} \operatorname{REV}(A; F).$$

This can be seen as a "vanilla" notion of robust revenue maximization, and it was considered in Azar and Micali [10] (where they proved maximin optimality for deterministic mechanisms); it was also of central interest in the work by Carrasco et al. [34] and in other works in the economics and management science literature (e.g., [35, 107, 146]).

It is perhaps subjective to ask which, if any, of the two quantities is "better", as both have their merits. From a theoretical perspective, the *absolute revenue* in (3.7) is a simpler quantity (e.g. it behaves linearly with respect to convex combinations of mechanisms and distributions) and thus probably easier to extend to other settings; furthermore, it might be more appealing to an economist. On the other hand, the *approximation ratio* in (3.1) is "dimensionless" or "scale-free", and arguably rather natural for a computer scientist.

Consider the following thought experiment, that highlights this comparison from a more practical perspective. You are the head of a selling platform and your marketing team offers you two possible selling mechanisms:

- Mechanism A (in expectation) guarantees 10$ on each item for sale, but only 25% of the optimal revenue.

- Mechanism B (in expectation) guarantees 50% of the optimal revenue, but only 5$ on each item for sale.

One possible answer could be that "it (almost) doesn't matter": for single-item randomized mechanisms, we proved that the maximin optimal lottery of Carrasco et al. [34] yields asymptotically the best possible guarantee for the robust approximation ratio. However, it is not at all clear if, in general settings, the maximin optimal auction always achieves a guarantee "similar to" (say, a constant away of) the robust ratio-optimal auction. Providing an answer to the debate above is of course beyond the scope of this thesis. Nevertheless, we briefly presented it here as a potentially stimulating topic for future work and discussion, both from a theoretical and an empirical/behavioural point of view.

Multiple bidders We would also like to point out a qualitative change between the many-items and many-bidders settings, when moving to them from the basic single-bidder, single-item scenario: for a single bidder and many items, the approximation guarantee does not degrade; it is essentially bounded by the approximation guarantee of the "worst" item (see Theorem 4). For a single item and many bidders, however, even with the assumption of independent, regular distributions, we gain an extra factor of 2 (see Corollary 2), coming from the general black box reduction in Azar et al. [9]. It would be interesting to see if this factor can be dropped (or alternatively, provide stronger lower bounds). We believe that a promising way to attack this question would be to study existing or novel bounds on the coefficient of variation of the *maximum order statistic* of random variables, which may be of independent interest to statisticians. Of course, the most ambitious extension would be to consider multi-dimensional, multi-bidder settings (a generalization of both our work and that of Azar et al. [9], Azar and Micali [10]).

Finally, below we propose alternative, or more general, models of limited statistical information that might be interesting for future work:

Broader classes of value functions An interesting next case would be to study the setting of, say, a single unit-demand bidder and many items, or perhaps more generally, other valuation models such as constrained additivity or submodularity.

Higher-order moments Carrasco et al. [34] already looked at a single-item, single-bidder case for the "vanilla" revenue maximization problem (3.7) under knowledge of the first N moments of the valuation distribution; they characterized the solution in terms of an N-dimensional optimization problem, and briefly described it for the case of $N = 3$. The most intriguing question in this line of work would be to understand the dependence of the approximation guarantee on the number of moments N and, specifically, whether it converges to optimality and at what rate. In other words, what would be *the "moment complexity" of robust revenue maximization*?

4 The Secretary Problem with Independent Sampling

4.1 Introduction

The secretary problem, in which we search for the best secretary of an online sequence of candidates, is probably the most well-studied optimal stopping problem. These problems, motivated as *decision-making under uncertainty*, are characterized by a decision-maker who needs to decide when to stop an input sequence of information and take an action upon stopping. Optimal stopping problems, and in particular the secretary problem, originally arose in connection to labor markets, which is also insinuated by the name of the secretary problem. However, they have applications in many subfields of economics and management, such as monetary theory, industrial organization, e-commerce, and finance.

In finance, a well-known application of high-dimensional optimal stopping is in option pricing, such as swing and American options [45, 46, 49]. Ideas from optimal stopping, often closely related to prophet inequalities, have been employed to design posted price mechanisms in various scenarios [21, 39, 47, 117]. Recently, Derakhshan et al. [62] consider the problem of computing personalized reserve prices in online advertising using a dataset of past bids. Moreover, Ma et al. [117] use techniques from optimal stopping in assortment optimization. Babaioff et al. [12] and Kleinberg [104] show that generalizations of the classic secretary problem serve as a framework for online auctions. In industrial organization, extensions of the secretary problem have modeled situations where a group decision within a firm has to be made [3], or firms are competing to hire employers from a pool of candidates [58, 95]. Finally, because of its simplicity and broad applicability, variations of the secretary problem have been studied experimentally. Such papers usually describe the optimal policy for the scenario they are studying, and then through field experiments, try to explain the cognitive strategies that the agents develop. Some examples include the classical secretary problem [140], a cardinal i.i.d. setting [5], choosing which apartment to rent [153], trying to buy a plane ticket online [16], and learning when to stop by playing a repeated secretary problem [82].

Mathematically, in the secretary problem, we are faced with a randomly permuted sequence of n elements with arbitrary values. The elements' values are revealed one at a time. Upon receiving an element, we need to make an irrevocable decision of whether we keep the value and stop the sequence or drop the value forever and continue observing the next. The goal is to maximize the probability of stopping with the largest value. For this problem the best possible success guarantee has long been known to be $1/e$. The optimal algorithm is remarkably simple: Look at the first n/e values without taking any of them, and then stop with the first value larger than all values seen so far [67, 73, 113]. As mentioned also in Chapter 2, in the last decades, the secretary problem, its variants, and related basic optimal stopping problems such as the prophet inequality and the Pandora's box problem have been considered fundamental building blocks of online selection problems [20, 64, 109, 110, 150].

An essential limitation of the secretary problem for modeling real-world situations is the assumption that the values of the elements that have not yet been revealed are completely unknown. This is a very pessimistic assumption, as in realistic situations one would expect to have some available information, coming, for instance, from the context or past data. As a consequence, the best possible $1/e$ success probability for the secretary problem can be substantially improved in many settings. This gives rise to the following natural question: what is a reasonable model to take into account this additional available information? A first approach is to assume that the numbers originate from a distribution that is known to the algorithm. This assumption is relevant when the process at hand has been repeated many times, and past data can be aggregated into a distribution. Along these lines, already in the sixties, Gilbert and Mosteller [80] considered the so-called full information secretary problem in which we additionally know that the elements' values are i.i.d. random variables from a known distribution. For this variant, they showed how to compute the optimal stopping rule by dynamic programming and were able to conclude, numerically, that the best possible success probability is $\gamma \approx 0.5801$. In subsequent work, Samuels [138] finds an explicit expression for this quantity. Esfandiari et al. [71] relaxed the i.i.d.-ness assumption, considering the problem when the elements' values are arbitrary independent random variables. They show that one can guarantee a success probability of 0.517, which, quite surprisingly, was very recently improved to γ by Nuti [129]. Interestingly, in this full information model with independent but not necessarily identical values, Allaart and Islas [2] showed that if the order is not random but adversarial, the optimal stopping rule guarantees a success probability of $1/e$.[1]

While assuming no knowledge about the values seems too pessimistic, assuming that the full distribution is known might be too optimistic for most scenarios. Indeed, a typical situation would be that we have access to past data, but not enough to safely reconstruct a distribution. These informational issues in optimal stopping have given rise to a stream of research aiming at understanding the tradeoffs between the amount of information available and the success probabilities that can be derived. In this context, Azar et al. [11] pioneered the study of data-driven versions of optimal stopping problems. Recently, Rubinstein et al. [135] established a notable result in this direction for the classic prophet inequality. They prove that a single sample from each distribution, rather than its full knowledge, is enough to achieve the optimal guarantee. Also, for the prophet secretary problem, the variant of the prophet inequality when the elements come in random order, one sample has been proved to be quite effective [57].

However, this *sampling* approach still assumes that there is an underlying distribution from which we can effectively sample. In many situations, particularly when unexpected

[1]Esfandiari et al. [71] also obtained this result.

events may happen, this assumption may still feel a bit strong. Ideally, we would like to combine the idea of having samples representing past data with having arbitrary values chosen adversarially, to ensure maximum robustness while requiring no additional assumption. Recently, Kaplan et al. [98] study such a model.[2] In their model, there are n arbitrary values, and they sample a fraction p of them at random. Then the non-sampled values are presented to the decision-maker in either random order or adversarial order. Kaplan et al. [98] design algorithms for maximizing the expectation, rather than the probability of picking the maximum, that translate into algorithms for data-driven versions of prophet inequalities.

In this chapter, we consider an alternative sampling model, inspired by that of Campbell and Samuels [33] and Kaplan et al. [98]. The main difference is that in our model, the sampling of each element is performed independently with the same fixed probability. Such data-driven versions are well-motivated from several perspectives. First, in many applications, the decision-maker has access to historical data that give some insight into the distribution of future values. In our model, this information is captured in the form of samples that the decision-maker knows a priori. Second, the model is robust in the sense that only minimal knowledge of the involved data is needed. And, third, the general idea is closely related to machine learning methods that use predictors to learn the distribution. The insight here is that for problems that can be modeled as data-driven versions of the secretary problem, these learning procedures are overly complicated: The simple combinatorial model presented in this chapter already makes it possible to significantly increase the overall solution quality.

Of course, for large n our model is essentially equivalent to the model of Campbell and Samuels [33] and Kaplan et al. [98]. However, our independent sampling has two crucial advantages. On the one hand, independence makes many mathematical calculations a lot simpler and thus allows to obtain simpler expressions. On the other hand, it allows dealing with instances of unknown size, which is often the case in practical applications. In particular, several of our results hold if we do not know n. A slight disadvantage of the independent sampling model is that we may end up sampling all n elements. For consistency in this case, we assume, by vacuity, that we *win* (i.e., pick the maximum). However, this is not very restrictive since, as we will see, the difficult instances involve large values of n for a fixed value of p.

Our main result is to obtain the best possible algorithms, i.e., those maximizing the probability of selecting the largest element, for any prescribed sampling probability p and whether the order in which the elements are presented is either random or adversarial. These results uncover interesting connections between the quality of the solution and the amount of past data available to a decision-maker.

4.1.1 The problem

We are given n elements with values $\alpha_1, \ldots, \alpha_n$, which are unknown to us, and an order $\sigma : [n] \to [n]$. Each element is sampled independently with probability p. Let S be the

[2] Interestingly, the model was proposed much earlier by Campbell and Samuels [33] and recently rediscovered.

(random) set of sampled elements and V be the remaining elements, also referred to as the online set or the set of online elements. The elements in V are then presented to us in the order dictated by σ. Once an element is revealed we either pick it and stop the sequence or drop it forever and continue. The goal is to maximize the probability of picking the maximum valued element in V. In the *adversarial order secretary problem with p-sampling* (AOSp) the order σ is chosen by an adversary that knows all values $\alpha_1, \ldots, \alpha_n$ and the random sets S and V.[3] In the *random order secretary problem with p-sampling* (ROSp) the order σ is just a uniform random permutation.

Given n and an algorithm we define its *success probability* as the infimum over all values $\alpha_1, \ldots, \alpha_n$ of the probability that the algorithm stops with the maximum $\alpha_i \in V$. Moreover, the *success guarantee* of an algorithm is the infimum over all values of n of its success probability.

All algorithms considered in this chapter are *ordinal*, i.e., algorithms whose decision to stop at a given point depend only on the relative rankings of the values seen so far, and not on the actual values that have been observed, plus, possibly, on some external randomness. We observe that this is without loss of generality as for AOSp and ROSp general algorithms cannot perform better than ordinal algorithms. Indeed, as noted by Kaplan et al. [98, Theorem 2.3], a result of Moran et al. [122] implies the existence of an infinite subset of the natural numbers where general algorithms behave like ordinal algorithms (for single selection ordinal objective functions such as ours). Therefore, and because the worst case performance of our algorithms is attained as $n \to \infty$, our bounds apply to general algorithms.

4.1.2 Our results

For AOSp we consider the following very simple algorithm. Upon observing the sample set S we take as threshold the value of its k-th largest element for $k = \left\lfloor \frac{1}{1-p} \right\rfloor$. Then we stop with the first element in V whose value surpasses the threshold. If there are less than k samples, the algorithm accepts the first online value (we define the k-th largest element from a set of less than k elements as $-\infty$). We show that this algorithm achieves a success guarantee of $\left\lfloor \frac{1}{1-p} \right\rfloor p^{\left\lfloor \frac{1}{1-p} \right\rfloor}(1-p)$, so for instance for $p = 1/2$ the guarantee evaluates to $1/4$. Although the proof of this fact is relatively easy, what is more surprising is that this guarantee is best possible. To prove the latter we analyze a related optimal stopping problem, which we call the *last zero problem*. Suppose an adversary picks a number of identical blank cards n. Then independently with probability p each card is marked and you are informed about the total number of marked cards, but you ignore their position in the deck. Finally, one by one, you get to see the cards and whether they are marked or not. When you stop the sequence, you win if the card was the last blank card, otherwise you lose. By using a related conflict graph over possible sequences, we show that for this problem no ordinal algorithm can guess the last blank card with

[3] Our results, and in particular the upper bounds on the success probability, remain true if the adversary knows all values $\alpha_1, \ldots, \alpha_n$ but not the result of the sampling process, i.e., she does not know the random sets S and V.

probability better than $\left\lfloor \frac{1}{1-p} \right\rfloor p^{\left\lfloor \frac{1}{1-p} \right\rfloor}(1-p)$. Then, we relate this problem to a different one, in which the objective is to guess the last number of an increasing sequence of unknown length. Finally, we go back to the original AOSp by considering an adversary that picks a growing sequence which at some point in time decreases to a low value, and this time is difficult to guess.

It is worth noting that this simple best possible algorithm does not use knowledge of n and, as opposed to most variants of the secretary problem, for AOSp knowledge of n is irrelevant in worst case terms. Moreover, we discuss the case in which n is known but p is unknown. Here it is quite natural that the algorithm works again by simply estimating p using the size of the sample set. However, if neither n nor p are known, then no nontrivial success guarantee can be obtained.

For ROSp we obtain a randomized algorithm with best possible success guarantee that works as follows. First, we assign to each of the n elements a uniformly random arrival time in the interval $[0, 1]$, which implies that the elements arrive in uniform random order. All elements whose arrival time is less than p are placed in the sample set S. Then we find a sequence of time thresholds $0 < t_1 < t_2 < \cdots < 1$, dictating that if an element's arrival time is between t_i and t_{i+1}, we stop if its value is the maximum among elements arriving after p and it is among the i largest values of all elements seen so far. To obtain the success guarantee of this algorithm we first prove that for a fixed sequence $0 < t_1 < t_2 < \cdots < 1$, the success guarantee of the algorithm decreases with n. Then we write the optimization problem over the time thresholds, and interestingly, this turns out to be a separable concave optimization problem with a very simple solution. Moreover, the solution is universal in the sense that it does not depend on p. The resulting guarantee is thus easily computed and grows from $1/e$ when $p = 0$ to $\gamma \approx 0.58$ as $p \to 1$.[4] We also prove that this is a best possible algorithm. To this end we first argue that ordinal algorithms in our model are essentially equivalent to a ranking function that determines what *global* ranking an element, which is a *local* maximum, should have in order to accept it. Here, by global ranking we mean the ranking an element has among all samples and values revealed so far, and local ranking refers only to the values revealed and not to the samples. Finally, as n grows, this ranking function converges to a sequence of time thresholds as we defined them.

Fig. 4.1 illustrates the success guarantee for our problems. For AOSp it can be observed that the success guarantee can be bounded below by the function $p^{1/(1-p)}$ and bounded above by $\frac{p-1}{\log p} \cdot p^{-1/\log p}$ (see also Appendix B.2 for some details).

4.1.3 Further related literature

An interesting connection arises between our model and results when p is close to 1, and the so-called full information case. First, recall that Gilbert and Mosteller [80] obtained the optimal algorithm with worst case performance γ (see also Samuels [137, 138]), in

[4] We should note that after completion of this work we became aware of the work of [33] who obtain very similar results. Indeed they consider the dependent sampling version described earlier and obtain that the optimal success guarantee converges to γ as the fraction sampled grows to 1. Their methods however are very different from ours and are significantly more complicated.

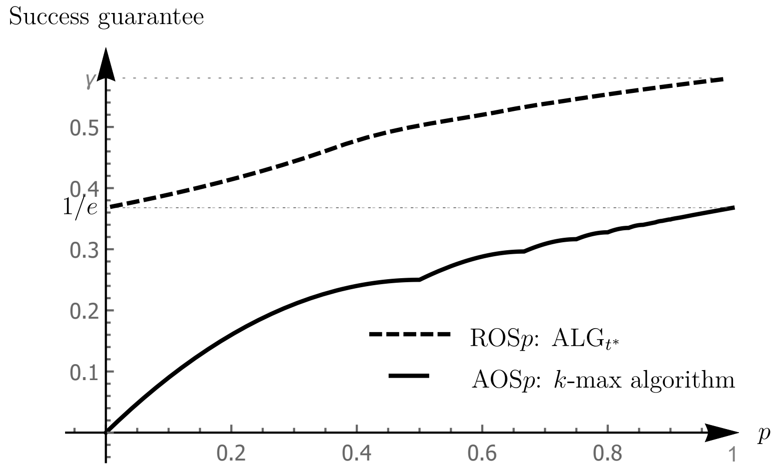


Figure 4.1: The best possible success guarantee for ROSp and AOSp as a function of p.

the secretary problem where the elements' values are taken as i.i.d. random variables from a known distribution. It may thus seem natural that our guarantee matches this quantity as $p \to 1$. However, this is far from obvious. Indeed, for the prophet inequality with i.i.d. values from an unknown distribution (a model that arguably gives more information than ours) Correa et al. [53] proved that with $O(n^2)$ samples, one can achieve the best possible performance guarantee of the case with known distribution, and only very recently Rubinstein et al. [135] improved this to $O(n)$ samples. This is in line with our result here since for p close to, but strictly less than 1, the size of the sample set is linear in the size of V.

A more intriguing connection to the full information case pops up in the adversarial order case. In this context, Allaart and Islas [2], and independently Esfandiari et al. [71], considered the adversarial order secretary problem in which an adversary chooses n distributions $F_1, \dots, F_n$. Then, independent values are drawn from these distributions and sequentially uncovered. A decision-maker who knows $F_1, \dots, F_n$ needs to stop at the maximum realization. They prove that the optimal stopping rule is a simple single threshold algorithm and the best possible success guarantee equals $1/e$. Although this problem has a similar flavor as our AOSp, and the optimal guarantee is the same, we are unaware of a precise connection.

On the other hand, our last zero problem, used as a tool for AOSp, is related to an old optimal stopping problem first studied by Bruss [28]. We face a sequence of n independent Bernoulli random variables where we know n and the distributions, and we want to stop with the last zero. Bruss obtains the optimal stopping rule for this problem, which also turns out to be a simple threshold rule. Our last zero problem is simpler in that the Bernoulli random variables are homogeneous. However, rather than knowing n, we only know the total number of ones. This subtle difference makes the problem substantially different.

Another very recent line of work studies robust or semi-random versions of the classical secretary problem [25, 101]. The main idea is that the problem input should be a mix of stochastic and adversarial parts. More specifically, in their (similar) models, some of the elements arrive at adversarially chosen times, and the rest at times uniformly randomly drawn from $[0, 1]$. Their objective functions (and in some cases also the benchmarks) are quite different from ours. Kesselheim and Molinaro [101] consider the knapsack secretary problem in this mixed model, while Bradac et al. [25] design algorithms for selecting k items or maximizing the expectation under various matroid or knapsack constraints. It would be interesting to incorporate their ideas in our setting and study a problem that interpolates between ROSp and AOSp.

Finally, we mention that our results may help explain some behavioral issues raised by Goldstein et al. [82]. They set up an experiment in which people play repeated secretary problems. The values come from any of three possible distributions, unknown to the players (the distribution is fixed for all games played by a person). They analyze a total of 48,336 games played by 6,537 players. Among other issues, Goldstein et al. study how close to optimal people play. However, they find difficulty in establishing what optimal means in their context since for the first game played optimal means simply the secretary algorithm while after playing many games, optimal should mean something close to the dynamic program of Gilbert and Mosteller [80]. They thus consider several candidate models for the players' behavior and conclude that the closest to actual play is a multi-threshold algorithm that is very much in the spirit of that of Gilbert and Mosteller. Interestingly, they find that by the fifth game, players have essentially learned the optimal thresholds [82, Figure 9]. However, they also find an apparent dichotomy between the strategy players use in the first few games and that used later on. Indeed they state that: *"One possible explanation for the apparent change in strategy is that players spent the first few games primarily collecting information about the distribution and then switched to trying to actually win the game only in later games: that is, they spent the first few games exploring and then switched to exploiting only later."* We believe that our model and results for ROSp can provide a different explanation, simply that players are playing close to optimal all along, using the information they gain while also optimizing the success probability. For this, we note that the first game the players face is just the normal secretary problem, or ROS0, the second closely corresponds to ROS$\frac{1}{2}$, the third to ROS$\frac{2}{3}$, and so on. And the induced thresholds of the fifth game, which would correspond to ROS$\frac{4}{5}$ are indeed very close to those of Gilbert and Mosteller [80].

Organization of the chapter.

Section 4.2 presents the techniques and results for the adversarial order case, while Section 4.3 does the same for the random order case. Then, Section 4.4 presents the results that can be obtained if we assume different knowledge of the parameters. We briefly discuss two straightforward applications and extensions of our results in Section 4.5. We conclude in Section 4.6 with some additional insights and open questions.

4.2 Adversarial order

In this section, we study the adversarial order secretary problem with p-sampling (AOSp). We present the k-max algorithm and prove that it is optimal (in the worst-case sense) for this setting.

Recall that we defined the *k-max algorithm* as follows: the k-th largest value of the sampled elements is set as a threshold, and the algorithm accepts the first element in the set V of online values whose value surpasses this threshold. If there are less than k sampled elements, then the algorithm accepts the first online element[5]. From now on, we take the k-max algorithm with $k = \left\lfloor \frac{1}{1-p} \right\rfloor$. This section is dedicated to proving the following theorem.

Theorem 7. *Let $k = \left\lfloor \frac{1}{1-p} \right\rfloor$. Then the k-max algorithm achieves a guarantee of $kp^k(1-p)$ for AOSp. Furthermore, no algorithm can achieve a better success guarantee.*

When p tends to 0, the guarantee naturally tends to zero: If there are very few samples, the problem becomes the secretary problem with adversarial order, where basically nothing can be done. What is more surprising is that when p is close to 1, the success guarantee approaches $1/e$ (see Fig. 4.1), which is the performance obtained for the secretary problem when one knows the distribution of the values of the elements [2, 71].

The proof of the guarantee of the algorithm is easy and appears in Section 4.2.1. The proof of its optimality is more advanced and requires new tools. We first introduce the concepts that we are going to use and then give the proof in the remainder of this section. A surprising fact of this proof is the following: when proving the negative result, it is enough to focus on the special case where the values of the elements are increasing (thus where the player aims to get the last element), with the twist that the player does not know the total size n of the instance.

4.2.1 The success guarantee of the k-max algorithm

Being a simple threshold algorithm, the main question to answer is what value of k is appropriate. Intuitively, the bigger the value of p, the higher the probability that the largest valued elements are sampled. Therefore, we should lower the threshold as p grows. As is the case for many threshold algorithms, there is a tradeoff between (1) setting the threshold too low and risking acceptance of an element that does not have the maximum online value, and (2) setting it too high and risking finishing the game without selecting any element. The following lemma is the first part of Theorem 7 and establishes the performance of the algorithm for the value $k = \left\lfloor \frac{1}{1-p} \right\rfloor$.[6]

Lemma 7. *For a given p, the k-max algorithm chooses the element of the online set with maximum value with probability $\left\lfloor \frac{1}{1-p} \right\rfloor p^{\left\lfloor \frac{1}{1-p} \right\rfloor}(1-p)$.*

[5] Recall that we define the k-th largest element from a set of less than k elements as $-\infty$.

[6] Observe that this lemma still holds in the setting where the order of the online elements is determined by the adversary after sampling, since our algorithm is order oblivious.

Proof. Note that the k-max algorithm wins in an instance if *exactly one* of the k largest values of the adversarial input ends up in the online set and the $(k+1)$-th largest ends up in the sample set. For the purpose of analysis, assume the values are sequenced in non-decreasing order. Thus, an instance in which the algorithm is successful is exactly a sequence ending in k sampled elements plus one online element that is somewhere in the last k entries of the sequence. The probability that this happens is $kp^k(1-p)$ because of the independent sampling. The lemma follows by substituting the value $k = \left\lfloor \frac{1}{1-p} \right\rfloor$. □

4.2.2 The negative result

We now focus on the proof for the negative result of Theorem 7, which consists of several steps. We start by considering the special case where *the algorithm does not know* n. Let us make precise what we mean by this. Consider an algorithm and two instances I_1 and I_2 of different sizes n_1 and n_2 respectively, but with the same value of p. Suppose that the algorithm happens to face the exact same set of samples and non-sampled elements in both instances, and is currently facing an online element of the same value in both instances. Thus, up to this point, A has access to exactly the same information (and possible beliefs over the size of the instance). Therefore, the algorithm needs to make the exact same (possibly randomized) decision in both situations, independent of n_1 or n_2.

For our main steps, we start by showing that we can focus on a simpler problem that we call the *last zero problem*. For this problem, we prove the negative result with some additional assumptions. We then remove the assumptions one by one, each time generalizing the proof one step further, until we retrieve the proof of Theorem 7 for the case where n is unknown. Finally, in the second phase, we show that allowing the algorithm to know n basically does not help (in worst case terms).

We start by defining the last zero problem and showing how negative results in this setting imply negative results in AOSp as well. Let the *norm* of a sequence of bits be the number of 1s it contains. The total number of bits in such a sequence s is called its length or size. The numbering of the entries of a sequence s is counted starting from 1.

Definition 4. The *last zero problem* with probability p is the following:

1. An adversary picks a size n.
2. A sequence of bits of length n is generated, where in each position independently the bit equals 1 with probability p and 0 otherwise.
3. The player is given the norm of the sequence.
4. The player observes the bits one after the other, and for each of them decides whether to stop the sequence there or to continue.
5. The player wins if she stops on the last 0 of the sequence.

Note that the fact that the player does not know the size n is crucial, as otherwise the game is trivial. Thus, it does not make sense to analyze the algorithm for a given size; we need to prove that no algorithm can perform well on all sizes.

The following proposition highlights the connection between the last zero problem and AOSp. With the increasing case of AOSp we mean the special case of the problem AOSp where the elements are presented to the algorithm in non-decreasing order of their values.

Proposition 4. *The last zero problem and the increasing case of AOSp are equivalent. Therefore, any negative result for the last zero problem also holds for AOSp.*

Proof of Proposition 4. We show that an algorithm for picking the element with maximum value in the increasing case of AOSp has the same success probability in the last zero problem, and the other way round.

($\Rightarrow$) Assume that we know that in AOSp the adversary is going to present the online set in increasing order. Therefore we need to fix an ordinal algorithm with the goal of picking the last element in the increasing sequence. Every time an element in V is revealed, the algorithm knows how many online elements it saw in total and how many sampled elements have larger or smaller values compared to the value of this online element. Moreover, the value of p creates some possible beliefs over the size of the instance. This knowledge guides the (possibly randomized) decision of the algorithm on whether to stop with the element just observed.

In the last zero problem, each revealed 0 of the binary sequence corresponds to an online element. Furthermore, since we are given the total number of 1s beforehand, we know how many 1s are before and after each revealed 0 in the sequence. This information corresponds to the relative ranking of an elements value in V among the values of sampled elements. Finally, p equals the probability that a 1 was written on a card, independently of the others.

An algorithm for AOSp takes as input the relative ranking of the values $r_1 > r_2 > \cdots > r_t$ in S and V seen so far at each time step t and outputs a stopping rule τ which gives a certain success probability. In particular, since the algorithm is ordinal, it does not even need to see the actual values of the sampled elements; all it needs to know is the ranking of a revealed element among the sampled ones. If we apply the same algorithm to the last zero problem (with the input now being the total number of 0s and 1s seen so far and the total number of 1s), we get the same success probability of picking the last 0.

($\Leftarrow$) Consider an algorithm for maximizing the probability of picking the last zero in the last zero problem. At each time step t, an algorithm ALG'_τ takes as input the given probability p, the total number k of 1s (also given) and how many 0s and 1s have been seen so far. Consider a stopping rule τ' that decides whether to stop at each revealed 0, and that attains a certain success probability.

In the increasing case of AOSp each element in S corresponds to a 1 and each online element to a 0. The total number of 1s represents the cardinality of the set S. Each time a 0 is observed (and we know its rank among the 1s), it translates to learning how many samples have smaller and how many have larger value than the online element just observed. Remember that since the sequence in AOSp is increasing, we win if we stop with the last online element. We can now conclude that an algorithm for the last zero problem with a certain success probability can be used as an ordinal algorithm to solve the increasing case of AOSp with the same success probability.

Since the increasing case is a specific instance for AOSp, a negative result for the last zero problem implies a hardness result for AOSp. □

For the remainder of this section, we consider the last zero problem. An instance in the last zero problem can be described by a finite string of bits. We introduce the shorthand notation 0^ℓ and 1^ℓ for the string of length ℓ consisting of only zeros and ones respectively.

Next, we obtain an upper bound for the special case of deterministic algorithms for $p = 1/2$. For this case of the last zero problem we introduce the *no-zero rule*, which specifies that if there are no online elements (i.e., all n elements are sampled), the player loses. This will be a useful rule for the sake of the proofs. As we will see, this decision actually becomes irrelevant for the generalization of the proof. Therefore, it poses no problem that this contradicts the assumption made for AOSp where we win in such an instance, as stated in the introduction of the problem.

We will show that the following proposition holds under the no-zero rule and starting from $n = 1$.

Proposition 5. *For the last zero problem with* $p = 1/2$*, no deterministic algorithm can achieve a better success guarantee than* k*-max (with the no-zero rule) for AOSp.*

The goal of the discussion and the proof sketch presented here is to introduce the tools we will use informally. The claim can be generalized to consider instances of size larger than some chosen N_0 (cf. Proposition 6).

For $p = 1/2$, the k-max algorithm achieves a guarantee of $1/4$. Suppose that there is an algorithm that achieves a guarantee strictly better than $1/4$. As a start, consider the decision of the algorithm when the adversary chooses $n = 1$. Then, there are two instances (after sampling), which both occur with probability $1/2$. The first possibility is that the instance is 0. Then the player knows that there is no 1 in the instance, and is first presented a 0. The second possibility is that the instance is 1. Then the player knows there is a 1 in the instance, and is announced from the start that the game is finished.

In the second case, the player loses because of the no-zero rule. Thus, to achieve at least $1/4$ for every n, the player needs to win in the first case (recall that we restrict ourselves to deterministic algorithms for now). This means that when the player is presented with not a single 1, and sees a first 0, she stops.

Here comes the key observation. Suppose that the adversary chose $n = 2$ and the sampling resulted in the instance 00. Now again the player is presented with not a single 1, and again sees a first 0. From the above, we already deduced that she needs to stop at this first 0. Indeed, from the point of view of the player, this is exactly the same situation as in the case where the instance was 0, because the player does not know n. In other words, these two situations are *indistinguishable* from the perspective of the player, and she has to make the same decision. In the case of 00, this decision is wrong

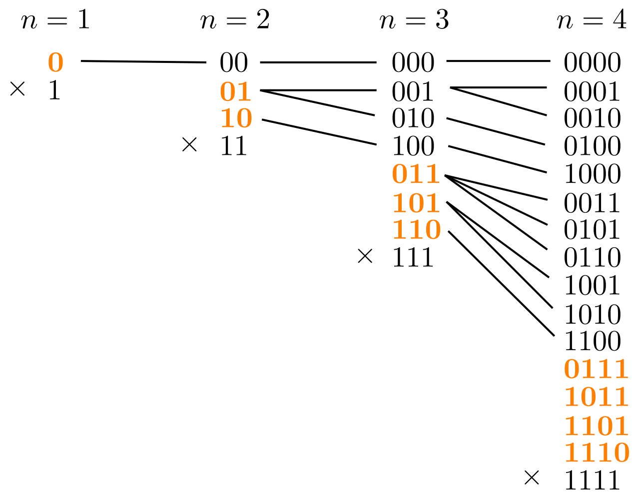


Figure 4.2: An illustration of the first four layers of the conflict graph. We use orange for the instances in which an optimal algorithm stops with the last zero.

as the last 0 is the second 0, hence the player loses. We call such a situation a *conflict* between the instances 0 and 00.

Note that conflict works in both directions. If the player had a strategy that would make her win in 00, then after the first 0, she would wait, which would make her lose in the instance 0.

Let us give yet another example of conflict, for the instances 01 and 001. On instance 001 the player receives a first 0, and knows that there is one 1. This is exactly the same information as in the instance 01 when it starts. If she stops on this element then she wins in 01 but loses in 001. On the other hand, if she waits and then stops on the next 0, she loses in 01 but wins in 001. Moreover, if she continues to wait she loses in both instances.

More generally, for every pair of instances there is a fairly simple criterion in each of the two directions to see if they are in conflict or not (cf. Lemma 10). In particular, it is enough to decide the conflict between instances whose sizes differ only by 1. Indeed, two instances s and s' of sizes n and $n+q$ respectively are in conflict if and only if there is a series of conflicts (s, s_1), (s_1, s_2), ..., (s_{q-1}, s'), where s_i has size $n+i$ (cf. Lemma 9). Then we can define the (infinite) *conflict graph* whose nodes are all possible instances and the edges represent the conflict between nodes of adjacent sizes. The conflict graph for size $n = 1$ to $n = 4$ is represented in Fig. 4.2. On this graph, we can represent an algorithm as a choice of instances in which it wins. Such *selected instances* cannot be in conflict. In other words, they cannot be linked by a monotone path, where monotone means that the path goes from left to right without changing direction.

For $n = 1$, we denote by a cross the fact that the player will never win in the instance which consists of one 1, because of the no-zero rule. We write 0 in orange to denote that the player wins in this instance, as she decides to select the last (and only) 0.

Let us now consider more systematically all the instances of size 2. They all have a probability of occurring of 1/4. We already know that both 00 and 11 cannot be selected (because of the conflict to the left and the no-zero rule, respectively). Thus, to achieve strictly more than 1/4, the player needs to win in both 01 and 10. Consequently, these instances need to be selected in the conflict graph.

Now, for $n = 3$, the player loses in 000, 001, 010 and 100 because of conflicts, and on 111 because of the no-zero rule. Therefore, she *must* win in 011, 101, and 110, since each instance has probability 1/8. Finally, for size 4, we can use the same kind of argument as before, to show that the player loses in all instances except 0111, 1011, 1101 and 1110. But these are only four cases out of sixteen and thus, the player cannot strictly beat the 1/4 bound if the adversary chose $n = 4$. And this is a contradiction. Therefore, Proposition 5 is true.

Of course, there are several limitations to this first proof of the claim:

1. The no-zero rule is arbitrary and it should be removed.
2. The fact that the proof is only considering small sizes is a weakness, in the sense that it does not take into account algorithms which could possibly have a better success guarantee than the k-max algorithm, if it would consider only instances of size at least some N_0.
3. The sampling probability is fixed to 1/2 instead of taking any value in $(0, 1)$.
4. The bound only applies to deterministic algorithms.

We continue by addressing the two first problems. We design a proof that also works by starting from an arbitrary N_0 and not necessarily from 1. This also solves the first problem, as it makes the probability of the case with no zeros negligible for large enough N_0 (for size n, this case is just one out of 2^n). But first, we formally define and state the properties of the conflict graph.

Preliminaries: Conflict graph

We now formalize the intuition that we built about the conflict graph. We first describe its generic structure, independent of the sampling probability p, without weights on the instances or any reference to success guarantees. Then we continue by describing how to measure the performance of a deterministic algorithm in this framework using probabilistic weights on the instances.

Conflict graph structure We first define what it means that two instances are in conflict. For an instance I we denote by $I[a, b]$ the instance I restricted to the positions a to b (both included). Consider two instances I_1 and I_2 of size n_1 and n_2 respectively with $n_1 < n_2$, both containing at least one 0. Let r be the position of the last 0 in I_1. The instances I_1 and I_2 are *in conflict* if they have the same norm and $I_1[1, r] = I_2[1, r]$. The following lemma outlines why we care about this notion.

Lemma 8. *No deterministic algorithm can win in two conflicting instances.*

Proof. Consider a deterministic algorithm that wins in I_1, meaning, this algorithm stops at position r. Note that at any position $j \leq r$, the knowledge of the algorithm up to that point consists of the norm of the instance and $I[1, j]$. Now run the same algorithm on I_2. Since the algorithm is deterministic and has the same information available at every point in time, it must make the exact same decision at every $j \leq r$. In particular, it stops at position r. However, since I_2 has the same norm as I_1 but a larger size, there must be a zero after position r in I_2, and the algorithm loses in I_2. □

We now define the conflict graph, which is the formal object described by Fig. 4.2.

Definition 5. The *conflict graph* is an infinite graph in which the nodes correspond to all finite strings of bits. There is an edge between nodes s_1 and s_2 if and only if the corresponding instances of the last zero problem are in conflict and the size of s_2 is one bit larger than the size of s_1.

As every node corresponds to a unique instance and vice versa, we will use these terms interchangeably.

When we draw the conflict graph, we order the nodes by increasing size as in Fig. 4.2. We define a *monotone path* as a (possibly infinite) path in the conflict graph where the nodes correspond to consecutive increasing sizes. For example, in Fig. 4.2, (01, 001, 0010) is a monotone path.

Lemma 9. *Two instances are in conflict if and only if they are linked by a monotone path in the conflict graph.*

Proof. Let I_1 and I_2, be two instances of size n_1 and n_2 respectively that are in conflict, with $n_1 < n_2$. By definition, they have the same norm, and have the same substring up to the last zero of I_1. Consider the following instance I_3: take I_2, and remove the last zero.

This instance (if it is not I_1) satisfies the two conditions above, thus is in conflict with I_1. It is also in conflict with I_2: they share the same prefix up to the last zero of I_3 and have the same norm. By repeating this operation (removing the last zero) until we get I_1, we get a series of instances (including I_1 and I_2), that are in conflict with one another, and can be ordered in increasing consecutive sizes. These instances form a monotone path in the conflict graph. The other direction of the proof follows similarly. □

This lemma and its proof have several consequences for the structure of the conflict graph. The following lemma is immediate.

Lemma 10. *Given an instance I of size n, the instances of size $n+1$ in conflict with I are the nodes that can be obtained by inserting a new zero anywhere after the last 0 of I. In the other direction, I is in conflict with only one instance of size $n-1$: the one where the last zero has been removed.*

Proof. By Lemma 10, an instance of size $n+1$ is in conflict with an instance of size n if we add a 0 anywhere after the last 0 of the instance of size n. Therefore, every instance that ends in a 0 has degree one, since the new 0 can only be inserted in one place. This is true for half of the nodes. Similarly, we see that every instance that ends in 01 has degree two and this is a quarter of the nodes. In general, every instance that has suffix 01^i has degree $i+1$.

A node with degree k has a suffix 01^{k-1}. To create a child, we need to add a zero anywhere after the last 0. If we insert this 0 at the very end, we create a node of degree one. If we insert this 0 before the last 1, we create a node of degree two. In general, if we insert the 0 after the i-th 1 from the end, we create a node of degree i. $\square$

This lemma implies that a node has only one edge on its left. We refer to this node as its *parent*. We define the *degree* of a node in the conflict graph as the number of neighbors it has on its right, which we refer to as its *children*. Furthermore, for a given size n each node corresponds to a different instance of zeros and ones, so we have 2^n nodes in total. The degrees adhere to the following structure.

Lemma 11. *Consider all 2^n nodes corresponding to instances of size n. For every $i \in \{1, \dots, n-1\}$, there are 2^{n-i} nodes of degree i. Concretely, half of these nodes have degree one, a quarter of the nodes have degree two, and so on until one node has degree n.*

Moreover, a node with degree k has exactly one child of degree i for every $i \in \{1, \dots, k\}$.

Proof. By Lemma 10, an instance of size $n+1$ is in conflict with an instance of size n if we add a 0 anywhere after the last 0 of the instance of size n. Therefore, every instance that ends in a 0 has degree one, since the new 0 can only be inserted in one place. This is true for half of the nodes. Similarly, we see that every instance that ends in 01 has degree two and this is a quarter of the nodes. In general, every instance that has suffix 01^i has degree $i+1$.

A node with degree k has a suffix 01^{k-1}. To create a child, we need to add a zero anywhere after the last 0. If we insert this 0 at the very end, we create a node of degree one. If we insert this 0 before the last 1, we create a node of degree two. In general, if we insert the 0 after the i-th 1 from the end, we create a node of degree i. $\square$

Algorithms and weights in the conflict graph We now turn to the connection between algorithms and the conflict graph. We start by linking the structure of the conflict graph to deterministic algorithms.

Lemma 12. *A deterministic algorithm can win in at most one of the instances of any monotone path in the conflict graph.*

Proof. By Lemma 9, any two instances that are in a monotone path are in conflict, and by Lemma 8 an algorithm can win in at most one instance of a pair of conflicting instances. $\square$

One can think of an algorithm for the problem as a partition of the nodes of the conflict graph into the nodes for which it wins and the nodes for which it loses. Lemma 12 gives a constraint on the structure of such a partition. Note that not all partitions correspond to a finite algorithm, but this is not an issue as we look for impossibility results (we will abuse terminology and use the word “algorithm” nevertheless).

More precisely, we will consider such a partition in the following structured way. We start from some size N_0, and ask the algorithm which nodes of this size it *selects*, that is, in which instances it wins. This implies that the algorithm will not be able to select some instances of larger size, namely the instances in conflict with any node of this selection. We say that these nodes that cannot be selected later are *removed*. Then we will move on to the next size, and ask the algorithm to select instances among those that have not been removed yet. We continue this in an iterative fashion.

We now continue by adapting the conflict graph to reflect the quality of an algorithm. For this, we extend the conflict graph to weighted nodes. We first define this properly and show how to measure the quality of an algorithm in the conflict graph. For now, we restrict ourselves to deterministic algorithms, which select a specific node either always or never. Afterwards we show that the arguments extend to randomized algorithms as well, that are allowed to select nodes with some probability.

We define the *weight* of a node as the probability that the corresponding instance results from the sampling process where we sample each of the n elements independently with probability p. In particular, if an instance has size n and norm m, then the weight of the corresponding node in the conflict graph is $p^m(1-p)^{n-m}$. Note that for a fixed size n, the weights of the instances of size n sum to 1.

With this definition of the weights, the *performance* of a deterministic algorithm for a fixed size n in terms of the weighted conflict graph is the sum of the weights of the instances in which it wins. Then, the *worst case performance* of an algorithm is the infimum of the performance of the algorithm over all sizes n. Note that the worst case performance of an algorithm for the last zero problem implies a bound on the success guarantee of any algorithm for AOSp, which is exactly the negative result we aim to prove in this section.

Fix a size n and let $V_{n,i}$ be the nodes of size n with degree i. Define w_i as the total weight of the nodes in $V_{n,i}$ (as the notation suggests, we will see this is indeed independent of n). Moreover, define w_{ij} as the sum of the weights of the nodes of size $n+1$ and degree j that are in conflict with any node in $V_{n,i}$. Note that w_{ij} is only positive for $j \leq i$ because of Lemma 11. The following lemma can be seen as the weighted version of this lemma.

Lemma 13. *For any size n, $w_i = p^{i-1}(1-p)$ and $w_{ij} = (1-p)w_i = p^{i-1}(1-p)^2$ for all $1 \leq j \leq i \leq n$.*

Proof. From the proof of Lemma 11, we see that the instances of degree one are exactly these which have a 0 in the end. Summing over their individual weights will give us $w_1 = 1-p$, which is the probability of having a 0 as the last bit of an instance. In general, a node of degree i ends in a 0 followed by $i-1$ ones. Accordingly, the probability of having an instance that ends with this suffix is $w_i = p^{i-1}(1-p)$.

Now consider an instance I_1 of size n with degree i. It starts with $n-i$ unrestricted bits and its suffix is 01^{i-1}. Now consider an instance I_2 of size $n+1$ with degree j that is in conflict with I_1. Because it is in conflict with I_1, the first $n-i$ unrestricted bits are the same as I_1, as well as the 0 in entry $n - i + 1$. Then, we interrupt the suffix of 1^{i-1} with an additional 0 such that I_2 has a suffix of 1^{j-1} in order to have degree j. Therefore, I_2 has the following structure. It starts with $n - i$ unrestricted bits, followed by $01^{i-j}01^{j-1}$.

Now consider the set of all instances that have the form of instance I_1, with certain bits in its unrestricted prefix of length $n - i$. Because of the suffix 01^{i-1}, the weight of these instances can be computed as $p^{i-1}(1-p)$. On the other hand, the weight of all instances that have the form of the instance I_2 can be computed as $(1 - p)p^{i-j}(1 - p)p^{j-1} = p^{i-1}(1 - p)^2$. □

This allow us to describe how the conflict graph can reflect a randomized algorithm. The difference is that it labels each instance with a *selection probability* q, while a deterministic algorithm labels each instance either with a one or a zero (we either always select it, or we never do). Concretely, this means the following: Suppose the algorithm is faced with the last 0 in this instance (it is not aware of this, of course). Then the algorithm stops with probability q (and wins in this instance). It continues the sequence with probability $1-q$, meaning it loses in this instance (but it might still win in instances of larger size that are in conflict with this instance).

The following lemma is the non-binary version of Lemma 8. For its statement, we define the *descendants* of an instance I as one would expect: The set of nodes in the conflict graph that are connected to I through a monotone path and that have a larger size than I.

Lemma 14. *For any algorithm, if for some instance I it picks a selection probability q, then the probability of winning at any descendant is at most $1 - q$.*

Proof. This follows from similar arguments as Lemma 8. □

It is important to note that this *removed fraction* adds up: if an instance has selection probability q, and one of its descendants has selection probability r, then for any descendant of the second instance its removed fraction is $q + r$ and its selection probability is at most $1 - q - r$. In other words, when a randomized algorithm reaches this particular descendant, it can assign at most a selection probability of $1 - q - r$ to it.

Similarly, we define the performance of a randomized algorithm as its quality for a given size n, i.e., the product of the weight of a node multiplied by its selection probability, summed over all instances of size n. The worst case performance is then the infimum over n of these performances. The worst case performance of an algorithm for the last zero problem provides a bound on the success guarantee of any algorithm for AOSp.

Finally, note that the success guarantee of the k-max algorithm, proved in Lemma 7, can be also shown now using the alternative perspective of the conflict graph. The k-max algorithm roughly selects low degree nodes in every size n of the conflict graph in order to remove as little weight as possible from instances of larger size. A careful analysis indeed gives the same success guarantee $kp^k(1 - p)$. For details, see Appendix B.1.

Warm up: Proof for the case of $p = 1/2$

As a warm up that introduces the main ideas behind the general proof, this section proves the special case of Theorem 7 for deterministic algorithms for the case where n is unknown but larger than some constant, and $p = 1/2$. Note that for $p = 1/2$, all nodes of size n have the same weight, namely $1/2^n$. The total fraction of selected nodes is therefore equal to the total weight of the selected nodes.

Proposition 6. *For the last zero problem with $p = 1/2$, no deterministic algorithm can have a better success guarantee than the k-max algorithm, even if we consider only instances of size larger than N_0, for any N_0.*

To prove Proposition 6 we will bound the worst case performance of any deterministic algorithm by considering a special class of algorithms.

Canonical algorithms. More precisely, we consider a deterministic algorithm that starts by selecting some nodes in the conflict graph for a certain size N_0. Consequently, all descendants of the selected nodes will be removed. The algorithm will then continue to the nodes of size $N_0 + 1$ and select a subset of the nodes of this size that have not been removed. Then it will continue to the next size and iterate this procedure. We will show that if the algorithm consistently selects at least a $1/4 + \varepsilon$ fraction of the nodes for each size, this process cannot run forever, reaching a contradiction.

Before we proceed to the proof, we make a crucial observation. Note that Lemma 11 implies that two nodes of the same degree have children with the same degree distribution, and the same holds for their further descendants. By construction, it follows that the subtrees to the right of any two nodes of the same degree are *isomorphic*. With this important observation at hand, we can prove that it suffices to restrict our attention to algorithms of a canonical form, in order to reduce the large variety of possible algorithms.

Lemma 15. *Consider the last zero problem for $p = 1/2$. Let I_1 and I_2 be two instances of the same size* and *the same degree that have not been removed, and consider an algorithm that selects I_1 but does not select I_2. Then there exists another algorithm that selects I_2 instead of I_1 and achieves the exact same success guarantee.*

Proof. Consider the instances I_1 and I_2 and an algorithm A that selects I_1 but not I_2. Since A selects I_1, the nodes to its right are removed. On the other hand, as I_2 is not selected and is alive, it can be that A selects some node in its subtree. As observed above, the subtrees rooted at I_1 and I_2 are isomorphic. Since any node has at most one edge to the left, these trees are also disjoint.

Now consider the algorithm B that selects the same nodes as A except for the following. It selects I_2 instead of I_1, deselects every node that A selected in the subtree of I_2 and instead selects the corresponding (according to the isomorphism) nodes in the subtree of I_1. By construction, for every given size n, the nodes that both algorithms select carry the same weight, so the success guarantees are equal. □

We can reduce the algorithms of interest even further by introducing the following important notion. We say that an algorithm follows a *small degrees first strategy* if for any size considered, among the nodes that are not yet removed, it selects the nodes with the smallest degrees.

Note that this strategy does not define a single algorithm: many nodes have the same degree. Indeed, the k-max algorithm is closely related to these small degrees first strategies – we will elaborate on this in the paragraph of Lemma 19.

Lemma 16 (Small degrees first strategy)**.** *Consider the last zero problem for $p = 1/2$. For every algorithm, there exists an algorithm using the small degrees first strategy that achieves the same performance for every n.*

Proof of Lemma 16. Consider an algorithm that does not follow the small degrees first strategy. Then there exists a size n where it selects an instance I_1 of degree k_1 and does not select an instance I_2 of degree $k_2 < k_1$.

Consider the part of the subtree rooted at I_1 that consists of its k_2 children of smallest degree and their subtrees. By the structure given by Lemma 11, this subtree is isomorphic to the subtree of I_2. Then the same swapping argument as in the proof of Lemma 15 between the subtree of I_1 and the tree of I_2 exhibits another algorithm with the same success guarantee that does follow the small degrees first strategy. □

From now on, we restrict ourselves to considering algorithms that follow the small degrees first strategy.

The cover ratio. In order to reach a contradiction and prove Proposition 6, we define the *cover ratio* ρ for an algorithm and a certain size n. It is defined as the sum of the weights of the instances of size n that the algorithm either selects or removes. The removal of an instance is due to selecting an instance of smaller size that is connected by a monotone path to this instance. Denoting the set of selected and removed instances of size n by S and R respectively, and the weight of an instance I by $w(I)$, we can write $\rho = \sum_{I:|I|=n, I \in S \cup R} w(I)$. Observe that this expression with a sum over only S instead of $S \cup R$ equals the performance of the algorithm for size n.

Note that in the special case that $p = 1/2$, all instances have equal weight and therefore $\rho = (r + s)/2^n$ is just the fraction of the total number of instances of size n that are either selected (s) or removed (r).

The proof sketch of Proposition 5 showed the intuition behind the proof. Here we state the formal arguments. The idea behind the proof is to show that selecting strictly more than 1/4 of the instances for many successive sizes implies that the cover ratio ρ increases in such a way that at some point it is impossible to select that many instances. This shows by contradiction that there is no deterministic algorithm that has a success guarantee of $1/4 + \varepsilon$.

Lemma 16 implies that we can restrict ourselves to a unique strategy for the algorithm as follows. For a size n, select a $1/4+\varepsilon$ fraction of the non-removed instances in increasing order of degrees (with an arbitrary order for the instances of same degree). Then the algorithm repeats this for the non-removed instances in the next size $n + 1$, which we refer to as the next *step*. Without loss of generality, we can assume that we start at size N_0 with no removed nodes.

We now analyze the dynamics of the process, and in particular the dynamics of the cover ratio ρ. First, observe that at size N_0, no nodes have been removed so far. As the algorithm selects a $1/4+\varepsilon$ fraction of the nodes and half of all these nodes have a degree of 1, the algorithm selects only nodes of degree 1. For a certain number of sizes, starting from N_0, the algorithm can select only degree 1 nodes. We call this the *first phase* of the algorithm.

Claim 1. Consider the last zero problem for $p = 1/2$ and an algorithm as described above. After t steps in the first phase of the algorithm, the cover ratio ρ is $(\frac{1}{4}+\varepsilon)\cdot\sum_{i=1}^{t}\frac{1}{2^{i-1}}$.

Proof of Claim 1. We prove the claim by induction. For the base case $n = N_0$ we have $\rho = 1/4+\varepsilon$, which corresponds to the formula of the claim. Now suppose that the lemma holds for some size $n+t-1$, so $\rho = (\frac{1}{4}+\varepsilon)\cdot\sum_{i=1}^{t-1}\frac{1}{2^{i-1}}$. We first determine the fraction of removed nodes in the next size $n+t$. Since each node of degree 1 removes one node of the next size, the number of nodes removed for size $n+t$ is the same. However, as there are twice as many instances in total of size $n+t$, the fraction is half this number, namely $(\frac{1}{4}+\varepsilon)\cdot\sum_{i=1}^{t-1}\frac{1}{2^i}$. The fraction of selected nodes is $1/4+\varepsilon$, thus in total the cover ratio becomes

$$\rho = \left(\frac{1}{4}+\varepsilon\right)\cdot\left(\sum_{i=1}^{t-1}\frac{1}{2^i}+1\right) = \left(\frac{1}{4}+\varepsilon\right)\cdot\left(\sum_{i=1}^{t}\frac{1}{2^{i-1}}\right).$$

□

Note that the term $(\frac{1}{4}+\varepsilon)\cdot\sum_{i=1}^{k}\frac{1}{2^{i-1}}$ goes asymptotically to $\frac{1+\varepsilon}{2}$ as k grows, for some $\varepsilon > 0$. In particular, this means that at some point it exceeds the value of $1/2$, which is the total fraction of nodes with degree 1. This implies, in turn, that the algorithm is forced at some point to start selecting degree 2 nodes in addition to degree 1 nodes. This is the start of a *second phase*, where the algorithm needs to select degree 2 nodes, in order to keep selecting a $1/4+\varepsilon$ fraction of the nodes for each size.

Claim 2. Consider the last zero problem for $p = 1/2$ and an algorithm as described above. In the second phase of the algorithm, the cover ratio ρ grows by at least ε at each step.

Proof of Claim 2. Let us consider a size n where $\rho > 1/2$, say $\rho = 1/2+\delta$ for some $\delta > 0$. Then for size $n+1$ the situation is the following: First, the $1/2$-fraction of nodes of size n remove $1/4$ of the nodes of size $n+1$ (since all these nodes have degree 1). Then, by Lemma 11, the δ fraction of degree 2 nodes remove one instance of degree 1 and one instance of degree 2 in the next size. That is, in size $n+1$, a $(1/4+\delta/2)$-fraction of the degree 1 nodes and a $\delta/2$-fraction of the degree 2 nodes are removed in total.

The algorithm must now select a $(1/4+\varepsilon)$-fraction of the nodes that have not been removed. Following the small degrees first strategy, the algorithm chooses the remaining

$1/4 - \delta/2$ fraction of degree 1 nodes, and a $\delta/2 + \varepsilon$ fraction of the degree 2 nodes. In total, for size $n + 1$ we have $\rho = 1/2 + \delta + \varepsilon$, and the claim follows. □

These two claims imply Proposition 6. Intuitively, if the cover ratio grows by the same additive factor in each step, at some point it will exceed the value 1, which completes the proof by contradiction.

Proof of Proposition 6. In the second phase of the algorithm, ρ increases by ε in each step. Therefore, at some point the cover ratio becomes too large to select only degree 1 and 2 nodes and the algorithm is forced to start selecting degree 3 nodes. Note that in this third phase ρ also grows by at least ε at each step, since selecting a node of degree 3 is even worse than selecting a node of degree 2: It removes the same number of degree 1 and 2 nodes, but in addition it removes degree 3 nodes.

The same holds true for further phases of the algorithm in which it selects nodes of even higher degree. Due to this increase of at least ε in each step, at some point ρ becomes strictly larger than $3/4 - \varepsilon$. Therefore, the algorithm cannot select an $1/4 + \varepsilon$ fraction of the nodes any more. Therefore, no algorithm can achieve a success guarantee of $1/4 + \varepsilon$ for any $\varepsilon > 0$. □

Generalization to any value of p

In this section, we generalize the previous results beyond the case of deterministic algorithms for $p = 1/2$. This case is a bit more complicated, because when $p \neq 1/2$ the instances of the same size do not have the same probability of occurring. For example, for $p = 3/4$, it is better for an algorithm to succeed in the instance 1^k0 than to succeed in the instance 0^{k+1}, as the first has probability $(3/4)^k(1/4)$ to occur and the second probability $(1/4)^{k+1}$. From a technical perspective, this means that in the conflict graph the nodes now have weights.

Building on the intuition of the previous section, but using quite different techniques, we show what is the best possible success guarantee that *any* algorithm can achieve. We then link our k-max algorithm to the conflict graph, such that we finally reach the main takeaway point of the section: The k-max algorithm, although very simple, is optimal for all values of p. We first focus on the family of deterministic algorithms and prove the optimality of k-max there. Then, we show how one can adapt the proof to include also randomized algorithms.

Local operators and average performance The main reason the proof techniques of the previous section need to be adapted is the fact that instances of a given size do not have the same weight anymore, and therefore, the swapping argument used in Lemma 15 and Lemma 16 is no longer true. Thus, we transform a strategy using moves that select and deselect nodes from instances of different sizes; we will call such moves *local operators*. These local operators might decrease the fraction of selected nodes in a specific size while increasing it for another size. To resolve this, we introduce the notion of the *average performance* of an algorithm in the window $[n, n + t]$, which is simply the average of its performance on sizes $s \in [n, n+t]$. We will show that there exists a set of local operators that can be used to improve the average performance.

Informally, the argument is then as follows. The k-max algorithm is very consistent in the sense that it selects the same total weight for every size. This means that its average performance is approximately equal to the infimum of its performance for every size (i.e., its success guarantee). Therefore, if a strategy would outperform the k-max algorithm, it would also exceed the average performance in every window. In this section, we show that the latter is a contradiction.

To prove that certain local operators improve the average performance in the next lemma, we say an algorithm is *valid* if it selects at most one node along each monotone path in the conflict graph.

Lemma 17. *Consider a valid deterministic algorithm with a certain average performance in a window $[n, n+t]$. Applying the following local operators yields a new valid algorithm whose average performance in this window is at least as good as the former algorithm.*

1. *If the algorithm selects a node of degree $d > 1/(1-p)$ for some size $s \in [n, n+t-1]$: Deselect it and select all its children.*
2. *If the algorithm has not selected nor removed a node of degree $d \leq 1/(1-p)$: Select it and remove all its descendants (in particular, deselect its selected descendants).*

Proof. The fact that the resulting algorithm is valid again is clear. We prove that these local operators do not decrease the average performance.

Consider the first local operator and a node of degree d and weight w of size $s \in [n, n+t-1]$. After applying the operator, the performance of the algorithm in size s is decreased by w. By Lemma 13, the total weight of its children is $dw(1-p)$, which is larger than w for $d > 1/(1-p)$.

Now consider the second local operator. Let A_1 be the algorithm before applying the second local operator and A_2 the resulting strategy afterwards. We will construct a reversed sequence of valid algorithms that starts at A_2, iteratively selects and deselects some nodes and ends in A_1, where in every step the average performance does not increase. This will prove the claim.

Consider a valid algorithm A in this reversed sequence (the "current" algorithm) from which we will construct its predecessor algorithm A'. Let v be the node that A_1 neither selects nor removes and consider the subtree T rooted at v for the remainder of this argument. Let S be the set of nodes in T that A_1 does not select, but that the current algorithm A does select. Among the nodes in S, let v' be an arbitrary node of minimum size. There are two cases to consider.

First, if A_1 does not select any of the descendants of v', deselect v' in the newly constructed algorithm A'. This clearly does not improve the average performance from A to its predecessor A'.

Second, consider the other case where A_1 selects at least one of the descendants of v'. Denote the weight of v' by w'. Then, to turn A into A', deselect v' and selects all its descendants. Note that this replaces a node of degree $d' \leq d \leq 1/(1-p)$ and weight w' by a set of at most d' nodes of weight $w'(1-p)$, having total weight $d'w'(1-p) \leq w'$. So the average performance of A' is at most the average performance of A.

By starting at algorithm A_2 and iteratively applying these two cases, we create a sequence of valid algorithms that converge to the initial algorithm A_1. Since the average performance does not increase in this direction, this means that from A_1 to A_2 the average performance does not decrease and the proof is complete. □

Fill-in strategy Using these local operators that improve the average performance, we can define the following: The *fill-in strategy* for a window $[n, n+t]$ scans the sizes in increasing order, selects all the non-removed instances of degree up to $\left\lfloor \frac{1}{1-p} \right\rfloor$ for each size $s \in [n, n+t-1]$, and all the non-removed instances for size $n+t$.

Lemma 18. *The fill-in strategy achieves optimal average performance for any window $[n, n+t]$.*

Proof. Consider an optimal strategy that is not the fill-in strategy. There are three cases. In the first case, a non-removed node of size $s \in [n, n+t-1]$ of degree at most $\left\lfloor \frac{1}{1-p} \right\rfloor$ is not selected. But in this case, applying the second operator of Lemma 17 improves the average performance, which is a contradiction. In the second case, a non-removed node of size $s \in [n, n+t-1]$ of degree strictly larger than $\left\lfloor \frac{1}{1-p} \right\rfloor$ is selected. Now we can apply the first operator of Lemma 17 to improve the average performance, and we have a contradiction. In the last case, a non-removed node of size $n+t$ is not selected. But selecting it will also improve the average performance, which is again a contradiction. □

With the optimal fill-in strategy at hand, we now proceed to describe the k-max algorithm in the conflict graph and finally show that the worst case performance of the fill-in strategy does not exceed the success guarantee of the k-max algorithm to conclude the proof of the negative results of Theorem 7.

The k-max algorithm in the conflict graph To link the fill-in strategy to the k-max algorithm, we need to analyze the dynamics of the k-max algorithm in the conflict graph. As a starting point, we will describe which instances the algorithm selects for $p \in [1/2, 2/3)$. Note that for such a value, $k = \left\lfloor \frac{1}{1-p} \right\rfloor = 2$, so the algorithm sets the second largest sampled value as a threshold (i.e., stops with the first 0 after the second-to-last 1 in the last zero problem). This implies that for any given size n, it obtains the last zero (i.e., the online element with the maximum value) in the instances which end in 110 or 101. Similarly, for $p \in [2/3, 3/4)$, the algorithm successfully selects the last zero in instances that end in 1110, 1101 or 1011.

We analyze its dynamics in the conflict graph in the following lemma.

Lemma 19. *Consider the instances in the conflict graph of size $n \geq k+1$ and consider the k-max algorithm that starts at size $k+1$ and iteratively considers instances of increasing size. For every size, it selects the non-removed nodes that have norm at least k as well as degree at most k.*

Proof. First of all, for this proof we will need the concept of the *m-cut suffix* of an instance, which is the last m bits in case the instance has at least m bits and the entire instance otherwise.

Consider the conflict graph for $n \geq k+1$ with selected and removed nodes by the k-max algorithm and suppose by contradiction that the lemma is false. Then either a node of norm less than k is selected, or a node of degree more than k is selected, or a node that has norm at least k as well as degree at most k is not selected.

In the first case, there are less than k samples, thus the algorithm sets a threshold of zero and accepts the first online value. So the algorithm only wins in this instance if the first online value is the maximum online value, i.e., the instance contains only one 0. But since there are less than k samples, this instance has size at most k. Contradiction.

In the second case, note that a node that has degree more than k has a suffix consisting of one 0 followed by at least k 1s. In such an instance, however, the k-max algorithm loses, so it does not select such a node. Contradiction.

In the third case, consider a node v of norm at least k and degree at most k that is selected. Without loss of generality we assume that v is the node with these properties of smallest size among all nodes with these properties. Let the degree of v be $d \leq k$ such that its suffix is 01^{d-1}. Consider the k-cut suffix of v and note that it contains at least one 0. Now, as long as the k-cut suffix of v contains more than one 0, remove the last 0 of v. Consider the unique resulting instance v' of this procedure whose k-suffix contains exactly one 0. Note that the size of v' is at least $k+1$ as its norm is at least k. Since v' has norm at least k as well as degree at most k, and v was the smallest (in terms of size) such node that was not selected, the k-max algorithm already selected v'. But then v, being a descendant of v', was removed and therefore could not be selected in the first place, contradiction. □

Now that the behavior of the k-max algorithm on the conflict graph is clear, it is possible to analyze its success guarantee using the conflict graph. The possibility to analyze the success guarantee of an algorithm through the conflict graph is one of its key properties. Indeed, such an analysis yields the same success guarantee as the one claimed in Lemma 7 (see also Appendix B.1).

Connecting the fill-in strategy to the k-max algorithm The previous lemma allows us to compare the performance of the fill-in strategy to the performance of the k-max algorithm. In fact, they select almost the same nodes in the conflict graph.

Lemma 20. *Consider the fill-in strategy and the k-max algorithm for window $[n, n+t]$.*

If $n > 1$, then for every size $s \neq n, n+t$, the fill-in strategy and the k-max algorithm select the same set of nodes. For sizes $s = n$ and $s = n+t$, the k-max algorithm selects a strict subset of the set of nodes selected by the fill-in strategy.

If $n = 1$, they select the same set of nodes for size $s = n = 1$ as well.

Proof. Suppose that we start with size $n > 1$. This means that none of the instances of size n have been removed. The fill-in strategy selects all nodes of degree up to

$k = \lfloor 1/(1-p) \rfloor$. The k-max algorithm selects only such nodes that have norm at least k as well, which is a strict subset.

We will now prove that for sizes $n < s < n+t$, the set S_1 of nodes selected by the fill-in strategy is the same as the set S_2 of nodes selected by the k-max algorithm. It is clear that $S_2 \subseteq S_1$. We prove $S_1 \subseteq S_2$ by contradiction, so we assume there is a $v \in S_1 \setminus S_2$, i.e., v has degree at most k and norm less than k. Without loss of generality, we assume that v has the smallest size among nodes in the set $S_1 \setminus S_2$. We consider two cases: v has a parent w of size $s-1$ or v has no parent.

In the first case, note that the degree d of node w is at most k. Otherwise, it would have suffix 01^d for $d \geq k$. But then its norm would be at least k and the norm of its child v would also be at least k, contradiction. So assume that the degree of w is at most k. Then w was selected by the fill-in strategy if it was not removed earlier. If w was selected, v was removed so could not be selected by the fill-in strategy, so $v \notin S_1$, contradiction. If w was not selected, that is because it was removed earlier. But it can only be removed earlier in case it is a descendant of a node that was selected by the fill-in strategy before. But in that case, v was also removed, contradiction.

In the second case, note that nodes without a parent are exactly the nodes that have at most one 0. In the single instance that contains no zeros, the k-max algorithm and the fill-in strategy make the same decision by definition, so we restrict ourselves to instances that contain exactly one 0. Since the norm of v is less than k, the k-max strategy sets a threshold of 0 and wins, since the only 0 is the maximum 0. But then $v \in S_2$, contradiction.

We wrap up the first part of the proof by considering the size $s = n + t$. Here, the fill-in strategy selects all non-removed nodes, while the k-max algorithm selects all non-removed nodes that have degree at most k and norm at least k. The set of removed nodes is the same and the set of non-removed nodes contains nodes of degree more than k or norm less than k, so the fill-in strategy indeed selects more nodes.

Finally, if $n = 1$, both the fill-in strategy and the k-max algorithm select instance 0 and cannot win in instance 1, so in this case they select exactly the same nodes also in the first size of the window. □

Combining everything, we can now prove the negative result for deterministic algorithms.

Theorem 8 (Negative result of Theorem 7 for deterministic algorithms)**.** *No deterministic algorithm can achieve a better success guarantee than the k-max algorithm.*

Proof. First, note that Lemma 20 implies that the performance of the fill-in strategy and the k-max algorithm for the sizes N_0 and N_0+t differs by at most 1 for each size, so the average performance of the fill-in strategy in $[N_0, N_0+t]$ is at most $2/(t+1) \leq 2/t$ more than the average performance of the k-max algorithm. As argued before, for some interval, the average performance of the k-max algorithm is arbitrarily close to $kp^k(1-p)$, since that is its worst case performance. Consider this interval.

To prove the theorem, suppose by contradiction that there exists an algorithm A that achieves a performance of $kp^k(1-p)+\varepsilon$ for some $\varepsilon > 0$ for every size n (larger than some size N_0), where $k = \lfloor 1/(1-p) \rfloor$. Consider a window $[n, n+t]$ (with $n \geq N_0$) for some $t > 0$. Then the average performance of A in $[n, n+t]$ is at least its worst case performance, which is $kp^k(1-p)+\varepsilon$. However, the average performance of the fill-in strategy in this window is (arbitrarily close to) $kp^k(1-p)+2/t$ and this is optimal by Lemma 18. Therefore, for $t > 2/\varepsilon$, this is a contradiction since A cannot be better. □

Finally, we adapt the above proof to randomized algorithms by generalizing Lemma 17 to the randomized setting. The rest of the proof follows immediately from the same arguments as for deterministic algorithms, so extending this lemma suffices to extend the negative results to randomized algorithms.

For a node v, let $q_s(v)$ and $q_r(v)$ be its selection probability and its removed fraction (cf. Lemma 14), respectively. Recall that a node selected with probability $q_s(v)$ removes a fraction $q_s(v)$ of its descendants. We call a randomized algorithm *valid* if the sum of $q_s(v)$ over all vertices v of a monotone path in the conflict graph is at most 1.

Lemma 21. *Consider a valid randomized algorithm with a certain average performance in a window $[n, n+t]$. Applying the following local operators yields a new valid algorithm whose average performance in this window is at least as good as the former algorithm.*

1. *If the algorithm selects a node v of degree $d \geq 1/(1-p)$ for some size $s \in [n, n+t-1]$ with probability $q_s(v) > 0$, set $q_s(v) = 0$ and increase the success probability of its children by $q_s(v)$.*
2. *If for a node v of degree $d \leq 1/(1-p)$ the algorithm sets $q_s(v)+q_r(v) < 1$, increase $q_s(v)$ by $\varepsilon = 1 - q_s(v) - q_r(v)$. Then for every descendant v', set $q_s(v') = 0$ and $q_r(v') = 1$.*

Proof. For both local operators, the claim that applying them does not decrease the average performance follows from the arguments of Lemma 17, so in this proof we will show that both local operators result in a valid algorithm. Let v be the node under consideration and for any node w denote by $q'_s(w)$ and $q'_r(w)$ its selection probability and its removed fraction, respectively, after applying one of the local operators.

Consider the first local operator and any monotone path $P = (v, v_1, v_2, \ldots)$. Note that every monotone path contains exactly one child of v. Then

$$\sum_{w \in P} q'_s(w) = q'_s(v) + q'_s(v_1) + \sum_{i \geq 2} q'_s(v_i) = 0 + (q_s(v_1) + q_s(v)) + \sum_{i \geq 2} q_s(v_i) = \sum_{w \in P} q_s(w)\,.$$

So if the original algorithm was valid, so is the algorithm after applying this operator.

For the second operator, note that we change $q_s(v)$ to $q_s(v) + \varepsilon = q_s(v) + 1 - q_s(v) - q_r(v) = 1 - q_r(v)$. Therefore, after applying the operator, we have $q_s(v) + q_r(v) = 1$. Since in general for any child w of v we have $q_r(w) = q_s(v) + q_r(v)$, we see that $q_s(w) \leq 1 - q_r(w) = 1 - 1 = 0$. The proof is complete.

With this lemma, the proof of the negative results of Theorem 7 is complete, except for one last assumption we need to drop: The proof also holds for known values of n.

Generalization for known n

We now prove that even exact knowledge of the size n that the adversary picks for the instance does not help asymptotically. To do so, we first introduce a variant of the last zero problem.

Definition 6. The *colored last zero problem* is the following:

1. An adversary picks two integers m and n, with $m \leq n$.
2. A sequence of bits of length n is created where every entry independently has value 1 with probability p and 0 otherwise.
3. We color the entries 1 to m with red, while the entries $m+1$ to n are colored blue.
4. The player is given the size n, the number of red 1s and the number of blue 1s.
5. Then the player is presented with the bits one after the other, and for each of them decides whether to stop or to continue.
6. The player wins if she stops on the last red 0 of the sequence.

Note that now the player has three numbers to start with: the number of red samples r, the number of blue samples b, and the size n.

Proposition 7. *The colored last zero problem is equivalent to a specific instance of AOSp with known size n. Therefore, any negative result for the colored last zero problem also holds for AOSp.*

Proof. (Analogue of Proposition 4.) The player again only wins if she stops with the element of the online set with the largest value, only that now she knows in advance how many online elements she is going to observe. Imagine now that she is facing an instance of the following form: The first m elements are assigned a series of positive strictly increasing values, and the remaining $n-m$ take arbitrary negative values. Thus, in this instance the player is aiming for the last non-sampled element among the first m. This is basically the same game as the colored last zero problem, where the red values correspond to the positive values and the blue values correspond to the negative ones. □

With this analog of Proposition 4, we are going to prove the following theorem in the remainder of this section.

Theorem 9. *In the colored last zero problem, no algorithm can achieve performance $kp^k(1-p)+\varepsilon$ on every size $n \geq N_0$ (for some $N_0 > 0$).*

Intuitively, the colored last zero problem should not be much different from the case without colors: there is still an unknown point in the sequence where the player should stop, and there is still a sequence of bits before this point (the red bits). The only difference is that now n is known and we are also given the total number of 1s in the last $n-m$ bits (the blue bits). At first sight these blue 1s seem useless, because the player wants to stop before reaching them. On the other hand, the fact that we know how many they are, gives an indication about the size of $n - m$ and this could be already enough to improve the performance. We show that this is not the case. To do so, we define a slightly different conflict graph, and study its structure to show that up to negligible terms the dynamics are the same as for the standard conflict graph.

Modified conflict graph For the colored last zero problem, m basically plays the role that n was playing before. Therefore, the different layers of the conflict graph correspond to the various sizes of m in this case, and there is a separate conflict graph for each value of n. Note that the conflict graph has a finite number of layers as m varies between 1 and n.

A node of the graph is a couple (S, b), where S is a sequence of bits of length m, that represents the sequence of red bits, and b is an integer that represents the number of blue 1s. The exact positions of 0s and 1s in the blue bits are irrelevant for our proof, only the total number of blue 1s matters.

Finally, just as before, the nodes have different weights, with the difference here that the weights also depend on b. In particular, the weight of a node (S, b) is

$$p^{r+b}(1-p)^{n-r-b}\binom{n-m}{b}.$$

Indeed, the probability of having $r + b$ 1s in an instance of size n when sampling with probability p is $p^{r+b}(1-p)^{n-r-b}$, where r is the number of red 1s. As we group together all the instances with b blue 1s, we multiply by the total number of such instances.

Conflict structure Now let us consider the conflicts. One can see that two nodes (S, b) and (S', b') are in conflict if and only if $b = b'$, and S is in conflict with S' (in the sense of the standard conflict graph). Note that for an instance and its descendants the values b, r and n are the same. In other words, to move from size m to size $m + 1$ we can add a 0 in the appropriate position, just as in Lemma 10.

We now study the relation between the weights of an instance and its children. Let I_1 be a node with a sequence S of size m and let I_2 be one of its children (note that I_2 has size $m + 1$ and is in conflict with I_1). Let p_1 and p_2 be the weights associated with these nodes. We derive from the formula above that:

$$\frac{p_2}{p_1} = \frac{\binom{n-m-1}{b}}{\binom{n-m}{b}} = \frac{n-m-b}{n-m}$$

Having defined the modified conflict graph, we are now ready to show the main result of this section. The key insight is that with high probability, the modified conflict graph has the same weight distribution as in Lemma 13, and therefore, the arguments of the proof sketch in Section 4.2.2 show a contradiction.

Proof of Theorem 9. Consider again the ratio p_2/p_1. The expected value of b is of course $(n - m)p$, but this will not be the case for all instances that we consider. For large values of $n - m$ though, we can apply standard concentration arguments (see also **??**) and obtain that with high probability we have

$$\frac{(n-m)-(n-m)p-\varepsilon}{n-m} \leq \frac{p2}{p1} \leq \frac{(n-m)-(n-m)p+\varepsilon}{n-m} \qquad \iff$$
$$1-p-\varepsilon' \leq \frac{p2}{p1} \leq 1-p+\varepsilon',$$

where $\varepsilon' = \frac{\varepsilon}{n-m}$. From here it is easy to observe that when ε takes a value very close to 0, so does ε'. Furthermore, as $n - m$ grows, ε' vanishes. Thus the modified conflict graph has the same weight distribution as in Lemma 13 with high probability.

Therefore, with high probability, the modified conflict graph is (almost) the same as the weighted conflict graph from Section 4.2.2. Thus, we can follow again the arguments in Section 4.2.2, since they all hold in this case too. We end up with the same impossibility results, which hold here as well both for deterministic and for randomized algorithms. □

4.3 Random order

In this section, we study the second problem of this chapter: the random-order secretary problem with p-sampling, ROSp. To analyze this case, it is useful to have the following equivalent point of view: We assign a uniformly random arrival time τ_i to each of the n elements in the interval $[0, 1]$. If $\tau_i < p$ we add i to S and otherwise we add it to V. Then the elements in V are revealed in the order of the τ_i's. Clearly, $\tau_i < p$ with probability p, so each value is in S independently with probability p. It is also clear that the resulting order is uniformly random. Therefore, any algorithm for the original formulation can be applied to this one. Conversely, an algorithm for this formulation can be transformed into a randomized algorithm for the original one, by sampling $|S|$ uniform arrival times in $[0, p]$ and $|V|$ uniform arrival times in $[p, 1]$.

Consider the following family of algorithms: We fix a sequence $t = (t_i)_{i\in\mathbb{N}}$ such that $0 \leq t_1 < t_2 < \cdots < 1$. Between times t_k and t_{k+1} the algorithm ALG_t sets as a threshold the k-th largest sampled value. More precisely, suppose the value α_i is revealed and assume $t_k \leq \tau_i < t_{k+1}$. ALG_t accepts α_i if it is the largest among the values from V seen so far, and is greater than the k-th largest value from S. For simplicity, if $|S| < k$ we define the k-th largest value of S as $-\infty$. We prove that the best possible success guarantee is attained in this family.

Theorem 10. *There exists a universal sequence t, independent of p and n, such that ALG_t obtains the best possible success guarantee for ROSp. Furthermore, when $p = 0$ this guarantee is equal to $1/e$, and when p tends to 1, the guarantee tends to $\gamma \approx 0.58$, the optimal success guarantee in the full-information secretary problem.*[7]

[7] The optimal guarantee $\gamma \approx 0.58$ was first obtained numerically by Gilbert and Mosteller [80]. An explicit formula for γ was later found by Samuels [137, 138].

We prove this theorem in two main steps. First, we find the sequence t^* that maximizes the success guarantee of ALG_t. Then, we find an expression for the optimal success probability when p and n are given, and prove that for fixed p it converges to the success guarantee of ALG_{t^*} when n tends to infinity.

In order to find the optimal sequence t^* we start by studying the success probability of algorithm ALG_t, for any sequence t, sample rate p and instance size n. We prove that in fact the worst case for this class of algorithms is when n is very large. The approach of approximating the problem when n is large by a continuous time problem was pioneered by Bruss [27] and has been used for different optimal stopping problems (see e.g. Chan et al. [38] and Immorlica et al. [95]).

Lemma 22. *For any sequence t and sampling probability p, the success probability of ALG_t in ROSp decreases with n.*

Proof. Fix a sequence t and a sampling probability p. We use a coupling argument between realizations of the arrival times in instances with n and $n+1$ values. We start with an instance $\alpha_1, \dots, \alpha_{n+1}$, and assume the values are indexed in decreasing order. Consider a realization of the arrival times $\tau_1 = \tau'_1, \dots, \tau_{n+1} = \tau'_{n+1}$ and couple it with the corresponding realization $\tau_1 = \tau'_1, \dots, \tau_n = \tau'_n$ in the instance $\alpha_1, \dots, \alpha_n$. Assume that in the instance with n values and for this particular realization of the arrival times, ALG_t fails. This means that $V \setminus \{\alpha_{n+1}\}$ is non-empty and either ALG_t never stops or it accepts a value that is not the maximum of $V \setminus \{\alpha_{n+1}\}$. Note that regardless of τ'_{n+1}, the rankings of the values in $V \setminus \{\alpha_{n+1}\}$ are the same in both instances because α_{n+1} is smaller than all other values. Thus, if $\tau'_{n+1} < p$, ALG_t does not succeed either when applied in the instance of $n+1$ values. On the other hand, if $\tau'_{n+1} > p$, we have to distinguish between two cases. If ALG_t accepts α_{n+1}, it fails, because $V \setminus \{\alpha_{n+1}\}$ is non-empty and then α_{n+1} cannot be the largest in V. If ALG_t does not accept α_{n+1}, then the behavior of ALG_t in the rest of the variables is the same as in the instance with n values and then it fails.

Since the distribution of $\tau_1, \dots, \tau_n$ is the same in both instances, we conclude with this argument that the probability that ALG_t fails in the instance with $n+1$ values is at least as large as in the instance with n values. □

To prove the lemma the idea is to inductively couple the realizations of the arrival times in instances of sizes n and $n+1$. We show that if ALG_t fails for a given realization of the arrival times of the largest n values in the instance of size n, then ALG_t also fails for any possible realization of the arrival time of the smallest (the $n+1$-th largest) value, in the instance of size $n+1$. This implies that the probability of failure increases with n.

By Lemma 22 the success guarantee of ALG_t is simply the limit of its success probability when n grows to infinity. We calculate these probabilities and obtain an explicit formula for the limit in the following lemma. Interestingly, the formula turns out to be quite simple.

Lemma 23. *Fix a sequence t and a sampling probability p. The success guarantee of ALG_t in ROSp is given by*

$$\sum_{i=1}^{\infty} p^{i-1}\left(1-\max\{p,t_i\}-\int_{\max\{p,t_i\}}^{1}\sum_{j=1}^{i}\frac{t-\max\{p,t_i\}}{t^j}\,dt\right). \tag{4.1}$$

Proof of Lemma 23. We first calculate the success probability of ALG_t for fixed p and n and then take the limit when n tends to infinity.

We say a value α_i is *acceptable* for ALG_t (for a particular realization of the arrival times) if $p < \tau_i$, for some $j \in \mathbb{N}$ we have that $t_j \le \tau_i < t_{j+1}$, and α_i is larger than the j-th largest value in S. Now, note that if $\max V$ is not acceptable for ALG_t, then ALG_t does not stop. This is because we restricted the sequence t to be increasing, so values that arrive before $\max V$ are not acceptable, and values arriving after $\max V$ will not be the best seen so far from V. We use this to decompose the success probability as follows.

$$\Pr(ALG_t \text{ succeeds}) = \Pr(\max V \text{ is acceptable}) - \Pr(ALG_t \text{ stops before seeing } \max V)\,. \tag{4.2}$$

In this definition, if V is empty we also say $\max V$ is acceptable. We first calculate the probability that $\max V$ is acceptable. Assume that the values are indexed in decreasing order, i.e., that $\alpha_1 > \cdots > \alpha_n$.

$$\begin{aligned}\Pr(\max V \text{ is acceptable}) &= \Pr(V=\emptyset) + \sum_{i=1}^{n}\Pr(\max V = \alpha_i)\cdot\Pr(t_i \le \tau_i \mid \max V = \alpha_i)\\ &= p^n + \sum_{i=1}^{n} p^{i-1}(1-p)\cdot\frac{1-\max\{p,t_i\}}{1-p}\\ &= p^n + \sum_{i=1}^{n} p^{i-1}\left(1-\max\{p,t_i\}\right)\,. \end{aligned} \tag{4.3}$$

By the same argument, ALG_t stops before seeing $\max V$ if and only if at least one value arrives after p and before the arrival time of $\max V$, and the maximum such value is acceptable.

$$\begin{aligned}&\Pr(ALG_t \text{ stops before seeing } \max V)\\ &= \sum_{j=1}^{n}\Pr(\max V=\alpha_j)\cdot\Pr(\text{maximum before } \max V \text{ is acceptable} \mid \max V=\alpha_j)\\ &= \sum_{j=1}^{n}\Pr(\max V=\alpha_j)\sum_{i=j}^{n-1}\Pr\Big(\text{max. in } [p,\tau_j) \text{ has rank } i \text{ among elements}\\ &\qquad\qquad \text{that arrive in } [0,\tau_j),\ \text{and arrives in } [t_i,\tau_j)\ \Big|\ \max V=\alpha_j\Big)\end{aligned}$$

$$= \sum_{j=1}^{n} p^{j-1}(1-p) \sum_{i=j}^{n-1} \frac{1}{1-p} \int_p^1 \Pr\Big(\text{max. in } [p,t) \text{ has rank } i \text{ among elements}$$
$$\text{that arrive in } [0,t), \text{ and arrives in } [t_i,t) \;\Big|\; \max V = \alpha_j, \tau_j = t\Big)\, dt$$
$$= \sum_{j=1}^{n} p^{j-1}(1-p) \sum_{i=j}^{n-1} \frac{1}{1-p} \int_{\max\{p,t_i\}}^1 \left(\frac{p}{t}\right)^{i-j} \cdot \frac{(t-\max\{p,t_i\})}{t}$$
$$\cdot \Pr(\text{at least } i \text{ values arrive before } t \mid \max V = \alpha_j, \tau_j = t)\, dt$$
$$= \sum_{j=1}^{n} p^{j-1} \sum_{i=j}^{n-1} \int_{\max\{p,t_i\}}^1 \left(\frac{p}{t}\right)^{i-j} \cdot \frac{(t-\max\{p,t_i\})}{t} \Big(1 - B_{t,n-j}(i-j+1)\Big)\, dt$$
$$= \sum_{i=1}^{n-1} p^{i-1} \int_{\max\{p,t_i\}}^1 \sum_{j=1}^{i} \frac{t-\max\{p,t_i\}}{t^j} \Big(1 - B_{t,n-j}(i-j+1)\Big)\, dt\,, \tag{4.4}$$

where $B_{p,n}(x) = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}$ is the CDF of a Binomial distribution of parameters p and n. Note that for any fixed integers i and j, and time $t \in (0,1)$, $B_{t,n-j}(i-j+1)$ converges to 0 when n tends to infinty. Therefore, replacing Eq. (4.3) and Eq. (4.4) in the identity (4.2), and taking the limit when n tends to infinity, we conclude the proof of the lemma. □

We then focus our attention on optimizing this success guarantee. Surprisingly, it turns out the problem of maximizing Eq. (4.1) is separable and concave, so we can simply impose the first-order conditions to obtain the optimum. Perhaps even more surprising is that these first-order conditions are independent of p, and therefore, the optimal sequence t^* is also independent of p, as the following lemma shows.

Lemma 24. *Fix a sampling probability p. The sequence t^* defined as the unique solution of the equations*

$$\ln\left(\frac{1}{t_i^*}\right) + \sum_{j=1}^{i-1} \frac{(1/t_i^*)^j - 1}{j} = 1, \quad \textit{for all } i \in \mathbb{N}\,, \tag{4.5}$$

maximizes Eq. (4.1)*. In particular, t^* does not depend on p.*

Proof. First, we relax the monotonicity constraint on the sequence of t_i's. The resulting relaxed optimization problem is separable, i.e., optimizing over the entire sequence is equivalent to optimizing over each variable independently. For each t_i we get the following equivalent problem.

$$\max_{t_i \in [0,1]} p^{i-1} \left(1 - \max\{p,t_i\} - \int_{\max\{p,t_i\}}^1 \sum_{j=1}^{i} \frac{t-\max\{p,t_i\}}{t^j}\, dt\right).$$

Equivalently, we can remove the factor p^{i-1} and restrict t_i to be in $[p,1]$, obtaining

$$\max_{t_i\in[p,1]} 1 - t_i - \int_{t_i}^{1} \sum_{j=1}^{i} \frac{t-t_i}{t^j}\, dt\,.$$

Denoting by $G_i(t_i)$ this objective function, we get that

$$\frac{d}{dt_i}G_i(t_i) = -1 + \int_{t_i}^{1} \sum_{j=1}^{i} \frac{1}{t^j}\, dt\,, \text{ and } \quad \frac{d^2}{dt_i^2}G_i(t_i) = -\sum_{j=1}^{i} \frac{1}{t_i^j}\,.$$

Therefore, $G_i(t_i)$ is a concave function and then the optimum is $\max\{p, t_i^*\}$, where t_i^* is the solution of $\frac{d}{dt_i}G_i(t_i) = 0$. In the original objective function t_i appears always as $\max\{p, t_i\}$ so there we can simply take t_i^* as the solution. Now we prove that t_i^* is actually increasing in i, so it is also the optimal solution before doing the relaxation. In fact, t_i^* satisfies

$$\int_{t_i^*}^{1} \sum_{j=1}^{i} \frac{1}{t^j} = 1\,.$$

Note that the left-hand side of this equation is decreasing in t_i^*, and is increasing in i. Thus, necesarily $t_i^* \le t_{i+1}^*$, for all $i \ge 1$. We conclude that t_i^* satisfies Eq. (4.5) by simply integrating on the left-hand side of the last equation. □

Now that we have the best algorithm in the family, we prove that its success guarantee is actually the best possible. To do this, we first characterize the algorithm that achieves the highest success probability for fixed sampling probability p and instance size n.

For a non-decreasing function $\ell : [n] \to [n]$, we define the *sequential-ℓ-max algorithm* in the following way: The algorithm accepts the i-th observed value (considering the values from S and the ones that have been revealed from V) if it is the largest seen so far from V and it is larger that the $\ell(i)$-th largest value from S. We prove that the optimal algorithm is in this class.

Lemma 25. *Fix a sampling probability p and an instance size n. There is a function ℓ such that the sequential-ℓ-max algorithm obtains the best possible success probability for instances of size n of ROSp.*

Proof. We study the optimal ordinal policy obtained with backward induction, and prove that it is in fact a sequential-ℓ-max algorithm for certain ℓ. Recall that we can assume the optimal policy is ordinal, so this algorithm will be optimal not only among ordinal algorithms.

Denote by $X_i = \alpha_{\pi(i)}$ the i-th value, in the order of increasing arrival times. Denote by $R(X_1,\dots,X_j)$ the relative ranks of values $X_1,\dots,X_j$. In what comes, we use the notation $R(X_1,\dots,X_j) = x$ to condition on a particular realization x of the relative ranks. Let x be a realization of the ranks such that X_j is the maximum in V so far, i.e., $X_j = \max V \cap \{X_1, \cdots, X_j\}$, and has rank r among $X_1, \cdots, X_j$. Then,

$$\begin{aligned}
&\Pr\left(X_j = \max V \;\middle|\; R(X_1, \ldots, X_j) = x\right) \\
&= \Pr\left(X_{j+1}, \ldots, X_n \text{ have overall rank at most } r+1 \;\middle|\; R(X_1, \ldots, X_j) = x\right) \\
&= \Pr\left(X_{j+1}, \ldots, X_n \text{ have overall rank at most } r+1\right) \\
&= \prod_{s=0}^{r-1} \frac{j-s}{n-s}\,.
\end{aligned}$$

The optimal policy is to accept X_j if this probability is larger or equal than the probability of picking $\max V$ after rejecting X_j if from $j+1$ onwards we use the optimal policy, conditional on $R(X_1, \ldots, X_j) = x$.

Let now x' be a realization of $R(X_1, \ldots, X_{j+1})$ such that the relative rank of the best of V up to step $j+1$ is r. Suppose that conditional on $R(X_1, \ldots, X_{j+1}) = x'$, the probability of winning if we use the optimal strategy from $j+2$ onwards depends solely of n, $j+1$ and the relative rank r, for all possible ranks r. Denote this conditional probability by $W(n, j+1, r)$. We want to inductively prove that this is in fact true for all n, j and r. It is of course true in the last step, when $j+1 = n$, so we do induction on j. Let x'' be a realization of $R(X_1, \ldots, X_j)$ such that the relative rank of the best of V up to step j is r. We have that

$$\begin{aligned}
&\Pr\left(\text{win after } j \;\middle|\; R(X_1, \ldots, X_j) = x''\right) \\
&= \Pr\left(X_{j+1} \text{ has relative rank } \geq r+1 \;\middle|\; R(X_1, \ldots, X_j) = x''\right) \cdot W(n, j+1, r) \\
&\quad + \sum_{r'=1}^{r} \Pr\left(X_{j+1} \text{ has relative rank } r' \;\middle|\; R(X_1, \ldots, X_j) = x''\right) \\
&\quad \cdot \max\left\{W(n, j+1, r'), \prod_{s=0}^{r'-1} \frac{j+1-s}{n-s}\right\}\,. \qquad (4.6)
\end{aligned}$$

But for all x,

$$\Pr\left(X_{j+1} \text{ has relative rank } r' \;\middle|\; R(X_1, \ldots, X_j) = x\right) = \frac{1}{j+1}\,.$$

This proves the inductive step. Therefore, $W(n, j, r)$ is well defined for all n, j and r, and the optimal policy accepts X_j that has relative rank r and is the maximum so far in V if and only if

$$\prod_{s=0}^{r-1} \frac{j-s}{n-s} \geq W(n, j, r)\,.$$

From Eq. (4.6) it is easy to check that $W(n, j, r)$ is decreasing in j for fixed n, r and increasing in r for fixed n, j.[8] Therefore the optimal policy is the sequential-ℓ-max algorithm, for ℓ defined as

$$\ell(j) = \max \left\{ r : \prod_{s=0}^{r-1} \frac{j-s}{n-s} \geq W(n, j, r) \right\} .$$

This concludes the proof of the lemma. □

To conclude the optimality of ALG_{t^*} we show that the success probability of the best sequential-ℓ-max algorithm for each n converges to Eq. (4.1) for some sequence t, when n grows to infinity. To this end we first calculate the success probability of a sequential-ℓ-max algorithm.

Lemma 26. *Fix n, p and a non-decreasing function ℓ. Consider an integer h such that $0 \leq h < n$, and define $\hat{\ell}(i) = \min\{\ell(i), h+1\}$ for all $i \in [n]$. The success probability of the sequential-ℓ-max algorithm, conditional on $|S| = h$, is given by*

$$\frac{1}{n-h}\left(1 - \prod_{j=0}^{\hat{\ell}(h+1)-1} \frac{h-j}{n-j}\right)$$
$$+ \sum_{i=h+1}^{n-1} \left(\sum_{r=h+1}^{i} \frac{1}{n-i} \left(\frac{1}{i-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{i-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(i+1)-1} \frac{h-j}{n-j} \right) . \tag{4.7}$$

Proof. We calculate first the probability of some events. For $i \in \{h+1, \ldots, n\}$, denote by A_i the event that the i-th element is the largest of V and the algorithm never stops. Notice that A_i is equivalent to the event that the overall largest $\hat{\ell}(i)$ elements are in S, and the i-th element is the largest of V (for this equivalence it is necessary that ℓ is non-decreasing). Therefore, we have that

$$\Pr(A_i) = \frac{1}{n-h} \prod_{j=0}^{\hat{\ell(i)}-1} \frac{h-j}{n-j} .$$

Note that this is 0 if $\hat{\ell}(i) = h+1$. Now, for $h+1 \leq r \leq i \leq n$, define $B_{r,i}$ the event that the r-th element is the largest among positions $\{h+1, \ldots, i\}$ and the algorithm does not stop before $i+1$. This is equivalent to the event that the r-th element is the largest among positions $\{h+1, \ldots, i\}$ and the largest $\hat{\ell}(r)$ elements among positions $\{1, \ldots, i\}$ are in S. Thus,

$$\Pr(B_{r,i}) = \frac{1}{i-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{i-j} .$$

[8] At an intuitive level it is also easy to be convinced of this: as time passes it is harder to win, and if only low values (with large rank) have appeared, it is easier to win in the future.

Now, note that $B_{r,i} \setminus A_r$ is the event that the r-th element is the largest among positions $\{h+1, \ldots, i\}$, but not of V, and the algorithm does not stop before $i+1$. Note also that $A_r \subseteq B_{r,i}$. Therefore, the probability that the algorithm does not stop before $i+1$ and the maximum of V is among positions $\{i+1, \ldots, n\}$ is

$$\sum_{r=h+1}^{i} \Pr(B_{r,i}) - \Pr(A_r) = \sum_{r=h+1}^{i} \frac{1}{i-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{i-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j}.$$

Conditional on this event, the probability that the number in the $i+1$-th position is the largest of V is $1/(n-i)$, because the relative order within positions $\{i+1, \ldots, n\}$ is independent of this event. Thus, we obtained the probability that the $i+1$-th element is the largest of V and the algorithm does not stop before $i+1$. To obtain the probability of winning in step $i+1$, we have to subtract the probability that the $i+1$-th element is the largest of V, but the algorithm never stops, i.e., $\Pr(A_{i+1})$. Therefore, the probability of winning at step $i+1$ is

$$\frac{1}{n-i} \sum_{r=h+1}^{i} \left(\frac{1}{i-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{i-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(i+1)-1} \frac{h-j}{n-j}.$$

The probability of winning at step $h+1$ is slightly different, because the algorithm never stops before it. In that case the probability of winning is

$$\frac{1}{n-h} \left(1 - \prod_{j=0}^{\hat{\ell}(h+1)-1} \frac{h-j}{n-j} \right).$$

Adding these expressions concludes the proof of the lemma. $\square$

We then show that there is a limit for the optimal ℓ in a continuous space, and use a Riemann sum analysis to obtain Eq. (4.1) in the limit.

Lemma 27. *Fix a sampling probability p. For each $n \in \mathbb{N}$, choose $\ell_{p,n}$ so that the sequential-$\ell_{p,n}$-max algorithm achieves the best possible success probability for fixed p and n. There exists a sequence t such that the success probability of the sequential-$\ell_{p,n}$-max algorithm converges to Eq.* (4.1) *when n grows to infinity.*

Proof. First we show that the function ℓ that maximizes Eq. (4.7), in a certain sense converges to a function $\tilde{\ell} : (0,1) \to \mathbb{N}$. Then, we do a Riemann sum analysis to show that the success probability of the sequential-ℓ-max algorithm converges to an expression in terms of $\tilde{\ell}$, and then we show that this can be equivalently expressed as Eq. (4.1) for some sequence t.

Except for terms that vanish when n tends to infinity, Eq. (4.7) can be rewritten as

$$\sum_{r=h+1}^{n} \left(\sum_{i=r}^{n} \frac{1}{n-i} \left(\frac{1}{i-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{i-j} - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{\hat{\ell}(r)-1} \frac{h-j}{n-j} \right). \tag{4.8}$$

To find the optimal $\ell(r)$ we simply maximize the following term as a function of s.

$$F_n(r,s) = \sum_{i=r}^{n} \frac{1}{n-i} \left(\frac{1}{i-h} \prod_{j=0}^{s-1} \frac{h-j}{i-j} - \frac{1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \right) - \frac{1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} .$$

Between s and $s+1$ the change is

$$\begin{aligned}
&F_n(r,s+1) - F_n(r,s) \\
&= \sum_{i=r}^{n} \frac{1}{n-i} \left(\frac{\frac{h-s}{i-s}-1}{i-h} \prod_{j=0}^{s-1} \frac{h-j}{i-j} - \frac{\frac{h-s}{n-s}-1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \right) - \frac{\frac{h-s}{n-s}-1}{n-h} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \\
&= \sum_{i=r}^{n} \frac{1}{n-i} \left(-\frac{1}{i-s} \prod_{j=0}^{s-1} \frac{h-j}{i-j} + \frac{1}{n-s} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \right) + \frac{1}{n-s} \prod_{j=0}^{s-1} \frac{h-j}{n-j} \\
&= \beta(n,s,h) \left(\sum_{i=r}^{n} \frac{1}{n-i} \left(1 - \frac{n-s}{i-s} \prod_{j=0}^{s-1} \frac{n-j}{i-j} \right) + 1 \right) ,
\end{aligned}$$

where $\beta(n,s,h)$ is a positive term, so the sign of this difference is not affected by it. The other term is decreasing in s, so $F_n(r,s)$ is maximized when this differences changes sign. In other words, it is maximized in

$$\ell_n^*(i) = \min \left\{ s \in [n] : \sum_{i=r}^{n} \frac{1}{n-i} \left(1 - \prod_{j=0}^{s} \frac{n-j}{i-j} \right) + 1 \leq 0 \right\} .$$

Now, doing a Riemann sum analysis, we have that $\tilde{\ell}(\tau) = \lim_{n\to\infty} \ell_n^*(\lfloor \tau n \rfloor)$ satisfies

$$\tilde{\ell}(\tau) = \min \left\{ s \in \mathbb{N} : \int_{\tau}^{1} \frac{1}{1-t} \left(1 - \frac{1}{t^{s+1}} \right) + 1 \leq 0 \right\} . \tag{4.9}$$

Thus, interpreting Eq. (4.8) as a Riemann sum, and noting that $|S|/n$ converges to p almost surely, we have that the success guarantee of the optimal policy converges to

$$\int_p^1 \int_\tau^1 \frac{1}{1-t} \left(\frac{1}{t-p} \left(\frac{p}{t} \right)^{\tilde{\ell}(\tau)} - \frac{1}{1-p} p^{\tilde{\ell}(\tau)} \right) dt - \frac{1}{1-p} p^{\tilde{\ell}(\tau)} \, d\tau .$$

From Eq. (4.9) it is clear that $\tilde{\ell}$ is non-decreasing, so we can define the sequence $t_i = \inf \left\{ \tau \in [p,1] : \tilde{\ell}(\tau) \geq i \right\}$ and rewrite the limiting success guarantee in terms of it. Thus, we obtain

$$\sum_{i=0}^{\infty} \left(\int_{t_i}^{t_{i+1}} \int_\tau^1 \frac{1}{1-t} \left(\frac{1}{t-p} \left(\frac{p}{t} \right)^i - \frac{1}{1-p} p^i \right) dt \, d\tau - \frac{t_{i+1}^i - t_i^i}{1-p} \right) .$$

If we rearrange the terms, turning the integral from t_i to t_{i+1} into the difference between the integral from t_i to 1 and the integral from t_{i+1} to 1, we obtain

$$
\begin{aligned}
&\int_p^1 \int_\tau^1 \frac{1}{(t-p)(1-p)}\, dt\, d\tau - \frac{p}{1-p} \\
&+\sum_{i=1}^{\infty} \left(\int_{t_i}^1 \int_\tau^1 \frac{1}{1-t} \left(\frac{\left(\frac{p}{t}\right)^i - \left(\frac{p}{t}\right)^{i-1}}{t-p} - \frac{p^i - p^{i-1}}{1-p} \right) dt\, d\tau + \frac{t_i \left(p^i - p^{i-1}\right)}{1-p} \right) \\
=&\frac{1}{1-p} - \sum_{i=1}^{\infty} p^{i-1} \left(\int_{t_i}^1 \int_\tau^1 \frac{1}{1-t} \left(\frac{t-p}{t^i(t-p)} - \frac{1-p}{1-p} \right) dt\, d\tau + t_i \frac{1-p}{1-p} \right) \\
=&\frac{1}{1-p} - \sum_{i=1}^{\infty} p^{i-1} \left(\int_{t_i}^1 \int_\tau^1 \frac{1}{t^i(1-t)} \left(1-t^i\right)\, dt\, d\tau + t_i \right) \\
=&\frac{1}{1-p} - \sum_{i=1}^{\infty} p^{i-1} \left(\int_{t_i}^1 \int_\tau^1 \sum_{j=0}^{i-1} \frac{t^j}{t^i}\, dt\, d\tau + t_i \right) \\
=&\sum_{i=1}^{\infty} p^{i-1} \left(1 - t_i - \int_{t_i}^1 \int_\tau^1 \sum_{j=1}^{i} \frac{1}{t^j}\, dt\, d\tau \right) \\
=&\sum_{i=1}^{\infty} p^{i-1} \left(1 - t_i - \int_{t_i}^1 \sum_{j=1}^{i} \frac{t-t_i}{t^j}\, dt \right) .
\end{aligned}
$$

This concludes the proof, since we defined the t_i's in a way that they satisfy $t_i = \max\{p, t_i\}$. □

Finally, we study the success guarantee of ALG_{t^*} in the border values of p, and show that it actually becomes equal to the best possible among all algorithms. It is easy to see that the success guarantee is $1/e$ when $p = 0$. Note that when $p = 0$, Eq. (4.1) simplifies to $t_1 \ln(1/t_1)$, and that from Eq. (4.5) we obtain that $t_1^* = 1/e$. Replacing gives the success guarantee of $1/e$. The case when p tends to 1 is a bit more involved and requires some tedious calculations. We evaluate Eq. (4.1) with the first order approximation $t_i^* \approx t_i' := 1 - c/i$, where c is a constant. To fix c we impose that (t_i') satisfies Eq. (4.5) in the limit when $i \to \infty$. More precisely, we take c such that

$$
\begin{aligned}
1 &= \lim_{i\to\infty} \ln\left(\frac{1}{1-c/i}\right) + \sum_{j=1}^{i-1} \frac{(1-c/i)^{-j} - 1}{j} \\
&= \int_0^1 \frac{e^{cx} - 1}{x}\, dx\,.
\end{aligned}
$$

With this in hand, we do then a Riemann sum analysis to show the next lemma, which states that when p tends to 1, this approximation converges to the explicit expression of Samuels [137, 138] for γ.

Lemma 28. *Let $t'_i = 1 - c/i$, where c is the solution of $\int_0^1 \frac{e^{cx}-1}{x} dx = 1$. When evaluated in t', Eq. (4.1) tends to*

$$\gamma = e^{-c} + (e^{-c} - 1 - c) \int_1^{\infty} x^{-1} e^{-cx} \, dx \approx 0.5801 \,, \tag{4.10}$$

when p tends to 1.

Proof. We analyze separately the sum when $p = \max\{p, t'_i\}$ and when $t'_i = \{p, t'_i\}$. We call the first part V_1, which includes the terms up to $i = \lfloor \frac{c}{1-p} \rfloor$, and V_2 the rest.

$$\begin{aligned}
V_1 &= \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} p^{i-1} \left(1 - p - \int_p^1 \sum_{j=1}^{i} \frac{t-p}{t^j} \, dt \right) \\
&= \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} p^{i-1} \left(1 - p - \int_p^1 dt + \int_p^1 \frac{dt}{t^i} - \int_p^1 \sum_{j=1}^{i} \frac{1-p}{t^j} \, dt \right) \\
&= \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} p^{i-1} \left(\frac{p^{-(i-1)} - 1}{i-1} - (1-p) \ln(1/p) - (1-p) \sum_{j=2}^{i} \frac{p^{-(j-1)} - 1}{j-1} \right) \\
&= \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} \frac{1 - p^{i-1}}{i-1} - \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} (p^{i-1} - p^i) \sum_{j=2}^{i} \frac{e^{-(j-1) \ln p} - 1}{j-1} \\
&= \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} \frac{1 - (p^{\frac{1}{1-p}})^{(i-1)(1-p)}}{(i-1)(1-p)} (1-p) - \lim_{p \to 1} \sum_{i=1}^{\lfloor \frac{c}{1-p} \rfloor} (p^{i-1} - p^i) \sum_{j=2}^{i} \frac{e^{-\frac{(j-1)}{i} i \ln p} - 1}{(j-1)/i} \cdot \frac{1}{i}
\end{aligned}$$

Interpreting these two sums as Riemann sums, we obtain

$$\begin{aligned}
V_1 &= \int_0^c \frac{1 - e^{-x}}{x} \, dx - \int_{e^{-c}}^1 \int_0^1 \frac{e^{-x \ln y} - 1}{x} \, dx \, dy \\
&= \int_0^c \frac{1 - e^{-x}}{x} \, dx - \int_{e^{-c}}^1 \int_0^1 \frac{e^{-x \ln y} - 1}{-x \ln y} (-\ln y) \, dx \, dy \\
&= \int_0^c \frac{1 - e^{-x}}{x} \, dx - \int_{e^{-c}}^1 \int_0^{-\ln y} \frac{e^x - 1}{x} \, dx \, dy \\
&= \int_0^c \frac{1 - e^{-x}}{x} \, dx - \int_0^c \int_{e^{-c}}^{e^{-x}} \frac{e^x - 1}{x} \, dy \, dx \\
&= \int_0^c \frac{1 - e^{-x} - (e^{-x} - e^{-c})(e^x - 1)}{x} \, dx \\
&= e^{-c} \int_0^c \frac{e^x - 1}{x} \, dx
\end{aligned}$$

$$
\begin{aligned}
&= e^{-c} \int_0^1 \frac{e^{cx}-1}{x}\, dx \\
&= e^{-c} ,
\end{aligned}
$$

where the last step comes from the definition of c. On the other hand, we have that

$$
\begin{aligned}
V_2 &= \lim_{p\to 1} \sum_{i=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} p^{i-1} \left(\frac{c}{i} - \int_{1-\frac{c}{i}}^{1} \sum_{j=1}^{i} \frac{t-1+c/i}{t^j}\, dt \right) \\
&= \lim_{p\to 1} \sum_{i=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} p^{i-1} \left(\frac{c}{i} - \int_{1-c/i}^{1} dt + \int_{1-c/i}^{1} \frac{1}{t^i}\, dt - \int_{1-c/i}^{1} \sum_{j=1}^{i} \frac{c/i}{t^j}\, dt \right) \\
&= \lim_{p\to 1} \sum_{i=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} p^{i-1} \left(\frac{(1-c/i)^{-(i-1)}-1}{i-1} + \frac{c}{i}\ln(1-c/i) - \sum_{j=2}^{i} c\frac{(1-c/i)^{-(j-1)}-1}{i(j-1)} \right) \\
&= \lim_{p\to 1} \sum_{i=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} (p^{i-1}-p^i) \frac{(1-c/i)^{-(i-1)}-1}{\frac{1-p}{-\ln p}(i-1)(-\ln p)} \\
&\quad - \lim_{p\to 1} \sum_{i=\left\lfloor \frac{c}{1-p} \right\rfloor+1}^{\infty} \frac{p^{i-1}-p^i}{\frac{1-p}{-\ln p} i(-\ln p)} \sum_{j=2}^{i} \frac{c\left((1-c/i)^{-i\frac{j-1}{i}}-1\right)}{j/i} \cdot \frac{1}{i} ,
\end{aligned}
$$

where in the last equality we omitted a term that vanishes when p tends to 1. We again interpret the sums as Riemann sums.

$$
\begin{aligned}
V_2 &= \int_0^{e^{-c}} \frac{e^c-1}{\ln(1/x)}\, dx - c\int_0^{e^{-c}} \frac{1}{\ln(1/x)} \int_0^1 \frac{e^{cy}-1}{y}\, dy\, dx \\
&= (e^c - 1 - c) \int_0^{e^{-c}} \frac{1}{\ln(1/x)}\, dx \\
&= (e^{-c} - 1 - c) \int_1^{\infty} x^{-1} e^{-cx}\, dx .
\end{aligned}
$$

In the second equality we used the definition of c and in the third one we performed a change of variables. Summing V_1 and V_2 we get Eq. (4.10). □

4.4 Robustness with respect to the knowledge of the parameters

In this section, we briefly discuss the impact of the knowledge of the parameters on the guarantees that can be obtained. There are two parameters for both AOSp and ROSp: the number of elements n and the sampling probability p. The performance of an algorithm can vary a lot depending on its presumed knowledge about these parameters.

For AOSp we already discussed that knowledge of n is irrelevant in worst case terms. To complete the picture, we turn our attention to the cases when p is unknown. First, if p is unknown but n is known, we show that the ratio of the number of samples to the total number of elements gives a good estimate of p, and that using k-max with this estimate is basically optimal. More specifically, assume we are given a set S of h samples, drawn independently from an initial set consisting of n values in total, using some (unknown) value of p. The remaining $n-h$ samples form the online set V. In this setting we adapt the k-max algorithm as follows by simply setting the threshold to the k-th largest sample, where $k = \left\lfloor \frac{n}{n-h} \right\rfloor$, and accepts the first value of the online set that is above the threshold. This variation of the k-max algorithm boils down to simply estimating p as $\hat{p} = h/n$ and using $\hat{p}$ to determine the desired value of k. By standard concentration arguments, we can prove that the estimate $\hat{p}$ is accurate with high probability, and thus we obtain the following theorem.

Theorem 11. *For AOSp with known n and unknown p, the variation of the k-max algorithm for unknown p achieves the best possible success guarantee up to a factor $1-\varepsilon$ with high probability.*

Before formally proving the theorem, we need to find the expression for the success probability.

Lemma 29. *For a given sample set S with h values and an online set V with $n-h$ values, the k-max algorithm chooses the maximum value of the online set with probability*

$$\Pr[\mathit{Win}] = \sum_{h=0}^{n} \left\lfloor \frac{n}{n-h} \right\rfloor \left(\frac{h}{n}\right)^{\left\lfloor \frac{n}{n-h} \right\rfloor} \frac{n-h}{n} \binom{n}{h} p^h (1-p)^{n-h} ,$$

where p is the probability of independently sampling a value from the initial set.

Proof. Assume that the values of the adversarial input $\mathcal{A}$ are sorted in decreasing order $\alpha_1 > \alpha_2 > \ldots > \alpha_n$. Let us call p_h the probability that the k-max algorithm succeeds in a particular instance with h samples and S_h the event where $|S| = h$. Then the total probability that the k-max algorithm succeeds equals

$$\begin{aligned} \Pr[\text{Win}] &= \sum_{h=0}^{n} \Pr[k\text{-max algorithm wins} \mid S_h] \cdot \Pr[S_h] \\ &= \sum_{h=0}^{n} p_h \binom{n}{h} p^h (1-p)^{n-h} , \end{aligned}$$

since each value of the initial set is sampled independently with probability p. It remains to determine p_h. Conditioned on the fact that we end up with h samples, all the different labelings (as a sample or online value) of the initial n values are equally likely to happen. There are $\binom{n}{h}$ different labelings, and each α_i is labeled as a sample in an h/n-fraction of the possible labelings and as an online value in the rest.

Observe, as in the proof of Lemma 7, that the algorithm succeeds only if *exactly one* of the $\left\lfloor \frac{n}{n-h} \right\rfloor$ largest values of the adversarial input ends up in the online set and the $(\left\lfloor \frac{n}{n-h} \right\rfloor + 1)$-th largest ends up in the sample set. To compute the number of such labelings, first consider those such that $\alpha_1, \alpha_2, \ldots, \alpha_{\lfloor \frac{n}{n-h} \rfloor + 1}$ are all labeled as samples except for exactly one. From those, we can exclude the labelings that mark $\alpha_{\lfloor \frac{n}{n-h} \rfloor + 1}$ as an online value, since in this case the $\left\lfloor \frac{n}{n-h} \right\rfloor$-th largest sample is larger than all the online values. Therefore, we obtain

$$\begin{aligned} p_h &= \left(\left\lfloor \frac{n}{n-h} \right\rfloor + 1 \right) \left(\frac{h}{n} \right)^{\lfloor \frac{n}{n-h} \rfloor} \left(\frac{n-h}{n} \right) - \left(\frac{h}{n} \right)^{\lfloor \frac{n}{n-h} \rfloor} \left(\frac{n-h}{n} \right) \\ &= \left\lfloor \frac{n}{n-h} \right\rfloor \left(\frac{h}{n} \right)^{\lfloor \frac{n}{n-h} \rfloor} \left(\frac{n-h}{n} \right), \end{aligned}$$

and the lemma follows. □

We will also use the following well-known concentration bound in proving the main theorem of this section.

Lemma 30 (Hoeffding's inequality for i.i.d. Bernoulli random variables [93])**.** *Let $X_1, X_2, \ldots, X_n$ be i.i.d. Bernoulli random variables with parameter p and let $\bar{X} = \left(\sum_{i=1}^{n} X_i\right)/n$. Then for any $\varepsilon > 0$,*

$$\Pr\left[\left|\bar{X} - pn\right| \geq \varepsilon\right] \leq 2e^{-2n\varepsilon^2}.$$

Alternatively, by setting $\delta = 2e^{-2n\varepsilon^2}$ we get that

$$\left|\bar{X} - pn\right| \leq \sqrt{\frac{1}{2n} \ln \frac{2}{\delta}} \qquad \textit{with probability at least } 1 - \delta.$$

With these two lemmas at hand, we are ready to prove the theorem.

Proof of Theorem 11. Consider an instance of AOSp for a fixed unknown value of p where the player is faced with h samples.

Let ε_1 and ε_2 be such that

$$\varepsilon_1 \leq 1 - \frac{\left(\frac{h}{n}\right)^{\frac{n}{n-h}}}{p^{\frac{1}{1-p}}} \quad \text{and} \quad \varepsilon_2 \leq 2e^{-2n}.$$

Note that the first value is chosen such that

$$\left\lfloor \frac{n}{n-h} \right\rfloor \left(\frac{h}{n} \right)^{\lfloor \frac{n}{n-h} \rfloor} \frac{n-h}{n} \geq \left(\left\lfloor \frac{1}{1-p} \right\rfloor \right) p^{\left\lfloor \frac{1}{1-p} \right\rfloor} (1-p) \cdot (1 - \varepsilon_1),$$

while the second is chosen such that Hoeffding's inequality yields $\Pr\left[\left|\bar{X} - pn\right| < 1\right] \geq 1 - 2e^{-2n} \geq 1 - \varepsilon_2$. Therefore, with probability at least $1 - \varepsilon_2$, we have

$$\sum_{h=pn-\varepsilon}^{pn+\varepsilon} \binom{n}{h} p^h (1-p)^{n-h} = \binom{n}{h} p^h (1-p)^{n-h} \bigg|_{h=pn} \geq 1 - \varepsilon_2.$$

Therefore, we can bound the success guarantee given by Lemma 29 as follows.

$$\begin{aligned}
\Pr[\mathrm{Win}] &= \sum_{h=0}^{n} \left\lfloor \frac{n}{n-h} \right\rfloor \left(\frac{h}{n}\right)^{\left\lfloor \frac{n}{n-h} \right\rfloor} \frac{n-h}{n} \binom{n}{h} p^h (1-p)^{n-h} \\
&\geq \left\lfloor \frac{1}{1-p} \right\rfloor p^{\left\lfloor \frac{1}{1-p} \right\rfloor} (1-p) \cdot (1-\varepsilon_1) \cdot \sum_{h=pn-\varepsilon}^{pn+\varepsilon} \binom{n}{h} p^h (1-p)^{n-h} \\
&\geq \left\lfloor \frac{1}{1-p} \right\rfloor p^{\left\lfloor \frac{1}{1-p} \right\rfloor} (1-p) \cdot (1-\varepsilon_1) \cdot (1-\varepsilon_2)\,.
\end{aligned}$$

For any given $\varepsilon > 0$, one can take ε_1 and ε_2 that adhere to the bounds above and such that $(1-\varepsilon_1)(1-\varepsilon_2) \leq (1-\varepsilon)$. This yields a success guarantee that is at least $1-\varepsilon$ times the success guarantee of the k-max algorithm for known p. □

Second, for AOSp where both p and n are unknown, we show that no non-trivial guarantee can be obtained. The intuition behind this strong negative result results from the situation in which the algorithm is given very few samples. In this case, it does not know whether the instance is very short (in which case it should stop early), or the sampling probability is very low (in which case it should wait longer).

Theorem 12. *When both p and n are unknown, no algorithm can get positive success guarantee for AOSp.*

Proof. We prove that for any $\varepsilon > 0$, it is not possible to achieve a success guarantee of ε.

Consider the following new game for any $\delta > 0$. The adversary selects a size n and generates an instance of this size with increasing values. Then, the adversary again selects p appropriately, so that the probability that there is at least one sample is at most δ and the probability that there are no samples is at least $1-\delta$. Then the sampling process happens and the player faces the sequence. If at least one value is sampled, the player automatically wins, otherwise, she wins if and only if she selects the last non-sampled value.

Consider the case where there are no sampled values. Since the player does not learn anything along the game, any deterministic algorithm waits $t-1$ values before it selects the t-th value. A randomized algorithm can be thought of as a distribution over the stopping times t. Since the domain of t are all positive integers, it is not possible that this distribution has weight at least λ for every size, for any constant $\lambda > 0$. Therefore, on instances with stopping probability less than λ, the player only wins with probability at most λ. Such an instance occurs trivially with probability at most 1.

Overall, in this new game, the player wins in at most $\delta + \lambda$ values. Taking e.g. δ and λ slightly smaller than $\varepsilon/2$, the success guarantee of this game is less than ε.

The proof for AOSp with unknown p and n follows easily now. The adversary chooses values of n and p as above. In case there are no sampled values, both games are the same, since in both cases the player has the same information and the same available strategies. In case there is at least one sampled value, the player wins in the new game with probability 1 and in AOSp with probability strictly less than 1. Therefore, the success guarantee of AOSp is at most the success guarantee of the new game, which is less than ε. □

For ROSp, we have shown that the optimal algorithm ALG_{t^*} does not depend on p, and knowledge of the uniform random arrivals suffices to obtain the optimal guarantee. Therefore, ALG_{t^*} achieves the best possible success guarantee, even when n is unknown. On the other hand, if p is unknown and n is known and large, then we can sample uniform random arrival times for each value and obtain with ALG_{t^*} the best success guarantee. Indeed, the sampled arrival times themselves will provide a sharp estimate of p.

On a more applied note, whenever it is reasonable to assume that the values come in random order, it is usually also safe to assume that this random order comes from random arrival times. In case the arrival times are random but not uniform, the time thresholds t^* can be transformed using the distribution function of the arrival times and again it is possible to obtain the optimal success guarantee.

4.5 Extensions

We expect that the ideas developed in this chapter will prove useful in other contexts related to online decision-making. For example, very recently Kaplan et al. [99] study a sample-based version of the online edge-weighted bipartite matching with vertex arrivals. For their analysis they use our independent sampling model. Our model can also be incorporated in well-established problems with combinatorial constraints. In what remains, the extensions are concerned with the well-studied matroid secretary problem. Among the most promiment problems in the area are the knapsack secretary problem (Babaioff et al. [12]) and the rank-k uniform matroid secretary problem. The latter corresponds to the scenario in which an auctioneer wants to sell at most k items to a random stream of bidders, and it was studied in Kleinberg [104] In order to motivate further work we discuss more in detail two relatively straightforward extensions of our model and results. Here, the decision-maker faces a sequence of the elements of a given ground set and needs to select a subset, subject to the constraint that the selected set has to be an independent set of an underlying matroid.

4.5.1 Graphical matroid secretary problem with independent sampling

First, consider the graphical matroid secretary problem, in which the underlying matroid is graphical (i.e., the independent sets are forests of an undirected graph). For this problem, Korula and Pál [108] gave a $1/(2e)$-competitive algorithm for the case in which the elements are presented to the decision-maker in random order. In a nutshell, Korula and Pál fix an ordering of the vertices and with probability $1/2$ they orient all edges from the lower numbered vertex to the higher numbered vertex, and with probability $1/2$ they orient all edges in the other direction. Then they run for each vertex independently a standard secretary algorithm to find the maximum-weight edge leaving this vertex. It is not difficult to see that this gives a $1/(2e)$-approximation.

Now consider the random order graphical matroid secretary problem with independent sampling, in which every edge is sampled independently a priori with a fixed probability p and the goal is to select the maximum-weight independent set of the non-sampled elements. Let $\alpha_R^*(p)$ be the success guarantee of ROSp that we obtained before. The analysis of Korula and Pál [108] immediately yields a success guarantee of $\alpha_R^*(p)/2$ for this extension to graphical matroids.

4.5.2 Laminar matroid secretary problem with independent sampling

Second, consider the laminar matroid secretary problem, in which underlying matroid is laminar[9]. For this class, a sequence of papers have obtained constant factor guarantees [97, 116] until the currently best known factor of 5.16 [144]. All these papers rely on the idea of first (binomially) sampling a fraction of the elements of the matroid to guide the posterior decisions. However, the final goal is to compare to the optimal solution that includes even the sampled elements. An interesting direction will be to study the performance guarantees of these algorithms when the benchmark is the optimal solution of the online set as in this chapter.

Moreover the technique of first using independent sampling is ubiquitous in secretary problems with combinatorial constraints. Therefore we believe that understanding the performance guarantees when compared to the optimum of the online set is interesting not only from a theoretical perspective, but also from a practical viewpoint since these samples can be interpreted as historical data.

4.6 Potential directions and open questions

Finally, we discuss some of the most promising future directions and provide some insights wherever possible. We conclude with a number of other interesting open questions.

4.6.1 Maximizing the expectation

In our work, we didn't consider the objective of maximizing the expected performance of the best algorithm compared to the expected performance of a prophet as in Kaplan et al. [98]. It is therefore natural to ask how this objective behaves in our model for all different values of p. Let us create a starting point by briefly studying the problem for the special case of $p = 1/2$ and adversarial order. Before analyzing a first algorithm, let's recall how Kaplan et al. proved tight bounds in their model. Remember that as $n \to \infty$ their model and ours are essentially equivalent. When less than half of the total elements are sampled, namely $|S| < |V|$, they use $T = \max_{i \in S}$ as their threshold. When $|S| \geq |V|$ they sample $|V| - 1$ elements uniformly at random from $|S|$ and again set the one with the maximum value as their thresold. For completeness, we state the guarantee they obtain.

[9] A laminar family is a collection $\mathcal{A}$ of subsets of a ground set E such that, for any two intersecting sets, one is contained in the other. For a capacity function c on $\mathcal{A}$, a laminar matroid is given by the family of independent sets $\{I : |I \cap A| \leq c(A), \text{ for all } A \in \mathcal{A}\}$.

Theorem 13 (Kaplan et al. [98]). *When $|S| \geq |V|$ the above algorithm achieves a competitive ratio of* $1/2$.

Through nice techniques and carefully constructed instances they provide matching upper bounds for both cases. We revisit their upper bound when $|S| \geq |V|$, and give a slightly simpler proof for that case which still conveys the intuition. We believe that similar ideas could be useful in deriving hardness results in other online bayesian selection problems.

Theorem 14 (Kaplan et al. [98]). *No algorithm can achieve competitive ratio better than* $1/2$.

Proof. The adversary constructs two instances, $\mathcal{I}_1$ and $\mathcal{I}_2$, and chooses them for the game with probability $1-\varepsilon$ and ε, respectively. Instance $\mathcal{I}_1$ consists of n elements in total (for some large n); half of them have value ε and the rest have value 0. Instance $\mathcal{I}_2$ is the same as $\mathcal{I}_1$, except that one of the elements with value 0 is substituted uniformly at random with an element α' with value $\ell > 0$. Then, a fraction $q \geq 1/2$ for some given q of the chosen instance is sampled. When fixing an algorithm, we can even assume that the decision-maker knows exactly the two instances and with which probability she will face each one of them. Still, we will show that even with such a restricted adversary, she cannot hope for an asymptotic guarantee better than $1/2$.

We assume that n is sufficiently large so that all the elements with value ε are sampled with vanishing probability (i.e., $\Pr[\text{no element with value } \varepsilon \text{ in } V] = q^{\frac{n}{2}} = \delta$, for small $\delta > 0$). A first observation is that given the samples the decision-maker cannot always distinguish between the two instances. If $\alpha' \notin S$ she cannot tell if she is facing $\mathcal{I}_1$ or $\mathcal{I}_2$ until either α' appears or she observes all the elements of the online sequence. Therefore, the optimal algorithm will simply do the following: If $\alpha' \in S$ she will always stop with value ε. If $\alpha' \notin S$ the decision-maker faces two bad scenarios. Either she stops with value ε, but she is in $\mathcal{I}_2$ and she should have continued till she encounters value $\ell > \varepsilon$ or she skips all elements with value ε, but she is in $\mathcal{I}_1$ and she ends up with value 0. Thus, when $\alpha' \notin S$ any algorithm will pick an element with value ε with some probability p, and continue with $1-p$ (and collect 0 or ℓ). The crucial point is to optimally set this probability p and get the best possible competitive ratio. Note that the decision-maker can possibly also change p dynamically, if her beliefs change as the online instance is revealed.

Now that we defined the family of algorithms where the optimal lies, we can express the expected performance of any algorithm as a function of p. In Table 4.1 we state the expected performance of the optimal online algorithm $\mathrm{OPT}_{\mathrm{on}}$ and the optimal offline algorithm $\mathrm{OPT}_{\mathrm{off}}$ for the three different scenarios that we will encounter. We can find the optimal algorithm by simply maximizing the competitive ratio over p:

$$\begin{aligned}\frac{\mathbb{E}[\mathrm{OPT}_{\mathrm{on}}]}{\mathbb{E}[\mathrm{OPT}_{\mathrm{off}}]} &= \max_{p\in[0,1]} \frac{(1-\varepsilon)\varepsilon p + \varepsilon^2 q + \varepsilon q\,(p\varepsilon + (1-p)\ell)}{(1-\varepsilon)\varepsilon + \varepsilon^2 q + \varepsilon q \ell} \\ &= \max_{p\in[0,1]} \frac{\varepsilon^2 + \varepsilon\ell + p\left(\frac{(1-\varepsilon)\varepsilon}{q} - \varepsilon\ell + \varepsilon^2\right)}{\varepsilon^2 + \varepsilon\ell + \frac{(1-\varepsilon)\varepsilon}{q}}\,.\end{aligned}$$

Instances	$\mathrm{OPT_{on}}$	$\mathrm{OPT_{off}}$
$\mathcal{I}_1$	$(1-\varepsilon)\cdot\varepsilon\cdot p$	$(1-\varepsilon)\cdot\varepsilon$
$\mathcal{I}_2, \alpha' \in S$	$\varepsilon\cdot\varepsilon\cdot q$	$\varepsilon\cdot\varepsilon\cdot q$
$\mathcal{I}_2, \alpha' \notin S$	$\varepsilon\cdot q\cdot(p\cdot\varepsilon+(1-p)\cdot\ell)$	$\varepsilon\cdot q\cdot\ell$

Table 4.1: The expected performance of the optimal online and offline algorithm on the two instances.

The adversary can choose the value ℓ of element α' so that $\left(\frac{(1-\varepsilon)\varepsilon}{q} - \varepsilon\ell + \varepsilon^2\right) \leq 0$. In this case, the expression is maximized for $p = 0$ and the chosen ℓ must satisfy $\ell \geq \frac{1-\varepsilon}{q} + \varepsilon$. The competitive ratio now becomes

$$\frac{\mathbb{E}[\mathrm{OPT_{on}}]}{\mathbb{E}[\mathrm{OPT_{off}}]} = \frac{\varepsilon+\ell}{\varepsilon+\ell+\frac{1-\varepsilon}{q}} \underset{\varepsilon\to 0}{\approx} \frac{1}{1+\frac{1}{q\ell}} ,$$

and for $\varepsilon \to 0$ the chosen ℓ must now satisfy $\ell \geq \frac{1}{q}$. If the adversary chooses the sampled fraction q of the total elements and the value ℓ of element α' to satisfy the relation $q = \frac{1}{\ell}$ (that is, with equality), then for the competitive ratio it always holds that $\frac{\mathbb{E}[\mathrm{OPT_{on}}]}{\mathbb{E}[\mathrm{OPT_{off}}]} \approx \frac{1}{2}$, and the proof is complete. □

Now let us assume that for the case of $p = 1/2$ we use the same algorithm as we used for the secretary objective, namely take $T = \max_{i\in S} a_i$ as the threshold and accept the first online element above the threshold. Following the approach of Rubinstein et al. [135] we can analyze the performance of this algorithm.

Theorem 15. *Assume that an adversary picks the order and the (non-negative) values of n elements $a_1, a_2, \dots, a_n$, and each a_i is sampled independently with $p = 1/2$. For the algorithm* ALG *described above it holds that* $\frac{\mathbb{E}[\mathrm{ALG}]}{\mathbb{E}[\max_{i\in V} a_i]} \geq 1/2$.

Proof. We begin by sorting the elements in decreasing order $a_{\sigma(1)} \geq a_{\sigma(2)} \geq \cdots \geq a_{\sigma(n)}$ and studying the distribution of the expected maximum. For $a_{\sigma(1)}$ to be the maximum valued online element, it has to end up in set V, which happens with $p = 1/2$. Following the same reasoning, for $a_{\sigma(k)}$ to be the maximum in the online set, it must be that all $a_{\sigma(\ell)}$ with $\ell < k$ were sampled and $a_{\sigma(k)}$ is the first non-sampled element. Since each of these events happens independently with $p = 1/2$ we have that $\Pr[\max_{i\in V} a_i = a_{\sigma(k)}] = 1/2^k$. Then it's easy to see that

$$\mathbb{E}[\max_{i\in V} a_i] = \sum_{i=1}^{n} a_{\sigma(i)} \cdot \Pr[\max_{i\in V} a_i = a_{\sigma(k)}] = \sum_{i=1}^{n} a_{\sigma(i)} \cdot \frac{1}{2^i} .$$

Next we analyze the probability of our algorithm picking each of the n elements. The algorithm picks $a_{\sigma(1)}$ only if $a_{\sigma(1)} \in V$ and $a_{\sigma(1)} \in S$, which occurs with probability $1/4$. Similarly, the algorithm picks element $a_{\sigma(k)}$ if $\mathrm{T} = a_{\sigma(k+1)}$. This happens when $a_{\sigma(1)}, a_{\sigma(2)}, \dots a_{\sigma(k)} \in V$ and $a_{\sigma(k+1)} \in S$. Therefore, it holds that $\Pr[\mathrm{ALG} = a_{\sigma(k)}] =$

$\Pr[\mathrm{T} = a_{\sigma(k+1)}] = \frac{1}{2^{k+1}}$. Finally, note that $\Pr[\mathrm{ALG} = a_{\sigma(n)}] = \Pr[\mathrm{T} = 0] = \frac{1}{2^n}$. Putting everything together we get

$$\begin{aligned}\mathbb{E}[\mathrm{ALG}] &= \sum_{i=1}^{n} a_{\sigma(i)} \cdot \Pr[\mathrm{ALG} = a_{\sigma(k)}] \\ &= \sum_{i=1}^{n-1} a_{\sigma(i)} \cdot \frac{1}{2^{i+1}} + \frac{1}{2^n} \\ &\geq \sum_{i=1}^{n} a_{\sigma(i)} \cdot \frac{1}{2^{i+1}} \\ &= \frac{1}{2}\mathbb{E}[\max_{i \in V} a_i] \ .\end{aligned}$$

□

Next we show through a simpe example that the previous lower bound is actually tight.

Lemma 31. *For $p = 1/2$, no algorithm can achieve a better competitive ratio than $1/2$.*

Proof. We use again the notation $\mathrm{OPT_{on}}$ and $\mathrm{OPT_{off}}$ to denote the optimal online and optimal offline algorithm, respectively. The adversary creates an instance with two elements $\mathcal{I} = \{a_1, a_2\}$ with values ε and 1, respectively. In the table below we can see what any algorithm and the prophet achieve for each of the possible scenarios.

Probability	Elements		Algorithms	
	ε	1	$\mathrm{OPT_{on}}$	$\mathrm{OPT_{off}}$
1/4	S	V	1	1
1/4	V	S	0	ε
1/4	S	S	0	0
1/4	V	V	ε	1

Table 4.2: Performance of $\mathrm{OPT_{on}}$ and $\mathrm{OPT_{off}}$ on instance $\mathcal{I}$ for $p = \frac{1}{2}$.

Recall that a crucial point is that we do not know the total number of elements n in the instance when we design our algorithm. Recall also that we announce our algorithm to an adversary, who then in turn fixes the instance. In the case that we get no samples, as in the last row of Table 4.2, from our perspective we know that $p = 1/2$ and an unknown number of elements is in the online set. The best we can do is always pick the first element since we do not know if there are more elements in the online sequence. Note that in this specific instance $\mathcal{I}$, in case we pick the first online element with some probability q and then the next one with the remaining $1 - q$ (in case there is a second one), we gain more in the last row, but we lose the gained fraction in the first row. Therefore the best algorithm obtains $\mathbb{E}[\mathrm{OPT_{on}}] = \frac{1+\varepsilon}{4}$ and the prophet $\mathbb{E}[\mathrm{OPT_{off}}] = \frac{2+\varepsilon}{4}$, resulting in $\frac{\mathbb{E}[\mathrm{OPT_{on}}]}{\mathbb{E}[\mathrm{OPT_{off}}]} \to \frac{1}{2}$ as $\varepsilon \to 0$. □

On the other hand, if we know n beforehand we can do better, at least for small values of n.

Example 2. Consider the above example with the extra information that $n = 2$. In this case we want to do something different when we have no samples, and since we know that the adversary is always going to pick the worst-case order, we choose each of the elements with probability $q = 1/2$. This will make the adversary indifferent on the ordering of the two elements. The above table now is as follows:

Probability	Elements		Algorithms	
	ε	1	$\mathrm{OPT_{on}}$	$\mathrm{OPT_{off}}$
1/4	S	V	1	1
1/4	V	S	ε	ε
1/4	S	S	0	0
1/4	V	V	1/2	1

Table 4.3: Performance of $\mathrm{OPT_{on}}$ and $\mathrm{OPT_{off}}$ on instance $\mathcal{I}$ for known small values of n.

This results in an improved performance $\mathbb{E}[\mathrm{OPT_{on}}] = \frac{3/2+\varepsilon}{4}$ and the competitive ratio now becomes approximately 3/4.

The observations above motivate further exploration of the expectation maximization objective. How does the objective behave for all different values of p? How does the performance compare to the results Kaplan et al. [98]? Apart from obtaining asymptotic results (i.e., $n \to \infty$) is there a chance that we can characterize the performance for relatively small values of n and the algorithms that achieve best possible guarantees in this case? Can the results extend to well-known Bayesian online selection problems where the objective is to maximize the expected performance of an algorithm against a prophet?

4.6.2 Potential connections to the full information case

In this section we want to further expand on one of the most intriguing open questions of our work, which is the connection of AOSp and the problem studied by Allaart and Islas [2] and Esfandiari et al. [71]. We already pointed at this direction in Section 4.1.3; although our guarantees converge to the full information case and conceptually the two problems are similar, we are not aware of a precise connection.

The algorithm that they use for solving the problem is a simple single-threshold rule: Find the value T for which $\Pr\left[\max_{i\in[n]} a_i \leq T\right] = \frac{1}{e}$ and set it as a threshold. Accept the first element whose value is above T. They show that the probability of picking the maximum value is at least $\frac{1}{e}$ and then they give an instance with distributions $F_1, F_2, \ldots, F_n$, such that as $n \to \infty$ no algorithm can do better than $\frac{1}{e}$. Therefore, the guarantee that they obtain for their full-information model match ours when $p \to 1$. We are still far from understanding whether and how the two models are connected, but our intuition is that it should be the case. In this section, we provide a different instance and a different proof technique for showing that $\frac{1}{e}$ is tight. We do this because the instance we provide resembles the ones that we used to prove tight results in AOSp (there is an increasing sequence of expected values), and because the new proof uncovers also the specific algorithm that we would use for the instance to achieve this guarantee. This might facilititate a bit finding the precise connections. Finally, we give an informal idea of how we could simulate the full information case through AOSp.

We construct the following instance I: We have n distributions $F_1, F_2, \ldots, F_n$ which have just two point masses. The random variable a_i drawn from F_i takes value i with probability $\frac{1}{n}$ and 0 otherwise. We will prove that no algorithm can be the maximum element with probability better than $\frac{1}{e}$ and that an optimal algorithm always rejects the first $\frac{1}{e}$-fraction of the values and then picks the first non-zero value, if any.

Let $q_i = \Pr[\text{ALG stops at } a_i \mid a_i > 0]$. In the following lemmas we will also denote by V^* the maximum value of the sequence, i.e., $V^* = \max_{i \in [n]} a_i$. We start with the following lemma.

Lemma 32. *For instance I the probability that any algorithm ALG picks the element with maximum value is given by*

$$\frac{q_1}{n}\left(1-\frac{1}{n}\right)^{n-1} + \sum_{i=2}^{n}\left(\prod_{j=1}^{i-1}\left(1-\frac{q_j}{n}\right)\frac{q_i}{n}\left(1-\frac{1}{n}\right)^{n-i}\right). \tag{4.11}$$

Proof. Observe first that

$$\begin{aligned}
\Pr[a_i = V^*] &= \Pr\left[a_i = i \ \wedge \ a_j = 0, \ \forall j > i\right] \\
&= \Pr[a_i = i] \cdot \prod_{j=i+1}^{n} \Pr[a_j = 0] \\
&= \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-i},
\end{aligned}$$

where the second equality is due to the independence of the random variables.

Now we will try to come up with an optimal algorithm for the instance. Any candidate algorithm, deterministic or randomized, adaptive or not, does the following: Whenever it sees a positive value α_i, it decides with probability q_i whether to stop or not. Since we know the order of arrival, if we find the optimal sequence of such probabilities $(q_1^*, q_2^*, \ldots, q_n^*)$ from the optimization problem, then what remains is to design an algorithm which implements this sequence of probabilities. For an algorithm ALG we can write down the maximization problem for our objective:

$$\begin{aligned}
\Pr\left[\text{ALG picks } V^*\right] &= \sum_{i=1}^{n} \Pr[\text{ALG stops at } a_i \ \wedge \ a_i = V^*] \\
&= \sum_{i=1}^{n} \Pr[\text{ALG does not stop at time } j, \ \forall j < i]
\end{aligned}$$

$$\cdot \Pr[\text{ALG stops at } a_i \mid a_i > 0] \cdot \Pr[a_i = V^*]$$
$$= \frac{q_1}{n}\left(1-\frac{1}{n}\right)^{n-1} + \sum_{i=2}^{n}\left(\prod_{j=1}^{i-1}\left(1-q_j\cdot\frac{1}{n}\right) q_i \cdot \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-i}\right) .$$

□

Now we should optimize Eq. (4.11) over the sequence $(q_1, q_2, \ldots, q_n)$. It turns out that the optimization problem has a specific form with respect to the values of the arguments q_j. The following observation significantly reduces the space of potential optimal solutions.

Lemma 33. *Eq.* (4.11) *is maximized when* $q_j \in \{0, 1\}$, $\forall j \in [n]$.

Proof. If we want to separate q_k for some $k \in [n]$ from the rest of the expression, we can rearrange the terms in Eq. (4.11) and obtain

$$\frac{q_1}{n}\left(1-\frac{1}{n}\right)^{n-1} + \frac{q_k}{n}\prod_{j=1}^{k-1}\left(1-\frac{q_j}{n}\right) + \left(1-\frac{q_k}{n}\right)\sum_{i=2}^{n}\frac{q_i}{n}\prod_{\substack{j=1\\ j\neq k}}^{i-1}\left(1-\frac{q_j}{n}\right)\left(1-\frac{1}{n}\right)^{n-i} .$$

To simplify a bit the expression, let $A = \frac{q_1}{n}\left(1-\frac{1}{n}\right)^{n-1}$, $B = \prod_{j=1}^{k-1}\left(1-\frac{q_j}{n}\right)$ and $C = \sum_{i=2}^{n}\frac{q_i}{n}\prod_{\substack{j=1\\ j\neq k}}^{i-1}\left(1-\frac{q_j}{n}\right)\left(1-\frac{1}{n}\right)^{n-i}$. We will prove now by contradiction that each q_j should take value 0 or 1 in order for the expression to be maximized. Let's assume that there is an optimal assignment $(q_1, q_2, \ldots, q_n) \in [0, 1]^n$. Therefore we cannot increase the objective by changing the value of any q_j. Starting from the beginning find the first $q_k \in (0, 1)$ and assume without loss of generality that $k \neq 1$. As written above, from the perspective of q_k, the optimization problem is

$$A + \frac{q_k}{n}\cdot B + \left(1-\frac{q_k}{n}\right)\cdot C .$$

Note that expressions A, B, C are all independent of q_k. Moreover, apart from the constant term A, the rest of the expression is a convex combination of the two points B and C. Therefore , if $B > C$ we set $q_k = 1$ and the objective will increase. Likewise, if $B < C$ then we set $q_k = 0$. If $B = C$ the value of the objective will not be affected from the choice of q_k so we can again set either $q_k = 0$ or $q_k = 1$. We proceed in the same way for all other $q_l \in (0, 1)$ after k. We conclude that by changing the values of q_i's to 0 or 1 the objective cannot decrease. Thus, Eq. (4.11) attains its maximum for some $(q_1^*, q_2^*, \ldots, q_n^*) \in \{0, 1\}^n$. □

We managed to make a crucial step towards restricting the form of the optimal solution. The next observation tells us that we can focus in an even more restricted family of solutions to find the optimal one.

Lemma 34. *Without loss of generality, the optimal solution can be of the following form: For some i^*, $q_j = 0$ if $j < i^*$ and $q_j = 1$ otherwise.*

Proof. We use an exchange argument and show that the objective increases if we swap the values of $q_m = 1$ and $q_k = 0$ for some $l < m$. Assume without loss of generality that we have fixed a solution $\vec{q}$. Denote by m the smallest index for which $q_m = 1$ and by k the first index after m for which $q_k = 0$. For this assignment fixed by some algorithm ALG_1, the probability of picking the optimal is

$$\begin{aligned}
\Pr[\text{ALG}_1 \text{ picks } V^*] &= \sum_{i=m}^{k-1} \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-i} \prod_{j=1}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) \\
&\quad + \sum_{i=l}^{n} \frac{\mathbb{1}_{\{q_i=1\}}}{n}\left(1-\frac{1}{n}\right)^{n-i} \prod_{j=1}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) \\
&= \frac{1}{n}\sum_{i=m}^{k-1}\left(1-\frac{1}{n}\right)^{n-i}\left(1-\frac{1}{n}\right)^{i-m} \\
&\quad + \frac{1}{n}\sum_{i=l+1}^{n}\left(1-\frac{1}{n}\right)^{n-i+k-m} \prod_{j=l}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) \\
&= \frac{1}{n}(k-m)\left(1-\frac{1}{n}\right)^{n-m} \\
&\quad + \frac{1}{n}\sum_{i=l+1}^{n}\left(1-\frac{1}{n}\right)^{n-i+k-m} \prod_{j=l}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) .
\end{aligned}$$

Now consider the assignment which swaps the values of q_m and q_k. Moreover, assume that there exists an algorithm ALG_2 which does exactly that, i.e., induces the same assignment as $\vec{q}$ with the difference that now $q_m = 0$ and $q_k = 1$. In this case, we have that

$$\begin{aligned}
\Pr[\text{ALG}_2 \text{ picks } V^*] &= \sum_{i=m+1}^{k} \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-i} \prod_{j=1}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) \\
&\quad + \sum_{i=l}^{n} \frac{\mathbb{1}_{\{q_i=1\}}}{n}\left(1-\frac{1}{n}\right)^{n-i} \prod_{j=1}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) \\
&= \frac{1}{n}(k-m)\left(1-\frac{1}{n}\right)^{n-m-1} \\
&\quad + \frac{1}{n}\sum_{i=l+1}^{n}\left(1-\frac{1}{n}\right)^{n-i+k-m} \prod_{j=l}^{i-1}\left(1-\frac{\mathbb{1}_{\{q_j=1\}}}{n}\right) .
\end{aligned}$$

Subtracting the two probabilities we get

$$\Pr[\text{ALG}_2 \text{ picks } V^*] - \Pr[\text{ALG}_1 \text{ picks } V^*] = \frac{1}{n}(k-m)\left(1-\frac{1}{n}\right)^{n-m-1} - \frac{1}{n}(k-m)\left(1-\frac{1}{n}\right)^{n-m} > 0 \,,$$

which means that the objective increases by swapping the values of q_m and q_k when $q_m = 1$, $q_k = 0$, and $m < k$. By applying this process iteratively we conclude that the optimal solution $\vec{q}^*$ has $q_j = 0$ for $j < i^*$ and $q_j = 1$ for $j \geq i^*$. □

Now we are ready to prove the hardness result as $n \to \infty$ for the instance I we constructed and show an optimal algorithm for this case.

Theorem 16. *No algorithm can pick the element with maximum value with probability better than* $\frac{1}{e}$. *Moreover, for the instance* I *defined earlier, an algorithm that achieves* $\frac{1}{e}$ *just accepts the first element with non-zero value.*

Proof. From Lemma 34 we want to find the value i^* which maximizes Eq. (4.11). We can rewrite the objective for $i^* \geq 2$ as follows:

$$\begin{aligned}
\Pr[\text{ALG picks } V^*] &= \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-i^*} + \frac{1}{n}\sum_{i=i^*+1}^{n}\left(1-\frac{1}{n}\right)^{n-i}\prod_{j=i^*}^{i-1}\left(1-\frac{1}{n}\right) \\
&= \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-i^*} + \frac{1}{n}(n-i^*)\left(1-\frac{1}{n}\right)^{n-i^*} \\
&= \frac{1}{n}(n-i^*+1)\left(1-\frac{1}{n}\right)^{n-i^*} . \qquad (4.12)
\end{aligned}$$

In the case of $i^* = 1$ following similar calculations we end up with

$$\Pr[\text{ALG picks } V^*] = \left(1-\frac{1}{n}\right)^{n-1} ,$$

and thus we can use Eq. (4.12) for all values of i^*. We claim that Eq. (4.12) is maximized for $i^* = 1$ because it is a decreasing function for all values of n. Indeed, fix two consecutive values k and $k+1$ for i^*, and note that

$$\begin{aligned}
&\Pr[\text{ALG picks } V^* \mid i* = k] > \Pr[\text{ALG picks } V^* \mid i* = k+1] \Leftrightarrow \\
&\frac{1}{n}(n-k+1)\left(1-\frac{1}{n}\right)^{n-k} > \frac{1}{n}(n-k)\left(1-\frac{1}{n}\right)^{n-k-1} \Leftrightarrow \\
&(n-k+1)\left(1-\frac{1}{n}\right) > n-k \Leftrightarrow \\
&\frac{n-k+1}{n} > 1 \Leftrightarrow k > 1 \,.
\end{aligned}$$

Thus, the probability of picking the element with maximum value is decreasing as i^* increases, except when $i^* = 1$ and $i^* = 2$ in which case the objective obtains the same value for both values of i^*. This means that setting $i^* = 1$ or $i^* = 2$ maximizes the probability. The algorithm which implements the solution that assigns $q_i = 1$ for all i, essentially accepts the first element with non-zero value that appears in the online sequence of instance I . □

Now that we have specified an algorithm which gives a $1/e$ guarantee, we will briefly try to make a high-level connection between our AOSp model and the setting we are studying here. We mention again that we are not aware of a precise connection between the two, and we believe that it would be an interesting direction to explore. The sampling process could be simulated in two different ways. In the first scenario, we know that there are n unknown distributions. Then we sample (in expectation) r values from each distribution, so that in total we have roughly $n \cdot r$ samples. The online set consists of one realization from each distribution (i.e., n values in total) which are adversarially presented. A second, similar way of conducting the sampling process is to sample $n \cdot r$ values in expectation, each one drawn independently from one of the n distributions at random. The question is whether we can use something along the lines of the k-max algorithm to recover the approximate guarantee of $1/e$ as r grows large (which corresponds to learning the distributions with smaller and smaller error). The instance that we analyzed before could serve as an indication that something in the spirit of k-max could work. Remember that an algorithm that achieves a guarantee of $1/e$ in this case accepts the first element with non-zero value. A variant of k-max would probably work in a similar way: Because of these two point-mass distributions, we expect a very big fraction of the sampled elements to have value 0. As r grows (which would informally correspond to $p \to 1$ in our AOSp model), a variant of the k-max algorithm will pick an element with small value as the (unique) threshold. In this instance, it it very likely that for large r the chosen sample will have value 0; this also implies that a k-max variant would accept the first online element with positive value, and the asymptotic guarantee could be close to $1/e$.

4.6.3 Further directions

Finally, we mention a few promising open questions which would either help us gain insight into the powers and limitations of our model, or extend it in ways that it could capture a number of important applications.

Different assumptions on the arrival order In this work, we studied the secretary problem with independent sampling first for adversarial and then for uniform random arrival order of the online elements. These two different assumptions on the order of arrival are very common in the analysis of online algorithms. The main critique of the adversarial order assumption is that it might be too pessimistic; on the other hand, the uniform random order assumption usually leads to significantly improved guarantees, but it is sometimes perceived as quite strong and unrealistic. Moreover, it is often the case that algorithms designed under the random order assumption are not robust to small perturbations of the arrival model. Therefore, a rich open direction would be to understand how the guarantees of our model and the algorithms change under different assumptions in the arrival order. Such ideas have already been explored in the literature; Bradac et al. [25] study a robust version of the secretary problem by mixing adversarial and random order, and Kesselheim et al. [102] design novel arrival models that are considerably weaker than the uniform random order.

Possibility of recall The field of sequential decision-making under uncertainty (or online algorithms in computer science) is traditionally based on the assumption that the decisions are immediate and irrevocable, and the classical secretary problem also assumes that. Often though in practical scenarios , such as in many hiring procedures, the decision-maker might still have the chance to recall a candidate; the candidate then might reject the offer if she has already accepted another position. It would be interesting to incorporate in our model a (probably time-dependent) probability of recalling an already rejected online element.

Moving beyond ordinal and single selection objectives Since we want to draw connections to problems in pricing and mechanism design, it is natural to study objective functions that are better suited to these problems. A first natural step would be to consider the objective function of Kleinberg [104]: The decision-maker is allowed to choose k elements, and the goal is to maximize the sum of their values. The benchmark is the offline optimum where always the top k elements are selected. A relaxed version of the secretary objective is also of interest: The decision-maker can now again choose only one element, but the objective is to maximize the probability of picking any of the top k elements, instead of just the top one.

Unify different models Since the secretary problem is one of the most fundamental models in sequential decision-making under uncertainty, numerous variants of the classical setting have been extensively studied. The variants might make different assumptions on the information the decision-maker possesses (e.g. whether the values of the online elements are drawn from known or unknown distributions), the arrival order of the online sequence, the game-theoretic considerations (e.g. the online elements might be strategic agents who might misreport their value), and more. It is an interesting direction from a theoretical point of view to try and uncover similarities between some of these variants. It might be possible to design general-purpose models which manage to capture different settings studied in the literature and recover their guarantees. In very recent work, Dütting et al. [68] unify different informational assumptions under a single formulation, including also our sampling model for the random arrival order.

5 The Impact of Ordering on the Pandora's Box Problem

5.1 Introduction

In the previous chapter, we presented a sampling model for the secretary problem, one of the most fundamental problems in online decision-making. We also mentioned several recent results in prophet inequalities, a closely related problem which also involves immediate and irrevocable decisions. Furthermore, recall that in the secretary problem, the values of the online elements are fixed by an adversary, while in the prophet inequality they are drawn from known distributions. A common point between these two problems is that the information, i.e., the revealed values, is acquired without cost.

However, in many real-life scenarios learning the hidden information might require monetary transfers or be computationally costly. For example, when looking to buy a house, the interested buyer has to search for potential purchases. By looking at real-estate advertisements, she shapes some prior belief about a house and makes an initial screening. To inspect a house and learn exactly how much she values it, she has to undertake a costly action. This might involve investing time and paying a price to a real-estate agent. Moreover, the revealed information after each inspection will play an important role in which house she is going to inspect next. This procedure raises a tradeoff between exploration and exploitation: Given the time pressure and other potential factors, should she keep inspecting after finding a decent choice? When should she stop the search and obtain the best option so far among the inspected ones? Of course, here we consider a simplified motivating scenario, where we assume that the houses will still be available for sale for at least a short period of time following inspection (see also Section 2.3 for more details on the aforementioned scenario). Similar tradeoffs and costly evaluation processes arise when a company is looking to hire a skilled worker, or when trying to identify profitable investment opportunities.

Costly information acquisition gives rise to a new class of stochastic optimization problems; the introduced search costs crucially affect the structure of (optimal) solutions of such optimization problems. Note that these processes also involve sequential decision-making under uncertainty; the decision-maker tries to optimize an objective in an environment with limited information, and each new piece of information obtained influences how and whether the search will be continued. A characteristic example stemming from the field of microeconomics is *search theory*. The field of search theory dates back to the paper of Stigler [145], who introduced the idea of consumers' search costs as one of the explanations for the phenomenon of price dispersion. In recent decades, the idea of incorporating search costs in models of information acquisition spans the literature of operations research, economics, and computer science.

Perhaps the most well-established mathematical model of search theory is Weitzman's Pandora's box problem [150] (see also Section 2.3 for an informal description of the problem). We will now define some aspects of the problem more formally. We remind the reader that in the Pandora's box problem there are n boxes; each box is associated with a known distribution F_i and an inspection cost c_i. The distributions need not be identical, but they have to be independent. The decision-maker can now decide the order in which the boxes are opened. Every time she opens a box b_i, she pays the cost and observes the *reward* $v_i \sim F_i$. Based on the observed rewards and the costs paid, she decides whether to continue by opening another box or terminate the search. In the classical Pandora's box problem, the decision-maker collects the reward of at most one of the opened boxes. Given in advance the costs and the distributions, the decision-maker designs an adaptive policy in order to maximize her *expected gain*. Let $S \subseteq [n]$ denote the subset of opened boxes. Then, the expected gain can be written as

$$\mathbb{E}\left[\max_{i \in S} v_i - \sum_{i \in S} c_i\right] ,$$

where the randomness is over the independent draws from the distributions. As mentioned in Section 2.3, although it seems that the optimal policy should be very complex to describe, it turns out to be very simple: Order the boxes in decreasing order of a quantity termed the *reservation value* σ_i, and start opening them in this order. When the reward of an opened box is greater than the reservation values of the uninspected boxes, we stop and collect the maximum reward. This achieves the maximum expected gain among all adaptive policies. The reservation value σ_i can be found by simply solving the following equation for each box:

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

where $(v_i - \sigma_i)^+ = \max\{v_i - \sigma_i, 0\}$. It intuitively expresses the maximum reward that we need to have collected, so that we are indifferent between opening b_i and leaving it uninspected. Weitzman's policy has both a non-adaptive and an adaptive component; the order in which we open the boxes is fixed in advance, while the stopping time depends on the random rewards that we observe. Note also that the Pandora's box optimal policy can be seen as a special case of the Gittins index theorem (see e.g. [81]).

In this chapter, our main goal is to study the impact of exogenous (borrowing the term from the work of Kleinberg and Kleinberg [103]) ordering on the Pandora's box problem. We want to understand what type of policies we need to design when we do not have full control over which box to inspect next. Inspired by the field of online decision-making and recent research on online variants of the Pandora's box problem, some of our formulations have an online flavor; boxes are presented to us one by one, that is, we do not know in advance the exact order of exploration. We explore the impact of ordering by studying it under in two of the most common cases encountered in the literature; the adversarial and the random order case. We design approximately-optimal policies for a variety of natural models (but always for the classical Pandora's box objective) that could be applied in different practical scenarios. Note that slightly changing the assumptions of the model can significantly change the structure of a good policy. An important parameter is what we are comparing our policies to; we argue in the next section why the benchmarks we consider are strong and suitable for the respective formulations.

5.1.1 Our contributions

Before describing the results in more detail, let us briefly explore the different models we can consider and establish some terminology. We study the single-selection objective of the Pandora's box problem in two new classes of problems. The first is the adversarial order Pandora's box problem (AOPB for short), in which the order of exploration of the boxes is fixed by an adversary. Likewise, the second class of problems is the random order Pandora's box problem (ROPB for short), in which the boxes arrive in a uniform random order. The assumption on the arrival order significantly affects the the structure of the optimization problem. Following, we mention other important parameters that one can consider in order to come up with the desired formulation.

Commitment This parameter fixes whether we are allowed to take the maximum reward of any opened box (*without commitment*) when we stop the search, or we have to immediately and irrevocably decide whether to select a reward upon inspecting a box (*with commitment*). The former corresponds to the Pandora's box objective, while the latter resembles a prophet inequality setting with search costs.

Skipping Since we do not choose the order of exploration as in the classic model, we should specify whether we are allowed to ignore a box that is presented to us and let it go forever (*with skipping*) or we have to open each box, in which case we can either stop or pay the inspection cost of the next box in the sequence (*without skipping*). The former setting is perhaps more natural, but in order to gain further insights on the impact of ordering we will consider both.

Online arrivals The classic Pandora's box problem already involves some sort of online decision-making since we decide, possibly adaptively, when to stop the search. Moreover, we can add an extra layer of online decision-making by assuming that the boxes arrive online. The random order Pandora's box is by definition an online problem, but the adversarial (or fixed) order can be defined in two possible ways: Either the adversary fixes the order and we get to observe the whole sequence at the start, or the boxes arrive one by one and we cannot observe the whole sequence a priori. In the latter case, we can think of two different types of adversaries: an oblivious adversary (who decides on the order before we start playing) or an adaptive adversary (who can choose which box to present to us next based on the realized rewards and our decisions so far).

Later in this chapter we will see that some of the variants of the Pandora's box problem studied in the recent literature can be defined by a specific choice of the aforementioned parameters.

We start in Section 5.2 by studying ROPB without skipping and without commitment.We show that the strongest tractable benchmark that we can consider is the optimal (probably not polynomial-time) adaptive policy. Thus, we strive to find simple policies whose expected gain is a good approximation of that of the optimal adaptive policy. We show that a very natural exponential-time policy, which uses as a subroutine the optimal policy for the known fixed order case, can perform arbitrarily bad in a carefully constructed family of instances. We interpret this as an indication that a strong hardness result, such as finding a suitable reduction, might apply. The type of policies that need to be employed in order to achieve a meaningful approximation for this setting remain elusive to us.

Next, in Section 5.3 we study AOPB and ROPB when there is the possibility of skipping. In this setting, we argue that the most suitable benchmark is the performance of the optimal policy in the free order case, i.e., Weitzman's policy. Therefore, here we truly compare the impact that a different order of exploration has on our objective. We present a general reduction from the Pandora's box with commitment under any ordering to the prophet inequality under the same ordering with the Pandora's box. This reduction has already been employed for specific settings in Segev and Singla [141] and Kleinberg and Kleinberg [103]. Our proof follows similar steps to Kleinberg and Kleinberg [103, Section 4]. The reduction establishes lower bounds for the adversarial and the random order in the Pandora's box problem with commitment achieved by simple threshold-based algorithms.

More obstacles start to arise when we try to develop policies for AOPB and ROPB without commitment; they should still be simple, but at the same time take into account the observed rewards, since now we can always collect them after we stop. Intuitively, the commitment constraint is quite restrictive; a policy should be able to perform much better when it has the power to collect the reward of any opened box. Surprisingly, we show that for AOPB this is not the case: The fixed threshold policy designed for the variant with commitment is still optimal when we assume that there is no commitment! Next, we construct a hard instance inspired by Correa et al. [56, Section 5] to provide an upper bound for the performance of any policy in ROPB without commitment. Closing the small gap between the lower and upper bound in ROPB without commitment is an interesting open question.

We conclude in Section 5.4 by making some final observations and stating the current and future directions of the settings under study in this chapter.

5.1.2 Related work

Our work can be placed in the stream of literature that studies sequential search problems. As mentioned in Section 5.1.1, our models are also related to the literature on online selection problems. More specifically, throughout the chapter we refer many times to the classic prophet inequality, and some of its variants. One of the most interesting variants which has recently received a lot of attention is the prophet secretary, the variant of the prophet inequality in which the elements come in random order [1, 56, 69, 114]. The prophet inequality differs from our models (the ones without commitment) in two fundamental ways: first, there is no inspection cost and second, the decision of whether to collect the current reward must be immediate and irrevocable, i.e., there is no possibility for recall.

The surprising simplicity of Weitzman's optimal policy sparked further interest in the problem and its extensions. In some cases, the core concepts of Weitzman's optimal algorithm carry over (e.g. in [22]), but this is not always the case. One of the most natural and well motivated variants for which Weitzman's policy is not optimal is studied in Doval [64]. The difference to the Pandora's box problem is that the decision-maker can choose a box without necessarily inspecting it first (and subsequently stop the search). Doval shows the intractability of this model in the general case, in contrast to the classic Pandora's box problem. She then explores different ways of imposing additional structure on the model, so that the optimal policy can be fully or partially characterized. Due to the lack of structure of the optimal policy in the general case, Beyhaghi and Kleinberg [20] study simple, polynomial-time policies that achieve constant-factor approximations to the expected gain of the optimal adaptive policy. In a slightly different context, Attias et al. [7] also incorporate the idea of non-obligatory inspection in their model.

Other natural extensions include having a utility function that takes into account the rewards of all opened boxes instead of just the maximum one [130], having rewards that are a sum of a known and a hidden factor [48], considering a richer inspection model [6], and assuming correlations among the distributions of the rewards [41]. In the computer science literature, a recent line of work has generalized the Pandora's box model to various combinatorial optimization problems [76, 87, 142].

The importance of the Pandora's box problem can be illustrated by the multitude of applications in which it has been used as a building block. Derakhshan et al. [61] draw inspiration from it to develop a two-stage consumer search model in online platforms. Kleinberg et al. [106] develop a model based on the Pandora's box problem to study bidders' information acquisition in simultaneous auctions. They also present a novel interpretation and proof of the optimality of Weitzman's policy that inspired further work [20, 142]. Immorlica et al. [96] also use the Pandora's box problem for modeling costly information acquisition in matching markets. Their main case study is college admissions, where students have to gather information for colleges they wish to attend in order to learn their true preferences over them. Kleinberg and Kleinberg [103] and Bechtel and Dughmi [17] apply the Pandora's box problem to design delegation mechanisms, where a principal assigns the task of choosing from a set of alternatives to an agent. In this context we would like to quantify the cost of delegation, i.e., how much a principal loses by assigning a task to someone else.

Close to our work are the online variants of the Pandora's box problem that have appeared in the literature. In a well-studied online variant, we can still choose the order of inspection but we have the commitment constraint [75, 141]. Esfandiari et al. [70] present a general reduction that connects an online Pandora's box problem, where the rewards *and* the costs are jointly drawn from a distribution, and the prophet inequality.

The reduction also extends to more general objective functions with different feasibility constraints. The work of Boodaghians et al. [22] is also related to ours. They initiate the study of the Pandora's box problem when there are constraints in the exploration order of the boxes. They derive the optimal policy when the boxes are in a line (i.e., known fixed order), and when they form tree-like constraints. The optimal policy can also be computed in polynomial time. Then, they show a hardness result when considering slightly more general constraints in the order of exploration.

Finally, as mentioned in Section 5.1.1, one of the benchmarks we consider is the optimal adaptive online policy. Although the approximability of the optimal online policy by polynomial time algorithms is a natural question, it has not been extensively studied in the literature of online selection problems. Nevertheless, there has been recent work that tries to answer this question for well-known models [4, 127, 132].

5.2 Exogenous ordering without skipping

In this section, we discuss AOPB and ROPB when we have to pay the cost of an arriving box, or stop the search before inspecting it. Our focus will be mostly on the setting without commitment, since our primary goal is to understand the effect of changing the order of exploration, while keeping the same objective function. First, we should establish what our benchmark is. Ideally, we would like to find the optimal, computationally-efficient policy and compare its performance to that of Weitzman's policy in the free order case. Unfortunately, when we cannot skip it is easy to see that we cannot guarantee any fraction of the performance of Weitzman's policy. The following simple example for ROPB illustrates that.

Example 3. We consider n boxes that will be presented to us in random order. For simplicity, we choose deterministic rewards and costs. In particular:

- Boxes $b_1, b_2, \cdots, b_{n-1}$ have reward $v_1 = v_2 = \cdots = v_{n-1} = 0$ and cost $c_1 = c_2 = \cdots = c_{n-1} = H$, for some large positive number H.
- Box b_n has reward $v_n = M$ for some large positive number $M < H$, and cost $c_n = 0$.

In other words, boxes $b_1, b_2, \cdots, b_{n-1}$ are dummy boxes with no reward and huge cost and, thus, they will never be explored. Their reservation values are $\sigma_1 = \sigma_2 = \cdots = \sigma_{n-1} = 0$, and the reservation value of box b_n is $\sigma_n = M$. Weitzman's policy opens b_n and stops, for a gain of WEITZ $= M$.

Any optimal algorithm for the random order will only enter the game if the good box b_n appears first, in which case it will open it and stop, for a gain of M. In any other case, starting to inspect boxes will necessarily result in negative gain. Since the order is random, for a permutation $\pi : [n] \to [n]$, we have that $\Pr[b_{\pi(n)} = b_1] = 1/n$. The expected gain of any optimal algorithm then becomes $\mathbb{E}[\text{ALG}] = \frac{M}{n} + \frac{n-1}{n} \cdot 0 = \frac{1}{n}\text{WEITZ}$. As n grows, the expected gain of any algorithm becomes a vanishing fraction of that of Weitzman's policy. Therefore, we conlcude that there are instances in which $\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{WEITZ}]} \to 0$, and we cannot hope to get any guarantee against this benchmark.

Of course, the above example also shows that we cannot compete against any type of offline benchmark that always knows the realized rewards (as the prophet benchmark does in prophet inequalities). Thus, we aim to approximate the next strongest meaningful benchmark, which is the optimal online policy, using polynomial-time algorithms. This is our main goal in this section.

5.2.1 Random order

Before proceeding to the algorithms, let us mention that we can obtain the optimal solution for ROPB using dynamic programming. Denote by R the set of boxes that still remain unopened. Let $\text{OPT}(R, i)$ be the maximum reward we can collect minus the costs paid when the remaining set of items is R, and the next box is $i \in R$. With this notation at hand, the dynamic program can be formulated as follows:

$$\text{OPT}(R, i) = \max\left(\max_{j \notin R} v_j, -c_i + \mathbb{E}_{k \in R \setminus \{i\}, v_k \sim \mathcal{D}_k}\left[\text{OPT}(R \setminus \{i\}, k)\right]\right) . \tag{5.1}$$

The solution of the dynamic program is $\mathbb{E}\left[\text{OPT}([n], i)\right]$. The expectation is taken over the independent draws of the rewards and the uniform random permutation of the boxes. Note that the dynamic program decides whether to continue at each point without knowing which box will appear next for inspection. It is easy to see that the state space grows very quickly; indeed, in experiments we conducted with synthetic data we observed that for instances larger than nine boxes, each extra box adds a huge computational burden for solving the DP to optimality. Therefore, we need to find polynomial time algorithms with provably good guarantees to be able to tackle large instances. In Appendix C.1, we formally show that the DP we formulated returns the optimal expected gain.

Moreover, note that the following upper bound to the DP and the optimal online policy holds:

$$\mathbb{E}\left[\text{OPT}([n], i)\right] \le \frac{1}{n!} \sum_{\pi \in S_n} \text{OPT}_{\text{fixed}}(\pi_i) . \tag{5.2}$$

This follows since for known fixed order we can calculate the optimal expected gain (as shown in [22], this reduces in their setting to the line graph). Therefore, the optimal that has the *extra* power of knowing the sequence that follows would have run the algorithm of Boodaghians et al. [22], and since each permutation is equally likely to appear, the expected cost is averaging over all $n!$ optimal solutions.

Finally, we can formulate a linear program to describe our problem. Let again S_n be the set of all possible permutations with n boxes. Next, we define the two variables $x_{\pi(i)}$ and $z_{\pi(i)}$: Let $x_{\pi(i)} = 1$ if for permutation $\pi \in S_n$ the i-th box of this permutation is opened, and $z_{\pi(i)} = 1$ if for permutation π the i-th box is chosen.

$$
\begin{aligned}
\text{maximize} \quad & \frac{1}{n!}\sum_{\pi\in S_n}\sum_{i=1}^{n} -x_{\pi(i)}c_{\pi(i)} + z_{\pi(i)}v_{\pi(i)} & \\
\text{subject to} \quad & \sum_{i=1}^{n} z_{\pi(i)} = 1 & \forall \pi \in S_n, \\
& z_{\pi(i)} \le x_{\pi(i)} & \forall \pi \in S_n, i \in [n], \\
& x_{\pi(i-1)} \ge x_{\pi(i)} & \forall \pi \in S_n, i \in [n], \\
& x_{\pi(i)}, z_{\pi(i)} \in [0,1] & \forall \pi \in S_n, i \in [n].
\end{aligned}
$$

The first constraint of the LP says that we can collect the reward from one box in total for every permutation π. The second constraint indicates that we cannot collect the reward of a box if we haven't opened it first (unlike the non-obligatory inspection model where there is no such constraint). The third constraint expresses the restriction of not being allowed to skip boxes; we cannot have an unopened box with a smaller index than an opened one. With the last constraint we relax the problem to capture also randomized algorithms that probabilistically open a box and collect its reward. However, the opposite direction, i.e., translating a non-integer solution of the LP into a randomized strategy is not very clear. Assuming that we have solved the exponential size LP and obtained the (optimal) values for the decision variables $z_{\pi(i)}$, it is not straightforward how to obtain a randomized algorithm from them (e.g., by interpreting them as probabilities and aggregating them in a clever way). Note that the way the LP is written, it is implied that the decision-maker must enter the game and open at least one box, but we can easily adapt it to make this optional by offering an outside option.

In the following, we study algorithms that we intuitively expect to perform well against the optimal online policy. However, we show that for carefully constructed instances they can be arbitrarily bad, indicating that we have to come up with very different policies so that we achieve good approximation for any instance.

5.2.2 Impossibility results for specific algorithms

For the moment, we do not restrict ourselves to only algorithms with polynomial runtime. Instead, we want to consider algorithms that are likely to perform well for any instance, ideally achieving expected gain very close to either the solution of the dynamic program or the upper bound in Eq. (5.2). Following, we describe two candidate algorithms that we term the *majority* algorithm and the *averaging* algorithm. We show that, although they seem natural, and adaptively decide based on the current gain and the future "value" of the instance, they fail to provide any guarantee for some instances. More concretely, we construct a family of instances and show that for specific choice of the parameters within this family the optimal online policy gets positive expected gain, while the expected gain of our algorithms is negative.

We begin by stating the algorithms; we explain the majority one in Algorithm 1 in detail, and the averaging algorithm is a simple adaptation in the decision phase. More

Algorithm 1: The majority algorithm ALG_{maj}

```
Input  : Set of boxes B with their distributions F_i and costs c_i, random
         permutation σ.
Output: The gain for the permutation σ.
while B is not empty do
    Observe the next uninspected box b_i in the sequence; Set B ← B \ {b_i};
    Initialize majority sum M = 0;
    for all independent uniform random permutations π of boxes in B do
        Fix the permutation π' ← {b_i} ∪ π;
        Calculate the optimal gain OPT_fixed of π' using the algorithm in [22];
        if OPT_fixed > 0 then
            M ← M + 1;  /* Add one point from this permutation to the
             sum since it gives positive gain */
        end
        else
            M ← M − 1;
        end
    end
    if M > 0 then
        Open b_i and update the gain with the realized reward v_i and the cost c_i;
        continue;
    end
    else
        break;
    end
end
```

specifically, in the majority algorithm, for each permutation π we add one point if the maximum gain from this fixed permutation π is positive and we remove one point if it is not. In the averaging algorithm, we take the maximum gain of each fixed permutation and we average over them (since each permutation is equally likely to appear). The algorithm opens the next box if the average expected gain is positive, and stops and returns the current gain otherwise. Remember that the boxes are presented to us in an online fashion. In particular, at each point we can observe the next box of the sequence (namely the distribution of its reward and its inspection cost), and decide whether to open it, or stop without opening it and collect the maximum reward so far.

The hard instance Next, consider both algorithms in the following instance. For simplicity, it consists of deterministic rewards only, which makes our analysis easier. Consider three different sets of boxes with different cardinalities.

- There are two boxes of type A. For this type of boxes, there is no inspection cost (i.e., $c_A = 0$) and the reward is $v_A = t$, where t is a big positive constant.

- There is one box of type B. For this type of boxes, there is a huge inspection cost $c_B \to \infty$ and very low reward; let us assume here that $v_B = 0$. Type B boxes just serve for stopping the search as we will never open them.

- Finally, we have a large number m_c of type C boxes. For inspecting these boxes we have to pay inspection cost $c_C = 1$ and we get value $v_C = 0$.

It is clear that when the first box is of type A then both the optimal online policy and the proposed algorithms would inspect the box and stop, resulting in a gain of t. It is also clear that when the first box is the one of type B, then we never inspect and we receive zero gain. Thus, the interesting case is when the first box of the sequence is of type C. We analyze this case and show why the majority algorithm does not work. Once we have proved this, the result for the averaging algorithm easily follows. We start with an important observation for the instance we just defined when the random sequence starts with a box of type C.

Lemma 35. *Let X be the total number of boxes of type C before the first box of type A or B. The majority algorithm will open all X boxes of type C if and only if $Pr[X \leq t-1] \geq 3/4$.*

Proof. Since the costs and values of type A and type C boxes are all integers, for a fixed permutation we add a point to the majority sum if the optimal gain is at least 1. When starting with a box of type C, we achieve positive optimal gain under the following two conditions: (1) X is at most $t-1$; and (2) the first box that is not of type C must be of type A. In order to continue and start opening the type C boxes, this has to be the case in at least half of the permutations that start with a type C box. Let E be the event that a type A box is presented to us before the type B box. For opening the type C boxes we need to satisfy $\Pr[(X \leq t-1) \cap E] \geq 1/2$. Notice that

$$\Pr[(X \leq t-1) \cap E] = \Pr[(X \leq t-1) \mid E] \cdot \Pr[E] = \frac{2}{3} \Pr[X \leq t-1],$$

where the second equality holds because the two events are independent and $\Pr[E] = \frac{2}{3}$ because there are two type A and one type B boxes, which are presented in a uniform random order. Since we want $\Pr[(X \leq t-1) \cap E] \geq 1/2$ to continue with the type C boxes, the condition $\Pr[X \leq t-1] \geq 3/4$ must be satisfied. $\square$

With this at hand, we can try to fix the two parameters c_m and t so that the condition from the previous lemma is satisfied, and the majority algorithm gets negative expected gain.

Theorem 17. *For $m_c = 1001$ and $t = 372$ the majority algorithm gets negative expected gain. On the other hand, the expected gain of the optimal online policy is positive.*

Proof. Let us first write the probability that the boxes of type C in the beginning are at most $t-1$ as a function of the parameters. Observe that $\Pr[X = 1] = \frac{3}{m_c+2}$, because we start the sequence with a type C box (which is a given) and the second box is of type A or B, which happens with probability $\frac{3}{m_c+2}$. Similarly, we can calculate the probability for other values of X. For example,

$$\Pr[X=2] = \frac{m_c-1}{m_c+2}\cdot\frac{3}{m_c+1}, \quad \Pr[X=4] = \frac{(m_c-2)\cdot(m_c-3)\cdot 3}{(m_c+2)\cdot(m_c+1)\cdot m_c}, \cdots$$

Thus, we can write it down for any value of t as follows:

$$\Pr[X=t-1] = \frac{(m_c+2-(t-1))\cdot(m_c+1-(t-1))\cdot 3}{(m_c+2)\cdot(m_c+1)\cdot m_c}\ .$$

Now we can calculate the desired probability by summing up the previous values:

$$\Pr[X\le t-1] = \sum_{k=1}^{t-1}\Pr[X=k] = \frac{3}{(m_c+2)\cdot(m_c+1)\cdot m_c}\sum_{i=m_c+2-(t-1)}^{m_c+1}(i-1)\cdot i\ .$$

The expected gain of the majority algorithm is simply

$$\mathbb{E}[\mathrm{ALG_{maj}}] = \frac{2}{m_c+3}\cdot t + \frac{m_c}{m_c+3}\cdot\left[\frac{2}{3}\cdot t - \left(\frac{m_c-1}{4}+1\right)\right]\cdot \mathbb{1}_{\Pr[X\le t-1]\ge 3/4}\ ,$$

since when the first box is of type C, the majority algorithm will open the type C boxes only if the condition from Lemma 35 is satisfied. In this case, it will encounter in expectation some boxes of type C, and then with probability 2/3 a box of type A. If we choose $t=372$ and $m_c=1004$ we get that $\Pr[X\le t-1] > 3/4$ and $\mathbb{E}\left[\mathrm{ALG_{maj}}\right] < 0$. We denote by OPT the optimal policy. Since $\mathbb{E}\left[\mathrm{OPT} \mid \text{first box is of type C and we open it}\right] < 0$, the optimal online policy will proceed in opening boxes only when the first box is of type A, collect its reward and stop. Thus, $\mathbb{E}\left[\mathrm{OPT}\right] = \frac{2}{m_c+3}\cdot t > 0$, and the proof is complete. □

Remark. Even if we assume that all the rewards and the reservation values of the boxes are positive, we still cannot circument the arbitrarily bad performance of the majority algorithm. We create a new instance in which the following things change compared to the previous one: The type C boxes have now $v_C = 1$ and $c_C = 1-\varepsilon$, and their reservation value becomes $\sigma_C = \varepsilon$, for some small $\varepsilon > 0$. Moreover, we remove the type B box since it has negative reservation value. Following the same reasoning as before, we come up again with two expressions that need to be satisfied simultaneously so that the expected gain of the majority algorithm is negative. We can show again that there is a range of values for m_c and t, for which the majority algorithm achieves negative expected gain, while the expected gain of the optimal online policy is positive, thus ensuring again an arbitrarily large gap.

Now we can extend the negative result above to the averaging algorithm as well by comparing the decisions of the two algorithms on each permutation.

Proposition 8. *For the expected gain of the averaging algorithm $\mathcal{A}$ on the same instance, it holds that $\mathbb{E}[\mathcal{A}] \le \mathbb{E}[\mathrm{ALG}_{maj}]$.*

Proof. The interesting case again is the one in which the very first box is of type C. For the other two cases, the two algorithms obviously make the same decision. We show that if the majority algorithm decides to open the type C boxes (which for some choices of c_m and t will result in negative expected gain), so will the averaging algorithm. Recall that for each fixed permutation we add one point to the majority sum when its gain is positive, and we remove one otherwise. If the majority sum M is positive, we open the type C boxes. Denote by Σ_{avg} the corresponding sum of the averaging algorithm. Again, if $\Sigma_{\text{avg}} > 0$ we inspect the next box, otherwise we stop.

Let us assume that we want to decide whether to open the first type C box. Consider a fixed permutation σ from the set of possible permutations. If its optimal gain is positive, we add 1 to M and $\text{OPT}_{\text{fixed}}(\sigma)$ to Σ_{avg}. By construction of the instance $\text{OPT}_{\text{fixed}}(\sigma) \geq 1$ since any positive gain is at least 1. If the optimal gain is negative, ALG_{maj} subtracts 1 from M, and $\mathcal{A}$ subtracts $\text{OPT}_{\text{fixed}}(\sigma)$ from $\Sigma_{\text{avg}} > 0$, where $\text{OPT}_{\text{fixed}}(\sigma) = -1$. The optimal algorithm for the fixed permutation will minimize the negative gain, since we want the optimal solution given that we open at least the first box of this permutation. Therefore, for all σ whose optimal gain is negative we have that $\text{OPT}_{\text{fixed}}(\sigma) = -1$.

From the above we can conclude that $\text{OPT}_{\text{fixed}}(\sigma) \geq M$, which means that whenever the majority algorithm decides to open the type C boxes until encountering an A or B, the averaging algorithm will make the same decision, resulting in a negative expected gain as well. □

5.3 Exogenous ordering with skipping

In this section, we study both AOPB and ROPB with skipping. In this variant, we can choose to not inspect a box that is presented to us and let it go forever. Thus, the policies we design here have to make two decisions: (1) when to open a box; and (2) when to stop. To start, we present a very useful reduction from the Pandora's box with skipping and commitment to the prophet inequality. The reduction holds for *any* order.

5.3.1 From Pandora's box with commitment to prophet inequalities

We show an approximation-preserving reduction to prophet inequalities. More specifically, we can transform algorithms that have provably good approximation to the prophet's reward in the adversarial, random, free, or any other order, to algorithms in Pandora's box with commitment (keeping the same assumption on the order) that have also good approximation to the expected gain of Weitzman's policy. The approximation guarantees apply to Pandora's box without commitment as well, since by dropping the commitment constraint we can only do better. In any case, we can always use algorithms that are designed for the setting with commitment when there is no commitment constraint, although they will probably be suboptimal. We now state the main result that we want to prove.

Theorem 18. *Given an α-approximation* threshold-based *algorithm for the prophet inequality problem, we can construct an α-approximation algorithm for the Pandora's box problem with skipping and commitment using the same set of thresholds.*

The first step is the following very important observation. This is one of the main steps for an alternative proof of the optimality of Weitzman's policy, discovered by Kleinberg et al. [106]. This proof is also the building block of the reduction.

Lemma 36 (Lemma 2 in [106])**.** *Let x_i be the indicator random variable that takes value 1 if we collect the reward from box b_i and z_i be the indicator random variable that takes value 1 if we open b_i. Then it holds that*

$$\mathbb{E}_{v_i \sim F_i}[x_i v_i - z_i c_i] \leq \mathbb{E}_{v_i \sim F_i}[x_i \min\{v_i, \sigma_i\}]. \tag{5.3}$$

It is satisfied with equality if and only if for each box b_i with $v_i > \sigma_i$, it holds that $x_i = z_i$.

Next, we relate Lemma 36 to the performance of Weitzman's optimal policy in the Pandora's box problem.

Lemma 37 (Merge of Corollary 1 and Theorem 1 in [106])**.** *No algorithm can achieve a better expected gain than $\mathbb{E}\left[\max_{i\in[n]} \min\{v_i, \sigma_i\}\right]$. Furthermore, the expected gain of Weitzman's policy is exactly this upper bound.*

Proof. By summing over the boxes in Eq. (5.3) and the fact that we collect the reward of only one box we get

$$\sum_{i\in[n]} \mathbb{E}[x_i v_i - z_i c_i] \leq \sum_{i\in[n]} \mathbb{E}[x_i \min\{v_i, \sigma_i\}] \leq \mathbb{E}\left[\max_{i\in[n]} \min\{v_i, \sigma_i\}\right], \tag{5.4}$$

where the expectation is over the independent draws from the distributions and possibly over the randomness of the exploration order of the boxes. Recall that Weitzman's policy opens the boxes in decreasing order of their reservation values, and collects the reward of the first box b_i for which v_i exceeds all the reservation values of the remaining uninspected boxes. This procedure by definition always collects the reward of the box b_i with the largest $\min\{v_i, \sigma_i\}$ and it also satisfies the property for which Eq. (5.3) holds with equality. Therefore, Weitzman's policy satisfies both inequalities of Eq. (5.4) with equality. □

Now we define a family of threshold-based policies for the Pandora's box problem with commitment. They can be applied regardless of the assumptions on the ordering of the boxes.

Definition 7 (A family of threshold-based algorithms)**.** Fix a sequence of thresholds $\tau_1, \tau_2, \ldots, \tau_n$. The algorithm inspects a box b_i if $\sigma_i > \tau_i$. It collects the reward of the first inspected box for which it also holds that $v_i > \tau_i$.

From the proof of the main theorem of this section it will become clear that these are the types of policies that arise when applying the algorithms from prophet inequalities in our problem. Notice that such algorithms satisfy Eq. (5.3) with equality, since by definition if $v_i > \sigma_i$ then x_i and z_i always agree in their values. In particular, there are two scenarios: either a box b_i with $v_i > \sigma_i$ is reached, inspected and selected (in which case $x_i = z_i = 1$) or the algorithm stops before reaching that box (in which case $x_i = z_i = 0$).

Now we can put everything together and prove Theorem 18.

Proof of Theorem 18. Consider a prophet inequality problem for an instance with distributions $G_1, G_2, \cdots, G_n$, where $\min\{v_i, \sigma_i\} \sim G_i$ for all i. Run a threshold-based algorithm with some α-approximation guarantee. This means that the expected value of the first random variable which exceeds its threshold will be at least an α-fraction of $\mathbb{E}\left[\max_{i\in[n]} \min\{v_i, \sigma_i\}\right]$ which is the expected reward of the prophet. From Lemma 37 we know that this is also the expected gain of Weitzman's optimal policy. Recall that the threshold-based algorithms for the Pandora's box problem from Definition 7 satisfy Eq. (5.3) with equality. Moreover, by this definition we observe that these algorithms will pick the first box b_i for which $\min\{v_i, \sigma_i\} > \tau_i$. Thus, we will always pick the same box in both problems, if we run their threshold-based algorithms with the same set of thresholds. Combining all the above, we conclude that when we have an α-approximation threshold-based algorithm for a prophet inequality problem, we get

$$\begin{aligned}\sum_{i\in[n]} \mathbb{E}_{v_i\sim F_i}[x_i v_i - z_i c_i] &= \sum_{i\in[n]} \mathbb{E}_{v_i\sim F_i}[x_i \min\{v_i, \sigma_i\}] \\ &\geq \alpha \cdot \mathbb{E}\left[\max_{i\in[n]} \min\{v_i, \sigma_i\}\right] \\ &= \alpha \cdot \mathbb{E}[\text{WEITZ}] \ ,\end{aligned}$$

where by $\mathbb{E}[\text{WEITZ}]$ we denote the expected gain of Weitzman's optimal policy. □

Finally, a couple of remarks are now in order. Note that the reduction holds for any assumption on the order of the boxes. There just has to exist a threshold-based algorithm whose expected reward guarantees an approximation to the expected reward of a prophet for the order we assumed.

Furthermore, the reduction holds for randomized thresholds. It could still hold for randomization over the selection of a box which exceeds its threshold as well; the only condition that needs to be always satisfied is that if $v_i > \sigma_i$ then $x_i = z_i$, which is always the case for the algorithms of Definition 7.

5.3.2 Adversarial order

In this section, we show that, perhaps surprisingly, for AOPB with skipping an approximation of 1/2 to Weitzman's optimal policy is the best possible guarantee, regardless of whether we impose the commitment constraint. First, we state the lower bound that comes directly from Theorem 18.

Corollary 4. *A single-threshold algorithm for AOPB with or without commitment guarantees at least half of the expected gain of Weitzman's policy.*

The result follows from the well-known bound for the classic prophet inequality; setting a single thresold guarantees at least half of the expected reward of the prophet (see Samuel-Cahn [136] and Kleinberg and Weinberg [105] for more details on how such a threshold can be calculated). The threshold-based algorithm is designed for a setting with commitment, but the result carries through when there is no commitment as well. Although one might expect that a different algorithm could break the barrier of $1/2$ when we can collect the reward of any opened box, the following result shows that this is not the case.

Theorem 19. *For AOPB with or without commitment, no algorithm can guarantee more than $1/2$ of the expected gain of Weitzman's optimal policy. Thus, the bound of $1/2$ is tight.*

Proof. Denote by FIX the gain of an optimal algorithm for the fixed order case. Consider the following instance with 2 boxes.
Box 1: $c_1 = 1 - 1/x$ and $v_1 = 1$.
Box 2: $c_2 = 1$ and $v_2 = \begin{cases} 0, & \text{w.p. } 1 - 1/x \\ x + \varepsilon, & \text{w.p. } 1/x \end{cases}$.

We will set x and ε such that $\sigma_2 > \sigma_1$. Then the optimal opening order is b_2, b_1 and Weitzman's policy gets expected gain

$$\mathbb{E}[\text{WEITZ}] = -1 + \frac{1}{x}(x+\varepsilon) + \left(1 - \frac{1}{x}\right)\left(-1 + \frac{1}{x} + 1\right) = \frac{x + x\varepsilon - 1}{x^2} .$$

The adversarial order is $b_1 b_2$ and any algorithm that observes this order has 3 possibilities; open only b_1 and stop, skip b_1 and open b_2, or open both. An optimal algorithm gets the maximum over these expected gains:

$$\mathbb{E}[\text{FIX}] = \max\left\{\frac{1}{x}, \frac{\varepsilon}{x}, -1 + \frac{1}{x} - 1 + \frac{1}{x}(x+\varepsilon) + \left(1 - \frac{1}{x}\right)\right\} = \max\left\{\frac{\varepsilon}{x}, \frac{1}{x}, \frac{\varepsilon}{x}\right\} .$$

By setting $x \to \infty$ and $\varepsilon \to 1$ we get $\varepsilon = \sigma_2 > \sigma_1 = 1/x$, so our condition for the reservation values is satisfied. Moreover, taking the limit we have that

$$\frac{\mathbb{E}[\text{FIX}]}{\mathbb{E}[\text{WEITZ}]} = \lim_{(x,\varepsilon)\to(\infty,1)} \frac{\max\left\{\frac{\varepsilon}{x}, \frac{1}{x}, \frac{\varepsilon}{x}\right\}}{\frac{x+x\varepsilon-1}{x^2}} = \frac{1}{2} ,$$

and the proof is complete.

□

5.3.3 Random order

In this section, we provide a lower bound for ROPB with commitment (which is also a lower bound for ROPB without commitment) and an upper bound for ROPB without commitment (which applies to ROPB with commitment as well). For completeness, we formulate again a dynamic program that solves ROPB with skipping and without commitment optimally. We use the same notation as we did in the random order without skipping (see Section 5.2.1). This dynamic program, apart from dealing with all possible permutations, has to also take into account all the combinations that arise from skipping or not skipping each arriving box. We denote by R the set of unopened boxes, and by v the maximum reward so far. Then the DP becomes

$$\mathrm{OPT}(R, v, i) = \max\left(-c_i + \mathbb{E}_{j\in R\setminus\{i\}}\mathrm{OPT}\left(R \setminus \{i\}, \max(v, \mathbb{E}v_i), j\right), \mathbb{E}_{j\in R\setminus\{i\}}\mathrm{OPT}(R \setminus \{i\}, v, j)\right)$$

and the optimal solution is $\mathbb{E}\left[\mathrm{OPT}([n], 0, i)\right]$.

Our final goal is to design good polynomial-time algorithms for ROPB without commitment. Again, from Theorem 18 we know that we can apply the state-of-the-art threshold-based policies for the prophet secretary problem and directly obtain the same approximation guarantee in ROPB without commitment when compared to the expected gain of Weitzman's policy. Correa et al. [56] present an algorithm that is a decreasing collection of randomized thresholds and guarantees a 0.669-fraction of the prophet's expected reward. This is currently the best known bound for the prophet secretary problem; they also show that no algorithm can obtain more than $\sqrt{3} - 1$. This upper bound just implies that when we restrict ourselves to policies with commitment, we cannot hope for a better guarantee than $\sqrt{3} - 1$ through the reduction to the prophet secretary problem; we need different techinques to break this barrier.

Corollary 5. *For ROPB with or without commitment, we can design an algorithm that is a decreasing collection of thresholds $\tau_1, \tau_2 \ldots, \tau_n$ and guarantees at least a 0.669-fraction of the expected gain of Weitzman's optimal policy.*

We complement this result with a (non-tight) upper bound that applies to any algorithm. The construction of the hard instance is inspired by Correa et al. [56, Section 5], but requires quite different ideas and observations to adapt it to a setting with costs and the possibility of collecting a previously observed reward. Before proceeding to the hardness result, note that an almost immediate upper bound of $\frac{3}{4}$ follows from Theorem 19. Consider again the same instance with $x \to \infty$ and $\varepsilon \to 1$, so that the optimal opening order is $b_2 b_1$. Since the order is random, the two possible permutations $b_2 b_1$ and $b_1 b_2$ appear each with probability 1/2. The optimal algorithm for ROPB gets the optimal gain in each of the permutations; this is WEITZ and FIX, respectively, as calculated in Theorem 19. In other words, half the time it gets the same as Weitzman's policy, and half the time it gets half of it, leading to an upper bound of $\frac{3}{4}$ overall. Next, we present our slightly improved upper bound.

Theorem 20. *For ROPB with or without commitment, no algorithm can obtain more than a $(\sqrt{3} - 1)$-fraction of the expected gain of Weitzman's optimal policy.*

Proof. Let us focus here on the ROPB without commitment, which is more general than the ROPB with commitment. Consider the following instance with $n+1$ boxes.

- The first n boxes with rewards $v_1, v_2, \dots, v_n \sim \begin{cases} n, & \text{w.p. } \frac{1}{n^2} \\ 0, & \text{w.p. } 1-\frac{1}{n^2} \end{cases}$ and zero costs, i.e., $c_1 = c_2 = \dots = c_n = 0$.
- The last box b_{n+1} has reward $v_{n+1} = \alpha_n$ and cost $c_{n+1} = c_{\alpha_n}$. When clear of context, we will just denote them α and c_α, respectively, knowing that both of those values depend on n.

First, let us describe what Weitzman's optimal policy will do. For that, we need to make some assumption about α and c_α, which will determine if b_{n+1} will be inspected first or last, since all other boxes have the same reservation value. Choose α and c_α such that $\alpha - c_\alpha := \Delta \in O(1)$. It is then clear that b_{n+1} will be inspected last (if we reach it), for an expected gain of

$$\mathbb{E}[\text{WEITZ}] = \left(1 - \left(1 - \frac{1}{n^2}\right)^n\right) \cdot n + \left(1 - \frac{1}{n^2}\right)^n \cdot (\alpha - c_\alpha).$$

Next, let us consider algorithms for the random order case. Denote by RAND the gain of an algorithm on a specific realization. We assume from now on that $\alpha_n < n$. With this assumption, the only decision of an algorithm is whether to inspect the deterministic (once n is fixed) box and continue inspecting the free boxes, or skip that box and continue. Intuitively, if there are a lot of boxes remaining, it is more likely that at least one of them will be realized, resulting in RAND $= n$. If few boxes are left uninspected we prefer to open the deterministic box so that we obtain at least RAND $\in O(1)$ if all the remaining rewards turn out to be 0. Let κ_n denote the index starting at which we start inspecting b_{n+1}. In particular, this is the first index for which opening the box obtains a larger expected future gain than skipping.

Assume that for a realized permutation we have $n - \kappa_n$ boxes remaining (meaning the first $\kappa_n - 1$ boxes had reward 0), and we have to decide whether to open box b_{n+1} at position κ_n. The expected future gain by skipping it is

$$\mathcal{R}_{\text{skip}} := \left(1 - \left(1 - \frac{1}{n^2}\right)^{n+1-\kappa_n}\right) \cdot n$$

and by opening it is

$$\mathcal{R}_{\text{open}} := \left(1 - \left(1 - \frac{1}{n^2}\right)^{n+1-\kappa_n}\right) \cdot (n - c_\alpha) + \left(1 - \frac{1}{n^2}\right)^{n+1-\kappa_n} (\alpha - c_\alpha).$$

Solving for the smallest such index κ_n^* for which we would decide to inspect, we get

$$\mathcal{R}_{\text{open}} \geq \mathcal{R}_{\text{skip}} \Leftrightarrow \kappa_n^* \geq \frac{\log\left(\frac{d^{n+1}\alpha}{c_\alpha}\right)}{\log d},$$

where $d_n = 1 - \frac{1}{n^2}$ (we will again often just use d and omit the subscript). Thus, κ_n^* will be the first index that satisfies the condition, i.e., $\kappa_n^* = \left\lceil \frac{\log\left(\frac{d^n \alpha}{c_\alpha}\right)}{\log d} \right\rceil$.

Now we are ready to calculate the expected gain of an optimal algorithm. We have that

$$\begin{aligned}
\mathbb{E}[\text{RAND}] &= \mathbb{E}\left[\text{RAND} \mid \pi(n+1) < \kappa_n^*\right] \Pr\left[\pi(n+1) < \kappa_n^*\right] + \frac{1}{n+1} \sum_{i=\kappa_n^*}^{n+1} \mathbb{E}\left[\text{RAND} \mid \pi(n+1) = i\right] \\
&= \frac{\kappa_n^* - 1}{n+1} \left(1 - d^n\right) n + \frac{1}{n+1} \sum_{i=\kappa_n^*}^{n+1} (n - c_\alpha)\left(1 - d^{n-i+1}\right) d^{i-1} + n\left(1 - d^{i-1}\right) + (\alpha - c_\alpha) d^n \\
&= \frac{\kappa_n^* - 1}{n+1} \left(1 - d^n\right) n + \frac{1}{n+1} \sum_{i=\kappa_n^*}^{n+1} \left(1 - d^n\right) n + d^n \alpha - d^{i-1} c_\alpha.
\end{aligned}$$

Remember that the assumptions that we made about α_n and c_{α_n} are that (1) their difference is some constant, and (2) $\alpha_n < n$. Then by choosing $\alpha_n = n - \epsilon$ for some very small $\epsilon > 0$ and $c_{\alpha_n} = n - \epsilon - \sqrt{3} + 1$, we have that $\Delta = \sqrt{3} - 1$ and taking the desired ratio as n grows and $\epsilon \to 0$ we get

$$\frac{\mathbb{E}[\text{RAND}]}{\mathbb{E}[\text{WEITZ}]} = \lim_{n \to \infty} \frac{\frac{\kappa_n^* - 1}{n+1} \left(1 - d^n\right) n + \frac{1}{n+1} \sum_{i=\kappa_n^*}^{n+1} \left(1 - d^n\right) n + d^n \alpha - d^{i-1} c_\alpha}{\left(1 - d^n\right) \cdot n + d^n \cdot (\alpha - c_\alpha)} = \sqrt{3} - 1 \approx 0.732,$$

and the proof is complete.

□

5.4 Current and future directions

We conclude with questions that directly arise from the previous results and discuss some more general future directions.

Hardness of AOPB and ROPB with skipping An equally well-motivated benchmark with the expected gain of Weitzman's optimal policy is the expected gain of the optimal online policy for the same exploration order. In particular, is it easy to compute the optimal online policy for AOPB or ROPB with skipping? If the problem turns out to be hard, can we design simple policies with good approximation guarantees for this benchmark? This direction is in line with the work that studies the approximability of the online optimum for other online selection problems (see also Section 5.1.2 where some of this related work is mentioned).

Different objective functions The current chapter tries to shed light on the impact of ordering in fundamental search processes, focusing on the common single-selection objective of the Pandora's box problem. With this work as a starting point, we could also consider more general objective functions. Examples include collecting the reward of up to k items or having a constraint on the number of boxes that we are allowed to open. From a mechanism design perspective, we can say that the decision-maker's valuation function is unit-demand (i.e., it selects at most one item, minus the costs paid for the inspected items). It is worth studying more general classes of valuation functions, such as submodular and XOS.

Showing separation between settings with and without commitment We saw that for the AOPB with skipping the commitment constraint does not play any role; a simple algorithm designed for the setting with commitment turns out to be the best possible when there is no commitment as well. In the ROPB with skipping there is still a gap in the lower and upper bound; this leaves open the question of whether we can design an algorithm for the ROPB without commitment which beats the threshold-based algorithm. Nevertheless, even if that is possible, the improvement in the guarantee will be small, which implies that this algorithm still performs very well even when we drop the commitment constraint. If we slightly modify other assumptions of the problem (e.g. the objective function, the exploration order or the independence of the distributions) does the commitment constraint crucially affect the structure of the optimal solutions? Or is it maybe that in most cases dropping the commitment constraint does not give much additional power to the policies we can design?

Non-obligatory inspection In Section 5.1.2 we mentioned another very interesting extension of the classic Pandora's box problem; the one in which we can select a box without necessarily inspecting it first (that is, only with the knowledge of the distribution of its reward). Note that the Pandora's box problem with non-obligatory inspection has not been fully solved even in the case that we are allowed to choose the order of exploration. We know that most likely it is difficult to compute the optimal policy, and we are aware of algorithms that achieve a good constant-factor approximation [20]. We do not have any general hardness result for this model, so we do not know if these factors are tight as well. In any case, it is an intriguing direction to investigate the non-obligatory inspection assumption in the adversarial and the random order case. The benchmarks could be either the optimal online policy, or a novel upper bound to the optimal policy in the free order case, which would be of independent interest.

Bibliography

[1] M. Abolhassani, S. Ehsani, H. Esfandiari, M. HajiAghayi, R. Kleinberg, and B. Lucier. Beating 1-1/e for ordered prophets. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, jun 2017. doi: 10.1145/3055399.3055479.

[2] P. Allaart and J. A. Islas. A sharp lower bound for choosing the maximum of an independent sequence. *Journal of Applied Probability*, 53:1041–1051, 2015.

[3] S. Alpern and V. Baston. The secretary problem with a selection committee: Do conformist committees hire better secretaries? *Management Science*, 63(4): 1184–1197, 2017. doi: 10.1287/mnsc.2015.2377.

[4] N. Anari, R. Niazadeh, A. Saberi, and A. Shameli. Nearly optimal pricing algorithms for production constrained and laminar bayesian selection. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, 2019. doi: 10.1145/3328526.3329652.

[5] A. Angelovski and W. Güth. When to stop — a cardinal secretary search experiment. *Journal of Mathematical Psychology*, 98:102425, 2020. doi: 10.1016/j.jmp.2020.102425.

[6] A. Aouad, J. Ji, and Y. Shaposhnik. The pandora's box problem with sequential inspections. *SSRN Electronic Journal*, 2021. doi: 10.2139/ssrn.3726167. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3726167`.

[7] C. Attias, R. Krauthgamer, R. Levi, and Y. Shaposhnik. Stochastic selection problems with testing. *SSRN Electronic Journal*, 2017. doi: 10.2139/ssrn.3076956. URL `https://doi.org/10.2139%2Fssrn.3076956`.

[8] P. Azar and S. Micali. Optimal parametric auctions. Technical Report MIT-CSAIL-TR-2012-015, MIT CSAIL, Oct. 2012. URL `http://hdl.handle.net/1721.1/70556`.

[9] P. Azar, C. Daskalakis, S. Micali, and S. M. Weinberg. Optimal and efficient parametric auctions. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 596–604, 2013. doi: 10.1137/1.9781611973105.43.

[10] P. D. Azar and S. Micali. Parametric digital auctions. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 231–232, 2013. doi: 10.1145/2422436.2422464.

[11] P. D. Azar, R. Kleinberg, and S. M. Weinberg. Prophet inequalities with limited information. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2014. doi: 10.1137/1.9781611973402.100.

[12] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 16–28. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-74208-1_2.

[13] M. Babaioff, N. Immorlica, B. Lucier, and S. M. Weinberg. A simple and approximately optimal mechanism for an additive buyer. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 21–30, 2014. doi: 10.1109/FOCS.2014.11.

[14] C. Bandi and D. Bertsimas. Optimal design for multi-item auctions: A robust optimization approach. *Mathematics of Operations Research*, 39(4):1012–1038, 2014. doi: 10.1287/moor.2014.0645.

[15] R. E. Barlow and F. Proschan. Failure distributions. In *Mathematical Theory of Reliability*, chapter 2. Society for Industrial and Applied Mathematics, 1996. doi: 10.1137/1.9781611971194.

[16] C. Baumann, H. Singmann, S. J. Gershman, and B. von Helversen. A linear threshold model for optimal stopping behavior. *Proceedings of the National Academy of Sciences*, 117(23):12750–12755, 2020. doi: 10.1073/pnas.2002312117.

[17] C. Bechtel and S. Dughmi. Delegated stochastic probing. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185 of *LIPIcs*, pages 37:1–37:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.ITCS.2021.37.

[18] X. Bei, N. Gravin, P. Lu, and Z. G. Tang. Correlation-robust analysis of single item auction. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 193–208, 2019. doi: 10.1137/1.9781611975482.13.

[19] D. Bergemann and K. Schlag. Robust monopoly pricing. *Journal of Economic Theory*, 146(6):2527 – 2543, 2011. doi: 10.1016/j.jet.2011.10.018.

[20] H. Beyhaghi and R. Kleinberg. Pandora's problem with nonobligatory inspection. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 131–132. Association for Computing Machinery, 2019. doi: 10.1145/3328526.3329626.

[21] H. Beyhaghi, N. Golrezaei, R. P. Leme, M. Pál, and B. Sivan. Improved revenue bounds for posted-price and second-price mechanisms. *Operations Research*, 69 (6):1805–1822, 2021. doi: 10.1287/opre.2021.2121.

[22] S. Boodaghians, F. Fusco, P. Lazos, and S. Leonardi. Pandora's box problem with order constraints. In *EC '20: The 21st ACM Conference on Economics and Computation*, pages 439–458. ACM, 2020. doi: 10.1145/3391403.3399501.

[23] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[24] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[25] D. Bradac, A. Gupta, S. Singla, and G. Zuzic. Robust algorithms for the secretary problem. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*, volume 151 of *LIPIcs*, pages 32:1–32:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.ITCS.2020.32.

[26] B. Brooks and S. Du. Optimal auction design with common values: An informationally robust approach. *Econometrica*, 89(3):1313–1360, 2021. doi: 10.3982/ecta16297.

[27] F. T. Bruss. A unified approach to a class of best choice problems with an unknown number of options. *The Annals of Probability*, 12(3), 1984. doi: 10.1214/aop/1176993237.

[28] F. T. Bruss. Sum the odds to one and stop. *The Annals of Probability*, 28(3), 2000. doi: 10.1214/aop/1019160340.

[29] J. Bulow and P. Klemperer. Auctions versus negotiations. *The American Economic Review*, 86(1):180–194, 1996. URL `http://www.jstor.org/stable/2118262`.

[30] Y. Cai and C. Daskalakis. Learning multi-item auctions with (or without) samples. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 516–527, Oct 2017. doi: 10.1109/FOCS.2017.54.

[31] Y. Cai and M. Zhao. Simple mechanisms for subadditive buyers via duality. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 170–183, 2017. doi: 10.1145/3055399.3055465.

[32] Y. Cai, N. R. Devanur, and S. M. Weinberg. A duality based unified approach to Bayesian mechanism design. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 926–939. ACM Press, 2016. doi: 10.1145/2897518.2897645.

[33] G. Campbell and S. M. Samuels. Choosing the best of the current crop. *Advances in Applied Probability*, 13(3):510–532, 1981. doi: 10.2307/1426783.

[34] V. Carrasco, V. F. Luz, N. Kos, M. Messner, P. Monteiro, and H. Moreira. Optimal selling mechanisms under moment conditions. *J. Econ. Theory*, 177:245–279, 2018. doi: 10.1016/j.jet.2018.05.005.

[35] G. Carroll. Robustness and separation in multidimensional screening. *Econometrica*, 85(2):453–488, 2017. doi: 10.3982/ECTA14165.

[36] G. Carroll. Robustness in mechanism design and contracting. *Annual Review of Economics*, 11(1):139–166, 2019. doi: 10.1146/annurev-economics-080218-025616.

[37] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006. doi: 10.1017/CBO9780511546921.

[38] T.-H. H. Chan, F. Chen, and S. H.-C. Jiang. Revealing optimal thresholds for generalized secretary problem via continuous LP: Impacts on online k-item auction and bipartite k-matching with random arrival order. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms.* Society for Industrial and Applied Mathematics, 2014. doi: 10.1137/1.9781611973730.78.

[39] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd ACM symposium on Theory of Computing (STOC)*, pages 311–320, 2010. doi: 10.1145/1806689.1806733.

[40] S. Chawla, H. Fu, and A. R. Karlin. Approximate revenue maximization in interdependent value settings. *CoRR*, abs/1408.4424, 2014. URL `https://arxiv.org/abs/1408.4424`.

[41] S. Chawla, E. Gergatsouli, Y. Teng, C. Tzamos, and R. Zhang. Pandora's box with correlations: Learning and approximation. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1214–1225, 2020. doi: 10.1109/FOCS46700.2020.00116.

[42] J. Chen, B. Li, Y. Li, and P. Lu. Bayesian auctions with efficient queries. *Artificial Intelligence*, 303:103630, 2022. doi: 10.1016/j.artint.2021.103630.

[43] X. Chen, I. Diakonikolas, A. Orfanou, D. Paparas, X. Sun, and M. Yannakakis. On the complexity of optimal lottery pricing and randomized mechanisms. In *Proceedings of 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1464–1479, Oct. 2015. doi: 10.1109/FOCS.2015.93.

[44] X. Chen, G. Matikas, D. Paparas, and M. Yannakakis. On the complexity of simple and optimal deterministic mechanisms for an additive buyer. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2036–2049, 2018. doi: 10.1137/1.9781611975031.133.

[45] Y. Chen and D. Goldberg. *Beating the curse of dimensionality in options pricing and optimal stopping.* Manuscript, 2018. URL `https://arxiv.org/abs/1807.02227`.

[46] Y. Chen and D. A. Goldberg. *A new approach to high-dimensional online decision-making.* Manuscript, 2020.

[47] Y. Chen, V. F. Farias, and N. Trichakis. On the efficacy of static prices for revenue management in the face of strategic customers. *Management Science*, 65(12):5535–5555, 2019.

[48] M. Choi and L. Smith. Optimal sequential search. *SSRN Electronic Journal*, 2020. doi: 10.2139/ssrn.3530526. URL `https://doi.org/10.2139%2Fssrn.3530526`.

[49] D. F. Ciocan and V. V. Mišić. Interpretable optimal stopping. *Management Science*, 2020. doi: 10.1287/mnsc.2020.3592.

[50] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.

[51] R. Cole and S. Rao. Applications of α-strongly regular distributions to bayesian auctions. *ACM Trans. Econ. Comput.*, 5(4):18:1–18:29, Dec. 2017. doi: 10.1145/3157083.

[52] R. Cole and T. Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 243–252, 2014. doi: 10.1145/2591796.2591867.

[53] J. Correa, P. Dütting, F. Fischer, and K. Schewior. Prophet inequalities for i.i.d. random variables from an unknown distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, 2019. doi: 10.1145/3328526.3329627.

[54] J. Correa, P. Foncea, R. Hoeksma, T. Oosterwijk, and T. Vredeveld. Recent developments in prophet inequalities. *ACM SIGecom Exchanges*, 17(1):61–70, 2019. doi: 10.1145/3331033.3331039.

[55] J. Correa, A. Cristi, L. Feuilloley, T. Oosterwijk, and A. Tsigonias-Dimitriadis. The secretary problem with independent sampling. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2047–2058. Society for Industrial and Applied Mathematics, 2021. doi: 10.1137/1.9781611976465.122.

[56] J. R. Correa, R. Saona, and B. Ziliotto. Prophet secretary through blind strategies. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 1946–1961. SIAM, 2019. doi: 10.1137/1.9781611975482.118.

[57] J. R. Correa, A. Cristi, B. Epstein, and J. A. Soto. The two-sided game of googol and sample-based prophet inequalities. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2066–2081. Society for Industrial and Applied Mathematics, 2020. doi: 10.1137/1.9781611975994.127.

[58] D. Cownden and D. Steinsaltz. Effects of competition in a secretary problem. *Operations Research*, 62(1):104–113, 2014. doi: 10.1287/opre.2013.1233.

[59] C. Daskalakis, A. Deckelbaum, and C. Tzamos. The complexity of optimal mechanism design. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1302–1318, 2013. doi: 10.1137/1.9781611973402.96.

[60] C. Daskalakis, A. Deckelbaum, and C. Tzamos. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017. doi: 10.3982/ECTA12618.

[61] M. Derakhshan, N. Golrezaei, V. Manshadi, and V. Mirrokni. Product ranking on online platforms. In *Proceedings of the 21st ACM Conference on Economics and Computation*. ACM, 2020. doi: 10.1145/3391403.3399483.

[62] M. Derakhshan, N. Golrezaei, and R. P. Leme. Linear program-based approximation for personalized reserve prices. *Management Science*, 68(3):1849–1864, 2022. doi: 10.1287/mnsc.2020.3897.

[63] P. Dhangwatnotai, T. Roughgarden, and Q. Yan. Revenue maximization with a single sample. *Games and Economic Behavior*, 91:318–333, 2015. doi: 10.1016/j.geb.2014.03.011.

[64] L. Doval. Whether or not to open pandora's box. *Journal of Economic Theory*, 175:127–158, 2018. doi: https://doi.org/10.1016/j.jet.2018.01.005.

[65] S. Dughmi, L. Han, and N. Nisan. Sampling and representation complexity of revenue maximization. In *Proceedings of the 10th International Conference on Web and Internet Economics (WINE)*, pages 277–291, 2014. doi: 10.1007/978-3-319-13129-0_22.

[66] P. Dütting, T. Roughgarden, and I. Talgam-Cohen. Simple versus optimal contracts. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 369–387, 2019. doi: 10.1145/3328526.3329591.

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

[68] P. Dütting, S. Lattanzi, R. P. Leme, and S. Vassilvitskii. Secretaries with advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. ACM, 2021. doi: 10.1145/3465456.3467623.

[69] S. Ehsani, M. Hajiaghayi, T. Kesselheim, and S. Singla. Prophet secretary for combinatorial auctions and matroids. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 700–714. Society for Industrial and Applied Mathematics, jan 2018. doi: 10.1137/1.9781611975031.46.

[70] H. Esfandiari, M. HajiAghayi, B. Lucier, and M. Mitzenmacher. Online pandora's boxes and bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1885–1892, 2019. doi: 10.1609/aaai.v33i01.33011885.

[71] H. Esfandiari, M. Hajiaghayi, B. Lucier, and M. Mitzenmacher. Prophets, secretaries, and maximizing the probability of choosing the best. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3717–3727. PMLR, 2020. URL `https://proceedings.mlr.press/v108/esfandiari20a.html`.

[72] T. Ezra, S. Leonardi, R. Reiffenhauser, M. Russo, and A. Tsigonias-Dimitriadis. Prophet inequalities via the expected competitive ratio. *CoRR*, abs/2207.03361, 2022. URL `https://arxiv.org/pdf/2207.03361.pdf`.

[73] T. S. Ferguson. Who solved the secretary problem? *Statistical Science*, 4(3), 1989. doi: 10.1214/ss/1177012493.

[74] H. Fu, N. Immorlica, B. Lucier, and P. Strack. Randomization beats second price as a prior-independent auction. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, pages 323–323, 2015. doi: 10.1145/2764468.2764489.

[75] H. Fu, J. Li, and P. Xu. A PTAS for a Class of Stochastic Dynamic Programs. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:14, 2018. doi: 10.4230/LIPIcs.ICALP.2018.56.

[76] B. Gamlath, S. Kale, and O. Svensson. Beating greedy for stochastic bipartite matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2841–2854. Society for Industrial and Applied Mathematics, 2019. doi: 10.1137/1.9781611975482.176.

[77] Y. Giannakopoulos and E. Koutsoupias. Duality and optimality of auctions for uniform distributions. *SIAM Journal on Computing*, 47(1):121–165, 2018. doi: 10.1137/16M1072218.

[78] Y. Giannakopoulos, D. Poças, and A. Tsigonias-Dimitriadis. Robust revenue maximization under minimal statistical information. In *Proceedings of the 16th Conference on Web and Internet Economics (WINE)*, pages 177–190, 2020. doi: 10.1007/978-3-030-64946-3_13.

[79] Y. Giannakopoulos, D. Poças, and K. Zhu. Optimal pricing for MHR and λ-regular distributions. *ACM Transactions on Economics and Computation*, 9(1): 1–28, 2021. doi: 10.1145/3434423.

[80] J. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61:35–73, 1966.

[81] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979.

[82] D. G. Goldstein, R. P. McAfee, S. Suri, and J. R. Wright. Learning when to stop searching. *Management Science*, 66(3):1375–1394, 2020. doi: 10.1287/mnsc.2018.3245.

[83] Y. A. Gonczarowski and S. M. Weinberg. The sample complexity of up-to-ϵ multi-dimensional revenue maximization. *Journal of the ACM*, 68(3):1–28, 2021. doi: 10.1145/3439722.

[84] N. Gravin and P. Lu. Separation in correlation-robust monopolist problem with budget. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2069–2080, 2018. doi: 10.1137/1.9781611975031.135.

[85] T. Groves. Incentives in teams. *Econometrica*, 41(4):617, 1973. doi: 10.2307/1914085.

[86] A. Gupta and S. Singla. Random-order models. In *Beyond the Worst-Case Analysis of Algorithms*, pages 234–258. Cambridge University Press, 2020. doi: 10.1017/9781108637435.015.

[87] A. Gupta, H. Jiang, Z. Scully, and S. Singla. The markovian price of information. In *Integer Programming and Combinatorial Optimization*, pages 233–246. Springer International Publishing, 2019. doi: 10.1007/978-3-030-17953-3_18.

[88] N. Haghpanah and J. Hartline. Reverse mechanism design. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, pages 757–758, 2015. doi: 10.1145/2764468.2764498.

[89] S. Hart and N. Nisan. The menu-size complexity of auctions. In *Proceedings of the 14th ACM Conference on Electronic Commerce (EC)*, pages 565–566, 2013. doi: 10.1145/2482540.2482544.

[90] S. Hart and N. Nisan. Approximate revenue maximization with multiple items. *J. Econ. Theory*, 172:313–347, 2017. doi: 10.1016/j.jet.2017.09.001.

[91] J. D. Hartline. Mechanism design and approximation. Manuscript, 2013. URL `http://jasonhartline.com/MDnA/`.

[92] J. D. Hartline and T. Roughgarden. Simple versus optimal mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, pages 225–234, 2009. doi: 10.1145/1566374.1566407.

[93] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

[94] Z. Huang, Y. Mansour, and T. Roughgarden. Making the most of your samples. *SIAM Journal on Computing*, 47(3):651–674, 2018. doi: 10.1137/16M1065719.

[95] N. Immorlica, R. Kleinberg, and M. Mahdian. Secretary problems with competing employers. In *Lecture Notes in Computer Science*, pages 389–400. Springer Berlin Heidelberg, 2006. doi: 10.1007/11944874_35.

[96] N. Immorlica, J. Leshno, I. Lo, and B. Lucier. Information acquisition in matching markets: The role of price discovery. *SSRN Electronic Journal*, 2020. doi: 10.2139/ssrn.3705049. URL `https://doi.org/10.2139%2Fssrn.3705049`.

[97] P. Jaillet, J. A. Soto, and R. Zenklusen. Advances on matroid secretary problems: Free order model and laminar case. In *Integer Programming and Combinatorial Optimization*, pages 254–265. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-36694-9_22.

[98] H. Kaplan, D. Naori, and D. Raz. Competitive analysis with a sample and the secretary problem. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2082–2095. Society for Industrial and Applied Mathematics, 2020. doi: 10.1137/1.9781611975994.128.

[99] H. Kaplan, D. Naori, and D. Raz. Online weighted matching with a sample. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1247–1272. Society for Industrial and Applied Mathematics, 2022. doi: 10.1137/1.9781611977073.52.

[100] M. G. Kendall. *The Advanced Theory of Statistics*, volume I. Charles Griffin, 4th edition, 1948.

[101] T. Kesselheim and M. Molinaro. Knapsack Secretary with Bursty Adversary. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:15. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.ICALP.2020.72.

[102] T. Kesselheim, R. Kleinberg, and R. Niazadeh. Secretary problems with non-uniform arrival order. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. ACM, 2015. doi: 10.1145/2746539.2746602.

[103] J. Kleinberg and R. Kleinberg. Delegated search approximates efficient search. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, page 287–302. Association for Computing Machinery, 2018. doi: 10.1145/3219166.3219205.

[104] R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, page 630–631. Society for Industrial and Applied Mathematics, 2005.

[105] R. Kleinberg and S. M. Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games and Economic Behavior*, 113:97–115, 2019. doi: 10.1016/j.geb.2014.11.002.

[106] R. Kleinberg, B. Waggoner, and E. G. Weyl. Descending price optimally coordinates search. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, page 23–24, New York, NY, USA, 2016. Association for Computing Machinery. doi: 10.1145/2940716.2940760.

[107] Ç. Koçyiğit, G. Iyengar, D. Kuhn, and W. Wiesemann. Distributionally robust mechanism design. *Management Science*, 66(1):159–189, 2020. doi: 10.1287/mnsc.2018.3219.

[108] N. Korula and M. Pál. Algorithms for secretary problems on graphs and hypergraphs. In *Automata, Languages and Programming*, pages 508–520. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-02930-1_42.

[109] U. Krengel and L. Sucheston. Semiamarts and finite values. *Bulletin of the American Mathematical Society*, 83:745–747, 1977.

[110] U. Krengel and L. Sucheston. On semiamarts, amarts, and processes with finite value. *Advances in Probability and Related Topics*, 4:197–266, 1978.

[111] X. Li and A. C.-C. Yao. On revenue maximization for selling multiple independently distributed items. *Proceedings of the National Academy of Sciences*, 110 (28):11232–11237, 2013. doi: 10.1073/pnas.1309533110.

[112] Y. Li, P. Lu, and H. Ye. Revenue maximization with imprecise distribution. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1582–1590, 2019. URL `http://dl.acm.org/citation.cfm?id=3306127.3331877`.

[113] D. V. Lindley. Dynamic programming and decision theory. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 10(1):39–51, 1961.

[114] A. Liu, R. P. Leme, M. Pál, J. Schneider, and B. Sivan. Variable decomposition for prophet inequalities and optimal ordering. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. ACM, 2021. doi: 10.1145/3465456.3467598.

[115] B. Lucier. An economic view of prophet inequalities. *ACM SIGecom Exchanges*, 16(1):24–47, 2017. doi: 10.1145/3144722.3144725.

[116] T. Ma, B. Tang, and Y. Wang. The simulated greedy algorithm for several submodular matroid secretary problems. *Theory of Computing Systems*, 58(4):681–706, 2016. doi: 10.1007/s00224-015-9642-4.

[117] W. Ma, D. Simchi-Levi, and J. Zhao. Dynamic pricing (and assortment) under a static calendar. *Management Science*, 67(4):2292–2313, 2021. doi: 10.1287/mnsc.2020.3671.

[118] C. L. Mallows and D. Richter. Inequalities of chebyshev type involving conditional expectations. *The Annals of Mathematical Statistics*, 40(6):1922–1932, 1969. URL http://www.jstor.org/stable/2239511.

[119] A. M. Manelli and D. R. Vincent. Multidimensional mechanism design: Revenue maximization and the multiple-good monopoly. *Journal of Economic Theory*, 137 (1):153–185, 2007. doi: 10.1016/j.jet.2006.12.007.

[120] P. Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511813825.009.

[121] A. M. Mood, F. A. Graybill, and D. C. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill, 3rd edition, 1974.

[122] S. Moran, M. Snir, and U. Manber. Applications of ramsey's theorem to decision tree complexity. *Journal of the ACM*, 32(4):938–949, 1985. doi: 10.1145/4221.4259.

[123] J. Morgenstern and T. Roughgarden. Learning simple auctions. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, pages 1298–1318, 2016. URL http://proceedings.mlr.press/v49/morgenstern16.html.

[124] S. Moriguti. Extremal properties of extreme value distributions. *The Annals of Mathematical Statistics*, 22(4):523–536, 1951. URL http://www.jstor.org/stable/2236921.

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

[126] R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6 (1):58–73, 1981. doi: 10.1287/moor.6.1.58.

[127] R. Niazadeh, A. Saberi, and A. Shameli. Prophet inequalities vs. approximating optimum online. In *Web and Internet Economics*, pages 356–374. Springer International Publishing, 2018. doi: 10.1007/978-3-030-04612-5_24.

[128] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007. doi: 10.1017/CBO9780511800481.

[129] P. Nuti. The secretary problem with distributions. In *Integer Programming and Combinatorial Optimization*, pages 429–439. Springer International Publishing, 2022. doi: 10.1007/978-3-031-06901-7_32.

[130] W. Olszewski and R. Weber. A more general pandora rule? *Journal of Economic Theory*, 160:429–437, 2015. doi: 10.1016/j.jet.2015.10.009.

[131] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* The MIT Press, 1994.

[132] C. Papadimitriou, T. Pollner, A. Saberi, and D. Wajc. Online stochastic max-weight bipartite matching: Beyond prophet inequalities. In *Proceedings of the 22nd ACM Conference on Economics and Computation.* ACM, 2021. doi: 10.1145/3465456.3467613.

[133] T. Roughgarden and I. Talgam-Cohen. Approximately optimal mechanism design. *Annual Review of Economics*, 11(1), 2019. doi: 10.1146/annurev-economics-080218-025607.

[134] A. Rubinstein and S. M. Weinberg. Simple mechanisms for a subadditive buyer and applications to revenue monotonicity. *ACM Trans. Econ. Comput.*, 6(3-4): 19:1–19:25, Oct. 2018. doi: 10.1145/3105448.

[135] A. Rubinstein, J. Z. Wang, and S. M. Weinberg. Optimal Single-Choice Prophet Inequalities from Samples. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:10, 2020. doi: 10.4230/LIPIcs.ITCS.2020.60.

[136] E. Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *The Annals of Probability*, 12(4), 1984. doi: 10.1214/aop/1176993150.

[137] S. Samuels. Exact solutions for the full information best choice problem. *Purdue Univ. Stat. Dept. Mimeo Series*, pages 82–17, 1982.

[138] S. Samuels. Secretary problems. In *Handbook of Sequential Analysis.* Chapter 16, (B. Ghosh, P. Sen, Eds.), CRC Press, 1991.

[139] N. Schweizer and N. Szech. Performance bounds for optimal sales mechanisms beyond the monotone hazard rate condition. *Journal of Mathematical Economics*, 82:202–213, 2019. doi: 10.1016/j.jmateco.2019.02.007.

[140] D. A. Seale and A. Rapoport. Sequential decision making with relative ranks: An experimental investigation of the "secretary problem". *Organizational Behavior and Human Decision Processes*, 69(3):221–236, 1997. doi: 10.1006/obhd.1997.2683.

[141] D. Segev and S. Singla. Efficient approximation schemes for stochastic probing and prophet problems. In *Proceedings of the 22nd ACM Conference on Economics and Computation.* ACM, 2021. doi: 10.1145/3465456.3467614.

[142] S. Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2523–2532. Society for Industrial and Applied Mathematics, 2018. doi: 10.1137/1.9781611975031.161.

[143] A. Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019. doi: 10.1561/2200000068.

[144] J. A. Soto, A. Turkieltaub, and V. Verdugo. Strong algorithms for the ordinal matroid secretary problem. *Mathematics of Operations Research*, 46(2):642–673, 2021. doi: 10.1287/moor.2020.1083.

[145] G. J. Stigler. The economics of information. *Journal of Political Economy*, 69(3): 213–225, 1961. doi: 10.1086/258464.

[146] A. Suzdaltsev. An optimal distributionally robust auction. *CoRR*, abs/2006.05192, 2020. URL `https://arxiv.org/abs/2006.05192`.

[147] V. Syrgkanis. A sample complexity measure with applications to learning optimal auctions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5358–5365, Red Hook, NY, USA, 2017.

[148] T. Tao. *An Introduction to Measure Theory*, volume 126 of *Graduate Studies in Mathematics*. American Mathematical Society, 2011.

[149] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961. doi: 10.1111/j.1540-6261.1961.tb02789.x.

[150] M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3): 641, 1979. doi: 10.2307/1910412.

[151] R. Wilson. Game-theoretic analyses of trading processes. In *Advances in Economic Theory: Fifth World Congress*, Econometric Society Monographs, pages 33–70. Cambridge University Press, 1987. doi: 10.1017/CCOL0521340446.002.

[152] A. C.-C. Yao. An n-to-1 bidder reduction for multi-item auctions and its applications. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 92–109, 2015. doi: 10.1137/1.9781611973730.8.

[153] R. Zwick, A. Rapoport, A. K. C. Lo, and A. V. Muthukrishnan. Consumer sequential search: Not enough or too much? *Marketing Science*, 22(4):503–519, 2003. doi: 10.1287/mksc.22.4.503.24909.

A Omitted Proofs from Chapter 3

A.1 Technical lemmas

Lemma 38. *For $r > 0$, let $\rho(r)$ denote the unique solution over $[1, \infty)$ of the equation*

$$\frac{(\rho-1)^3}{(2\rho-1)^2} = r^2.$$

Then it holds that

$$1 + 4r^2 \leq \rho(r) \leq 2 + 4r^2$$

for large enough values of r.

Proof. One can directly check that the expression $\frac{(\rho-1)^3}{(2\rho-1)^2}$ is increasing and goes from 0 at $\rho = 1$ to ∞ at $\rho \to \infty$, so that for any nonnegative r there is a unique solution $\rho \in [1, \infty)$ to the above equation. Moreover, we can write

$$r^2 = \frac{(\rho-1)^3}{(2\rho-1)^2} = \frac{1}{4}\rho - \frac{1}{4} - \frac{(\rho-\frac{3}{4})(\rho-1)}{4(\rho-\frac{1}{2})^2} \quad \Longleftrightarrow \quad \rho = 1 + 4r^2 + \frac{(\rho-\frac{3}{4})(\rho-1)}{(\rho-\frac{1}{2})^2};$$

since the fraction appearing on the right-hand side takes values between 0 and 1 (for $\rho \in [1, \infty)$), this gives us the desired global bounds. $\square$

Lemma 39. *For $r > 0$, let $\rho(r)$ denote the (unique) positive solution of the equation*

$$\frac{1}{\rho^2}\left(2e^{\rho-1} - 1\right) = r^2 + 1.$$

Then, for any $\varepsilon > 0$, it holds that

$$\rho(r) \leq 1 + (1+\varepsilon)\ln(1+r^2)$$

for large enough values of r.

Proof. Fix an $\varepsilon > 0$. For convenience, define the functions $f, g : (0, \infty) \longrightarrow \mathbb{R}$ with

$$f(x) = \frac{1}{x^2}\left(2e^{x-1} - 1\right) \qquad \text{and} \qquad g(x) = 1 + (1+\varepsilon)\ln(1+x^2).$$

By considering their derivatives, it is straightforward to see that both f and g are increasing functions. So, to prove our lemma, it is enough to show that

$$f(g(r)) \geq r^2 + 1$$

for large enough values of r.

Indeed, taking r large enough we can guarantee that

$$g(r) = 1 + (1+\varepsilon)\ln(1+r^2) \leq (1+r^2)^{\varepsilon/2},$$

since $\ln(1+x) = o(x^{\varepsilon/2})$. Thus we have

$$f(g(r)) = \frac{2e^{g(r)-1}-1}{[g(r)]^2} \geq \frac{2e^{(1+\varepsilon)\ln(1+r^2)}-1}{\left[(1+r^2)^{\varepsilon/2}\right]^2} = \frac{2(1+r^2)^{1+\varepsilon}-1}{(1+r^2)^{\varepsilon}} = 2(1+r^2) - \frac{1}{(1+r^2)^{\varepsilon}}$$

which is greater than $1+r^2$ for large enough r, since $\frac{1}{x^\varepsilon} = o(x)$. □

A.2 Asymptotics of the mechanism by Azar and Micali [8]

In this section we look at the upper bound proposed in Azar and Micali [8, Thm. 1]. They propose a deterministic mechanism with selling price $p = \mu - k(r)\sigma$, where $k(r)$ is the unique positive solution of the cubic equation $\frac{1}{r} = \frac{1}{2}(3k + k^3)$. They derive an approximation guarantee which in our setting can be expressed as

$$\mathrm{APX}(\mu,\sigma) \leq \frac{\mu}{\mathrm{REV}(p;F)} \leq \frac{1}{1-\frac{3}{2}rk(r)} \equiv \tilde{\rho}(r). \tag{A.1}$$

We have the following global bounds and asymptotics:

Lemma 40. *For any $\mu > 0$ and $\sigma \geq 0$, let $r = \sigma/\mu$ and let k denote the unique real solution of $\frac{1}{r} = \frac{1}{2}(3k+k^3)$. Furthermore, let $\tilde{\rho} = \frac{1}{1-\frac{3}{2}rk}$ and $p = \mu - k\sigma$. Then $\tilde{\rho}$ is the unique solution over $[1,\infty)$ of the equation*

$$\frac{27}{4}r^2 = \frac{(\tilde{\rho}-1)^3}{\tilde{\rho}^2},$$

and further satisfies

$$1 + \frac{27}{4}r^2 \leq \tilde{\rho} \leq 3 + \frac{27}{4}r^2 \qquad \text{and} \qquad p = \frac{\tilde{\rho}+2}{3\tilde{\rho}} \cdot \mu.$$

Proof. We begin by rewriting k in terms of $\tilde{\rho}$,

$$\tilde{\rho} = \frac{1}{1-\frac{3}{2}rk} \quad\Longleftrightarrow\quad k = \frac{2}{3r}\frac{\tilde{\rho}-1}{\tilde{\rho}};$$

plugging this in the cubic equation for k, and doing some manipulation, gives

$$\frac{1}{r} = \frac{1}{2}\left(\frac{2}{r}\frac{\tilde{\rho}-1}{\tilde{\rho}} + \frac{8}{27r^3}\frac{(\tilde{\rho}-1)^3}{\tilde{\rho}^3}\right) \quad\Longleftrightarrow\quad \frac{27}{4}r^2 = \frac{(\tilde{\rho}-1)^3}{\tilde{\rho}^2}.$$

One can directly check that the expression $\frac{(\tilde{\rho}-1)^3}{\tilde{\rho}^2}$ is increasing and goes from 0 at $\tilde{\rho} = 1$ to ∞ at $\tilde{\rho} \to \infty$, so that for any nonnegative r there is a unique solution $\tilde{\rho} \in [1, \infty)$ to the above equation. Moreover, we can write

$$p = \mu - k\sigma = \mu - \frac{2}{3}\frac{\sigma}{r}\frac{\tilde{\rho}-1}{\tilde{\rho}} = \frac{\tilde{\rho}+2}{3\tilde{\rho}} \cdot \mu$$

and

$$\frac{27}{4}r^2 = \frac{(\tilde{\rho}-1)^3}{\tilde{\rho}^2} = \tilde{\rho} - 1 - \frac{(2\tilde{\rho}-1)(\tilde{\rho}-1)}{\tilde{\rho}^2} \qquad \Longleftrightarrow \qquad \tilde{\rho} = 1 + \frac{27}{4}r^2 + \frac{(2\tilde{\rho}-1)(\tilde{\rho}-1)}{\tilde{\rho}^2}.$$

Since the fraction appearing on the right-hand side takes values between 0 and 2 (for $\tilde{\rho} \in [1, \infty)$), this gives us the desired global bounds. □

B Omitted Proofs from Chapter 4

B.1 Proof of Lemma 7 through the conflict graph

Following we state again Lemma 7 and we give an alternative proof for the success guarantee of the k-max algorithm. This approach highlights the clear properties of the conflict graph and gives a good understanding of the type of instances in which the k-max algorithm picks the maximum.

Lemma 41. *The success guarantee of the k-max algorithm equals $(1-p)kp^k$, where $k = \left\lfloor \frac{1}{1-p} \right\rfloor$.*

Proof. Recall again the scenario where $p \in [1/2, 2/3)$. For a given size n, the instances ending with 110 or 101 are picked by the algorithm, and these are instances with degree one and degree two, respectively. One can check that the rest of the instances of degree one and two (and size n) that are not selected by the algorithm belong to a monotone path linked to a selected instance of smaller size. Therefore, they have already been removed. Instances of larger degree have at least two 1s in the end, therefore for $p \in [1/2, 2/3)$ the k-max algorithm will never obtain the maximum in those instances. Thus, for each size n, it holds that all the instances of degree one and two are either removed or selected. Similarly, for $p \in [2/3, 3/4)$ all instances of degree one, two and three are either selected or removed.

Let r and s be the total number of removed and selected instances respectively. We will show that for the k-max algorithm s equals the claimed success guarantee by computing the value of $r+s$ and subtracting the value of r. From the above we see that, using the k-max algorithm, $r+s$ counts all instances of degrees 1 up to $k = \left\lfloor \frac{1}{1-p} \right\rfloor$. Therefore,

$$r + s = \sum_{i=1}^{k} w_i = (1-p) \sum_{i=0}^{k-1} p^i \,.$$

Now we compute the value of r. From the above, we know that, for a certain size n, all instances that have degrees 1 up to k are either selected or removed. In the first case, we need to remove the descendants of these instances, while in the latter case these descendants have already been removed. In total, of all instances of size $n+1$, the algorithm removes all children of nodes of size n that have degrees 1 up to k. Recalling the degree structure from Lemma 11, we see that a selected instance of degree i has one descendant of weight w_{ij} for all $j = 1, \ldots, i$. Therefore, the total removed weight equals

$$r = w_{11} + (w_{21} + w_{22}) + \ldots + (w_{k1} + \ldots + w_{kk})$$

$$
\begin{aligned}
&= (1-p)w_1 + 2(1-p)w_2 + \ldots + k(1-p)w_k \\
&= (1-p)\sum_{i=1}^{k} i w_i = (1-p)^2 \sum_{i=0}^{k-1} (i+1)p^i \,,
\end{aligned}
$$

and the instances that will be selected by the algorithm have total weight

$$
\begin{aligned}
s &= (r+s) - r \\
&= (1-p)\left[\sum_{i=0}^{k-1} p^i - (1-p)\sum_{i=0}^{k-1}(i+1)p^i\right] \\
&= (1-p)\sum_{i=0}^{k-1}\left[1-(i+1)(1-p)\right]p^i \\
&= (1-p)\sum_{i=0}^{k-1}\left[(i+1)p^{i+1} - ip^i\right] \\
&= (1-p)kp^k \,.
\end{aligned}
$$

□

B.2 Intuition behind threshold for AOSp

Next, we briefly show the intuition behind the choice of k in the k-max algorithm for AOSp. Assume that the samples are sorted in decreasing order $s_1 \geq s_2 \geq \cdots \geq s_r$. An algorithm ALG, given input $(\mathcal{S}, p)$, draws $t \sim \mathcal{D}$ and sets a threshold $\tau = s_t$. Here, $t \in [r]$ and $\mathcal{D}$ is a probability distribution over the indices of the samples. Thus, the algorithm wins at least in the instances where *exactly one* of the t largest values of the adversarial input ends up in the online set and the $(t+1)^{\text{th}}$ largest ends up in the sample set. Since the coin flips are independent, we obtain

$$
\Pr[\text{ALG stops at} \max_i v_i] = lp^l(1-p), \qquad \text{where } l = \mathbb{E}_{t\sim\mathcal{D}}[t].
$$

To find the maximizer of the function $f(p,l) = lp^l(1-p)$, consider its derivative:

$$
\frac{\partial f(p,l)}{\partial l} = \frac{\partial(lp^l - lp^{l+1})}{\partial l} = 0 \Leftrightarrow l = -\frac{1}{\ln p} \,.
$$

Substituting this value for l into $f(p,l)$ yields an upper bound $f_1(p) = \frac{p-1}{\log p} \cdot p^{-1/\log p}$. To turn it into a practical algorithm, we wish to have an integer value of l without a logarithmic term. The latter can be fixed by taking an approximation. Taking the Maclaurin series of $\ln(1+x)$, substituting $p = x+1$ and dropping the higher order terms yields

$$
\ln p = \sum_{n=1}^{\infty}(-1)^{n-1}\frac{(p-1)^n}{n} \approx p-1 \Leftrightarrow -\frac{1}{\ln p} \approx \frac{1}{1-p}, \qquad p \in (0,1)\,.
$$

Substituting this value for l into $f(p,l)$ yields a lower bound $f_2(p) = p^{1/(1-p)}$. To get to an integer value, we simply take the floor function of this expression and obtain the k-max algorithm as defined previously.

Let us be a bit more precise regarding the success guarantee after these modifications and show that it interpolates between the two functions mentioned above. Because $-\frac{1}{\ln p}$ is a maximizer of f, we see that $f\left(p, \left\lfloor \frac{1}{1-p} \right\rfloor\right) \leq f\left(p, -\frac{1}{\ln p}\right)$. Second, $f(p,l)$ is increasing for $l \leq -\frac{1}{\ln p}$ and decreasing in the rest of the domain. Moreover, one can easily check that $f\left(p, \frac{1}{1-p} - 1\right) = f\left(p, \frac{1}{1-p}\right)$ and $\frac{1}{1-p} - 1 < -\frac{1}{\ln p} \leq \frac{1}{1-p}$ for $p \in (0,1)$. Finally, since it holds that $\frac{1}{1-p} - 1 < \left\lfloor \frac{1}{1-p} \right\rfloor \leq \frac{1}{1-p}$, we get $f(p, \frac{1}{1-p}) \leq f(p, \left\lfloor \frac{1}{1-p} \right\rfloor)$. Combining all the above, we can conclude that $f\left(p, \left\lfloor \frac{1}{1-p} \right\rfloor\right) \approx f\left(p, -\frac{1}{\ln p}\right)$.

C Omitted Proofs from Chapter 5

C.1 Correctness of the DP

We show that the dynamic program from Section 5.2.1 finds the optimal solution for the random order Pandora's box problem without skipping and without commitment. In particular, we want to show that our recursion is correct. We denote by $J_\ell(b_i)$ the cost-to-go functions, where by ℓ we denote the number of the uninspected boxes, and b_i is the next box in the sequence that we want to decide whether to open it or not.

- The base case is when there is only one box b_i remaining, and the cost-to-go is

$$J_1(b_i) = \max\left(\max_{k\in\mathcal{P}} v_k, -c_i + v_i\right) ,$$

where we denote by $\mathcal{P}$ the set of opened boxes so far. If we had only the subproblem with the last box to solve, the cost-to-go function would be $J_1(b_i) = \max(0, -c_i + v_i)$, which is the optimal solution.

- Solving backwards: Assume that for $\ell-1$ uninspected remaining boxes the dynamic program solves optimally the subproblem starting at box with index $n-\ell+1$. We want to show that then the dynamic program returns the optimal expected gain for the subproblem which starts with ℓ uninspected boxes. The optimal solution in this case will be

$$J_\ell(b_i) = \max\left(0, -c_i + \mathbb{E}\left[J_{\ell-1}(b_j)\right]\right)$$

and the cost-to-go functions will be $J_s(b_i) = \max\left(\max_{k\in\mathcal{P}} v_k, -c_i + \mathbb{E}\left[J_{s-1}(b_j)\right]\right)$, for $s \in (1, \ell)$.

Let τ be the optimal stopping time for the subproblem with $\ell - 1$ boxes. When we stop at time $\tau = i$ it means that we stop *before* inspecting the box with index i in the sequence. By our hypothesis, this is the optimal stopping time. We consider two cases:

- $\tau = 1$: In this case, the dynamic program stops and collects the reward minus the cost before opening the first box, i.e., it stops without inspecting any boxes. This means that the following condition holds

$$\max_{r\in\mathcal{P}} v_r - \sum_{j=1}^{\tau} c_j < 0 \qquad ,\forall \tau \geq 1 , \tag{C.1}$$

and the optimal solution is 0. If we had one extra box now at the beginning of the sequence, then $J_\ell(b_i) = \max(0, -c_i + \mathbb{E}[J_{\ell-1}(b_j)])$ and

$$J_{\ell-1}(b_j) = \max\left(\max_{k\in\mathcal{P}} v_k, -c_j + \mathbb{E}[J_{\ell-2}(b_\kappa)]\right) = \max_{k\in\mathcal{P}} v_k$$

by Eq. (C.1). Thus $J_\ell(b_i) = \max(0, -c_i + v_i)$, which is the optimal solution for the subproblem with ℓ boxes, given that the subproblem for $\ell - 1$ is solved optimally.

- $\tau \neq 1$: In this case we open the first box for the subproblem with $\ell - 1$ boxes, and we stop optimally at a later box with index τ. The optimal solution of the DP then becomes

$$J_{\ell-1}(b_i) = \max\left(0, \max_{k\in\mathcal{P}} v_k - \sum_{j=1}^{\tau-1} c_j\right) = \max_{k\in\mathcal{P}} v_k - \sum_{j=1}^{\tau-1} c_j \ .$$

  For the subproblem with ℓ boxes, the cost-to-go functions for ℓ and $\ell-1$ uninspected boxes respectively, become $J_\ell(b_i) = \max(0, -c_i + \mathbb{E}[J_{\ell-1}(b_j)])$ and $J_{\ell-1}(b_j) = \max\left(\max_{k\in\mathcal{P}} v_k, \max_{r\in\mathcal{P}_f} v_r - \sum_{j=1}^{\tau-1} c_j\right)$, where the set $\mathcal{P}$ in this case contains just the reward of the first box of the sequence, and by $\mathcal{P}_f$ we denote the set of opened boxes up to the last box we open before stopping at time τ. Now we can observe that $J_{\ell-1}(b_j)$ returns either the optimal solution to the subproblem with $\ell - 1$ boxes or the reward of the first box. In turn, $J_\ell(b_i)$ returns the optimal solution, which is the max of either inspecting nothing, or opening the first box (and then maybe proceed optimally in the subproblem, or stop if the reward minus the cost of the first box is bigger).

We conclude that the DP always makes the optimal decision on when to stop inspecting. The optimal expected gain for an instance with n boxes is $J_n(b_i)$.