ALGORITHMS VS. MECHANISMS .: MECHANISM DESIGN FOR COMPLEX ENVIRONMENTS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Rad Niazadeh

August 2017

ALGORITHMS VS. MECHANISMS .:
MECHANISM DESIGN FOR COMPLEX ENVIRONMENTS

Rad Niazadeh, Ph.D.
Cornell University 2017

Prevalent internet marketplaces and crowdsourcing platforms have started facing new computational challenges. In point of fact, these challenges exist mostly due to the strategic behavior of users, the stochastic nature of these platforms, their real-time computations, and the demand for incorporating the large-scale users data. With these challenges in front of the mechanism designer, the job of the field of algorithmic mechanism design is to find new ways of tackling them. Having this objective in mind, my dissertation is taking a closer look into various aspects of mechanism design for (real-world) applications suffering from the aforementioned challenges. We informally refer to such applications as *complex environments*. In this dissertation, we particularly show how to develop general reductions from mechanism design to algorithm design, how to develop and analyze simple mechanisms for trade, and finally how to learn optimal revenue mechanisms in an online fashion by interacting with users in several rounds. Throughout this dissertation, we point out how we Incorporate techniques and ideas from combinatorial optimization, learning theory and applied probability to obtain our results.

BIOGRAPHICAL SKETCH

Rad Niazadeh was born in Tehran, Iran in 1986. His interest in mathematics originated during his teenhood, when he attended Allame-Helli high school, the selective school for the National Organization for Development of Exceptional Talents (NODET) in Iran. He was then ranked 15$^{\text{th}}$ out of more than 500000 participants in the national entrance exam in Iran, and started his bachelors in electrical engineering at Sharif University of Technology, Tehran, Iran in 2004. He finished his B.Sc. in 2008, while ranking 3$^{\text{rd}}$ out of more than 140 student in his class. He was also one of the recipients of the *Exceptional Talents Awards* for master of science programs at Sharif, which admitted him into the EE department MS.C. program without any exams. He successfully finished his masters in electrical engineering, majored in communication systems, in 2010. Upon graduation, he was ranked 1$^{\text{st}}$ out of more than 120 students in his class.

After obtaining his master degree, Rad received the *Jacobs Scholar Fellowship* from Cornell University in 2011, and he started his Ph.D. in computer science at Cornell under the supervision of Professor Robert D. Kleinberg in 2012. Since then, he is pursuing his graduate studies by conducting research in various areas of theoretical computer science. Rad is also the recipient of the *Google Ph.D. Fellowship* in market algorithms in 2016.

This dissertation is dedicated to my love of life *Saba Niaki*, and,
my amazing parents *Nasser Niazadeh* and *Mahboubeh Afshari*, without whom I would never be able to pass the hardships of my graduate studies and living far from my home.

ACKNOWLEDGEMENTS

Maybe the hardest part of writing this dissertation is expressing my gratitude and never-ending sense of appreciation towards Bobby Kleinberg, my PhD advisor, in one paragraph. I started my graduate studies in electrical engineering. Then, at the luckiest moment of my academic career, I met the mathematical genius Bobby Kleinberg after I took a graduate theory course with him. Through his academic brilliancy, high-value standards, and altruistic character, not only he showed me, quote, "the right path to fulfill my intellectual curiosity", unquote, but also he soon became a very close friend of mine. Since then, he has continued being a phenomenal mentor and has been a role-model even in non-academic aspects of my life. Indeed, finishing a PhD in theoretical CS after switching from EE is not easy; we went through many hardships in my research and faced many strategic obstacles in my academic career together. Confronting with these turmoils and barriers, he always helped me with the most supportive attitude I have seen from a person in my life, and he guided me through the process of making several of my key decisions with his wise and thoughtful advice. Without Bobby, I cannot imagine reaching to the point I am standing at now, and therefore I always feel infinitely thankful for his presence in my life.

Among all of my colleagues and co-authors, Jason Hartline and Nicole Immorlica have been very special to me. Jason is not only a great friend and a fabulous researcher, but also he has been like a second advisor to me. It is always fun to work with Jason due to his very specific character and taste in research, and I have always enjoyed receiving his very interesting points of view and advices. Nicole has been my mentor during an internship at Microsoft Research New England. While she is a very good friend of mine too, she also is a thoughtful mentor that I have learned a lot from her and enjoyed her advices several times at key decision making points of my career. I am very grateful that I had the opportunity to work with both of them, and I hope it continues in the future.

At Cornell, I was very fortunate to have the chance of getting guidance from Eva Tardos and Jon Kleinberg (who is also in my PhD committee) throughout the graduate school. With their academic excellency, infitnite amount of experience, and kind support, they helped me several times to advance in my research and achieve my career goals. I have learned many academic skills and gained research proficiency by taking their courses or being their teaching assistants, and I am grateful for having these opportunities. I also want to thank Emin G. Sirer for being in my PhD committee and supporting me all these years with his kind words and advices. I would also like to send my regards to Dexter Kozen and Bart Selman (at Cornell CS), and David Shmoys, David Willimson, Sid Banerjee and Kris Iyer (at Cornell ORIE) for their useful comments on my research and their never-ending support. I also like to thank the remaining faculty members of computer science department at Cornell for having me all these years and providing such a beautiful environment for conducting research and getting expertise in CS.

Besides people I have mentioned so far in the theoretical computer science community, I was very fortunate to work with a large group of brilliant and nice people in the area of algorithmic game theory , either through research internships, visiting their institutions or distant collaborations. In particular, I would like to thank Brendan Lucier (my internship mentor at MSR New England), Nikhil Devanur (my internship mentor at MSR Redmond), Chris Wilkens (my internship mentor at Yahoo Research labs) and Matt Weinberg (my host when I was visiting Princeton). I would also like to thank my co-authors Saeed Alaei, Thomas Kesselheim, Shaddin Dughmi, Shuchi Chawla, Zhiyi Huang, Manolis

Pountourakis, Giro Cavallo, Sam Taggart, and my dear office-mate and good friend Yang Yuan.

Besides faculty members at Cornell, I was delightful for having many good friends, student colleagues and classmates in Ithaca. I would like to thank my previous office-mates Hu Fu and Ashwin Kumar B. V., and most importantly my very dear friend Bruno Abrahao. I also want to thank my friends in the theory group at Cornell, and in particular Pooya Jalaly, Soroush Hosseini, Hedyeh Beyhaghi, Rahmtin Rotabi, Sam Hopkins, Thodoris Lykouris, Manish Raghavan, Rediet Abebe and Daniel Freund. I have also had the chance of knowing many Iranians at Cornell, some became very close friends. Payman (Nima) Taherkhani, Naser Nikandish, Sameen Kiayei and Sepehr Saroukhani were my great friends and I thank each one of them for being there always for me and providing an enormous amount of support.

Last and foremost, life outside of my country and studying abroad would not be possible without the infinite love I have received all these years from my family members, the old ones and the new. My parents Nasser and Mahboubeh, my grandmother Mamani Zahra and my beautiful love Saba were my major sources of encouragement and incentive to advance in my academic career, and it is a blessing to have these beautiful people in my life. I am also very grateful to my three closest friends Mohammad, Mahnoosh and Saeed, who have been a very important part of my life for ages, and I cannot imagine it without their endless support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Mechanism Design: the Past and Future

The world around us has become an interconnected network of economic and computational systems that have been broadly used by human users. As a result of this development, and due to strategic behavior of users of these systems, the need to design algorithms that are working properly with strategic input has emerged. Such algorithms are called *mechanisms*, and the discipline of designing such algorithms is widely known as *algorithmic mechanism design*. We consider mechanism design with money, i.e. when the designer is allowed to charge each user with a monetary *payment*. These payments will help the designer to control the strategic behavior of utility-maximizing users.

The applications of algorithmic mechanism design by giant web companies has spurred the development of the field. In fact, mechanism design, and *algorithmic game theory* more generally, have found major roles in designing and analyzing several fundamental components of the last decade's web platforms and search engines. The most canonical example is the *advertisement auction*, a.k.a. ad-auction, which *"still seems to be the undisputed queen of killer applications of algorithmic game theory"* (Nisan, 2010). However, the field of algorithmic mechanism design is continuing to advance, as web-based computational economic platforms are growing more and more, and it constantly finds new, yet more complex, applications beyond ad-auction.

More recently, prevalent internet marketplaces and crowdsourcing platforms such as ride-sharings (e.g. Uber), online freelancing markets (e.g. Upwork), internet sale services (e.g. eBay), and cloud computing pricing mechanisms (e.g. Microsoft Cloud) have started facing new rising computational challenges. In point of fact, these challenges exist mostly due to the strategic behavior of users, the stochastic nature of these platforms, their real-time computations, and the demand for incorporating the large-scale user data. With these challenges in front of the mechanism designer, the job of the field is to find new ways of tackling them. Having this objective in mind, this dissertation is taking a closer look into various aspects of (real-world) applications suffering from the aforementioned challenges. We informally refer to such applications as *complex environments*.

While current methods from theoretical computer science are broadly amplifying our understanding of many of these studied settings, we still require substantially new techniques from *probability theory*, *combinatorial optimization* and *statistical learning* to address many of the challenges in a complex environment. As part of these explorations, in this dissertation we aim to view several of these emerging challenges from different perspectives, and incorporate ideas from the mentioned areas of mathematics to find adequate solutions.

1.2 Towards Modern Mechanisms in Complex Environments

This dissertation gravitates towards problems with an emphasis on the themes of *handling strategic user misbehavior*, *simplicity of mechanisms* and *mechanisms with online input*. By the same token, it exploits - and sometimes develops - new tools that are applicable to problems under these three themes by mingling them with classic ideas from game theory, learning theory and theory of online algorithms. We elaborate more on each of these themes individually. For each one of them we explain what research questions have been solved, and how providing adequate solutions to those questions will guide us towards designing better mechanisms for complex environments.

1.2.1 Incentive Compatibility and User Misbehavior

A striking goal of computer science research in incentive compatible mechanism design is to understand the extent to which "strategic" efficient computation is less powerful than "non-strategic" efficient computation. Incentive compatibility of a mechanism requires that, though agents could misreport their preferences, it is not in any agent's best interest to do so. A key goal then is to design a computationally efficient procedure for transforming any algorithm into an incentive compatible mechanism with little or no loss in expected social welfare [1], a procedure also known as *black-box reduction* for welfare. This gives an affirmative answer to the question of whether algorithms' performance guarantees are robust (up to an efficient reduction) in the presence of users' misbehavior.

In Chapter 3 of this dissertation, we resolve a five-year-old open question in this area: there is a polynomial time reduction from Bayesian [2] incentive compatible mechanism design to Bayesian algorithm design for welfare maximization problems. This in turn shows the robustness to strategic behavior in Bayesian mechanism design, and indeed is a surprising victory for Bayesian

[1] In a mechanism design setting, each user has some specific value for any picked outcome by the mechanism. The total generated value after assigning an outcome is commonly refereed to as "social welfare".

[2] In a *Bayesian setting*, the valuations of the users are assumed to be drawn independently from common-knowledge prior distributions.

welfare optimization.

Unlike prior results, our reduction achieves exact incentive compatibility for problems with multidimensional and continuous type spaces. It turned out a key technical barrier preventing exact incentive compatibility in prior black-box reductions is giving a polynomial time algorithm to exactly sample from a distribution over outcomes that is a function of expectations of a set of input distributions, given only sample access to those input distributions. We overcome this barrier by employing and generalizing *Bernoulli Factories*, a compelling toolbox in applied probability that describes how to generate new random coins from old ones only by sampling. Finally, we incorporate techniques from online convex optimization and learning theory to demonstrate how efficient algorithms under the above mentioned computation model generate Bayesian black-box reductions.

1.2.2 The Need for Simple Mechanisms

One of the important aspects of mechanism design for complex real world environments such as ride sharing, online retail and cloud computing is the fact that the final mechanism will have human users. Accordingly, in most of these design scenarios it is critical to design mechanisms that are understandable by the users and have few parameters. Such mechanisms are informally called *simple* mechanisms. Interestingly, mechanisms for selling goods - commonly refereed to as *auctions* - that are considered to be simple are at the same time easy to optimize. As a result, especially for the objective of maximizing the revenue [3] of the

[3]In a mechanism design setting, the total collected payments by the auctioneer is commonly refereed to as "revenue".

designer, these mechanisms gained a lot of prevalence in electronic commerce. The quintessential examples are various forms of pricing mechanisms that are used for different electronic markets, e.g. selling items in online retails or selling computational resources in cloud computing platforms.

However, the theory of microeconomics, particularly the phenomenal work of Myerson (1981), proves more complicated forms for optimal revenue auctions in most of these settings. The mystery of this discrepancy between simple mechanisms used in practice versus optimal ones predicted by the theory initiates the following question: are there simple mechanisms whose performance is a good approximation to that of the optimal one? Chapter 4 of this dissertation tackles this question, as a continuation to great prior work on this subject initiated by Hartline and Roughgarden (2009). In particular, we look at simple pricing mechanisms for the *single-item sale*.

Consider the problem of selling one item to independent but non-identical buyers. A surprising result of Hartline and Roughgarden (2009) shows that using a universal price, a.k.a. *anonymous pricing*, guarantees no worse than a 4-approximation and it cannot be better than a 2-approximation for revenue maximization. The question of resolving the tight approximation factor (a number between 2 and 4) has remained open for the last half decade. In this dissertation, we make progress on this open problem by comparing the revenue of anonymous pricing to a standard upper bound on optimal revenue known as ex-ante pricing revenue. We prove that the worst-case ratio between these two quantities is e and our result is tight. A corollary of this theorem is the improved upper bounds by e (from 4) on the worst-case approximation factor of anonymous reserves and anonymous pricing with respect to the optimal auction. We also prove anonymous pricing cannot approximate the optimal revenue by a factor better than 2.23, which improves on the known lower bound of 2. The corollary relating anonymous pricing to the optimal auction also has implications on mechanism design for agents with multi-dimensional preferences (e.g., for multiple items; cf. Chawla et al. (2007)).

1.2.3 Online Nature and User Data

The traditional approach of microeconomic theory towards market design is to look at static settings, where all the agents/users are available during the trade and the market is not dynamic at all. However, in many real-world complex environments for mechanism design, e.g. airline pricing, ride sharing, or cloud service mechanisms, these assumptions are too simplistic. Indeed, these applications require an appropriate treatment that considers the online nature of these problems, e.g. users arrive over time and the trade happens in an online fashion, and takes into account the history of past trades to learn more about future trades. This provides an opportunity to incorporate techniques and ideas borrowed from *online learning*, i.e. the art of designing algorithms that make decisions on the fly and learn from past interactions, to design *online mechanisms*.

On a similar topic, designing near-optimal mechanisms with only samples from users' type distributions (rather than distributions themselves) is one of the emerging challenges of mechanism design for complex environments, due to the importance of incorporating users' data instead of prior information. As a canonical example, there has been great initial work by Cole and Roughgarden (2014) and Dhangwatnotai et al. (2014) to address the *sample complexity* of single-item revenue maximization. The objective of this line of work is to find out how many samples from buyers' distributions are needed to design near-optimal auctions. In Cole and Roughgarden (2014); Devanur et al. (2016); Huang et al. (2015) matching upper and lower sample complexity bounds for various auctions design problems have been proved, where these bounds are polynomials of the number n of buyers and $\epsilon > 0$ (so that a $(1 - \epsilon)$-approximation of the optimal revenue is achievable).

As mentioned earlier, many of the complex environments for mechanism design need online mechanisms. So, despite the success of the line of work on sample complexity of offline auctions, there is still an important research challenge stemming from a discrepancy between these results and the reality in many applications: holding all the samples in advance, versus collecting samples over time as the sequence of auctions progresses. In fact, online samples over time model an online market in which users arrive over time, rather than users being static. Now, the offline sample complexity problem requires a reformulation, where the objective is to design mechanisms that in the fastest time possible (i.e. minimum number of online samples) can achieve near-optimal revenue. In other words, we need online mechanisms with fast *convergence rate*.

In Chapter 5 of this dissertation, we start exploring the aforementioned challenge by using online learning theory. We consider an online version of the same problem when samples are chosen adversarially in an online fashion, and the seller runs a new auction every time a new sample arrives. Now, the main question is whether we can *learn* the optimal-in-hindsight auction in T rounds to get a $(1 - \epsilon)$-approximation with respect to this benchmark, while T is a polynomial in the number of bidders n and $\epsilon > 0$. In the full-information case, very surprisingly, we answer in the affirmative by proposing online learning algorithms for the single buyer case or multiple buyer case that achieve optimal polynomial upper-bounds for T, i.e. matching known offline sample complexity bounds. In our results in Chapter 5, the optimal learnability of the optimal revenue auction is a byproduct of expert-specific low-regret online learning algorithms, which we refer to as *multi-scale online learning*. The multi-scale online learning framework, which we will elaborate more in Chapter 5, should be also of independent interest for the theoretical machine learning community.

1.3 Organization

This dissertation is organized as follows. In Chapter 2 we review and introduce some tools and definitions that will act as the bedrock of future chapters. In Chapter 3 we discuss our treatment for handling incentive compatibility in complex environments, and explain how Bayesian welfare-maximizing mechanism design can be reduced to Bayesian welfare-maximizing algorithm design. In Chapter 4 we discuss simple vs. optimal mechanism design, and show how the simple anonymous pricing can be a constant approximation to the optimal revenue auction for single-item sale (under mild distributional assumptions). Finally, in Chapter 5 we discuss online mechanism design and show how online learning theory can be used to complement the line of work on sample complexity of offline auctions.

1.4 Bibliographical Note

The material in Chapter 3 is based on a joint work with Shaddin Dughmi, Jason Hartline, and Robert Kleinberg, in Proceedings of ACM 49th Symposium on Theory of Computing (STOC'17) (Dughmi et al., 2017). Chapter 4 is based on a joint work with Saeed Alaei, Jason Hartline, Manolis Pountourakis, and Yang Yuan, in Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS'15) (Alaei et al., 2015). Finally, the material in Chapter 5 is based on a joint work with Sébastien Bubeck, Nikhil Devanur and Zhiyi Huang, in Proceedings of ACM 18th Conference on Economics and Computation (EC'17) (Bubeck et al., 2017)

CHAPTER 2

BACKGROUND MATERIAL

2.1 General Basics of Bayesian Mechanism Design

Multi-parameter Bayesian setting. Suppose there are n agents, where agent k has private *type* t^k from *type space* $\mathcal{T}^k$. The *type profile* of all agents is denoted by $\mathbf{t} = (t^1, \ldots, t^n) \in \mathcal{T}^1 \times \ldots \times \mathcal{T}^n$. Moreover, we assume types are drawn independently from known prior distributions. For agent k, let F^k be the distribution of $t^k \in \mathcal{T}^k$ and $\mathbf{F} = F^1 \times \ldots \times F^n$ be the joint distribution of types. Suppose there is an *outcome space* denoted by $\mathcal{O}$. Agent k with type t^k has valuation $v(t^k, o)$ for outcome $o \in \mathcal{O}$, where $v : (\mathcal{T}^1 \cup \ldots \cup \mathcal{T}^n) \times \mathcal{O} \to [0, 1]$ is a fixed function. Note that we assume agent values are non-negative. In particular, in Chapter 3 we assume w.l.o.g. that values are bounded and in $[0, 1]$, and in Chapter 5 we assume values are bounded and in the range $[1, h]$ unless noted otherwise. Finally, we allow charging agents with non-negative money *payments* and we assume agents are *quasi-linear*, i.e. an agent with private type t has *utility* $u = v(t, o) - p$ for the outcome-payment pair (o, p).

Algorithms, mechanisms and interim rules. An *allocation algorithm* $\mathcal{A}$ is a mapping from type profiles $\mathbf{t}$ to outcome space $\mathcal{O}$. A (direct revelation) mechanism $\mathcal{M}$ is a pair of *allocation rule* and *payment rule* $(\mathcal{A}, \mathbf{p})$, in which $\mathcal{A}$ is an allocation algorithm and $\mathbf{p} = (p^1, \ldots, p^n)$ where each p^k (denoted by the payment rule for agent k) is a mapping from type profiles $\mathbf{t}$ to $\mathbb{R}_+$. In fact, one can think of the interaction between strategic agents and a mechanism as follows: agents submit their *reported types* $\mathbf{s} = (s^1, \ldots, s^n)$ and then the mechanism $\mathcal{M}$ picks the outcome $o = \mathcal{A}(\mathbf{s})$ and charges each agent k with its payment $p^k(\mathbf{s})$. We also consider *interim allocation rule*, which is the allocation from the perspective of one agent when the other agent's types are drawn from their prior distribution. More concretely, we abuse notation and define $\mathcal{A}^k(s^k) \triangleq \mathcal{A}(s^k, \mathbf{t}^{-k})$ to be the distribution over outcomes induced by $\mathcal{A}$ when agent k's type is s^k and other agent types are drawn from $\mathbf{F}^{-k}$. [1] Similarly, for agent k we define *interim payment rule* $p^k(s^k) \triangleq \mathbf{E}_{\mathbf{t}^{-k}\sim\mathbf{F}^{-k}}\left[p^k(s^k, \mathbf{t}^{-k})\right]$, and *interim value* $v^k(s^k) \triangleq \mathbf{E}_{\mathbf{t}^{-k}\sim\mathbf{F}^{-k}}\left[v(s^k, \mathcal{A}^k(s^k, \mathbf{t}^{-k}))\right]$. In most parts of this dissertation, we focus only on one agent, e.g. agent k, and we just work with the interim allocation algorithm $\mathcal{A}^k(.)$. When it is clear from the context, we drop the agent's superscript, and therefore $\mathcal{A}(s)$ denotes the distribution over outcomes induced by $\mathcal{A}(s, \mathbf{t}^{-k})$ when $\mathbf{t}^{-k} \sim \mathbf{F}^{-k}$.

Bayesian and dominant strategy truthfulness. We are mostly interested in designing mechanisms that are *interim truthful* (a.k.a. Bayesian truthful), i.e. every agent is best off by reporting her true type assuming all other agents' reported types are drawn independently from their prior type distributions. More precisely, a mechanism $\mathcal{M}$ is *Bayesian Incentive Compatible (BIC)* if for all agents k, and all types $s^k, t^k \in \mathcal{T}^k$,

$$\mathbf{E}_{\mathbf{t}^{-k}\sim\mathbf{F}{-k}}\left[v(t^k, \mathcal{A}^k(t^k))\right] - p^k(t^k) \geq \mathbf{E}_{\mathbf{t}^{-k}\sim\mathbf{F}{-k}}\left[v(t^k, \mathcal{A}^k(s^k))\right] - p^k(s^k) \tag{2.1}$$

As a stronger notion of truthfulness than Bayesian truthfulness, one can consider *dominant strategy truthfulness*. More precisely, a mechanism $\mathcal{M}$ is *Dominant Strategy Incentive Compatible (DSIC)* if for all agents k, and all types $s^k, t^k \in \mathcal{T}^k$ and all types $\mathbf{t}^{-k} \in \mathcal{T}^{-k}$,

$$v(t^k, \mathcal{A}(\mathbf{t})) - p^k(\mathbf{t}) \geq v(t^k, \mathcal{A}(s^k, \mathbf{t}^{-k})) - p^k(s^k, \mathbf{t}^{-k}) \tag{2.2}$$

[1] We use the notational convention that a superscript or subscript "$-k$" denotes omitting the k-th element of an n-tuple.

Moreover, an allocation algorithm $\tilde{\mathcal{A}}$ is said to be BIC (or DSIC) if there exists a payment rule $\tilde{\mathbf{p}}$ such that $\tilde{M} = (\tilde{\mathcal{A}}, \tilde{\mathbf{p}})$ is a BIC (or DSIC) mechanism. Throughout this dissertation, we use the terms Bayesian (or dominant strategy) truthful and Bayesian (or dominant strategy) incentive compatible interchangeably. For randomized mechanisms, DSIC and BIC solution concepts are defined by considering expectation of utilities of agents over mechanism's internal randomness.

Social welfare. In Chapter 3, we are considering mechanism design for maximizing *social welfare*, i.e. the sum of the utilities of agents and the mechanism designer. For quasi-linear agents, this quantity is in fact the sum of the valuations of the agents under the outcome picked by the mechanism. For the allocation algorithm $\mathcal{A}$, we use the notation $\text{val}(\mathcal{A})$ for the expected welfare of this allocation and $\text{val}^k(\mathcal{A})$ for the expected value of agent k under this allocation, i.e. $\text{val}(\mathcal{A}) \triangleq \mathbf{E}_{\mathbf{t}\sim\mathbf{F}}\left[\sum_k v(t^k, \mathcal{A}(\mathbf{t}))\right]$ and $\text{val}^k(\mathcal{A}) \triangleq \mathbf{E}_{\mathbf{t}\sim\mathbf{F}}\left[v(t^k, \mathcal{A}(\mathbf{t}))\right]$. If a mechanism maximizes social welfare, it is called an *efficient* mechanism.

Revenue of the Auctioneer. In Chapter 4 and Chapter 5, we are considering mechanism design for maximizing *auctioneer's expected revenue*, i.e. the expected sum of payments collected from agents by the principal who runs the mechanism. The expectation is normally taken over mechanism's internal randomness, and also the randomness of agent types. For mechanism M= $(\mathcal{A}, \mathbf{p})$, we use the notation $\text{Rev}(M)$ for the expected revenue of this mechanism, i.e. $\text{Rev}(M) \triangleq \mathbf{E}_{\mathbf{t}\sim\mathbf{F}}\left[\sum_k p^k(\mathbf{t})\right]$. If a (Bayesian) mechanism maximizes the generated revenue, it is called an *optimal* or *Bayesian optimal* mechanism.

Single-item Bayesian setting. In the special case of a single-item environment, we assume there are n agents and there is a single item that can only be sold to one of these agents. Each agent k has a private value v_k for the item, which is drawn from a distribution F_k with cumulative distribution function (CDF) denoted by $F_k(\cdot)$. In a *direct revelation* auction for the single-item setting, the agents submit their reported values, a.k.a. *bids*, to the mechanism. The mechanism, a.k.a. the *auction*, then allocates the item to one of the bidders and charges each agent k with a payment p^k.

Revenue curves and regular instances. In a Bayesian single-item setting, the *revenue curve* $R_i(q) = q \cdot F_i^{-1}(1-q)$ gives the expected revenue obtained by selling an item to agent i with probability exactly q, i.e., by posting price $F_i^{-1}(1-q)$. The agent is *regular* if its revenue curve $R_i(q)$ is concave in q. An n-agent instance $\mathcal{I} = \{F_i\}_{i=1}^n$ is regular if each agent's distribution is regular. The family of all regular instances for all $n \geq 1$ is denoted by $\textsc{Reg}$.

2.2 Basics and Notations Specific for Chapter 3

In Chapter 3, we heavily incorporate a tool from applied probability that is called *Bernoulli factory*. In this section, we introduce this toolbox by providing some basic notations and reviewing basic results from the literature. We will build parts of our result in Chapter 3 on top of these basic, yet important, results. We recommend referring back to this section concurrent with reading Chapter 3.

Bernoulli factory problem. Throughout this dissertation, we use the terms *Bernoulli* and *coin* to refer to distributions over $\{1, 0\}$ and $\{\texttt{heads}, \texttt{tails}\}$, interchangeably. The Bernoulli factory problem is about generating new coins from old ones.

Definition 2.2.1 (Keane and O'Brien, 1994)**.** Given function $f : (0, 1) \to (0, 1)$, the *Bernoulli factory* problem is to output a sample of a Bernoulli variable with bias $f(p)$ (i.e. an $f(p)$-coin), given black-box access to independent samples of a Bernoulli distribution with bias $p \in (0, 1)$ (i.e. a p-coin). [2]

To illustrate the Bernoulli factory model, consider the examples of $f(p) = p^2$ and $f(p) = e^{p-1}$. For the former one, it is enough to flip the p-coin twice and output 1 if both flips are 1, and 0 otherwise. For the latter one, the Bernoulli factory is still simple but more interesting: draw K from the Poisson distribution with parameter $\lambda = 1$, flip the p-coin K times and output 1 if all coin flips where 1, and 0 otherwise (see below).[3] The question of characterizing functions f for which there is an algorithm for sampling $f(p)$-coins from p-coins has been the main subject of interest in this literature (Keane and O'Brien, 1994; Nacu and Peres, 2005). In particular, Keane and O'Brien (1994) provide necessary and sufficient conditions for f, under which an algorithm for the Bernoulli factory exists. Moreover, Nacu and Peres (2005) suggest an algorithm for simulating an $f(p)$-coin based on polynomial envelopes of f. The canonical challenging problem of Bernoulli factories – and a primitive in the construction of more general Bernoulli factories – is the *Bernoulli Doubling* problem: $f(p) = 2p$ for $p \in (0, 1/2)$. See Łatuszyński (2010) for a survey on this topic.

[2] The desired algorithm is also assumed to have access to an unlimited number of samples from an unbiased coin, or in other words is randomized.

[3] The Poisson distribution with parameter λ has probability of $K = k$ as $\lambda^k e^{-\lambda}/k!$.

Questions in Bernoulli factories can be generalized to multiple input coins. Given $f : (0,1)^m \to (0,1)$, the goal is tom sample from a Bernoulli with bias $f(p_1, \ldots, p_m)$ given sample access to m independent Bernoulli variables with unknown biases $\mathbf{p} = (p_1, \ldots, p_m)$. Linear functions f were studied and solved by Huber (2015). For example, the special case $m = 2$ and $f(p_1, p_2) = p_1 + p_2$, a.k.a., *Bernoulli Addition*, can be solved by reduction to the Bernoulli Doubling problem (formalized below).

Building blocks to apply Bernoulli factories in mechanism design. Questions in Bernoulli factories can be generalized to allow input distributions over real numbers on the unit interval $[0,1]$ (rather than Bernoullis over $\{0,1\}$). In this generalization the question is to produce a Bernoulli with bias $f(\mu)$ with sample access to draws from a distribution supported on $[0,1]$ with expectation μ. These problems can be easily solved by reduction to the Bernoulli factory problem. Below we enumerate the important building blocks for Bernoulli factories:

0. *Continuous to Bernoulli*: One can implement Bernoulli with bias μ with one sample from distribution $\mathcal{D}$ with expectation μ. Algorithm:
   - Draw $Z \sim \mathcal{D}$ and $P \sim \text{Bern}[Z]$.
   - Output P.
1. *Bernoulli Down Scaling*: One can implement $f(p) = \lambda \cdot p$ for $\lambda \in [0,1]$ with one sample from $\text{Bern}[p]$. Algorithm:
   - Draw $\Lambda \sim \text{Bern}[\lambda]$ and $P \sim \text{Bern}[p]$.
   - Output $\Lambda \cdot P$ (i.e., 1 if both coins are 1, otherwise 0).

2. *Bernoulli Doubling:* One can implement $f(p) = 2p$ for $p \in (0, 1/2 - \delta]$ with $O(1/\delta)$ samples from Bern[p] in expectation. The algorithm is complicated, see Nacu and Peres (2005).

3. *Bernoulli Probability Generating Function:* Can implement $f(p) = \mathbf{E}_{k\sim\mathcal{D}}\left[p^k\right]$ for distribution $\mathcal{D}$ over non-negative integers with $\mathbf{E}_{K\sim\mathcal{D}}[K]$ samples from Bern[p] in expectation. Algorithm:

   - Draw $K \sim \mathcal{D}$ and $P_1, \ldots, P_K \sim$ Bern[p] (i.e., K samples).
   - Output $\prod_i P_i$ (i.e., 1 if all K coins are 1, otherwise 0).

4. *Bernoulli Exponentiation:* One can implement $f(p) = \exp(\lambda(p - 1))$ for $p \in [0, 1]$ and non-negative constant λ with λ samples from Bern[p] in expectation. Algorithm: Apply the Bernoulli Probability Generating Function algorithm for the Poisson distribution with parameter λ.

5. *Bernoulli Averaging:* One can implement $f(p_1, p_2) = (p_1 + p_2)/2$ with one sample from Bern[p_1] or Bern[p_2]. Algorithm:

   - Draw $Z \sim$ Bern[1/2], $P_1 \sim$ Bern[p_1], and $P_2 \sim$ Bern[p_2].
   - Output P_{Z+1}.

6. *Bernoulli Addition:* One can implement $f(p_1, p_2) = p_1 + p_2$ for $p_1 + p_2 \in [0, 1 - \delta]$ with $O(1/\delta)$ samples from Bern[p_1] and Bern[p_2] in expectation. Algorithm: Apply Bernoulli Doubling to Bernoulli Averaging.

It may seem counterintuitive that Bernoulli Doubling is much more challenging than Bernoulli Down Scaling. Notice, however, that for a coin with bias $p = 1/2$, Bernoulli Doubling with a finite number of coin flips is impossible. The doubled coin must be deterministically heads, while any finite sequence of coin flips of Bern[1/2] has non-zero probability of occuring. On the other hand a coin with bias $p = 1/2 - \delta$ for some small δ has a similar probability of each sequence but Bernoulli Doubling must sometimes output tails. Thus, Bernoulli Doubling must require a number of coin flips that goes to infinity as δ goes to zero.

2.3 Basics and Notations Specific for Chapter 4

In Chapter 4 of this dissertation, we compare various simple mechanisms for selling a single item to independent agents in the Bayesian setting, and also use various benchmarks. Moreover, we make use of a particular class of distributions in our analysis, which we call *triangular revenue curve distribution*. We elaborate on these definitions, notations and mechanisms below. We recommend referring back to this section concurrent with reading Chapter 4.

Anonymous pricings. In a Bayesian single-item setting, an anonymous pricing is a mechanism that posts a price p that is bought by an arbitrary agent whose value is at least the posted price (if one exists). The expected revenue of the anonymous pricing p for instance $\mathcal{I} = \{F_i\}_{i=1}^n$ is

$$\textsc{PriceRev}(\mathcal{I}, p) \triangleq p \cdot \left(1 - \prod\nolimits_i F_i(p)\right). \tag{2.3}$$

The expected revenue of the optimal anonymous pricing is

$$\textsc{OptPriceRev}(\mathcal{I}) \triangleq \max_{p \in \mathbb{R}_+} \textsc{PriceRev}(\mathcal{I}, p). \tag{2.4}$$

Ex ante relaxation and optimal auctions. In a Bayesian single-item setting, and in general in any single dimensional setting where the allocation $x_k(\mathbf{v})$ for each agent is in $\{0,1\}$, the optimal auction was characterized by Myerson (1981). In this characterization, for each agent k there exists a mapping $\bar{\phi}_k : \mathbb{R} \to \mathbb{R}$ that maps values to *(ironed) virtual values*. The Myerson optimal auction then would be maximizing the *ironed virtual welfare*, i.e. $\sum_{k=1}^{n} x_k(v_k)\bar{\phi}_k(v_k)$, and charging appropriate incentive compatible payments through *Myerson payment rule*. This characterization, though complex, is the foundation of modern auction theory. In the single-item setting, we also consider a non-implementable mechanism, called *ex ante relaxation*, as a relaxation to the optimal auction. The revenue of the *ex ante relaxation*, which allocates to one agent in expectation, gives an upper bound on the revenue of the optimal auction. For any instance $\mathcal{I}$, it can be easily expressed in terms of the revenue curves of the agents.

$$\begin{aligned} \textsc{ExAnteRev}(\mathcal{I}) \triangleq \quad & \max & & \sum\nolimits_{i=1}^{n} R_i(q_i) \\ & \text{subject to} & & \sum\nolimits_{i=1}^{n} q_i \leq 1 \\ & & & q_i \geq 0 \qquad \forall i \in \{1,\ldots,n\}\,. \end{aligned} \tag{2.5}$$

Triangular revenue curve instances. In Chapter 4, we heavily use distributions with triangular-shaped revenue curves. A *triangular revenue curve* distribution, denoted $\text{Tri}(\bar{v},\bar{q})$ with parameters $\bar{v} \in (0,\infty)$ and $\bar{q} \in [0,1]$, has a cumulative density function given by

$$F(p) = \begin{cases} 1 & p \geq \bar{v} \\ \frac{p\cdot(1-\bar{q})}{p\cdot(1-\bar{q})+\bar{v}\bar{q}} & 0 \leq p < \bar{v} \end{cases} \qquad \forall p \in \mathbb{R}_+. \tag{2.6}$$

The revenue curve corresponding to the above distribution has the form of a triangle with vertices at $(0,0)$, $(\bar{q},\bar{v}\bar{q})$, and $(1,0)$ as illustrated in Figure 2.1; the revenue curve's concavity implies that the distribution is regular. Note that the CDF is discontinuous at $\bar{v}$ which corresponds to a point mass of $\bar{q}$ at value $\bar{v}$.

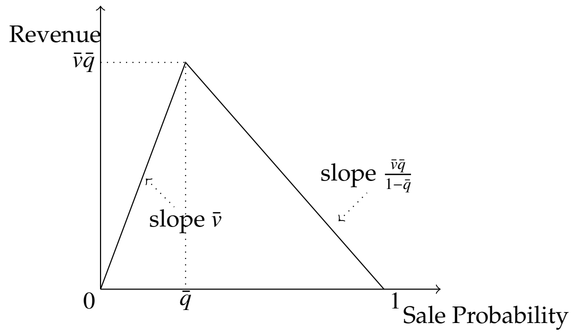

Figure 2.1: Revenue curve of distribution $\mathrm{Tri}(\bar{v}, \bar{q})$.

A *triangular revenue curve instance* is given by $\mathcal{I} = \{\mathrm{Tri}(\bar{v}_i, \bar{q}_i)\}_{i=1}^n$ with $\sum_{i=1}^n \bar{q}_i \leq 1$; with respect to it the revenue of anonymous pricing p and the ex ante relaxation are given by

$$\textsc{PriceRev}(\mathcal{I}, p) = p \cdot \left(1 - \prod\nolimits_{i:\bar{v}_i \geq p} \left(1 + \frac{\bar{v}_i \bar{q}_i}{p \cdot (1 - \bar{q}_i)}\right)^{-1}\right), \tag{2.7}$$

$$\textsc{ExAnteRev}(\mathcal{I}) = \sum\nolimits_{i=1}^n \bar{v}_i \bar{q}_i. \tag{2.8}$$

2.4 Basics and Notations Specific for Chapter 5

In Chapter 5 we heavily use a standard algorithm from the online learning literature called *Online Mirror Descent (OMD)*. In this section, we describe this algorithm for the basic full-information *best-expert* online learning problem. We recommend referring back to this section concurrent with reading Chapter 5.

Full information and bandit information online learning. In the *best-expert problem*, there are T discrete time instances $t = 1, \ldots, T$. We also have an action set A (we use actions and experts interchangeably). For simplicity, we mainly consider countable and finite actions/experts sets (although there are some exceptions, e.g. Chapter 5, Section 5.3.3). At each time t, given the history of past actions and rewards, the online algorithm selects an expert i_t (it can be randomized). The adversary then reveals a reward function $\mathbf{g}(t)$, and the expected gain of the algorithm will be $\mathbf{E}\left[g_{i_t}(t)\right]$. In the partial information version of this problem, known as *multi-armed bandit problem*, the adversary only reveals the reward of the specific action that has been picked by the algorithm at that round. We assume the adversary is *oblivious*, i.e. it picks the entire sequence $\mathbf{g}(1), \ldots, \mathbf{g}(T)$ adversarially before the game begins.

Online Mirror Descent (OMD) algorithm. Fix an open convex set $\mathcal{D}$ and its closure $\bar{\mathcal{D}}$, which in our case are $\mathbb{R}^A_{>0}$ and $\mathbb{R}^A_+$ respectively, and a closed, convex action set $\mathcal{A} \subset \bar{\mathcal{D}}$, which in our case is Δ_A, i.e. the set of all probability distributions over experts in A. At the heart of an OMD algorithm there is a *Legendre* function $F : \bar{\mathcal{D}} \to \mathbb{R}$, i.e. a strictly convex function that admits continuous first order partial derivatives on $\mathcal{D}$ and $\lim_{x \to \bar{\mathcal{D}} \setminus \mathcal{D}} ||\nabla F(x)|| = +\infty$, where $\nabla F(.)$ denotes the gradient map of F. One can think of OMD as a member of *projected gradient descent* algorithms, where the *gradient update* happens in the *dual space* $\nabla F(\mathcal{D})$ rather than in primal $\mathcal{D}$, and the *projection* is defined by using the *Bregman divergence* associated with F rather than ℓ_2-distance.

Definition 2.4.1 (Bregman Divergence)**.** Given a Legendre function F over Δ_A, the Bregman divergence associated with F, denoted as $D_F : \mathbb{R}^A \times \mathbb{R}^A \to \mathbb{R}$, is defined by

$$D_F(x, y) = F(x) - F(y) - (x - y)^T \nabla F(y) \ .$$

Definition 2.4.2 (Online Mirror Descent)**.** Suppose F is a Legendre function. At every time $t \in [T]$, the online mirror descent algorithm with Legendre function F selects an expert drawn from distribution $\mathbf{p}(t)$, and then updates $\mathbf{w}(t)$ and $\mathbf{p}(t)$

given rewards $\mathbf{g}(t)$ by:

Gradient update:

$$\nabla F(\mathbf{w}(t+1)) = \nabla F(\mathbf{p}(t)) + \eta \cdot \mathbf{g}(t) \Rightarrow \mathbf{w}(t+1) = (\nabla F)^{-1}\left(\nabla F(\mathbf{p}(t)) + \eta \cdot \mathbf{g}(t)\right) \tag{2.9}$$

Bregman projection:

$$\mathbf{p}(t+1) = \operatorname*{argmin}_{\mathbf{p} \in \Delta_A} \left(D_F(\mathbf{p}, \mathbf{w}(t+1))\right) \tag{2.10}$$

where $\eta > 0$ is called the learning rate of OMD.

We use the following standard *regret bound* of OMD [4] that compares the gain of the algorithm with any fixed (randomized) action in hindsight.

Lemma 2.4.1. *For any learning rate parameter $0 < \eta \leq 1$ and any benchmark distribution $\mathbf{q}$ over A, the OMD algorithm with Legendre function $F(.)$ admits the following:*

$$\sum_{t\in[T]} \mathbf{g}(t) \cdot \left(\mathbf{q} - \mathbf{p}(t)\right) \leq \frac{1}{\eta} \sum_{t\in[T]} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) + \frac{1}{\eta} D_F(\mathbf{q}, \mathbf{p}(1)) \tag{2.11}$$

[4] Refer to Bubeck (2011) for a thorough discussion on OMD. For completeness, a proof is also provided in the appendix, Section C.1.5

CHAPTER 3

REDUCING MECHANISM DESIGN TO ALGORITHM DESIGN

In this chapter, we investigate the computational complexity gap between Bayesian algorithm design and mechanism design for welfare maximization. To this end, we start by looking at the problem of *Bayesian black-box reductions* in mechanism design. In the black-box computational model, one is willing to design Bayesian truthful mechanisms while having access to a black-box allocation algorithm. The objective is to design such a mechanism in polynomial time, i.e. with polynomial (in number of agents) query calls to the black-box and polynomial extra computation time. Moreover, the designed mechanism should have negligible loss in expected social welfare compared to the expected social welfare of the allocation black-box.

Organization of the chapter. In Section 3.1 we define the Bayesian black-box reduction and review the literature. In Section 3.2 we summarize our approach and techniques. In Section 3.3 we give detailed proofs of our results. Finally, we conclude by summarizing the chapter and proposing some interesting open problems in Section 3.4.

3.1 Preliminary

In this section, we give an overview on the Bayesian black-box reduction problem. We define the problem formally in Section 3.1.1 and review some related work in the literature in Section 3.1.2.

3.1.1 Problem Definition : Black-box Reduction

A central question at the interface between algorithms and economics is on the existence of black-box reductions for mechanism design. Given black-box access to any algorithm that maps inputs to outcomes, can a mechanism be constructed that induces agents to truthfully report the inputs and produces an outcome that is as good as the one produced by the algorithm? The mechanism must be computationally tractable, specifically, making no more than a polynomial number of elementary operations and black-box calls to the algorithm.

A line of research initiated by Hartline and Lucier (2010, 2015) demonstrated that, for the welfare objective, Bayesian black-box reductions exist.[1] In the Bayesian setting, agents' types are drawn from a distribution. The algorithm is assumed to obtain good welfare for types from this distribution.[2] The constructed mechanism is an approximation scheme; For any ϵ it gives a mechanism that is Bayesian incentive compatible (Definition 3.3.4) and obtains a welfare that is an additive ϵ form the algorithms welfare. Before formalizing this problem, for further details on Bayesian mechanism design and our set of notations in this chapter, we refer the reader to Chapter 2, Section 2.1.

Definition 3.1.1 (BIC black-box reduction problem)**.** Given black-box oracle access to an allocation algorithm $\mathcal{A}$, simulate a Bayesian incentive compatible allocation algorithm $\tilde{\mathcal{A}}$ that approximately preserves welfare, i.e. for every agent a, $\mathrm{val}^{\mathsf{a}}(\tilde{\mathcal{A}}) \geq \mathrm{val}^{\mathsf{a}}(\mathcal{A}) - \epsilon$, and runs in time $\mathrm{poly}(n, \frac{1}{\epsilon})$.

[1] One could also consider approximately preserving objectives other than welfare. However, Chawla et al. (2012) have shown that BIC black-box reductions for the makespan objective cannot be computationally efficient in general.

[2] Although this assumption is not necessary for the reduction to work, the black-box reduction in algorithmic mechanism design makes more sense when the algorithm is assumed to obtain good welfare in a Bayesian sense.

3.1.2 Related Work

In this literature, Hartline and Lucier (2010, 2015) solve the case of single-dimensional agents and Hartline et al. (2011, 2015) solve the case of multi-dimensional agents with discrete type spaces. For the relaxation of the problem where only approximate incentive compatibility is required, Bei and Huang (2011) solve the case of multi-dimensional agents with discrete type space, and Hartline et al. (2011, 2015) solve the general case by (1) achieving exact BIC for discrete type spaces, and (2) achieving approximate BIC for general multi-dimensional type spaces. These reductions are approximation schemes that are polynomial in the number of agents, the desired approximation factor, and a measure of the size of the agents' type spaces (i.e., its dimension).

3.2 Our Approach in a Nutshell

We resolve a five-year-old open question from Hartline et al. (2011, 2015): *There is a polynomial time reduction from Bayesian incentive compatible mechanism design to Bayesian algorithm design for welfare maximization problems.* [3] The key distinction between our result and those of Hartline et al. (2011, 2015) is that both (a) the agents' preferences can be multi-dimensional and from a continuous space (rather than single-dimensional or from a discrete space), and (b) the resulting mechanism is exactly Bayesian incentive compatible (rather than approximately Bayesian incentive compatible).

[3]A Bayesian algorithm is one that performs well in expectation when the input is drawn from a known distribution. By polynomial time, we mean polynomial in the number of agents and the combined "size" of their type spaces.

A mechanism solicits preferences from agents, i.e., how much each agent prefers each outcome, and then chooses an outcome. *Incentive compatibility* of a mechanism requires that, though agents could misreport their preferences, it is not in any agent's best interest to do so. A quintessential research problem at the intersection of mechanism deign and approximation algorithms is to identify black-box reductions from approximation mechanism design to approximation algorithm design. The key algorithmic property that makes a mechanism incentive compatible is that, from any individual agent's perspective, it must be *maximal-in-range*, specifically, the outcome selected maximizes the agent's utility less some cost that is a function of the outcome (e.g., this cost function can depend on other agents' reported preferences.).

The black-box reductions from Bayesian mechanism design to Bayesian algorithm design in the literature are based on obtaining an understanding of the distribution of outcomes produced by the algorithm through simulating the algorithm on samples from agents' preferences. Notice that, even for structurally simple problems, calculating the exact probability that a given outcome is selected by an algorithm can be #P-hard. For example, Hartline et al. (2015) show such a result for calculating the probability that a matching in a bipartite graph is optimal, for a simple explicitly given distribution of edge weights. A black-box reduction for mechanism design must therefore produce exactly maximal-in-range outcomes merely from samples. This challenge motivates new questions for algorithm design from samples.

3.2.1 The Expectations from Samples Model

In traditional algorithm design, the inputs are specified to the algorithm exactly. In this chapter, we formulate the *expectations from samples* model. This model calls for drawing an outcome from a distribution that is a precise function of the expectations of some random sources that are given only by sample access. Formally, a problem for this model is described by a function $f : [0, 1]^n \to \Delta(X)$ where X is an abstract set of feasible outcomes and $\Delta(X)$ is the family of probability distributions over X. For any n input distributions on support $[0, 1]$ with unknown expectations $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$, an algorithm for such a problem, with only sample access to each of the n input distributions, must produce sample outcome from X that is distributed exactly according to $f(\mu_1, \ldots, \mu_n)$.

Producing an outcome that is approximately drawn according to the desired distribution can typically be done from estimates of the expectations formed from sample averages (a.k.a., Monte Carlo sampling). On the other hand, exact implementation of many natural functions f is either impossible for information theoretic reasons or requires sophisticated techniques. Impossibility generally follows, for example, when f is discontinuous. The literature on *Bernoulli Factories* (e.g., Keane and O'Brien, 1994), which inspires our generalization to the expectations from samples model and provides some of the basic building blocks for our results, considers the special case where the input distribution and output distribution are both Bernoullis (i.e., supported on $\{0, 1\}$).

We propose and solve two fundamental problems for the expectations from samples model. The first problem considers the biases $\mathbf{p} = (p_1, \ldots, p_m)$ of m Bernoulli random variables as the marginal probabilities of a distribution on $\{1, \ldots, m\}$ (i.e., $\mathbf{p}$ satisfies $\sum_i p_i = 1$) and asks to sample from this distribution. We develop an algorithm that we call the Bernoulli Race to solve this problem.

The second problem corresponds to the "soft maximum" problem given by a regularizer that is a multiple $1/\lambda$ of the Shannon entropy function $H(\mathbf{p}) = -\sum_i p_i \log p_i$. The marginal probabilities on outcomes that maximize the expected value of the distribution over outcomes less the cost of the negative entropy regularizer are given by exponential weights,[4] i.e., the function outputs i with probability proportional to $e^{\lambda p_i}$. A straightforward exponentiation and then reduction to the Bernoulli Race above does not have polynomial sample complexity. We develop an algorithm that we call the Fast Exponential Bernoulli Race to solve this problem.

3.2.2 Building on Top of Previous Black-box Reductions

A special case of the problem that we must solve to apply the standard approach to black-box reductions is the *single-agent multiple-urns problem*. In this setting, a single agent faces a set X of urns, and each urn contains a random object whose distribution is unknown, but can be sampled. The agent's type determines his utility for each object; fixing this type, urn i is associated with a random real-valued reward with unknown expectation μ_i. Our goal is to allocate the agent his favorite urn, or close to it.

As described above, incentive compatibility requires an algorithm for selecting a high-value urn that is maximal-in-range. If we could exactly calculate the expected values $\mu_1, \ldots, \mu_n$ from the agent's type, this problem is trivial both algorithmically and from a mechanism design perspective: simply solicit the

[4]This is a standard relationship that has, for example, been employed in previous work in mechanism design (e.g., Huang and Kannan, 2012).

agent's type t then allocate him the urn with the maximum $\mu_i = \mu_i(t)$. As described above, with only sample access to the expected values of each urn, we cannot implement the exact maximum. Our solution is to apply the Fast Exponential Bernoulli Race as a solution to the regularized maximization problem in the expectations from samples model. This algorithm – with only sample access to the agent's values for each urn – will assign the agent to a random urn with a high expected value and is maximal-in-range.

The multi-agent reduction from Bayesian mechanism design to Bayesian algorithm design of Hartline et al. (2011, 2015) is based on solving a matching problem between multiple agents and outcomes, where an agent's value for an outcome is the expectation of a random variable which can be accessed only through sampling.[5] Specifically, this problem generalizes the above-described single-agent multiple-urns problem to the problem of matching agents to urns with the goal of approximately maximizing the total weight of the matching (the social welfare). Again, for incentive compatibility we require this expectations from samples algorithm to be maximal-in-range from each agent's perspective. Using methods from Agrawal and Devanur's (2015) work on stochastic online convex optimization, we reduce this matching problem to the single-agent multiple-urns problem.

As stated in the opening paragraph, our main result – obtained through the approach outlined above – is a polynomial time reduction from Bayesian incentive compatible mechanism design to Bayesian algorithm design. The analysis assumes that agents' values are normalized to the $[0, 1]$ interval and gives additive loss in the welfare. The reduction is an approximation scheme and the

[5] Bei and Huang (2011) independently discovered a similar reduction based on solving a fractional assignment problem. Their reduction applies to finite, discrete type spaces and is approximately Bayesian incentive compatible.

dependence of the runtime on the additive loss is inverse polynomial. The reduction depends polynomially on a suitable notion of the size of the space of agent preferences. For example, applied to environments where agents have preferences that lie in high-dimensional spaces, the runtime of the reduction depends polynomially on the number of points necessary to approximately cover each agent's space of preferences. More generally, the bounds we obtain are polynomial in the bounds of Hartline et al. (2011, 2015) but the resulting mechanism, unlike in the proceeding work, is exactly Bayesian incentive compatible.

3.2.3 Organization of the Technical Parts

The organization of the Section 3.3, which includes all the technical details, separates the development of the expectations from samples model and its application to black-box reductions in Bayesian mechanism design. Section 3.3.1 defines two central problems in the expectations from samples model, sampling from outcomes with linear weights and sampling from outcomes with exponential weights, and gives algorithms for solving them. We return to mechanism design problems in Section 3.3.2 and solve the single-agent multiple urns problem. In Section 3.3.3 we give our main result, the reduction from Bayesian mechanism design to Bayesian algorithm design. Throughout the next section, we heavily use the Bernoulli factory toolbox that we introduced earlier in Chapter 2, Section 2.2.

3.3 Detailed Result: a BIC Black-box Reduction

In this section, we provide all the details needed to explain and prove our BIC black-box reduction, by exploiting different technical pieces and gluing them together at the end.

3.3.1 The Expectations from Samples Model

The expectations from samples model is a combinatorial generalization of the Bernoulli factory problem. The goal is to select an outcome from a distribution that is a function of the expectations of a set of input distributions. These input distributions can be accessed only by sampling.

Definition 3.3.1. Given function $f : (0,1)^n \to \Delta(X)$ for domain X, the *expectations from samples* problem is to output a sample from $f(\boldsymbol{\mu})$ given black-box access to independent samples from n distributions supported on $[0,1]$ with expectations $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \in (0,1)^n$.

Without loss of generality, by the Continuous to Bernoulli construction of Section 2.2, the input random variables can be assumed to be Bernoullis and, thus, this expectations of samples model can be viewed as a generalization of the Bernoulli factory question to output spaces X beyond $\{0,1\}$. In this section we propose and solve two fundamental problems for the expectations of samples model. In these problems the outcomes are the a finite set of m outcomes $X = \{1, \ldots, m\}$ and the input distributions are m Bernoulli distributions with biases $\mathbf{p} = (p_1, \ldots, p_m)$.

In the first problem, biases correspond to the marginal probabilities with which each of the outcomes should be selected. The goal is to produce random i from X so that the probability of i is exactly its marginal probability p_i. More generally, if the biases do not sum to one, this problem is equivalently the problem of *random selection with linear weights*.

The second problem we solve corresponds to a regularized maximization problem, or specifically *random selection from exponential weights*. For this problem the baiases of the m Bernoulli input distributions correspond to the weights of the outcomes. The goal is to produce a random i from X according to the distribution given by exponential weights, i.e., the probability of selecting i from X is $e^{\lambda p_i} / \sum_j e^{\lambda p_j}$.

Random Selection with Linear Weights

Definition 3.3.2 (Random Selection with Linear Weights). The *random selection with linear weights* problem is to sample from the probability distribution $f(\mathbf{v})$ defined by $\mathbf{Pr}_{I \sim f(\mathbf{v})}[I = i] = v_i / \sum_j v_j$ for each i in $\{1, \ldots, m\}$ with only sample access to distributions with expectations $\mathbf{v} = (v_1, \ldots, v_m)$.

We solve the random selection with linear weights problem by an algorithm that we call the *Bernoulli race* (Algorithm 1). The algorithm repeatedly picks a coin uniformly at random and flips it. The winning coin is the first one to come up heads in this process.

Theorem 3.3.1. *The Bernoulli Race (Algorithm 1) samples with linear weights (Definition 3.3.2) with an expected $m / \sum_i v_i$ samples from input distributions with biases $v_1, \ldots, v_n$.*

Algorithm 1: Bernoulli Race

input sample access to m coins with biases $v_1, \dots, v_m$.
loop
    Draw I uniformly from $\{1, \dots, m\}$ and draw P from input distribution I.
    If P is `heads` then output I and halt.
end loop

Proof. At each iteration, the algorithm terminates if the flipped coin outputs 1 and iterates otherwise. Since the coin is chosen uniformly at random, the probability of termination at each iteration is $\frac{1}{m}\sum_i v_i$. The total number of iterations (and number of samples) is therefore a geometric random variable with expectation $m/\sum_i v_i$.

The selected outcome also follows the desired distribution, as shown below.

$$\begin{aligned}\mathbf{Pr}[i \text{ is selected}] &= \sum_{k=1}^{\infty} \mathbf{Pr}[i \text{ is selected at time } k]\ \mathbf{Pr}\Big[\text{algorithm reaches time } k\Big] \\ &= \frac{v_i}{m}\sum_{k=1}^{\infty}\left(1 - \frac{1}{m}\sum\nolimits_j v_j\right)^{k-1} = \frac{\frac{v_i}{m}}{\frac{1}{m}\sum_j v_j} = \frac{v_i}{\sum_j v_j}.\end{aligned}$$

□

Random Selection with Exponential Weights

Definition 3.3.3 (Random Selection with Exponential Weights). For parameter $\lambda > 0$, the *random selection with exponential weights* problem is to sample from the probability distribution $f(\mathbf{v})$ defined by $\mathbf{Pr}_{I\sim f(\mathbf{v})}[I = i] = \exp(\lambda v_i)/\sum_j \exp(\lambda v_j)$ for each i in $\{1, \dots, m\}$ with only sample access to distributions with expectations $\mathbf{v} = (v_1, \dots, v_m)$.

The *Basic Exponential Bernoulli Race*, below, samples from the exponential weights distribution. The algorithm follows the paradigm of picking one of the input distributions, exponentiating it, sampling from the exponentiated distribution, and repeating until one comes up heads. While this algorithm does not generally run in polynomial time, it is a building block for one that does.

Algorithm 2: The Basic Exponential Bernoulli Race (with parameter $\lambda > 0$)

1: **input** Sample access to m coins with biases $v_1, \ldots, v_m$.

2: For each i, apply Bernoulli Exponentiation to coin i to produce coin with bias $\tilde{v}_i = \exp(\lambda(v_i - 1))$.

3: Run the Bernoulli Race on the coins with biases $\tilde{\mathbf{v}} = (\tilde{v}_1, \ldots, \tilde{v}_m)$.

Theorem 3.3.2. *The Basic Exponential Bernoulli Race (Algorithm 2) samples with exponential weights (Definition 3.3.3) with an expected $\lambda m e^{\lambda(1-v_{\max})}$ samples from input distributions with biases $v_1, \ldots, v_n$ and $v_{\max} = \max_i v_i$.*

Proof. The correctness and runtime follows from the correctness and runtimes of Bernoulli Exponentiation and the Bernoulli Race.

□

The Fast Exponential Bernoulli Race

Sampling from exponential weights is typically used as a "soft maximum" where the parameter λ controls how close the selected outcome is to the true maximum. For such an application, exponential dependence on λ in the runtime would be prohibitive. Unfortunately, when $v_{\max}$ is bounded away from one, the runtime of the Basic Logistic Bernoulli Race (Algorithm 2; Theorem 3.3.2) is exponential in λ. A simple observation allows allows the resolution of this issue: the exponential weights distribution is invariant to any uniform additive shift of all weights. This section applies this idea to develop the *Fast Logistic Bernoulli Race*.

Observe that for any given parameter ϵ, we can easily implement a Bernoulli random variable Z whose bias z is within an additive ϵ of $v_{\max}$. Note that, unlike the other algorithms in this section, a precise relationship between z and $v_1, \ldots, v_m$ is not required.

Lemma 3.3.3. *For parameter $\epsilon \in (0, 1]$, there is an algorithm for sampling from a Bernoulli random variable with bias $z \in [v_{\max} - \epsilon, v_{\max} + \epsilon]$, where $v_{\max} = \max_i v_i$, with $O(\frac{m}{\epsilon^2} \cdot \log(\frac{m}{\epsilon}))$ samples from input distributions with biases $v_1, \ldots, v_m$.*

Proof. The algorithm is as follows: Sample $\frac{4}{\epsilon^2} \log(\frac{4m}{\epsilon})$ times from each of the m coins, let $\hat{v}_i$ be the empirical estimate of coin i's bias obtained by averaging, then apply the Continuous to Bernoulli algorithm (Section 2.2) to map $\hat{v}_{\max} = \max_i \hat{v}_i$ to a Bernoulli random variable.

Standard tail bounds imply that $|\hat{v}_{\max} - v_{\max}| < \epsilon/2$ with probability at least $1 - \epsilon/2$, and therefore $z = \mathbf{E}[\hat{v}_{\max}] \in [v_{\max} - \epsilon, v_{\max} + \epsilon]$. □

Since we are interested in a fast logistic Bernoulli race as λ grows large, we restrict attention to $\lambda > 4$. We set $\epsilon = 1/\lambda$ in the estimation of $v_{\max}$ (by Lemma 3.3.3). This estimate will be used to boost the bias of each distribution in the input so that the maximum bias is at least $1 - 3\epsilon$. The boosting of the bias is implemented with Bernoulli Addition which, to be fast, requires the cumulative bias be bounded away from one. Thus, the probabilities are scaled down by a factor of $1 - 2\epsilon$, this scaling is subsequently counterbalanced by adjusting the parameter λ. The formal details are given below.

Algorithm 3: Fast Exponential Bernoulli Race (with parameter $\lambda > 4$)

1: **input** Sample access to m coins with biases $v_1, \ldots, v_m$.

2: Let $\epsilon = 1/\lambda$.

3: Construct a coin with bias $z \in [v_{\max} - \epsilon, v_{\max} + \epsilon]$ (from Lemma 3.3.3).

4: Apply Bernoulli Down Scaling to a coin with bias $1 - z$ to implement a coin with bias $(1 - 2\epsilon)(1 - z)$.

5: For all i, apply Bernoulli Down Scaling to implement a coin with bias $(1 - 2\epsilon)v_i$.

6: For all i, apply Bernoulli Addition to implement coin with bias $v'_i = (1 - 2\epsilon)v_i + (1 - 2\epsilon)(1 - z)$.

7: Run the Basic Exponential Bernoulli Race with parameter $\lambda' = \frac{\lambda}{1-2\epsilon}$ on the coins with bias $v'_1, \ldots, v'_m$.

Theorem 3.3.4. *The Fast Exponential Bernoulli Race (Algorithm 3) samples with exponential weights (Definition 3.3.3) with an expected $O(\lambda^4 m^2 \log(\lambda m))$ samples from the input distributions.*

Proof. The correctness and runtime follows from the correctness and runtimes of the Basic Exponential Bernoulli Race, Bernoulli Doubling, Lemma 3.3.3 (for estimate of $v_{\max}$), and the fact $\lambda' v'_i = \lambda(v_i + 1 - z)$ and the distribution given by exponential weights is invariant to additive shifts of all weights.

A detailed analysis of the runtime follows. Since the algorithm builds a number of sampling subroutines in a hierarchy, we analyze the runtime of the algorithm and the various subroutines in a bottom up fashion. Steps 3 and 4 implement a coin with bias $(1 - 2\epsilon)(1 - z)$ with runtime $O(\lambda^2 m \cdot \log(\lambda m))$ per sample, as per the bound of Lemma 3.3.3. The coin implemented in Step 5 is sampled in constant time. Observe that $v'_i \le (1-2\epsilon)(1+v_i-v_{\max}+\epsilon) \le 1-\epsilon$, and the runtime of Bernoulli Doubling implies that $O(\lambda)$ samples from the coins of Steps 4 and 5 suffice for sampling Bern[v'_i]; we conclude that a v'_i-coin can be sampled in time $O(\lambda^3 m \cdot \log(\lambda m))$. Finally, note that for $v'_{\max} = \max_i v'_i$, we have $v'_{\max} \ge 1-3\epsilon$; Theorem 3.3.2 then implies that the Basic Exponential Bernoulli Race samples at most $\lambda' m e^{\lambda' 3\epsilon} \le 2e^6 \lambda m = O(\lambda m)$ times from the $\mathbf{v}'$-coins; we conclude the claimed runtime. □

3.3.2 The Single-Agent Multiple-Urns Problem

We investigate incentive compatible mechanism design for the *single-agent multiple-urns* problem. Informally, mechanism is needed to assign an agent to one of many urns. Each urn contains objects and the agent's value for being assigned to an urn is taken in expectation over objects from the urn. The problem asks for an incentive compatible mechanism with good welfare (i.e., the value of the agent for the assigned urn).

Problem Definition and Notations

A single agent with type t from type space $\mathcal{T}$ desires an object o from outcome space $\mathcal{O}$. The agent's value for an outcome o is a function of her type t and denoted by $v(t,o) \in [0,1]$. The agent is a risk-neutral quasi-linear utility maximizer with utility $\mathbf{E}_o[v(t,o)] - p$ for randomized outcome o and expected payment p. There are m urns. Each urn j is given by a distribution $\mathcal{D}_j$ over outcomes in $\mathcal{O}$. If the agent is assigned to urn j she obtains an object from the urn's distribution $\mathcal{D}_j$.

A mechanism can solicit the type of the agent (who may misreport if she desires). We further assume (1) the mechanism has black-box access to evaluate $v(t, o)$ for any type t and outcome o, (2) the mechanism has sample access to the distribution $\mathcal{D}_j$ of each urn j. The mechanism may draw objects from urns and evaluate the agent's reported value for these objects, but then must ultimately assign the agent to a single urn and charge the agent a payment. The urn and payment that the agent is assigned are random variables in the mechanism's internal randomization and randomness from the mechanisms potential samples from the urns' distributions.

The distribution of the urn the mechanism assigns to an agent, as a function of her type t, is denoted by $\mathbf{x}(t) = (x_1(t), \ldots, x_m(t))$ where $x_j(t)$ is the marginal probability that the agent is assigned to urn j. Denote the expected value of the agent for urn j by $v_j(t) = \mathbf{E}_{o \sim \mathcal{D}_j}[v(t, o)]$. The expected welfare of the mechanism is $\sum_j v_j(t)\, x_j(t)$. The expected payment of this agent is denoted by $p(t)$. The agent's utility for the outcome and payment of the mechanism is given by $\sum_j v_j(t)\, x_j(t) - p(t)$. Incentive compatibility is defined by the agent with type t preferring her outcome and payment to that assigned to another type t'.

Definition 3.3.4. A single-agent mechanism $(\mathbf{x}, p)$ is *incentive compatible* if, for all $t, t' \in \mathcal{T}$:

$$\sum\nolimits_j v_j(t)\, x_j(t) - p(t) \geq \sum\nolimits_j v_j(t)\, x_j(t') - p(t') \tag{3.1}$$

A multi-agent mechanism is Bayesian Incentive Compatible (BIC) if equation (3.1) holds for the outcome of the mechanism in expectation of the truthful reports of the other agents.

Incentive Compatible Approximate Scheme

If the agent's expected value for each urn is known, or equivalently mechanism designer knows the distributions $\mathcal{D}_j$ for all urns j rather than only sample access, this problem would be easy and admits a trivial optimal mechanism: simply select the urn maximizing the agent's expected value $v_j(t)$ according to her reported type t, and charge her a payment of zero. What makes this problem interesting is that the designer is restricted to only *sample* the agent's value for an urn. In this case, the following Monte-carlo adaptation of the trivial mechanism is tempting: sample from each urn sufficiently many times to obtain a close estimate $\tilde{v}_j(t)$ of $v_j(t)$ with high probability (up to any desired precision $\delta > 0$), then choose the urn j maximizing $\tilde{v}_j(t)$ and charge a payment of zero. This mechanism is not incentive compatible, as illustrated by a simple example.

Example Consider two urns. Urn A contains only outcome o_2, whereas B two contains a mixture of outcomes o_1 and o_3, with o_1 slightly more likely than o_3. Now consider an agent who has (true) values 0, 1, and 2 for outcomes o_1, o_2, and o_3 respectively. If this agent reports her true type, the trivial Monte-carlo mechanism — instantiated with any desired finite degree of precision — assigns her urn A most of the time, but assigns her urn B with some nonzero probability. The agent gains by misreporting her value of outcome o_3 as 0, since this guarantees her preferred urn A.

The above example might seem counter-intuitive, since the trivial Monte-carlo mechanism appears to be doing its best to maximize the agent's utility, up to the limits of (unavoidable) sampling error. One intuitive rationalization is the following: an agent can slightly gain by procuring (by whatever means) more precise information about the distributions $\mathcal{D}_j$ than that available to the mechanism, and using this information to guide her strategic misreporting of her type. This raises the following question:

Question: *Is there an incentive-compatible mechanism for the single-agent multiple-urns problem which achieves welfare within ϵ of the optimal, and samples only poly$(m, \frac{1}{\epsilon})$ times (in expectation) from the urns?*

We resolve the above question in the affirmative. We present approximation scheme for this problem that is based on our solution to the problem of random selection with exponential weights (Section 3.3.1). The solution to the single-agent multiple-urns problem is a main ingredient in the Bayesian mechanism that we propose in Section 3.3.3 as our black-box reduction mechanism.

To explain the approximate scheme, we start by recalling the following standard theorem in mechanism design.

Theorem 3.3.5. *For outcome rule $\mathbf{x}$, there exists payment rule p so that single-agent mechanism $(\mathbf{x}, p)$ is incentive compatible if and only if $\mathbf{x}$ is maximal in range, i.e., $\mathbf{x}(t) \in \operatorname{argmax}_{\mathbf{x}'} \sum_j v_j(t)\, x'_j - c(\mathbf{x}')$, for some cost function $c(\cdot)$.*[6]

The payments that satisfy Theorem 3.3.5 can be easily calculated with black-box access to outcome rule $\mathbf{x}(\cdot)$. For a single-agent problem, this payment can be calculated in two calls to the function $\mathbf{x}(\cdot)$, one on the agent's reported type t and the other on a type randomly drawn from the path between the origin and t. Further discussion and details are given in Appendix A.1. It suffices, therefore, to identify a mechanism that samples from urns and assigns the agent to an urn

[6]The "only if" direction of this theorem requires that the type space $\mathcal{T}$ be rich enough so that the induced space of values across the urns is $\{(v_1(t), \ldots, v_m(t)) : t \in \mathcal{T}\} = [0, 1]^m$.

that induces an outcome rule $\mathbf{x}(\cdot)$ that is good for welfare, i.e., $\sum_i v_j(t)\, x_j(t)$, and is maximal in range. The following theorem solves the problem.

Theorem 3.3.6. *There is an incentive-compatible mechanism for the single-agent multiple-urns problem which achieves an additive ϵ-approximation to the optimal welfare in expectation, and runs in time $O(m^2(\frac{\log m}{\epsilon})^5)$ in expectation.*

Proof. Consider the problem of selecting a distribution over urns to optimize welfare plus (a scaling of) the Shannon entropy function, i.e., $\mathbf{x}(t) = \operatorname{argmax}_{\mathbf{x}'} v_j(t)\, x'_j - (1/\lambda) \sum_j x'_j \log x'_j$.[7] It is well known that the optimizer $\mathbf{x}(t)$ is given by exponential weights, i.e., the marginal probability of assigning the jth urn is given by $x_j(t) = \exp(\lambda v_j(t)) / \sum_{j'} \exp(\lambda v_{j'}(t))$. In Section 3.3.1 we gave a polynomal time algorithm for sampling from exponential weights, specifically, the Fast Exponential Bernoulli Race (Algorithm 3). Proper choice of the parameter λ controls trades off faster runimes with decreased loss due to entropy term. The entropy is maximized at the uniform distribution $\mathbf{x}' = (1/m, \ldots, 1/m)$ with entropy $\log m$. Thus, choosing $\lambda = \log m/\epsilon$ guarantees that the welfare is within an additive ϵ of the optimal welfare $\max_j v_j(t)$. The bound of the theorem then follows from the analysis of the Fast Exponential Bernoulli Race (Theorem 3.3.4) with this choice of λ. □

[7] The additive entropy term can be interpreted as a negative cost vis-à-vis Theorem 3.3.5.

3.3.3 A Bayesian Incentive Compatible Black-box Reduction

Surrogate Selection and the Replica-Surrogate Matching

A main conclusion of the literature on Bayesian reductions for mechanism design is that the multi-agent problem of reducing Bayesian mechanism design to algorithm design, itself, reduces to a single-agent problem of *surrogate selection*. Consider any agent in the original problem and the *induced algorithm* with the inputs form other agents hardcoded as random draws from their respective type distributions. The induced algorithm maps the type of this agent to a distribution over outcomes. If this distribution over outcomes is maximal-in-range then there exists payments for which the induced algorithm is incentive compatible (Theorem 3.3.5). If not, the problem of surrogate selection is to map the type of the agent to an input to the algorithm to satisfy three properties:

(a) The composition of surrogate selection and the induced algorithm is maximal-in-range,

(b) The composition approximately preserves welfare,

(c) The surrogate selection preserves the type distribution.

Condition (c), a.k.a. *stationarity*, implies that fixing the non-maximaility-of-range of the algorithm for a particular agent does not affect the outcome for any other agents. With such an approach each agent's incentive problem can be resolved independently from that of other agents.

Theorem 3.3.7 (Hartline et al., 2015)**.** *The composition of an algorithm with a profile of surrogate selection rules, that maps the profile of agent types to an input to the algorithm, is Bayesian incentive compatible and approximately preserves the algorithms* *welfare (the loss in welfare is the sum of the losses in welfare of each surrogate selection rule).*

The surrogate selection rule of Hartline et al. (2015) is based on setting up a matching problem between random types from the distribution (replicas) and the outcomes of the algorithm on random types from the distribution (surrogates). The true type of the agent is one of the replicas, and the surrogate selection rule outputs the surrogate to which this replica is matched. This approach addresses the three properties of surrogate selection rules as (a) if the matching selected is maximal-in-range then the composition of the surrogate selection rule with the induced algorithm is maximal-in-range, (b) the welfare of the matching is the welfare of the reduction and the optimal matching approximates the welfare of the original algorithm, and (c) any maximal matching gives a stationary surrogate selection rule. For a detailed discussion on why maximal-in-range matching will result in a BIC mechanism after composing the corresponding surrogate selection rule with the allocation algorithm, we refer the interested reader to look at Lemma A.2.1 and Lemma A.2.2 in Appendix A.2.

Definition 3.3.5. The *replica-surrogate matching* surrogate selection rule; for a k-to-1 matching algorithm M, a integer market size m, and load k; maps a type t to a surrogate type as follows:

1. Pick the real-agent index i^* uniformly at random from $\{1, \ldots, km\}$.
2. Define the *replica type profile* $\mathbf{r}$, an km-tuple of types by setting $r_{i^*} = t$ and sampling the remaining $km - 1$ replica types $\mathbf{r}_{-i^*}$ i.i.d. from the type distribution F.
3. Sample the *surrogate type profile* $\mathbf{s}$, an m-tuple of i.i.d. samples from the type distribution F.

4. Run matching algorithm M on the complete bipartite graph between replicas and surrogates.

5. Output the surrogate j^* that is matched to i^*.

The value that a replica obtains for the outcome that the induced algorithm produces for a surrogate, henceforth, *surrogate outcome*, is a random variable. The analysis of Hartline et al. (2015) is based on the study of an ideal computational model where the value of any replica for any surrogate outcome $\mathbf{E}_{o \sim \mathcal{A}(s_j)}[v(r_i, o)]$ is known exactly. In this computationally-unrealistic model and with these values as weights, the maximum weight matching algorithm can be employed in the replica-surrogate matching surrogate selection rule above, and it results in a Bayesian incentive compatible mechanism. Hartline et al. (2015) analyze the welfare of the resulting mechanism in the case where the load is $k = 1$, prove that conditions (a)-(c) are satisfied, and give (polynomial) bounds on the size m that is necessary for the expected welfare of the mechanism to be an additive ϵ from that of the algorithm.

Remark Given a BIC allocation algorithm $\tilde{\mathcal{A}}$ through a replica-surrogate matching surrogate selection, the payments that satisfy Bayesian incentive compatibility can be easily calculated with black-box access to $\tilde{\mathcal{A}}$ as implicit payments (Appendix A.1).

If M is maximum matching, conditions (a)-(c) clearly continue to hold for our generalization to load $k > 1$. Moreover, the welfare of the reduction is monotone non-decreasing in k.

Lemma 3.3.8. *In the ideal computational model (where the value of a replica for being* *matched to a surrogate is given exactly) the per-replica welfare of the replica-surrogate maximum matching is monotone non-decreasing in load k.*

Proof. Consider a non-optimal matching that groups replicas into k groups of size m and finds the optimal 1-to-1 matching between replicas in each group and the surrogates. As these are random $(k = 1)$−matchings, the expected welfare of each such matching is equal to the expected welfare of the $(k = 1)$−matching. These matchings combine to give a feasible matching between the mk replicas and m surrogates. Thus, the total expected welfare of the optimal k-to-1 matching between replicas and surrogates is at least k times the expected welfare of the $(k = 1)$−matching. Thus, the per-replica welfare, i.e., normalized by mk, is monotone in k. □

Our main result is an approximation scheme for the ideal reduction of Hartline et al. (2015). We identify a $k > 1$ and a polynomial (in m and $1/\epsilon$) time k-to-1 matching algorithm for the black-box model and prove that the expected welfare of this matching algorithm (per-replica) is within an additive ϵ of the expected welfare per-replica of the optimal matching in the ideal model with load $k = 1$ (as analyzed by Hartline et al., 2015). The welfare of the ideal model is monotone non-decreasing in load k; therefore it will be sufficient to identify a polynomial load k where there is a polynomial time algorithm in the black-box model that has ϵ loss relative to the ideal model for that same load k.

In the remainder of this section we replace this ideal matching algorithm with an approximation scheme for the black-box model where replica values for surrogate outcomes can only be estimated by sampling. For any ϵ our algorithm gives an ϵ additive loss of the welfare of the ideal algorithm with only a polynomial increase to the runtime. Moreover, the algorithm produces a perfect (and so maximal) matching, and therefore the surrogate selection rule is stationary; and the algorithm is maximal-in-range for the true agent's replica, and therefore the resulting mechanism is Bayesian incentive compatible.

Entropy Regularized Matching

In this section we define an entropy regularized bipartite matching problem and discuss it's solution. We will refer to the left-hand-side vertices as replicas and the right-hand-side vertices as surrogates. The weights on the edge between replica $i \in \{1, \ldots, km\}$ and surrogate $j \in \{1, \ldots, m\}$ will be denoted by $v_{i,j}$. In our application to the replica-surrogate matching defined in the previous section, the weights will be set to $v_{i,j} = \mathbf{E}_{o \sim \mathcal{A}(s_j)}[v(r_i, o)]$ for $(i, j) \in [km] \times [m]$.

Definition 3.3.6. For weights $\mathbf{v} = [v_{i,j}]_{(i,j)\in[km]\times[m]}$, the entropy regularized matching program for parameter $\delta > 0$ is:

$$\begin{aligned} \max_{\{x_{i,j}\}_{(i,j)\in[km]\times[m]}} \quad & \sum\nolimits_{i,j} x_{i,j}\, v_{i,j} - \delta \sum\nolimits_{i,j} x_{i,j} \log x_{i,j}, & \\ \text{s.t.} \quad & \sum\nolimits_{i} x_{i,j} \leq k & \forall j \in [m], \\ & \sum\nolimits_{j} x_{i,j} \leq 1 & \forall i \in [km]. \end{aligned}$$

The optimal value of this program is denoted $\mathrm{OPT}(\mathbf{v})$.

The dual variables for right-hand-side constraints of the matching polytope can be interpreted as *prices* for the surrogate outcomes. Given prices, the *utility* of a replica for a surrogate outcome given prices is the difference between the replica's value and the price. The following lemma shows that for the right choice of dual variables, the maximizer of the entropy regularized matching program is given by exponential weights with weights equal to the utilities.

Observation 1. *For the optimal Lagrangian dual variables $\boldsymbol{\alpha}^* \in \mathbb{R}^m$ for surrogate feasibility in the entropy regularized matching program (Definition 3.3.6), namely,*

$$\boldsymbol{\alpha}^* = \text{argmin}_{\boldsymbol{\alpha}\geq\mathbf{0}} \max_{\mathbf{x}} \left\{\mathcal{L}(\mathbf{x},\boldsymbol{\alpha}) : \sum\nolimits_j x_{i,j} \leq 1 \ , \ \forall i\right\}$$

where $\mathcal{L}(\mathbf{x},\boldsymbol{\alpha}) \triangleq \sum_{i,j} x_{i,j}\, v_{i,j} - \delta \sum_{i,j} x_{i,j} \log x_{i,j} + \sum_j \alpha_j(k - \sum_i x_{i,j})$ *is the Lagrangian objective function; the optimal solution* $\mathbf{x}^*$ *to the primal is given by exponential weights:*

$$x^*_{i,j} = \frac{\exp\left(\frac{v_{i,j}-\alpha^*_j}{\delta}\right)}{\sum_{j'} \exp\left(\frac{v_{i,j'}-\alpha^*_{j'}}{\delta}\right)}, \quad \forall i, j.$$

Observation 1 recasts the entropy regularized matching as, for each replica, sampling from the distribution of exponential weights. For any replica i and fixed dual variables $\boldsymbol{\alpha}$ our Fast Exponential Bernoulli Race (Algorithm 3) gives a polynomial time algorithm for sampling from the distribution of exponential weights in the expectations from samples computational model.

Lemma 3.3.9. *For replica i and any prices (dual variables) $\boldsymbol{\alpha} \in [0,h]^m$, allocating a surrogate j drawn from the exponential weights distribution*

$$x_{i,j} = \frac{\exp\left(\frac{v_{i,j}-\alpha_j}{\delta}\right)}{\sum_{j'} \exp\left(\frac{v_{i,j'}-\alpha_{j'}}{\delta}\right)}, \quad \forall j \in [m], \tag{3.2}$$

is maximal-in-range, as defined in Definition 3.3.5, and this random surrogate j can be sampled with $O\left(\frac{h^4 m^2 \log(hm/\delta)}{\delta^4}\right)$ samples from replica-surrogate-outcome value distributions.

Proof. To see that the distribution is maximal-in-range when assigning surrogate outcome j to replica i, consider the regularized welfare maximization

$$\text{argmax}_{\mathbf{x}'} \sum_j v_{i,j}\, x'_j - \delta \sum_j x'_j \log x'_j - \sum_j \alpha_j x'_j$$

for replica i. Similar to Observation 1, it is easy to see that the exponential weight distribution in (3.2) is the unique maximizer of this concave program by looking at the first-order conditions.

To apply the Fast Exponential Bernoulli Race to the utilities, of the form $v_{i,j} - \alpha_j \in [-h, 1]$, we must first normalize them to be on the interval $[0, 1]$. This normalization is accomplished by adding h to the utilities (which has no effect on the exponential weights distribution, and therefore preserves being maximal-in-range), and then scaling by $1/(h + 1)$. The scaling needs to be corrected by setting λ in the Fast Exponential Bernoulli Race (Algorithm 3) to $(h + 1)/\delta$. The expected number of samples from the value distributions that are required by the algorithm, per Theorem 3.3.4, is $O(h^4 m^2 \log(hm/\delta)\delta^{-4})$. □

If we knew the optimal Lagrangian variables $\boldsymbol{\alpha}^*$ from Observation 1, it would be sufficient to define the surrogate selection rule by simply sampling from the true agent i^*'s exponential weights distribution (which is polynomial time per Lemma 3.3.9). Notice that the wrong values of $\boldsymbol{\alpha}$ correspond to violating primal constraints (for the surrogates) and thus the outcome from sampling from exponential weights for such α would not correspond to a maximal-in-range matching. In the next section we give a polynomial time approximation scheme that is maximal-in-range for each replica, and therefore true agent i^*, and approximates sampling from the correct $\boldsymbol{\alpha}^*$.

Online Entropy Regularized Matching

In this section, we reduce the entropy regularized matching problem to the problem of sampling from exponential weights (as described in Lemma 3.3.9) in an online fashion. Consider replicas being drawn adversarially, but in a random order, over times $1, \ldots, km$. The basic observation is that approximate dual variables $\boldsymbol{\alpha}$ are sufficient for an online assignment of each replica to a surrogate via Lemma 3.3.9 to approximate the optimal (offline) regularized matching. Recall, the replicas are independently and identically distributed in the original problem.

Our construction borrows techniques used in designing online algorithms for stochastic online convex programming problems (Agrawal and Devanur, 2015; Chen and Wang, 2013), and stochastic online packing problems (Agrawal et al., 2009; Devanur et al., 2011; Badanidiyuru et al., 2013b; Kesselheim et al., 2014). Our online algorithm (Algorithm 4, below) considers the replicas in order, updates the dual variables using multiplicative weight updates based on the current allocation, and allocates to each agent by sampling from the exponential weights distribution as given by Lemma 3.3.9. The algorithm is parameterized by δ, the scale of the regularizer; by η, the rate at which the algorithm learns the dual variables $\boldsymbol{\alpha}$; and by scale parameter γ, which we set later.

The algorithm needs to satisfy four properties to be useful in a polynomial time reduction. First, it needs to produce a maximal matching so that the replica-surrogate matching surrogate selection rule is stationary, specifically via condition (c). It needs to be maximal-in-range for the real agent (replica i^*). In fact, all replicas are treated symmetrically and allocated by sampling from an exponential weights distribution that is maximal-in-range via Lemma 3.3.9. Third, it needs to have good welfare compared to the ideal matching. Fourth, it's runtime needs to be polynomial. The first two properties are immediate and imply the theorem below. The last two properties are analyzed below.

Algorithm 4: Online Entropy Regularized Matching Algorithm (with parameters $\delta, \eta, \gamma \in \mathbb{R}_+$)

1: **input:** sample access to replica-surrogate matching instance values $\{v_{i,j}\}$ for replicas $i \in \{1, \ldots, mk\}$ and surrogates $j \in \{1, \ldots, m\}$.

2: **for all** $i \in \{1, \ldots, km\}$ **do**

3: Let k_j be the number of replicas previously matched to each surrogate j and $J = \{j : k_j < k\}$ the set of surrogates with availability remaining.

4: Set $\boldsymbol{\alpha}^{(i)}$ according to the exonential weights distribution with weights $\eta \cdot k_j$ for available surrogates $j \in J$ ($\alpha_j^{(i)} = 0$ for unavailable surrogates).

5: Match replica i to surrogate $j \in J$ drawn according to the exponential weights distribution with weights $(v_{i,j} - \gamma\, \alpha_j^{(i)})/\delta$ with the Fast Exponential Bernoulli Race (Algorithm 3).

6: **end for**

Theorem 3.3.10. *The mechanism that maps types to surrogates via the replica-surrogate matching surrogate selection rule with the online entropy regularized matching algorithm (with payments from Theorem 3.3.5) is Bayesian incentive compatible.*

Social Welfare Loss

We analyze the welfare loss of the online entropy regularized matching algorithm (Algorithm 4) with regularizer parameter δ, learning rate η, and scale parameter γ set as a k-fraction of an estimate of the value of the offline program (Definition 3.3.6).

Theorem 3.3.11. *There are parameter settings for online entropy regularized matching algorithm (Algorithm 4) for which (1) its per-replica expected welfare is within an additive ϵ of the optimal welfare of the replica surrogate matching, and (2) given oracle access to $\mathcal{A}$, the running time of this algorithm is polynomial in m and $1/\epsilon$.*

To prove this theorem, we first argue how to set γ to be a constant approximation to the k-fraction of optimal value of the convex program with high probability, and with efficient sampling. Second, we argue that the online and offline optimal entropy regularized matching algorithms have nearly the same welfare. Finally, we argue that the offline optimal entropy regularized matching has nearly the welfare of the offline optimal matching. The proof of the theorem is then given by combining these results with the right parameters.

Parameter γ and approximating the offline optimal. Pre-setting γ to be an estimate of the optimal objective of the convex program in Definition 3.3.6 is necessary for the competitive ratio guarantee of Algorithm 4. Also, γ should be set in a symmetric and incentive compatible way across replicas, to preserve stationarity property. To this end, we look at an instance generated by an independent random draw of mk replicas (while fixing the surrogates). In such an instance, we estimate the expected values by sampling and taking the empirical mean for each edge in the replica-surrogate bipartite graph. We then solve the convex program exactly (which can be done in polytime using an efficient separation oracle). Obviously, this scheme is incentive compatible as we do not even use the reported type of true agent in our calculation for γ, and it is symmetric across replicas. In Appendix A.3 we show how this approach leads to a constant approximation to the optimal value of the offline program in 3.3.6 with high probability.

Lemma 3.3.12. *If $k = \Omega(\frac{\log(\eta^{-1})}{\delta^2 m(\log m)^2})$, then there exist a polytime approximation scheme to calculate γ (i.e. it only needs polynomial in m, k, δ^{-1} and η^{-1} samples to black-box allocation $\mathcal{A}$) such that*

$$\mathrm{OPT}(\mathbf{v})/k \leq \gamma \leq O(1)\,\mathrm{OPT}(\mathbf{v})/k$$

with probability at least $1 - \eta$*.*

Competitive ratio of the online entropy regularized matching algorithm. Assuming γ is set to be a constant approximation to the k-fraction of the optimal value of the offline entropy regularized matching program, we prove the following lemma.

Lemma 3.3.13. *For a fixed regularizer parameter* $\delta > 0$*, learning rate* $\eta > 0$*, regularized welfare estimate* γ*, and market size* $m \in \mathbb{N}$ *that satisfy*

$$\frac{m \log m}{\eta^2} \leq k \quad \textit{and} \quad \mathrm{OPT}(\mathbf{v})/k \leq \gamma \leq O(1)\,\mathrm{OPT}(\mathbf{v})/k\,,$$

the online entropy regularized matching algorithm (Algorithm 4) obtains at least an $(1 - O(\eta))$ *fraction of the welfare of the optimal entropy regularized matching (Definition 3.3.6).*

Proof. Recall that $\mathrm{OPT}(\mathbf{v})$ denotes the optimal objective value of the entropy regularized matching program. We will analyze the algorithm up to the iteration τ that the first surrogate becomes unavailable (because all k copies are matched to previous replicas).

Define the contribution of replica i to the Lagrangian objective of Observation 1 for allocation $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m})$ and dual variables $\boldsymbol{\alpha}$ as

$$\mathcal{L}^{(i)}(\mathbf{x}_i, \boldsymbol{\alpha}) \triangleq \sum\nolimits_j v_{i,j}\, x_{i,j} - \delta \sum\nolimits_j x_{i,j} \log x_{i,j} + \sum\nolimits_j \gamma \alpha_j \left(\tfrac{1}{m} - x_{i,j}\right). \tag{3.3}$$

The difference between the outcome for replica i in the online algorithm and the solution to the offline optimization is that the algorithm selects the outcome with respect to dual variables $\gamma\boldsymbol{\alpha}^{(i)}$ while the offline algorithm selects the outcome with respect to the optimal dual variables $\boldsymbol{\alpha}^*$ (Observation 1). Denote the outcome of the online algorithm by

$$\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m}) = \text{argmax}_{\mathbf{x}'_i \in \Delta^m} \mathcal{L}^{(i)}(\mathbf{x}'_i, \gamma\alpha^{(i)}) \,,$$

and its contribution to the objective by

$$\texttt{ALG}_i \triangleq \sum_j v_{i,j}\, x_{i,j} - \delta \sum_j x_{i,j} \log x_{i,j} \,.$$

Likewise for the outcome of the offline optimization by $\mathbf{x}_i^*$ and OPT_i. Denote by $\hat{\mathbf{x}}_i$ the indicator vector for the online algorithm sampling from $\mathbf{x}_i$.

Optimality of $\mathbf{x}_i$ for dual variables $\gamma\boldsymbol{\alpha}^{(i)}$ in equation (3.3) implies

$$\texttt{ALG}_i + \sum\nolimits_j \gamma\alpha_j^{(i)}\Big(\tfrac{1}{m} - x_{i,j}\Big) \geq \text{OPT}_i + \sum\nolimits_j \gamma\alpha_j^{(i)}\Big(\tfrac{1}{m} - x_{i,j}^*\Big)$$

so, by rearranging the terms and taking expectations conditioned on the observed history, we have

$$\begin{aligned}
\mathbf{E}[\texttt{ALG}_i \mid \mathcal{H}_{i-1}] &\geq \gamma\mathbf{E}\Big[\boldsymbol{\alpha}^{(i)} \cdot \mathbf{x}_i \mid \mathcal{H}_{i-1}\Big] + \mathbf{E}[\text{OPT}_i \mid \mathcal{H}_{i-1}] - \gamma\mathbf{E}\Big[\boldsymbol{\alpha}^{(i)} \cdot \mathbf{x}_i^* \mid \mathcal{H}_{i-1}\Big] \\
&= \mathbf{E}[\text{OPT}_i] - \gamma\boldsymbol{\alpha}^{(i)} \cdot \mathbf{E}\Big[\mathbf{x}_i^*\Big] + \gamma\boldsymbol{\alpha}^{(i)} \cdot \hat{\mathbf{x}}_i - (\mathbf{E}[\text{OPT}_i] - \mathbf{E}[\text{OPT}_i \mid \mathcal{H}_{i-1}]) \\
&\quad + \gamma\boldsymbol{\alpha}^{(i)} \cdot (\mathbf{E}\Big[\mathbf{x}_i^*\Big] - \mathbf{E}\Big[\mathbf{x}_i^* \mid \mathcal{H}_{i-1}\Big]) + \gamma\boldsymbol{\alpha}^{(i)} \cdot (\mathbf{E}[\mathbf{x}_i \mid \mathcal{H}_{i-1}] - \hat{\mathbf{x}}_i) \\
&\geq \frac{1}{mk}\text{OPT}(\mathbf{v}) + \gamma\boldsymbol{\alpha}^{(i)} \cdot (\hat{\mathbf{x}}_i - \frac{1}{m}\mathbf{1}) - L_i - L'_i
\end{aligned}$$

where

$$\begin{aligned}
L_i &\triangleq \gamma\, \boldsymbol{\alpha}^{(i)}.(\hat{\mathbf{x}}_i - \mathbf{E}[\mathbf{x}_i|\mathcal{H}_{i-1}]) \,, \\
L'_i &\triangleq |(\mathbf{E}\Big[r_i^*\Big] - \mathbf{E}\Big[r_i^*|\mathcal{H}_{i-1}\Big]| + \gamma\|\mathbf{E}\Big[\mathbf{x}_i^*\Big] - \mathbf{E}\Big[\mathbf{x}_i^*|\mathcal{H}_{i-1}\Big]\| \,.
\end{aligned}$$

By summing the above inequalities for $i = 1 : \tau - 1$ we have:

$$\sum_{i=1}^{\tau-1} \mathbf{E}[\texttt{ALG}_i|\mathcal{H}_{i-1}] \geq \frac{\tau-1}{mk}\text{OPT}(\mathbf{v}) + \gamma\sum_{i=1}^{\tau-1} \boldsymbol{\alpha}^{(i)} \cdot (\hat{\mathbf{x}}_i - \frac{1}{m}\mathbf{1}) - \sum_{i=1}^{\tau-1}(L_i + L'_i) \tag{3.4}$$

In order to bound the term $\gamma \sum_{i=1}^{\tau-1} \boldsymbol{\alpha}^{(i)} \cdot (\hat{\mathbf{x}}_i - \frac{1}{m}\mathbf{1})$, let $g_i(\boldsymbol{\alpha}) \triangleq \boldsymbol{\alpha} \cdot (\hat{\mathbf{x}}_i - \frac{1}{m}\mathbf{1})$. Then, by applying the regret bound of exponential gradient (or essentially multiplicative weight update) online learning algorithm for any realization of random variables $\{\hat{\mathbf{x}}_i\}$ (which will result in $\boldsymbol{\alpha}^{(i)}$ to be the exponential weights distributions with weights $\eta \cdot k_j$), we have

$$\sum_{i=1}^{\tau-1} g_i(\boldsymbol{\alpha}^{(i)}) \geq (1-\eta) \max_{\|\boldsymbol{\alpha}\|_1 \leq 1, \boldsymbol{\alpha} \geq \mathbf{0}} \sum_{i=1}^{\tau-1} g_i(\boldsymbol{\alpha}) - \frac{\log m}{\eta} \geq (1-\eta)(k - \frac{\tau-1}{m}) - \frac{\log m}{\eta} \tag{3.5}$$

where the last inequality holds because at the time $\tau-1$, either there exists j such that $\sum_{i=1}^{\tau-1} \hat{x}_{i,j} = k$, or $\tau - 1 = mk$ and all surrogate outcome budgets are exhausted. In the former case, we have

$$\max_{\|\boldsymbol{\alpha}\|_1 \leq 1, \boldsymbol{\alpha} \geq \mathbf{0}} \sum_{i=1}^{\tau-1} g_i(\boldsymbol{\alpha}) \geq \sum_{i=1}^{\tau-1} g_i(\mathbf{e}_j) \geq k - \frac{\tau-1}{m},$$

and in the latter case we have

$$\max_{\|\boldsymbol{\alpha}\|_1 \leq 1, \boldsymbol{\alpha} \geq \mathbf{0}} \sum_{i=1}^{\tau-1} g_i(\boldsymbol{\alpha}) \geq 0 \geq k - \frac{\tau-1}{m}.$$

Combining (3.4) and (3.5), letting $Q_i = L_i + L_i'$, and assuming $\hat{\mathbf{x}}_j = 0$ for $j \geq \tau$, we have:

$$\begin{aligned}
\sum_{i=1}^{mk} \mathbf{E}[\texttt{ALG}_i | \mathcal{H}_{i-1}] &\geq \sum_{i=1}^{\tau-1} \mathbf{E}[\texttt{ALG}_i | \mathcal{H}_{i-1}] \\
&\geq \frac{\tau-1}{mk} \mathrm{OPT}(\mathbf{v}) + \gamma(1-\eta)(k - \frac{\tau-1}{m}) - \gamma\frac{\log m}{\eta} - \sum_{i=1}^{\tau-1} Q_i \\
&\geq \mathrm{OPT}(\mathbf{v}) \left(\frac{\tau-1}{mk} + \frac{1}{k}(1-\eta)(k - \frac{\tau-1}{m}) - O(1) \cdot \frac{\log m}{k\eta} \right) - \sum_{i=1}^{mk} Q_i \\
&\geq (1-\eta)\,\mathrm{OPT}(\mathbf{v}) - O(\eta) \cdot \mathrm{OPT}(\mathbf{v}) - \sum_{i=1}^{mk} Q_i
\end{aligned} \tag{3.6}$$

where the last inequality holds simply because $k > \frac{\log m}{\eta^2}$. By taking expectations from both sides, we have

$$\mathbf{E}[\texttt{ALG}] \geq (1 - O(\eta)) \cdot \mathrm{OPT}(\mathbf{v}) - \sum_{i=1}^{mk} (\mathbf{E}[L_i] + \mathbf{E}\left[L_i'\right]) \tag{3.7}$$

We now bound each term separately. First, define $Y_i \triangleq \sum_{i' \leq i} L_i$, and then note that $\mathbf{E}[Y_i - Y_{i-1} | \mathcal{H}_{i-1}] = 0$, and therefore sequence $\{Y_i\}$ forms a martingale. Now, by using concentration of martingales we have the following lemma.

Lemma 3.3.14. $\mathbf{E}\left[\sum_{i=1}^{mk} L_i\right] \leq \gamma O(\sqrt{km \log km})$.

Proof of Lemma 3.3.14. Sequence $\{Y_i\}_{i=1}^{mk}$ forms a martingale, following the fact that $\mathbf{E}[Y_i - Y_{i-1} | \mathcal{H}_{i-1}] = 0$ and using Cauchy-Schwarz

$$|Y_i - Y_{i-1}| = \gamma |\boldsymbol{\alpha}^{(i)} \cdot (\hat{\mathbf{x}}_i - \mathbf{E}[\mathbf{x}_i | \mathcal{H}_{i-1}])| \leq \gamma \|\boldsymbol{\alpha}^{(i)}\| \cdot \|\hat{\mathbf{x}}_i - \mathbf{E}[\mathbf{x}_i | \mathcal{H}_{i-1}]\| \leq 2\gamma \ .$$

By using Azuma's inequality, we have

$$\Pr\{|Y_{mk}| \geq t\} \leq \exp\left(-\frac{t^2}{4km\gamma^2}\right).$$

Let $t = \gamma \sqrt{2km \log(km)}$, then $\Pr\{|Y_{mk}| \geq \gamma \sqrt{2km \log(km)}\} \leq \frac{1}{\sqrt{km}}$. Therefore,

$$\mathbf{E}\left[\sum_{i=1}^{mk} L_i\right] \leq \mathbf{E}[|Y_{mk}|] \leq \gamma \sqrt{2km \log(km)} + \frac{1}{\sqrt{km}}.2\gamma km = \gamma O(\sqrt{km \log km}).$$

□

To bound the second term, we use an argument based on Lemma 4.1 in Agrawal and Devanur (2015). In fact, we have the following lemma.

Lemma 3.3.15. $\mathbf{E}\left[\sum_{i=1}^{mk} L'_i\right] \leq \gamma O(\sqrt{k \log m})$.

Proof of Lemma 3.3.15. Using Lemma 4.1 in Agrawal and Devanur (2015) with $S = \{v \in \mathbb{R}^m : v \leq \frac{1}{m}\mathbf{1}\}$, we have $\mathbf{E}\left[\sum_{i=1}^{mk} L'_i\right] \leq \gamma O(\sqrt{skm \log m})$ where $s = \max_{v \in S} \max_{j \in [m]} v_j$. Obviously, $s = \frac{1}{m}$, which completes the proof. □

Using Lemmas 3.3.15 and 3.3.14, combined with the facts that $\gamma \leq O(1) \cdot \frac{\mathrm{OPT}(\mathbf{v})}{k}$ and $k \geq \frac{m \log m}{\eta^2}$, we have $\mathbf{E}\left[\sum_{i=1}^{mk} L_i + L'_i\right] \leq O(\eta)\,\mathrm{OPT}(\mathbf{v})$. Together with (3.7), we conclude that $\mathbf{E}[\texttt{ALG}] \geq (1 - O(\eta))\,\text{OPT}(\mathbf{v})$. This holds conditioned on $\frac{\text{OPT}(\mathbf{v})}{k} \leq \gamma \leq O(1) \cdot \frac{\text{OPT}(\mathbf{v})}{k}$. Moreover, γ is calculated by sampling such that this event happens with probability at least $(1 - \eta)$, which completes the proof. □

Lemma 3.3.16. *With parameter $\delta \geq 0$ the welfare (average for the replicas) of the optimal entropy regularized matching is within an additive $\delta \log m$ of the welfare of the optimal matching.*

Proof. The entropy $-\sum_{i,j} x_{i,j} \log x_{i,j}$ is non-negative and maximized with $x_{i,j} = 1/m$. The maximum value of the entropy term is thus $\delta mk \log m$. The optimal objective value of the entropy regularized matching exceeds that of the optimal matching; thus, its welfare is within an additvie $\delta mk \log m$ of the optimal matching. The average welfare per replica in the entropy regularized matching (recall, there are mk replicas) is within $\delta \log m$ of the average welfare per replica on the optimal matching. □

We conclude the section by combining Lemmas 3.3.12, 3.3.13, and 3.3.16 to prove the main theorem.

Proof of Theorem 3.3.11. Let $\delta = \frac{\epsilon}{3} \cdot \frac{1}{\log m}$ and $\eta = \frac{\epsilon}{3} \cdot \frac{1}{c}$, where c is a constant such that competitive ratio of Algorithm 4 is at least $1 - c \cdot \eta$ (Lemma 3.3.13). Moreover, let $k = \frac{m \log m}{\eta^2} = O(\frac{m \log m}{\epsilon^2})$, to satisfy the required condition in Lemma 3.3.13. The per-replica welfare of Algorithm 4 is within an additive $\delta \log m = \epsilon/3$ of its entropy regularized matching objective value, which in turn is a $1 - c \cdot \eta = 1 - \epsilon/3$ approximation to the per-replica optimal value of the entropy regularized matching due to Lemma 3.3.12 and 3.3.13. Following Lemma 3.3.16, the per-replica optimal value of the entropy regularized matching is within an additive $\delta \log m = \epsilon/3$ of the per-replica expected welfare of the optimal matching. As the per-replica welfare is bounded by 1, the per-replica welfare of the Algorithm 4 is within an additive $\epsilon/3 + \epsilon/3 + \epsilon/3 = \epsilon$ of the per-replica expected welfare of the optimal matching, as claimed. Finally, due to Lemma 3.3.12 and the fact that k is polynomial in m and $1/\epsilon$, the algorithm's running time is polynomial in m and $1/\epsilon$. □

3.3.4 The End-to-End BIC Black-box Reduction

We now summarize the proposed BIC black-box reduction. We incorporate our surrogate selection rule (By using Algorithm 4 as the matching algorithm in Definition 3.3.5) in the reduction under ideal-model proposed in Hartline et al. (2015) and we set the market size parameter m accordingly to maintain the welfare preservation property of this reduction.

Definition 3.3.7 (Hartline et al. (2015))**.** The doubling dimension of a metric space is the smallest constant Δ such that every bounded subset S can be partitioned into at most 2 subsets, each having diameter at most half of the diameter of S.

We now use the following theorem in Hartline et al. (2015), which states the welfare preservation of the maximum weight replica-surrogate matching in the ideal model if m is large enough.

Theorem 3.3.17 (Hartline et al. (2015))**.** *For any agent with type space $\mathcal{T}$ that has doubling dimension $\Delta \geq 2$, if*

$$m \geq \frac{1}{2\epsilon^{\Delta+1}},$$

then the expected per-replica welfare of the maximum matching in the ideal model of

Hartline et al. (2015) with load $k = 1$ is within an additive 2ϵ of the expected welfare of allocation $\mathcal{A}$ for that agent.

By using Theorem 3.3.17, we now have the following immediate corollary by combining Theorem 3.3.11 with Theorem 3.3.17.

Corollary 3.3.18 (*BIC black-box reduction*)**.** *If the market size parameter m is set to $\lceil \frac{1}{2\epsilon^{\Delta+1}} \rceil$, and the parameters of Algorithm 4 are set as stated in Theorem 3.3.11, then the composition of surrogate selection rule defined by Algorithm 4 with the allocation $\mathcal{A}$ is (1) a BIC mechanism, (2) the expected welfare is within an additive 3ϵ of the expected welfare of $\mathcal{A}$ for each agent, and (3) its running time is polynomial in n and $1/\epsilon$ given access to black-box oracle $\mathcal{A}$.*[8]

3.4 Conclusion

We conclude the chapter by summarizing our contributions and going over a list of related open problems.

3.4.1 Summary

We provided a polynomial time reduction from Bayesian incentive compatible mechanism design to Bayesian algorithm design for welfare maximization problems. Unlike prior results, our reduction achieves exact incentive compatibility for problems with multi-dimensional and continuous type spaces.

[8] Our result obviously holds when the doubling dimensions of type spaces are considered to be constant. For arbitrary large-dimensional type spaces, the running time is polynomial in n and $1/\epsilon^{\Delta}$.

The key technical barrier preventing exact incentive compatibility in prior black-box reductions is that repairing violations of incentive constraints requires understanding the distribution of the mechanism's output, which is typically #P-hard to compute. Reductions that instead estimate the output distribution by sampling inevitably suffer from sampling error, which typically precludes exact incentive compatibility. We overcame this barrier by employing and generalizing the computational model in the literature on Bernoulli Factories. As discussed in this chapter, in a Bernoulli factory problem one is given a function mapping the bias of an "input coin" to that of an "output coin", and the challenge is to efficiently simulate the output coin given only sample access to the input coin. We considered a generalization which we called the *expectations from samples* computational model, in which a problem instance is specified by a function mapping the expected values of a set of input distributions to a distribution over outcomes. The challenge is to give a polynomial time algorithm that exactly samples from the distribution over outcomes given only sample access to the input distributions.

In this model, we gave a polynomial time algorithm for the function given by *exponential weights*: expected values of the input distributions correspond to the weights of alternatives and we wish to select an alternative with probability proportional to an exponential function of its weight. This algorithm was the key ingredient in designing an incentive compatible mechanism for bipartite matching, which we could use to make the approximately incentive compatible reduction of Hartline et al. (2015) exactly incentive compatible.

3.4.2 Open Problems

1. One can think of generalizations of the Bernoulli race problem to other combinatorial settings. In fact, imagine we have a ground set of elements and a coin (with unknown bias) corresponding to each element. Moreover, a feasibility environment, i.e. a collection of subsets of the ground set, is given. The question is to pick a feasible set in a randomized fashion by flipping the coins (i.e. sampling), such that the marginal probability that coin i belongs to the set is proportional to its bias. We already solved the problem for $1-$uniform matroid (Bernoulli race), and solving this problem for other feasibility environments remain open.
2. Given access to an ϵ-BIC mechanism, can one produce an exact BIC mechanism through an efficient black-box reduction that preserves the revenue? In general, what other forms of exact BIC black-box reduction are possible in mechanism design?

CHAPTER 4
SIMPLE MECHANISMS FOR SINGLE-ITEM REVENUE MAXIMIZATION

Methods from theoretical computer science are amplifying the understanding of studied phenomena broadly. A quintessential example from auction theory is the following. Myerson (1981) is oft quoted as showing that the second-price auction with a reserve price is revenue optimal among all mechanisms for selling a single item. This result is touted as triumph for microeconomic theory as in practice reserve-pricing mechanisms are widely prevalent, e.g., eBay's auction. A key assumption in this result, though, is that the agents in the auction are a priori identical; moreover, relaxation of this assumption renders the theoretically optimal auction much more complex and infrequently observed in practice. Optimality of reserve pricing with agent symmetry, thus, does not explain its prevalence broadly in asymmetric settings, e.g., eBay's auction where agents can be distinguished by public bidding history and reputation. This chapter considers the approximate optimality of anonymous pricing and auctions with anonymous reserves, e.g., eBay's buy-it-now pricing and auction, and justifies their wide prevalence in asymmetric environments.

Organization of the chapter. In Section 4.1 we define the anonymous pricing revenue approximation problem and review the literature. In Section 4.2 we summarize our approach and techniques. In Section 4.3 we give detailed proofs of our results. Finally, we conclude by summarizing the chapter and proposing some interesting open problems in Section 4.4.

4.1 Preliminary

In this section, we give an overview of the problem we are solving in this Chapter. We start by formally defining the worst-case revenue approximation ratio of anonymous pricing compared to the optimal auction in Section 4.1.1. We then locate our result in the related literature for this problem in Section 4.1.2.

4.1.1 Problem Definition: Anonymous Pricing vs. Optimal

For selling a single item to agents with independent but non-identically distributed values, the revenue optimal auction is complex. However, with two agents, anonymous reserve pricing is a tight two approximation to the optimal auction; moreover, a surprising corollary of a main result of Hartline and Roughgarden (2009) showed that the second-price auction with anonymous reserves is generally no worse than a four approximation. The question of resolving the approximation factor within $[2, 4]$ has remained open for the last half decade. Technically, (a) tight methods for understanding symmetric solutions in asymmetric environments are undeveloped, and (b) the main method for analyzing auction revenue is by Myerson's virtual values but for this question virtual values give a mixed sign objective that renders challenging the analysis of approximation. As a way to resolve the anonymous reserve pricing vs. optimal auction question, we investigate the more demanding problem of approximating the revenue of the ex ante relaxation of the auction problem by posting an anonymous price (while supplies last) [1]. We consider the Bayesian mechanism

[1] We would prefer to compare the performance of anonymous pricing directly to the optimal auction of Myerson (1981); however, the standard formulation of the expected revenue of the optimal mechanism is difficult to analyze relative to the optimal anonymous pricing.

design setting for selling a single item described in Chapter 2, Section 2.1, and use the definitions and notations in Section 2.1 and Section 2.3 heavily in this chapter.

Definition 4.1.1 (Anonymous pricing vs. optimal revenue auction problem)**.** In a Bayesian single-item setting with independent and regular agents, analyze the worst-case approximation ratio of the revenue of the ex ante relaxation to the revenue of the optimal anonymous pricing over all regular instances, that is

$$\rho \triangleq \sup_{\mathcal{I} \in \textsc{Reg}} \frac{\textsc{ExAnteRev}(\mathcal{I})}{\textsc{OptPriceRev}(\mathcal{I})}, \tag{P1}$$

where $\textsc{Reg}$ denotes the space of all regular instances.

4.1.2 Related Work

This chapter is part of a central area of study at the intersection of computer science and economics that aims to quantify the performance of simple, practical mechanisms versus optimal mechanisms (see Hartline, 2013, for a survey). Immediately related results in this area fit in to three broad categories, (i) anonymous and discriminatory reserve pricing (Hartline and Roughgarden, 2009), (ii) while-supplies-last posted pricing (Chawla et al., 2010a), and (iii) increased competition with symmetric (Bulow and Klemperer, 1996) and asymmetric agents (Hartline and Roughgarden, 2009). The four approximation of Hartline and Roughgarden (2009) for anonymous reserve pricing is a corollary of their result for (iii) on (i). In comparison this chapter considers (ii), foremost, and obtain a lower bound for (i) as a corollary.

4.2 Our Approach in a Nutshell

The four approximation of Hartline and Roughgarden (2009) employs the only known approach for resolving (b), an approximate extension of the main theorem of Bulow and Klemperer (1996). Our approach directly takes on the challenge of (a) by giving a tight analysis of anonymous pricing versus a standard upper bound (described later in this chapter); corollaries of this analysis are tightened upper bounds on approximation of the optimal auction from four to $e \approx 2.718$ for both anonymous pricing and anonymous reserves. We conclude that, up to an e factor, discrimination and simultaneity are unimportant for driving revenue in single-item auctions.

4.2.1 Ex Ante Relaxation as a Benchmark

In the Bayesian single-item auction problem agents' values are drawn from a product distribution and expected revenue with respect to the distribution is to be optimized. Our development of the approximation bound for anonymous pricing and reserves is based on the analysis of four classes of mechanisms:

1. **Ex ante relaxation (a discriminatory pricing):** An ex ante pricing relaxes the feasibility constraint of the auction problem, from selling at most one item ex post, to selling at most one item in expectation over the draws of agents' values, i.e., ex ante. Fixing a probability of serving a given agent the optimal ex ante mechanism offers this agent a posted price irrespective of the outcome of the mechanism for the other agents. This relaxation was identified as a quantity of interest in Chawla et al. (2007) and its study was

refined by Alaei (2011) and Yan (2011).

2. **Auction:** An auction is any mechanism that maps values to outcome and payments subject to incentive and feasibility constraints. The optimal auction was characterized by Myerson (1981) and this characterization, though complex, is the foundation of modern auction theory.

3. **Anonymous reserve:** An anonymous reserve mechanisms is a variant of the second-price auction where bids below an anonymous reserve are discarded, the winner is the highest of the remaining agents, and the price charged is the maximum of the remaining agents' bids or the reserve if none other remain.

4. **Anonymous pricing**: An anonymous pricing mechanism posts an anonymous price and the first agent to arrive who is willing to pay this price will buy the item.

For any distribution over agents' values the optimal revenue attainable by each of these classes of mechanisms is non-increasing with respect to the above ordering. The final inequality of optimal anonymous reserve exceeding optimal anonymous pricing follows as with equal reserve and price, the former has only higher revenue as competition drives a higher price. The ex ante relaxation is a quantity for analysis only, while the other problems yield relevant mechanisms.

4.2.2 Anonymous Pricing vs. Ex Ante Relaxation

Our main technical theorem identifies the supremum over all instances of the ratio of the revenues of the optimal ex ante relaxation to the optimal anonymous pricing as the solution to an equation which evaluates to e. To our knowledge, this evaluation is not by any standard progressions or limits that where previously known to evaluate to e. The theorem assumes that the distribution of agents' values satisfies a standard *regularity* property that is satisfied by common distributions, e.g., uniform, normal, exponential; without this assumption we show that the approximation factor is n for n-agent environments (see Section 4.3.3).

Theorem 4.2.1 (Anonymous pricing versus ex ante relaxation). *For a single item environment with agents with independently (but non-identically) distributed values from regular distributions, the worst case approximation factor of anonymous pricing to the ex ante relaxation is* $\left(\mathcal{V}\left(\mathcal{Q}^{-1}(1)\right)+1\right)$ *which evaluates to* $e \approx 2.718$ *where*

$$\mathcal{V}(p) \triangleq p \cdot \ln\left(\frac{p^2}{p^2-1}\right), \qquad \mathcal{Q}(p) \triangleq \int_p^\infty -\frac{\mathcal{V}'(v)}{v}\,\mathrm{d}v.$$

Intuition for the theorem, as given by functions $\mathcal{V}(\cdot)$ and $\mathcal{Q}(\cdot)$, and its proof is as follows. We write a mathematical program to maximize the worst case approximation factor; a tight-in-the-limit continuous relaxation of this program gives the objective $1+\mathcal{V}(p)$ subject to $\mathcal{Q}(p) \leq 1$ which has the following interpretation. There is a continuum of agents and each agent value distribution is given by a pointmass at a value with some probability (and then a continuous distribution below the pointmass to minimally satisfy the regularity property). The function $\mathcal{V}(p)$ is the expected pointmass value from agents with pointmass value at least price p;[2] $\mathcal{Q}(p)$ is the expected number of these agents to realize their pointmass value. The optimal p^* meets the constraint with equality, i.e., $\mathcal{Q}(p^*) = 1$.

Corollaries of this theorem are the improved upper bounds by e (from 4) on the worst-case approximation factor of anonymous reserves and anonymous

[2] $\mathcal{V}(\cdot)$ excludes the contribution from the "highest valued agent" which is 1; hence the objective $1+\mathcal{V}(p)$.

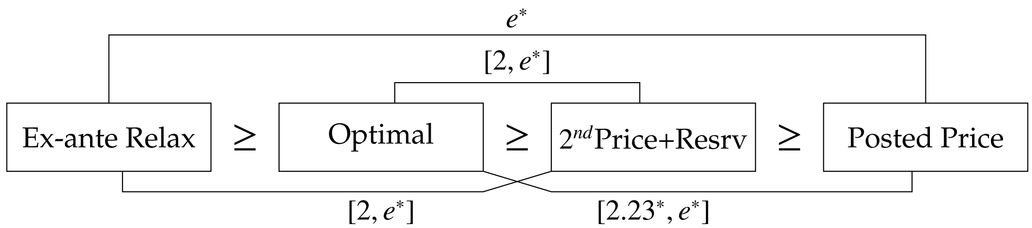

Figure 4.1: Revenue gap between mechanisms of study.

pricing with respect to the optimal auction. On the worst case instance of the theorem, however, the actual approximation factor of anonymous reserve and anonymous pricing are 2 and 2.23, respectively (see Section 4.3.4 of the Appendix). The latter improves on the known lower bound of two; the former does not improve the known lower bound. The question of refining our understanding of the revenue of anonymous reserves on worst-case instances and identifying a tight bound with respect to the optimal auction remains open. See Figure 4.1 (In this figure, the asterisk symbol * denotes new bounds derived in this chapter).

4.2.3 Implications on Multi-dimensional Mechanism Design

The corollary relating anonymous pricing to the optimal auction has implications on mechanism design for agents with multi-dimensional preferences (e.g., for multiple items; cf. Chawla et al., 2007). Understanding these problems, though there has been considerable recent progress, remains an area with fundamental open questions for optimization and approximation. Recently, Haghpanah and Hartline (2014) proved the optimality of uniform pricing for a single unit-demand buyer with values drawn from a large family of item-symmetric distributions. An immediate corollary of our anonymous pricing result is that, for a unit-demand buyer with values drawn from an asymmetric product distribution, uniform pricing is an e approximation (improved from four) to the optimal non-uniform pricing (cf. Cai and Daskalakis, 2011) and, via a result of Chawla et al. (2010b), a $2e$ approximation to the optimal pricing over lotteries (i.e., randomized allocations, improved from eight). Further refinement of this latter bound remains an important open question. These approximation results for a single agent automatically improve the approximation bounds for related multi-agent mechanism design problems based on uniform pricing, e.g., from Alaei et al. (2013). As one example, for selling an object that can be configured on sale in one of m configurations to n agents with independently (but non-identically) distributed values for each configuration (also satisfying a regularity property), the second-price auction with an anonymous reserve that configures the object as the winner most prefers is a $2e^2 \approx 14.8$ approximation to the optimal auction (which is sometimes randomized; improved from 32).

4.2.4 Worst-case Analysis in Mechanism Design

The field of algorithmic mechanism design contains many questions of constant approximation where tight bounds are not known. A key challenge of these problems is that the worst-case bounds are not given by small instances, e.g., $n = 2$ agents, but are instead approached in the limit with n. The field lacks general methods for analysis of this kind of problem. Our approach is similar to the recent successful approach of Chen et al. (2014) which identified the prior-free approximation ratio for digital good auctions as 2.42 (matching the lower bound of Goldberg et al., 2006). The approach at the first step writes the approximation ratio as the value of a mathematical program. In both our problem and that of Chen et al. (2014) the worst-case instance is attained in the limit with the number n of agents. With two success stories for this approach in algorithmic mechanism design, we are optimistic about the development of a set of tools for analyzing worst case approximation factors and that these tools will be useful for making progress on many other similar open questions in the area.

4.2.5 Organization of the Technical Parts

In Section 4.3.1, we provide details of our e-approximation in the worst-case for anonymous pricing versus ex ante relaxation when the valuations are independent and regular. Then, we prove this approximation ratio is indeed tight in Section 4.3.2. In Section 4.3.3 we consider the single item environment with independent but irregular agent value distributions and we show no better than an n-approximation is possible with respect to the optimal auction. In the same section, we also prove the n-approximation is tight. Finally, in Section 4.3.4, we develop lower-bounds on the approximation ratios of the anonymous pricing and reserve versus optimal auction (under regularity assumption) through simulating different hard instances. These hard instances are exactly the identified worst-case instances in Section 4.3.1 and Section 4.3.2 for the anonymous pricing versus ex ante problem.

4.3 Detailed Results: e-approximation in the Worst-case

In this section, we show matching upper and lower bounds for the approximation ratio of anonymous pricing versus ex ante relaxation. We then further investigate the necessity of regularity assumption. Finally, we conclude by providing a couple of simulations that lead us to find lower-bounds for approximation ratios of anonymous pricing and reserve auction with respect to the optimal revenue auction.

4.3.1 Upper-bound Analysis

Program (P1) defines a tight upper bound on the ratio, denoted by ρ, of the revenue of the ex ante relaxation to the revenue of the optimal anonymous pricing. This program can be thought of as a continuous optimization problem over regular distributions with the objective of maximizing the aforementioned ratio. To get our result in this section, we develop techniques to upper-bound the value of this program. In Section 4.3.2, we show our relaxation is indeed tight.

Overview of the Upper-bound Analysis

By normalizing the optimal anonymous pricing revenue to be one, (P1) is equivalent to the following program:

$$\begin{aligned}\rho = \quad & \sup_{\mathcal{I}\in\textsc{Reg}} \quad \textsc{ExAnteRev}(\mathcal{I}) && \text{(P2)}\\ & \text{subject to} \quad \textsc{PriceRev}(\mathcal{I}, p) \le 1 \qquad \forall p \ge 1. && \text{(P2.1)}\end{aligned}$$

Note that $\textsc{PriceRev}(\mathcal{I}, p) < 1$ for $p \in [0, 1)$, so it is safe to assume prices are in range $[1, +\infty)$. We show that for any fixed n the supremum of this program is approached even when restricting to triangular revenue curve instances, i.e., ones of the form $\{\text{Tri}(\bar{v}_i, \bar{q}_i)\}_{i=1}^{n}$ with $\sum_i \bar{q}_i \le 1$ as defined in Section 2.1. Consequently, the problem is reduced to a discrete optimization problem over variables $\bar{\mathbf{v}} \triangleq (\bar{v}_1, \ldots, \bar{v}_n)$ and $\bar{\mathbf{q}} \triangleq (\bar{q}_1, \ldots, \bar{q}_n)$. An *assignment* for this optimization problem refers to a pair $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$. This optimization problem is still of infinite dimension because n is itself a variable. It also turns out to be highly non-convex. Re-index $\bar{\mathbf{v}}$ such that $\bar{v}_1 \geq \ldots \geq \bar{v}_n$. We will show that, for any fixed n, inequality (P2.1) can be assumed without loss of generality to be tight for all $p \in \{\bar{v}_1, \ldots, \bar{v}_n\}$; otherwise an instance for which at least one of these constraints is not tight could be modified to make all these constraints tight while improving the objective. Thus,

$$\textsc{PriceRev}(\{\mathrm{Tri}(\bar{v}_i, \bar{q}_i)\}_{i=1}^{n}, \bar{v}_k) = 1 \qquad \forall k \in \{1, \ldots, n\}. \tag{4.1}$$

Observe that for each $k \in \{1, \ldots, n\}$ the left hand side of equation (4.1) only depends on the first k agents, because the valuations of the rest of the agents are always below $\bar{v}_k$. Consequently once $\bar{v}_1, \ldots, \bar{v}_n$ are fixed, we can compute $\bar{q}_1, \ldots, \bar{q}_n$ by solving equation (4.1) for $k \in \{1, \ldots, n\}$ and using forward substitution. Unfortunately, the resulting formulation of $\bar{q}_k$ in terms of $\bar{v}_1, \ldots, \bar{v}_k$ is difficult to analyze directly for $k \geq 2$. Instead, we relax inequality (P2.1) in such a way that it leads to a tractable formulation of $\bar{\mathbf{q}}$ in terms of $\bar{\mathbf{v}}$. We also show that the relaxed inequality is tight which implies the value of the relaxed program is equal to that of the original program. Finally, we show that the supremum of the relaxed program is attained when $n \to \infty$, and roughly speaking the instance converges to a continuum of infinitesimal agents with triangular revenue curve distributions. For this continuum of agents, ρ is given simply by the optimization of p in the objective $1 + \mathcal{V}(p)$ subject to the constraint $\mathcal{Q}(p) \leq 1$ for the two functions $\mathcal{V}(\cdot)$ and $\mathcal{Q}(\cdot)$ given in the statement of Theorem 4.2.1.

Reduction to Triangular Revenue Curve Instances

We begin by showing that without loss of generality we can restrict program (P2) to triangle revenue curve instances.

Lemma 4.3.1. *The supremum of program* (P2) *is approached by triangle revenue curve instances, i.e., of the form* $\hat{\mathcal{I}} = \{\mathrm{Tri}(\bar{v}_i, \bar{q}_i)\}_{i=1}^n$ *with* $\sum_{i=1}^n \bar{q}_i \leq 1$.

Proof. We will show that for any regular instance $\mathcal{I} = \{F_i\}_{i=1}^n$, there exists a corresponding instance $\hat{\mathcal{I}} = \{\mathrm{Tri}(\bar{v}_i, \bar{q}_i)\}_{i=1}^n$ with $\sum_{i=1}^n \bar{q}_i \leq 1$ yielding the same optimal ex ante revenue and (weakly) smaller expected revenue from the optimal anonymous price.

Let $\bar{\mathbf{q}}$ be an optimal assignment for the ex ante relaxation program (2.5) that computes $\textsc{ExAnteRev}(\mathcal{I})$. Set $\bar{v}_i \leftarrow R_i(\bar{q}_i)/\bar{q}_i$ for each $i \in \{1, \ldots, n\}$, where R_i is the revenue curve of F_i. We show changing agent i's valuation distribution to $\mathrm{Tri}(\bar{v}_i, \bar{q}_i)$ can only decrease the revenue of any anonymous pricing ($\textsc{PriceRev}$) while preserving the revenue of the ex ante relaxation ($\textsc{ExAnteRev}$), which implies the statement of the Lemma 4.3.1.

Let $\hat{R}_i$ be the revenue curve of $\mathrm{Tri}(\bar{v}_i, \bar{q}_i)$. Observe that the change of distributions does not affect $\textsc{ExAnteRev}$ because $\hat{R}_i(\bar{q}_i) = R_i(\bar{q}_i)$ for all $i \in \{1, \ldots, n\}$, and $\hat{R}_i$ is a lower bound on R_i elsewhere as R_i is concave (see Figure 4.2). Therefore the replacement preserves the optimal value of convex program of Definition 2.5 which implies $\textsc{ExAnteRev}(\hat{\mathcal{I}}) = \textsc{ExAnteRev}(\mathcal{I})$.

Next, we show the replacement may only decrease the value of $\textsc{PriceRev}(p)$ at any $p > 0$. Fix a price p, and consider the price line corresponding to p, that is, the line with slope p passing through the origin (see Figure 4.3). Observe that

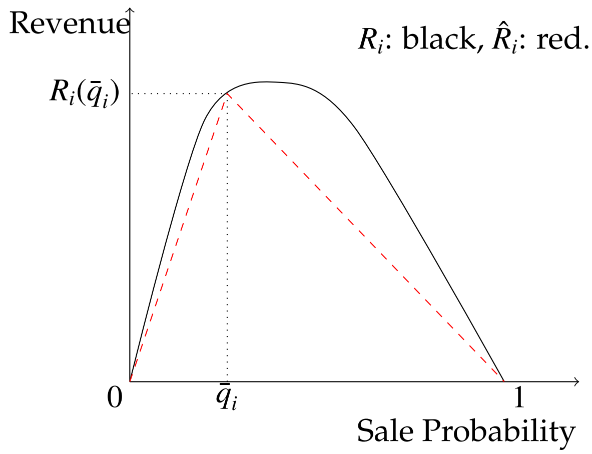

Figure 4.2: Replacing regular F_i with triangular revenue curve $\mathrm{Tri}(\bar{v}_i, \bar{q}_i)$.

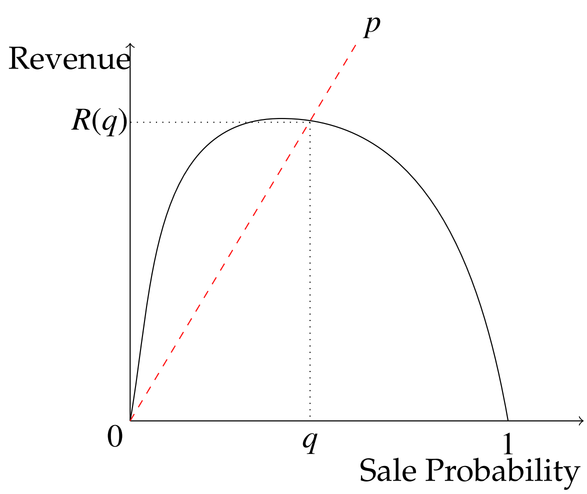

Figure 4.3: Intersection of revenue curve and price line p.

the probability of agent i's valuation being above p is equal to the q at which $R_i(q)$ intersects price line p. Given that $\hat{R}_i$ is a lower bound on R_i everywhere, the replacement may only decrease the probability of agent i's valuation being above p. Consequently, given that agents' valuations are distributed independently, the replacement may only decrease the revenue from sale at any anonymous price p, which implies $\textsc{PriceRev}(\hat{\mathcal{I}}) \leq \textsc{PriceRev}(\mathcal{I})$. $\square$

Combining Definition 4.3.1 with the algebraic formulation of PRICEREV and EXANTEREV from equations 2.8 and 2.7 (Chapter 2, Section 2.1) yields the following non-convex program for computing ρ:

$$\rho = \sup_{n\in\mathbb{N},\bar{\mathbf{v}},\bar{\mathbf{q}}} \quad \sum_{i=1}^{n} \bar{v}_i\bar{q}_i \tag{P3}$$

$$\text{subject to} \quad p\cdot\left(1-\prod_{i:\bar{v}_i\geq p}\frac{1}{1+\frac{\bar{v}_i\bar{q}_i}{p\cdot(1-\bar{q}_i)}}\right)\leq 1, \qquad \forall p\geq 1 \tag{P3.1}$$

$$\sum_{i=1}^{n}\bar{q}_i\leq 1$$

$$\bar{v}_i\geq 0, \bar{q}_i\geq 0 \qquad \forall i\in\{1,\ldots,n\}\,.$$

Relaxations and Canonical Assignments

In this section we find a relaxation of program (P3) where the corresponding *pricing revenue constraint* (P3.1) is tight for all $p \in \{\bar{v}_1,\ldots,\bar{v}_n\}$ and can thus be written as a program on variables $\bar{\mathbf{v}}$ alone (i.e., by solving for the appropriate $\bar{\mathbf{q}}$ in terms of $\bar{\mathbf{v}}$). To simplify the solution of $\bar{\mathbf{q}}$ in terms of $\bar{\mathbf{v}}$, we will first make a series of relaxations to the pricing revenue constraint (P3.1). We will point which of these relaxations are obviously tight, and the others we will prove to be tight in the limit with the number of agents n in Section 4.3.2, where we derive the matching lower bound.

Lemma 4.3.2 formalizes these relaxations as sketched below, the formal proof is given in the appendix, Section B.1. First, observe that the pricing revenue constraint (P3.1) can be rearranged as

$$\prod_{i:\bar{v}_i \geq p}\left(1+\frac{\bar{v}_i\bar{q}_i}{p\cdot(1-\bar{q}_i)}\right) \leq \left(\frac{p}{p-1}\right) \qquad \forall p \geq 1.$$

The first relaxation drops the constraint on $p \notin \{\bar{v}_1, \dots, \bar{v}_n\}$; this is without loss as the optimal anonymous price is always in $\{\bar{v}_1, \dots, \bar{v}_n\}$. We re-index such that $\bar{v}_1 \geq \cdots \geq \bar{v}_n$ and rephrase the relaxed constraint as

$$\prod_{i=1}^{k}\left(1+\frac{\bar{v}_i\bar{q}_i}{\bar{v}_k\cdot(1-\bar{q}_i)}\right) \leq \left(\frac{\bar{v}_k}{\bar{v}_k-1}\right) \qquad \forall k \in \{1, \dots, n\}.$$

As the second relaxation we drop the term $(1-\bar{q}_i)$ from the denominator of the left hand side and take the logarithm of both sides to get

$$\sum_{i=1}^{k} \ln\left(1+\frac{\bar{v}_i\bar{q}_i}{\bar{v}_k}\right) \leq \ln\left(\frac{\bar{v}_k}{\bar{v}_k-1}\right) \qquad \forall k \in \{1, \dots, n\}.$$

As the third relaxation we upper-bound $\ln\left(1+\frac{\bar{v}_i\bar{q}_i}{\bar{v}_k}\right)$ by $\frac{1}{\bar{v}_k}\ln(1+\bar{v}_i\bar{q}_i)$ for $i \geq 2$. Rearranging gives

$$\sum_{i=2}^{k} \ln\left(1+\bar{v}_i\bar{q}_i\right) \leq \bar{v}_k \ln\left(\frac{\bar{v}_k^2}{(\bar{v}_k-1)(\bar{v}_k+\bar{v}_1\bar{q}_1)}\right) \qquad \forall k \in \{2, \dots, n\}.$$

The previous relaxation uses the fact that $\frac{1}{a}\ln(1+b) \leq \ln(1+\frac{b}{a})$ for all $a \geq 1, b \geq 0$. In the proof, we will also show that $\bar{v}_1\bar{q}_1$ can be replaced with 1 both in the above constraint and in the objective function without loss of generality. Putting everything together, we will obtain the following program as a relaxation of program (P3).

$$\rho' = \sup_{n\in\mathbb{N},\bar{\mathbf{v}},\bar{\mathbf{q}}} \quad 1 + \sum_{i=2}^{n} \bar{v}_i \bar{q}_i \tag{P4}$$

$$\text{subject to} \quad \sum_{i=2}^{k} \ln\left(1 + \bar{v}_i\bar{q}_i\right) \le \mathcal{V}(\bar{v}_k) \qquad \forall k \in \{2,\dots,n\} \tag{P4.1}$$

$$\sum_{i=2}^{n} \bar{q}_i \le 1$$

$$\bar{v}_{i+1} \le \bar{v}_i, \qquad \forall i \in \{2,\dots,n-1\}$$

$$\bar{v}_i \ge 0, \bar{q}_i \ge 0 \qquad \forall i \in \{1,\dots,n\}.$$

where $\mathcal{V}(\cdot) = p \cdot \ln\left(\frac{p^2}{p^2-1}\right)$ is defined in Theorem 4.2.1.

Lemma 4.3.2. *The value of program* (P4)*, denoted by* ρ'*, is an upper bound on the value of program* (P3) *which is* ρ*.*

Next we show that we can assume without loss of generality the pricing revenue constraint (P4.1) is tight for all $k \in \{2,\dots,n\}$ in program (P4). That will allow us to specify one set of variables (e.g., $\bar{\mathbf{q}}$) in terms of the other set of variables (e.g., $\bar{\mathbf{v}}$), which consequently allows us to eliminate the former variables and drop the pricing revenue constraint (P4.1). To this end, we first define a *canonical* feasible solution for (P4), restriction to which is without loss as given by Lemma 4.3.3.

Definition 4.3.1. A feasible assignment $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ for (P4) is *canonical* if the pricing constraint (P4.1) is tight for all $k \in \{2,\dots,n\}$.

Lemma 4.3.3. *For any feasible assignment* $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ *for* (P4)*, there exists an equivalent canonical feasible assignment* $(\bar{\mathbf{v}}', \bar{\mathbf{q}}')$ *obtaining the same objective value, that is* $\sum_i \bar{v}_i\bar{q}_i = \sum_i \bar{v}'_i\bar{q}'_i$.

Proof. Without loss of generality assume $\bar{q}_k > 0$ for all $k \in \{2, \ldots, n\}$.[3] The right hand side of the pricing constraint (P4.1) is $\mathcal{V}(\bar{v}_k)$ which is decreasing in $\bar{v}_k$ (see Lemma 4.3.4) and approaches 0 as $\bar{v}_k \to \infty$, so for every $k \in \{2, \ldots, n\}$ there exists $\bar{v}'_k \geq \bar{v}_k$ such that

$$\sum_{i=2}^{k} \ln\left(1 + \bar{v}_i \bar{q}_i\right) = \mathcal{V}(\bar{v}'_k) \qquad \forall k \in \{2, \ldots, n\}.$$

Observe that by the above construction we always have $\bar{v}'_2 \geq \ldots \geq \bar{v}'_n$. We then decrease $\bar{q}_k$ to $\bar{q}'_k = \bar{q}_k \frac{\bar{v}_k}{\bar{v}'_k}$ for each $k \in \{2, \ldots, n\}$ to obtain the desired assignment $(\bar{\mathbf{v}}', \bar{\mathbf{q}}')$. $\square$

By Lemma 4.3.3, we can restrict our attention to canonical assignments of (P4) without loss of generality. In particular, we can fully identify such a canonical assignment by specifying only $\bar{\mathbf{v}} = (\bar{v}_1, \ldots, \bar{v}_n)$ since the corresponding $\bar{\mathbf{q}}$ is given by

$$\bar{q}_k = \frac{e^{\mathcal{V}(\bar{v}_k) - \mathcal{V}(\bar{v}_{k-1})} - 1}{\bar{v}_k} \qquad \forall k \in \{2, \ldots, n\}. \tag{4.2}$$

Therefore we can obtain from program (P4) the following program.

$$\begin{aligned}
\rho' = \quad & \sup_{n \in \mathbb{N}, \bar{\mathbf{v}}} & & 1 + \sum_{i=2}^{n} \bar{v}_i \bar{q}_i & & \text{(P5)}\\
& \text{subject to} & & \bar{q}_k = \frac{e^{\mathcal{V}(\bar{v}_k) - \mathcal{V}(\bar{v}_{k-1})} - 1}{\bar{v}_k} & \forall k \in \{2, \ldots, n\} \quad & \text{(P5.1)}\\
& & & \sum_{i=2}^{n} \bar{q}_i \leq 1 & & \text{(P5.2)}\\
& & & \bar{v}_{i+1} \leq \bar{v}_i, & \forall i \in \{2, \ldots, n-1\} & \\
& & & \bar{v}_i \geq 0,\ \bar{q}_i \geq 0 & \forall i \in \{1, \ldots, n\}. &
\end{aligned}$$

[3]If $\bar{q}_k = 0$, we can drop agent k without affecting feasibility or objective value.

Continuum of Agents.

Given that n itself is a variable, a solution to program (P5) can be practically specified by a finite subset $\bar{\mathbf{v}} \subset \mathbb{R}_+$ where $\bar{v}_i$ is the ith largest value in that subset. We now show that the optimal solution to program (P5) corresponds to $\bar{\mathbf{v}} = [p, \infty)$ (for some $p > 1$) which can be viewed as an instance with infinitely many infinitesimal agents (i.e., with $\bar{q}_i$ going to zero).

For any given $p' > p > 1$, we define a continuum of agents $[p, p')$ by defining for each $m \in \mathbb{N}$ a discrete family of agents of size m spanning $[p, p')$ and by taking the limit of this family as $m \to \infty$. Formally, for each $m \in \mathbb{N}$, we consider the family of agents with distributions $\{\text{Tri}(u_j, (e^{\mathcal{V}(u_j)-\mathcal{V}(u_{j-1})} - 1)/u_j)\}_{j=1}^m$ where $u_j = p' + \frac{j}{m}(p - p')$. Observe that the agents in these families satisfy equation (4.2). Furthermore, observe that

$$\lim_{\delta \to 0} \left(\frac{e^{\mathcal{V}(v)-\mathcal{V}(v+\delta)} - 1}{v} \cdot \frac{1}{\delta} \right) = -\frac{\mathcal{V}'(v)}{v}.$$

Therefore in a continuum of agents $[p, p')$ each infinitesimal agent $v \in [p, p')$ has a distribution of $\text{Tri}(v, -\frac{\mathcal{V}'(v)}{v}\,\mathrm{d}v)$, which implies that the contribution of $[p, p')$ to the objective value of (P5) is

$$\int_p^{p'} v \cdot (-\tfrac{\mathcal{V}'(v)}{v})\,\mathrm{d}v = \mathcal{V}(p) - \mathcal{V}(p'), \tag{4.3}$$

and the contribution of $[p, p')$ to the left hand side of the constraint (P5.2), i.e. $\sum_i \bar{q}_i \le 1$ which is referred to as the *capacity constraint*, is

$$\int_p^{p'} -\tfrac{\mathcal{V}'(v)}{v}\,\mathrm{d}v = Q(p) - Q(p'), \tag{4.4}$$

where $Q(p) = \int_p^\infty -\frac{1}{v}\mathcal{V}'(v)\,\mathrm{d}v$ as defined in Theorem 4.2.1.

Via the above derivation of a continuum of agents, program (P5), on the instance corresponding to the continuum $[p, \infty)$, simplifies as:

$$\rho'' = \max_{p \geq 1} \quad 1 + \mathcal{V}(p) \tag{P6}$$

$$\text{subject to} \quad \mathcal{Q}(p) \leq 1. \qquad .$$

Next we will sketch a construction that demonstrates that any feasible solution $\bar{\mathbf{v}} = (\bar{v}_1, \ldots, \bar{v}_n)$ to program (P5) can be replaced by a continuum of agents that corresponds to an interval $[p, \infty)$ (for some $p > 1$ to be determined) and the objective of (P5) is strictly increased. Note that $\bar{v}_1$ does not appear anywhere in (P5); for notational convenience we redefine it as $\bar{v}_1 = \infty$. Suppose for each $i \in \{2, \ldots, n\}$ we replace the agent $\bar{v}_i$ with the continuum of agents $[\bar{v}_i, \bar{v}_{i-1})$. It follows from equations 4.3 and 4.2 that this replacement changes the object value of (P5) by $\mathcal{V}(\bar{v}_i) - \mathcal{V}(\bar{v}_{i-1}) - \bar{v}_i\bar{q}_i = \ln(1 + \bar{v}_i\bar{q}_i) - \bar{v}_i\bar{q}_i < 0$ which is unfortunately always negative and thus the opposite of what we want to prove. On the other hand, it follows from equations 4.4 and 4.2 that this replacement also changes the left hand side of the capacity constraint (P5.2) by $\mathcal{Q}(\bar{v}_i) - \mathcal{Q}(\bar{v}_{i-1}) - \bar{q}_i$ which is also negative (as we will show later), and thus creates some slack in the capacity constraint (P5.2). Summing over the slack created in the capacity constraint (P5.2) from converting each agent to a continuum, we can add a new continuum of agents $[p, \bar{v}_n)$ where $p < \bar{v}_n$ is chosen to make the capacity constraint (P5.2) tight. As a consequence of the following claims, the net change in the objective value from this transformation is positive.

1. The amount of slack created in the capacity constraint (P5.2) by replacing $\bar{v}_i$ with $[\bar{v}_i, \bar{v}_{i-1})$ is more than the decrease in the objective value of (P5). By using equations 4.3 and 4.4, we can formally write this claim as

$$\bar{q}_i - (\mathcal{Q}(\bar{v}_i) - \mathcal{Q}(\bar{v}_{i-1})) > \bar{v}_i\bar{q}_i - (\mathcal{V}(\bar{v}_i) - \mathcal{V}(\bar{v}_{i-1})).$$

This is proved below in Lemma 4.3.6.

2. If there is a slack of $\Delta > 0$ in the capacity constraint (P5.2), it can be used to extend the last continuum of agents to increase the objective value by more than Δ. Using equations 4.3 and 4.4, we can formalize this claim as follows: if p is chosen such that $\mathcal{Q}(p) - \mathcal{Q}(\bar{v}_n) = \Delta$, then $\mathcal{V}(p) - \mathcal{V}(\bar{v}_n) > \Delta$. This is proved below in Lemma 4.3.4.

The suggestion from the above construction is that from any solution $\bar{\mathbf{v}}$ to program (P5), a price p can be identified such that the continuum of agents on $[p, \infty)$ has higher objective value. In other words, the optimal values of program (P5) and program (P6) are equal. This is proved below in Lemma 4.3.7; though we defer the proof that the solution of program (P6) corresponds to a limit solution of program (P5) to Section 4.3.2.

Algebraic Upper-bound Proof

The rest of this section develops a formal but purely algebraic proof that is based on the approach sketched in the previous paragraphs. The proofs of the first two lemmas, below, can be found in the appendix, Section B.1.

Lemma 4.3.4. *The functions $\mathcal{V}(p)$, $\mathcal{Q}(p)$, and $\mathcal{V}(p) - \mathcal{Q}(p)$ are all decreasing in p, for $p > 1$.*

Lemma 4.3.5. *for any $p' > p > 1$ the following inequality holds: $\mathcal{V}(p) - \mathcal{V}(p') < \ln(\frac{p}{p-1}) - \ln(\frac{p'}{p'-1})$.*

Lemma 4.3.6. *For any $p' > p > 1$ and $q = \frac{e^{\mathcal{V}(p)-\mathcal{V}(p')}-1}{p}$ the following inequalities hold:*

$$q - (\mathcal{Q}(p) - \mathcal{Q}(p')) \geq pq - (\mathcal{V}(p) - \mathcal{V}(p')) \geq 0 \tag{4.5}$$

Proof. Define

$$\begin{aligned} W(p,p') &\triangleq \mathcal{V}(p) - \mathcal{Q}(p) - \mathcal{V}(p') + \mathcal{Q}(p') + q - pq \\ &= \mathcal{V}(p) - \mathcal{Q}(p) - \mathcal{V}(p') + \mathcal{Q}(p') - (p-1)\Big(e^{\mathcal{V}(p)-\mathcal{V}(p')} - 1\Big)/p. \end{aligned}$$

Now, observe that proving the first inequality in the statement of the lemma is equivalent to proving $W(p,p') > 0$. We instead prove that $W(p,p')$ is increasing in p' which together with the trivial fact that $W(p,p) = 0$ implies $W(p,p') > 0$.

$$\begin{aligned} \tfrac{\partial}{\partial p'} W(p,p') &= -\mathcal{V}'(p') + \mathcal{V}'(p')/p' + (p-1)\,\mathcal{V}'(p')\,e^{\mathcal{V}(p)-\mathcal{V}(p')}/p \\ &= -\mathcal{V}'(p')\left[\frac{p'-1}{p'} - \frac{p-1}{p}\,e^{\mathcal{V}(p)-\mathcal{V}(p')}\right] \\ &> -\mathcal{V}'(p')\left[\frac{p'-1}{p'} - \frac{p-1}{p}\,e^{\ln(\frac{p}{p-1})-\ln(\frac{p'}{p'-1})}\right] = 0, \end{aligned}$$

where the final inequality follows from Lemma 4.3.4 and Lemma 4.3.5: by Lemma 4.3.4, $-\frac{\partial}{\partial p'}\mathcal{V}(p') > 0$; and by Lemma 4.3.5, $e^{\mathcal{V}(p)-\mathcal{V}(p')}$ is less than $e^{\ln(\frac{p}{p-1})-\ln(\frac{p'}{p'-1})}$, so replacing the former with the latter only decreases the value of the expression inside the brackets because its coefficient is $-\frac{p-1}{p}$ which is negative.

The second inequality in the statement of the lemma follows trivially from the fact that $\mathcal{V}(p) - \mathcal{V}(p') = \ln(1+pq)$ thus $pq - (\mathcal{V}(p) - \mathcal{V}(p')) = pq - \ln(1+pq) > 0$. □

Lemma 4.3.7. *The value of program* (P6)*, denoted by* ρ''*, is an upper bound on the value of program* (P5) *which is* ρ'*.*

Proof. Let $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ be any arbitrary feasible assignment for program (P5). We show there exists a feasible assignment for program (P6) with objective value upper bounding the objective value of $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ in program (P5). Define $p^* \triangleq Q^{-1}(1)$, a candidate solution to program (P6) that meets the feasibility constraint with equality. Note that such a p^* exists because $Q(\infty) = 0$, $Q(1) = \infty$, and $Q(\cdot)$ is continuous. Observe that the objective value of (P5) for $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ satisfies:

$$
\begin{aligned}
1 + \sum_{k=2}^{n} \bar{v}_k \bar{q}_k &\leq 1 + \sum_{k=2}^{n} \Big(\mathcal{V}(\bar{v}_k) - \mathcal{V}(\bar{v}_{k-1}) - (Q(\bar{v}_k) - Q(\bar{v}_{k-1})) + \bar{q}_k \Big) && [(4.3.6) \;\&\; \bar{v}_1 = \infty] \\
&= 1 + \mathcal{V}(\bar{v}_n) - Q(\bar{v}_n) + \sum_{k=2}^{n} \bar{q}_k && \text{as } \mathcal{V}(\infty) = Q(\infty) \\
&\leq 1 + \mathcal{V}(\bar{v}_n) - Q(\bar{v}_n) + 1 && \text{as } \sum_{k=2}^{n} \bar{q}_k \leq 1 \\
&< 1 + \mathcal{V}(p^*) - Q(p^*) + 1 && \text{as proved below} \qquad (*) \\
&= 1 + \mathcal{V}(p^*) && \text{as } Q(p^*) = 1.
\end{aligned}
$$

To prove inequality $(*)$ we show that $p^* < \bar{v}_n$ which together with Lemma 4.3.4 implies $\mathcal{V}(p^*) - Q(p^*) > \mathcal{V}(\bar{v}_n) - Q(\bar{v}_n)$. To prove $p^* < \bar{v}_n$ observe that Lemma 4.3.6 implies $\sum_{i=2}^{n} \bar{q}_i > \sum_{i=2}^{n} Q(\bar{v}_i) - Q(\bar{v}_{i-1}) = Q(\bar{v}_n)$. On the other hand $\sum_{i=2}^{n} \bar{q}_i \leq 1$. Therefore $Q(\bar{v}_n) < 1$ which implies $\bar{v}_n > p^*$ because $Q(p^*) = 1$ and, by Lemma 4.3.4, $Q(\cdot)$ is decreasing. □

We conclude the section with the proof of the upper-bound of Theorem 4.2.1.

Proof of upper-bound in Theorem 4.2.1. By putting all the pieces together, it follows from program (P1), Lemma 4.3.1, Lemma 4.3.2, Lemma 4.3.3, and the rest of the discussion in this section that ρ' which is computed by (P5) is an upper bound on the ratio of the ex ante relaxation to the expected revenue of the optimal anonymous pricing. Following Lemma 4.3.7, ρ' is upper bounded by the objective value of program (P6), i.e. ρ''. As $Q(\cdot)$ and $\mathcal{V}(\cdot)$ are decreasing (Lemma 4.3.4), the optimal solution to program (P6) is given by $\rho'' = 1 + \mathcal{V}(Q^{-1}(1))$ which numerically evaluates to $e \approx 2.718$. $\square$

4.3.2 Lower-Bound Analysis

In this section we show the tightness of our approximation, i.e. the matching lower-bound in Theorem 4.2.1. As a result of Lemma 4.3.1, it suffices to prove the following lemma.

Lemma 4.3.8. *For any $\epsilon > 0$ there exists a feasible assignment $(n, \bar{\mathbf{v}}, \bar{\mathbf{q}})$ of the program (P3) such that $\sum_{i=1}^{n} \bar{v}_i \bar{q}_i \geq 1 + \mathcal{V}(Q^{-1}(1)) - \epsilon$.*

Proof. Pick $\delta > 0$ such that:

$$(1-\delta)^2 \left(1 + \mathcal{V}\left(Q^{-1}\left(\tfrac{1}{(1+\delta)^2}\right) + \delta\right)\right) \geq 1 + \mathcal{V}(Q^{-1}(1)) - \epsilon$$

This is always possible as $\mathcal{V}$ and Q are decreasing. Lets define $\lambda = Q^{-1}(1)$. The proof is done in two steps:

Step 1: We find $\{v_i, q_i\}_{i=2}^{n}$ such that

$$\sum_{i=2}^{n} q_i \leq 1 \quad , \quad k = 2, \ldots, n: \quad \sum_{i=2}^{k} \ln(1 + v_i q_i) = \mathcal{V}(v_k)$$
$$\sum_{i=2}^{n} v_i q_i \geq \mathcal{V}\left(Q^{-1}\left(\tfrac{1}{(1+\delta)^2}\right) + \delta\right)$$

In our construction for $\{v_i, q_i\}_{i=2}^{n}$, we use two parameters $\Delta > 0$ and $V_T \geq \lambda$ which we fix later in the proof. Let $v_1 \triangleq \infty$, and for $i \geq 2$ set $v_i = V_T - (i-2)\Delta$ and $q_i = \frac{e^{\mathcal{V}(v_i) - \mathcal{V}(v_{i-1})} - 1}{v_i}$. Now, let $n = \max\{n_0 \in \mathbb{N} : \sum_{i=2}^{n_0} q_i \leq 1\}$. Obviously, $\sum_{i=2}^{n} q_i \leq 1$. Moreover, for any $2 \leq k \leq n$ we have $\sum_{i=2}^{k} \ln(1 + v_i q_i) = \sum_{i=2}^{k} (\mathcal{V}(v_i) - \mathcal{V}(v_{i-1})) =$

$\mathcal{V}(v_k) - \mathcal{V}(v_1) = \mathcal{V}(v_k)$. Now, pick $\delta' > 0$ small enough such that for $x \in [0, \delta']$ we have $\frac{e^x - 1}{x} \leq 1 + \delta$. Moreover, let Δ to be small enough and V_T to be large enough such that $\max\{\mathcal{V}(\lambda) - \mathcal{V}(\lambda + \Delta), \mathcal{V}(V_T), \Delta\} \leq \min\{\delta, \delta'\}$. First observe that due to Lemma 4.3.6 $q_i \geq Q(v_i) - Q(v_{i-1})$ which implies $2 \geq \sum_{i=1}^{n} q_i \geq Q(v_n) - Q(v_1) = Q(v_n)$. So, all values v_i are at least equal to λ. As $\mathcal{V}(.)$ is convex over $[1, \infty)$, we have $\mathcal{V}(v_i) - \mathcal{V}(v_{i-1}) \leq \mathcal{V}(\lambda) - \mathcal{V}(\lambda + \Delta) \leq \delta'$. As a result we have

$$\begin{aligned}
q_i &= \frac{e^{\mathcal{V}(v_i) - \mathcal{V}(v_{i-1})} - 1}{v_i} \leq (1+\delta)\frac{\mathcal{V}(v_i) - \mathcal{V}(v_{i-1})}{v_i} = (1+\delta)\int_{v_i}^{v_{i-1}} \frac{\mathcal{V}'(v)}{v_i}\,\mathrm{d}v. \\
&= (1+\delta)\int_{v_i}^{v_{i-1}} -\frac{v}{v_i}Q'(v)\,\mathrm{d}v = (1+\delta)\left(Q(v_i) - Q(v_{i-1}) + \int_{v_i}^{v_{i-1}} -\frac{v - v_i}{v_i}Q'(v)\,\mathrm{d}v\right) \\
&\leq (1+\delta)\left(Q(v_i) - Q(v_{i-1}) + \Delta\int_{v_i}^{v_{i-1}} -Q'(v)\,\mathrm{d}v\right) \leq (1+\delta)^2(Q(v_i) - Q(v_{i-1})) \qquad (4.6)
\end{aligned}$$

Based on the definition of n (number of distributions in our instance), we have $1 < \sum_{i=2}^{n+1} q_i$. By (4.6), we have $\sum_{i=2}^{n+1} q_i \leq (1+\delta)^2 \sum_{i=2}^{n+1}((Q(v_i) - Q(v_{i-1})) = (1+\delta)^2 Q(v_{n+1})$. Lets define $\lambda' \triangleq Q^{-1}\left(\frac{1}{(1+\delta)^2}\right)$. We conclude that $\lambda' \geq v_{n+1}$. Hence, $v_n \leq \lambda' + \Delta \leq \lambda' + \delta$. Moreover, using Lemma 4.3.1 we have

$$\sum_{i=2}^{n} v_i q_i \geq \sum_{i=2}^{n}(\mathcal{V}(v_i) - \mathcal{V}(v_{i-1})) = \mathcal{V}(v_n) \geq \mathcal{V}(\lambda' + \delta) = \mathcal{V}\left(Q^{-1}\left(\frac{1}{(1+\delta)^2}\right) + \delta\right) \qquad (4.7)$$

where the last inequality holds as $v_n \leq \lambda' + \delta$ and $\mathcal{V}$ is decreasing over $[1, \infty)$.

Step 2: Given $\{v_i, q_i\}_{i=2}^{n}$, we find an instance $\{\bar{v}_i, \bar{q}_i\}_{i=1}^{n}$ such that is feasible for program (P3) and $\sum_{i=1}^{n} \bar{v}_i \bar{q}_i \geq 1 + \mathcal{V}(Q^{-1}(1)) - \epsilon$. To do so, set $q_1 = \delta, v_1 = \frac{1}{\delta} - 1$. Now, for each $i, k \in [2:n]$ find $\gamma_{i,k}$ such that

$$1 + \frac{v_i q_i (1 - \gamma_{i,k})}{v_k} = (1 + v_i q_i)^{\frac{1}{v_k}} \qquad (4.8)$$

and then let $\bar{q}_i = (1-\delta)(1 - \max_{k \in [2:n]} \gamma_{i,k}) q_i$ and $\bar{v}_i = v_i$, for $i \in [2:n]$. Now we claim $\{\bar{v}_i, \bar{q}_i\}_{i=1}^{n}$ is a feasible assignment for the program (P3) . We have

$$\sum_{i=1}^{n} \bar{q}_i = \delta + (1-\delta)\sum_{i=2}^{n}(1 - \max_{k \in [2:n]} \gamma_{i,k}) q_i \leq \delta + (1-\delta)\sum_{i=2}^{n} q_i \leq 1. \qquad (4.9)$$

as $\sum_{i=2}^{n} q_i \leq 1$. Moreover, for $k \in [2:n]$ we have

$$
\begin{aligned}
\sum_{i=1}^{k} \ln\left(1 + \frac{\bar{v}_i \bar{q}_i}{\bar{v}_k(1-\bar{q}_i)}\right) &\leq \ln\left(1 + \frac{\bar{v}_1 \bar{q}_1}{\bar{v}_k(1-\bar{q}_1)}\right) + \sum_{i=2}^{k} \ln\left(1 + \frac{v_i q_i (1-\gamma_{i,k})}{v_k}\right) \\
&= \ln\left(\frac{v_k+1}{v_k}\right) + \frac{1}{v_k} \sum_{i=2}^{k} \ln(1 + v_i q_i) \\
&\leq \ln\left(\frac{v_k+1}{v_k}\right) + \ln\left(\frac{v_k^2}{v_k^2 - 1}\right) \\
&= \ln\left(\frac{\bar{v}_k}{\bar{v}_k - 1}\right)
\end{aligned}
$$

By taking exponents from both sides and rearranging the terms it is not hard to see $(n, \bar{\mathbf{v}}, \bar{\mathbf{q}})$ is a feasible assignment of program (P3). Additionally, for a fixed V_T all of the v_i's are bounded, i.e. $1 \leq v_i \leq V_T$. So, as Δ goes to zero we have $v_i q_i \to 0$ as $q_i \to 0$, and the left hand side of (4.8) converges to its right hand side. As a result, for small enough Δ, we can guarantee $\gamma_{i,k} \leq \delta$ for all i, k, and hence $\bar{q}_i \geq (1-\delta)^2 q_i$. So

$$
\begin{aligned}
\sum_{i=1}^{n} \bar{v}_i \bar{q}_i &\geq (1-\delta) + (1-\delta)^2 \left(\sum_{i=2}^{n} v_i q_i\right) \\
&\geq (1-\delta)^2 \left(1 + \sum_{i=2}^{n} v_i q_i\right) \\
&\geq (1-\delta^2)\left(1 + \mathcal{V}\left(Q^{-1}\left(\tfrac{1}{(1+\delta)^2}\right) + \delta\right)\right)
\end{aligned}
$$

which implies $\sum_{i=1}^{n} \bar{v}_i \bar{q}_i \geq 1 + \mathcal{V}(Q^{-1}(1)) - \epsilon$, as desired. □

4.3.3 Irregular inapproximability results

In this section we show that anonymous pricing and anonymous reserves are a tight n approximation to the optimal auction and ex ante relaxation. Specifically, we show a lower bound on the approximation factor of anonymous reserves to the optimal auction and an upper bound on the approximation factor of anonymous pricing to the ex ante relaxation. The ordering of these mechanisms by revenue then implies that all bounds are optimal and tight.

Proposition 4.3.9. *For n-agent, independent, non-identical, and irregular distributions the second-price auction with anonymous reserves is at best an n approximation to the optimal single-item auction.*

Proof. Consider the following value distribution

$$v_i = \begin{cases} h^i & \text{with probabiliy } h^{-i}, \\ 0 & \text{otherwise.} \end{cases}$$

On this distribution the ex ante relaxation has revenue $\sum_{i=0}^{n} h^i h^{-i} = n$ (and the optimal auction is no better). On the other hand, anonymous reserve and anonymous pricing of h^i for any $i \in \{1, \ldots, n\}$ gives revenue at least one. We will show that in the limit as h approaches infinity; these bounds are tight.

We first argue that in the limit of h the optimal auction revenue is n (the same as the ex ante relaxation). Consider the expected revenue of the following sequential posted pricing mechanism, which gives a lower-bound on the optimal revenue.[4] In decreasing order of price and until the first agent accepts her offered price, offer each agent i price h^i. This mechanism's revenue can be calculated as:

$$h^n \cdot \tfrac{1}{h^n} + \sum_{i=2}^{n} h^{n-i+1} \cdot \tfrac{1}{h^{n-i+1}} \prod_{j=1}^{i-1} (1 - \tfrac{1}{h^{n-j+1}}) = 1 + \sum_{i=2}^{n} \prod_{j=1}^{i-1} (1 - \tfrac{1}{h^{n-j+1}})$$

which converges to n as h goes to infinity.

[4]In fact, this sequential posted pricing mechanism is the optimal auction, but its optimality is unnecessary for the proof so we omit the details.

We now prove that the expected revenue of the second-price auction with any anonymous reserve in $\{h^i\}_{i=1}^n$ is at most one in the limit. (Any other reserve is only worse.) In the second-price auction with reserve, the winner pays the maximum of the highest agent value below her value and the reserve. An upper bound of this revenue is the sum of such a payment over all agents with values at least the reserve. So, for reserve h^i the contribution of $j \geq i$ to this upper bound is at most:

$$h^{-j}\,(j - i + h^i). \tag{4.10}$$

The first term, above, is the probability that agent j has high value h^j. Conditioned on her having the high value h^j, the second term bounds the agent's payment, the expected maximum of the highest lower-valued agent and the reserve h^i. It is at most $j - i + h^i$ as each agent between i and j has expected value one and the expectation of their maximum is at most the sum of their expectations. It follows from equation (4.10) that in the limit with h, the contribution from agent i to this bound is one and the contribution from agent $j \neq i$ is zero. Thus, the expected revenue of the second-price auction with reserve h^i is at most one in the limit. □

Proposition 4.3.10. *For independent, non-identical, irregular n-agent single-item environments, anonymous pricing is at worst an n approximation to the ex ante relaxation.*

Proof. Define $\{(\bar{v}_i, \bar{q}_i)\}_{i=1}^n$ as in equation (2.8) in Section 2.1 where the ex ante relaxation posts price $\bar{v}_i$ to agent i which is accepted with probability $\bar{q}_i$ and has total revenue $\sum_{i=1}^n \bar{v}_i\,\bar{q}_i$. For any i the anonymous pricing that posts price $\bar{v}_i$ obtains at least revenue $\bar{v}_i\,\bar{q}_i$. Thus, picking a uniformly random price from $\{\bar{v}_i\}_{i=1}^n$ gives an n approximation to the ex ante relaxation revenue $\sum_{i=1}^n \bar{v}_i\,\bar{q}_i$. The optimal anonymous price is no worse. □

4.3.4 Simulation results

In this section, we briefly discuss simulation results for the worst-case instance derived in Section 4.3.1. From these simulations we will see how fast, as a function of the number n of agents, the worst-case ratio of the ex ante relaxation to the expected revenue of optimal anonymous pricing converges to e. Moreover, for these worst-case instances we will be able to evaluate the approximation of the optimal auction by anonymous reserves and pricing.

We find a discrete approximation to the continuum instance of the upper bound as follows. As described in Section 4.3.1, it is without loss to assume that every anonymous pricing obtains the same revenue. For any $\{\bar{q}_i\}_{i=1}^n$ that sum to one, a decreasing sequence $\{\bar{v}_i\}_{i=1}^n$ can be identified inductively as $\bar{v}_k$ can be determined from $\{(\bar{v}_i, \bar{q}_i)\}_{i=1}^{k-1}$ by binary search and revenue calculations. Any sequence of $\{\bar{q}_i\}_{i=1}^n$, thus, gives a lower bound on the ratio of the ex ante relaxation to the anonymous pricing. One such sequence is $\bar{q}_i = 1/n$ for all i. We employ a sequence in our simulations that converges faster. Specifically we set $\{\bar{q}_i\}_{i=1}^n$ as the arithmetic progression that evenly divides $[0, \frac{2}{n}]$ (it can be easily verified that this sums to one as required).

After generating the instances, we also calculate the ratio of the revenue of the optimal mechanism to the anonymous pricing revenue, and the ratio of the revenue of the optimal mechanism to the revenue of the second price with anonymous reserve mechanism for these instances. We use sampling algorithm to calculate the revenue of the second price with anonymous reserve mechanism, while the calculation of the revenue of the optimal mechanism is exact. We report the results of our simulation in Table 4.1 and Figure 4.4 for various numbers n of agents. In the Table 4.1, ExAnteRev and OptPriceRev are the ex ante relaxation and optimal anonymous pricing revenues (as previously defined). OPTREV is the revenue of the optimal auction of Myerson (1981). OPTRESERVEREV is the revenue obtained by the second-price auction with an optimally chosen reserve price. In the Figure 4.4 the red sold line represents the ratio of ex-ante benchmark to anonymous pricing revenue. The blue line with star represents the ratio of the optimal revenue to anonymous pricing revenue. The black line with circle represents the ratio of the optimal revenue to the revenue of the second price auction with reserve.

n	2	10	100	500	1000	5000
EXANTEREV/OPTPRICEREV	2.000	2.622	2.710	2.717	2.718	2.718
OPTREV/OPTPRICEREV	2.000	2.187	2.227	2.231	2.231	2.232
OPTREV/OPTRESERVEREV	2.000	1.731	1.676	1.665	1.659	1.607

Table 4.1: The ratios of the revenues of various auctions and benchmarks.

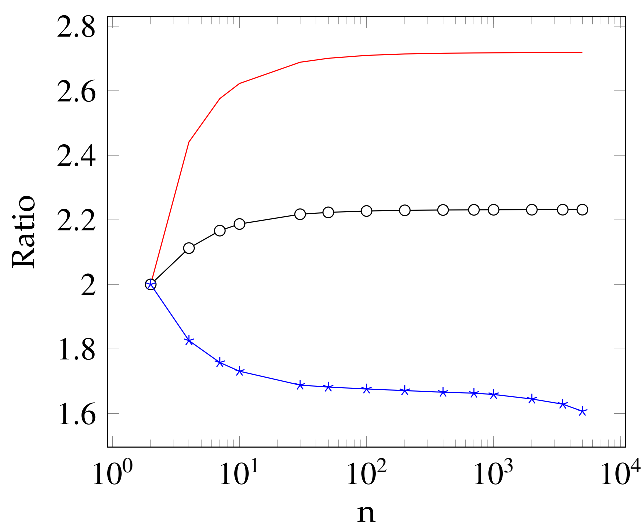

Figure 4.4: The ratios of the revenues of various auctions and benchmarks.

We conclude this section by highlighting the following theorem, derived from the table in Table 4.1.

Theorem 4.3.11. *There exists an instance for which anonymous pricing is a 2.232 approximation to the optimal auction.*

4.4 Conclusion

We conclude the chapter by summarizing our contributions and going over a list of related open problems.

4.4.1 Summary

For selling a single item to agents with independent but non-identically distributed values, the revenue optimal auction is complex. With respect to it, Hartline and Roughgarden (2009) showed that the approximation factor of the second-price auction with an anonymous reserve is between two and four. We considered the more demanding problem of approximating the revenue of the ex ante relaxation of the auction problem by posting an anonymous price (while supplies last) and proved that their worst-case ratio is e. As a corollary, the upper-bound of anonymous pricing or anonymous reserves versus the optimal auction improves from four to e. We concluded that, up to an e factor, discrimination and simultaneity are unimportant for driving revenue in single-item auctions.

4.4.2 Open Problems

1. In terms of expected revenue, what is the right approximation gap between anonymous pricing and optimal revenue auction? Our analysis shows the there is a tight multiplicative gap of e between the former and ex ante relaxation, and finding the tight gap with respect to the optimal auction remained open.

2. In terms of expected revenue, what is the right approximation gap between second price auction with anonymous reserve and either of optimal revenue auction or ex ante relaxation?
3. Can our style of worst-case analysis and characterizing the worst-case instance be used for other questions in mechanism design?

CHAPTER 5

ONLINE MECHANISMS FOR REVENUE MAXIMIZATION

In this chapter, we consider the following revenue maximization problem in a repeated setting, called the *online posted pricing* problem. In each period, the seller has a single item to sell, and a new prospective buyer. The seller offers to sell the item to the buyer at a given price; the buyer buys the item if and only if the price is below his private valuation for the item. The private valuation of the buyer itself is never revealed to the seller. We now cab ask the following questions: How should a monopolistic seller iteratively set the prices if he wishes to maximize his revenue? What if he also cares about market share?

Estimating price sensitivities and demand models in order to optimize revenue and market share is the bedrock of econometrics. The emergence of online marketplaces has enabled sellers to costlessly change prices, as well as collect huge amounts of data. This has renewed the interest in understanding best practices for data driven pricing. The extreme case of this when the price is updated for each buyer is the online pricing problem described above; one can always use this for less frequent price updates. Moreover this problem is intimately related to classical experimentation and estimation procedures.

In this chapter, we also consider the “full information” version of the problem, or what we call the *online auction* problem, where the valuations of the buyers are revealed to the algorithm after the buyer has made a decision. Such information may be available in a context where the buyers have to bid for the items, and are awarded the item if their bid is above a hidden price. How should an auction designer iteratively picks new auctions (or new hidden prices) if he wishes to maximize his revenue? What if he also cares about market share?

The objective of this chapter is to study the above problems through the lens of online learning, but from a different perspective that we call *multi-scale online learning*. In the multi-scale online learning framework, the objective is to design learning algorithms that are scale-free, i.e. their performance does not scale up with the range of rewards. We make bridges between this framework and our online mechanism design problems. By the help of these connections, we are able to find tight multi-scale regret bounds (which will be defined later in this chapter) for different online mechanism design questions that we are considering.

Organization of the chapter. In Section 5.1 we define the multi-scale online learning framework, define the online auction and pricing problem, and review the related literature for both of these topics. In Section 5.2 we summarize our approach and techniques. In Section 5.3 we give detailed proofs of our results. Finally, we conclude by summarizing the chapter and proposing some interesting open problems in Section 5.4.

5.1 Preliminary

In this section, we give an overview of the two main problems we are solving in this chapter, which are designing algorithms for online multi-scale learning, and designing mechanisms for online auctions/pricing. We formally define these two problems and the objectives we are pursuing in Section 5.1.1, and then locate our result in the related literature for this problem in Section 5.1.2.

5.1.1 Problem Definition: Auctions and Multi-scale Learning

We consider a variety of online algorithmic problems that are all parts of the *multiscale online learning* framework. We start by defining this framework and mentioning *action-specific* regret bounds for this general problem. Achieving this style of bounds for full-information (Arora et al., 2012) and bandit information setting (Auer et al., 1995; Audibert and Bubeck, 2009) is one of the main goals of this chapter. Next, we define different auction design problems that are covered by this framework. Here, the objective is to get multiplicative cum additive approximations for these problems [1] by the help of the multi-scale learning framework.

Multi-scale online learning framework

Our multiscale online learning framework is basically the classical learning from expert advice problem (under full-information) (Littlestone and Warmuth, 1994; Vovk, 1995; Arora et al., 2012) or multi-armed bandit problem (under partial-information) (Auer et al., 1995). The main difference is that the *range* of different experts/arms could be different. Suppose there is a set of actions A. The problem proceeds in T rounds, and in each round $t \in [T]$:[2]

- The algorithm picks an action $i_t \in A$
- The adversary picks a reward function $\mathbf{g}(t)$ simultaneously, where action i has reward $g_i(t)$.
- The algorithm gets the reward $g_{i_t}(t)$.

[1] Formally defined in Section 5.2.1.
[2] We use the notation $[n] := \{1, 2, \ldots, n\}$, for any $n \in \mathbb{N}$.

- In the *full information* setting, the algorithm sees the entire reward function $\mathbf{g}(t)$. In the *bandit* setting, the algorithm sees only its own reward, $g_{i_t}(t)$.

The total reward of the algorithm is denoted by

$$G_{\text{ALG}} := \sum_{t=1}^{T} g_{i_t}(t).$$

The standard "best fixed action" benchmark is

$$G_{\text{MAX}} := \max_{i \in A} \sum_{t=1}^{T} g_i(t).$$

We consider both full-information and the bandit setting:

- **Multi-scale experts:** The action set is countable. If the action set is finite of size k, we identify $A = [k]$. The reward $\mathbf{g}(t)$ is such that for all $i \in A$, $g_i(t) \in [0, c_i]$. The entire reward function $\mathbf{g}(t)$ is revealed to the algorithm after round t.
- **Multi-scale bandit learning:** The same as before but in the bandit information setting, i.e. only the reward of the action that has been picked by the algorithm will be revealed.

The goal is to obtain *action-specific regret* bounds, which we call also *multi-scale regret guarantees*. Towards this end, we define the following quantities.

$$G_i \quad := \quad \sum_{t \in [T]} g_i(t) \,, \tag{5.1}$$

$$\text{REGRET}_i \quad := \quad G_i - G_{\text{ALG}} \,. \tag{5.2}$$

Then, an action specific regret bound w.r.t. action i is an upper bound on $\mathbf{E}\left[\text{REGRET}_i\right]$ that only depends on the range c_i, as well as any *prior* distribution $\boldsymbol{\pi}$ over the action set A. We occasionally refer to this style of bound as *multi-scale regret bound*.

Online auction design

We investigate into the revenue maximization problem in online auctions and pricing. In our setting a seller sells an identical item in each period to a new buyer, or a new set of buyers, by running an auction or posting a price. Formally, we consider the following online auction design problems.

- **Online single buyer auction:** The action set $A = [1, h]$ [3]. The reward function is such that the adversary picks a *value* $v(t) \in [1, h]$ and for any *price* $i \in A$, the reward $g_i(t) := p \cdot \mathbf{1}(v(t) \geq i)$. This is the full information setting, where the value $v(t)$ is revealed to the algorithm after round t.
- **Online posted pricing:** The same as above, in the bandit setting. The algorithm only learns the indicator function $\mathbf{1}(v(t) \geq i_t)$ where i_t is the price it picks in round t.
- **Online multi buyer auction:** The action set is the set of all "Myerson-type" mechanisms for n buyers, for some $n \in \mathbb{N}$. (See Definition 5.3.2.) The adversary picks a valuation vector $\mathbf{v}(t) \in [1, h]^n$ and the reward of a mechanism M is its revenue when the valuation of the buyers is given by $\mathbf{v}(t)$; this is denoted by $\text{REV}_M(\mathbf{v}(t))$. The algorithm sees the full vector of valuations $\mathbf{v}(t)$.

By looking at the above problems as online learning problems, the technical goal is to obtain multiplicative cum additive approximations for these problems with G_{MAX} as the benchmark, à la Blum et al. (2004); Blum and Hartline (2005). The main improvement over these results that we are looking for in this chapter is that the additive term scales with the best price rather than maximum range

[3] As will be clear later, we discretize this action set to be able to use our learning algorithms.

h. Such a bound would be the auction version of the multi-scale style of regret bounds we have discussed earlier.

5.1.2 Related Work

The online pricing problem, also called *dynamic pricing*, is a much studied topic, across disciplines such as operations research and management science (Talluri and Van Ryzin, 2006), economics (Segal, 2003), marketing, and of course computer science. The multi-armed bandit approach to pricing is particularly popular. See den Boer (2015) for a recent survey on various approaches to the problem.

The online pricing problem has been studied from an *online learning* perspective, as a variant of the *multi-armed bandit* problem. The revenue of a pricing algorithm is compared to the revenue of the best fixed posted price, in hindsight, and the difference between the two, called the *regret*, is analyzed. No assumption is made on the distribution of values; the regret bounds are required to hold for the *worst case* sequence of values. Blum et al. (2004) assume that the buyer valuations are in $[1, h]$, and show the following multiplicative-plus-additive bound on the regret: for any $\epsilon \in (0, 1)$, the regret is at most ϵ times the revenue of the optimal price, + $O(\epsilon^{-2} h \log h \log\log h)$. Blum and Hartline (2005) show that the additive factor can be made to be $O(\epsilon^{-3} h \log\log h)$, trading off a $\log h$ factor for an extra ϵ^{-1} factor. Bar-Yossef et al. (2002), Blum et al. (2004), and Blum and Hartline (2005) also consider the online auction problem, as a variant of the *best expert* problem, in which they showed the additive part of the bound is $O(\epsilon^{-1} h \log(\epsilon^{-1}))$.

Kleinberg and Leighton (2003) consider the online pricing problem, under the assumption that the values are in $[0, 1]$, and considered purely additive factors. They showed that the minimax additive regret is $\tilde{\Theta}(T^{2/3})$, where T is the number of periods. This is similar in spirit to regret bounds that scale with h, since one has to normalize the values so that they are in $[0, 1]$. The finer distinction about the magnitude of the best fixed price is absent in this paper. Recently, Syrgkanis (2017) also consider the online auction problem, with an emphasis on a notion of "oracle based" computational efficiency. They assume the values are all in $[0, 1]$ and don't consider the scaling issue that we do; this makes their contribution orthogonal to ours.

Starting with Dhangwatnotai et al. (2014), there has been a spate of recent results analyzing the *sample complexity* of pricing and auction problems. In this category of problems, the designer receives i.i.d. samples from the buyer(s) value distribution and the objective is to obtain $(1 - \epsilon)$-approximation mechanism for the optimal revenue by using as few samples as possible. Cole and Roughgarden (2014) and Devanur et al. (2016) consider multiple buyer auctions with regular distributions (with unbounded valuations) and give sample complexity bounds that are polynomial in n and ϵ^{-1}, where n is the number of buyers. Morgenstern and Roughgarden (2015) consider arbitrary distributions with values bounded by h, and gave bounds that are polynomial in n, h, and ϵ^{-1}. Roughgarden and Schrijvers (2016); Huang et al. (2015) give further improvements on the single- and multi-buyer versions respectively; tables 5.1 and 5.2 give a summary of these results, for the problems we consider. The dynamic pricing problem has also been studied when there are a given number of copies of the item to sell (limited supply) (Agrawal and Devanur, 2014; Babaioff et al., 2015a; Badanidiyuru et al., 2013a; Besbes and Zeevi, 2009). There are also variants where the seller interacts with the same buyer repeatedly, and the buyer can strategize to influence his utility in the future periods (Amin et al., 2013; Devanur et al., 2014).

Foster et al. (2017) also consider the multi-scale online learning problem motivated by a model selection problem. They consider additive bounds, for the symmetric case, for full information, but not bandit feedback. Their regret bounds are not comparable to ours in general; our bounds are better for the pricing/auction applications we consider, and their bounds are better for their application.

5.2 Our Approach in a Nutshell

An undesirable aspect of the bounds in Blum and Hartline (2005) is that they scale *linearly with h*; this is particularly problematic when h is an estimate and we might set it to be a generous upper bound on the range of prices we wish to consider. A typical use case is when the same algorithm is used for many different products, with widely varying price ranges. We may not be able to manually tune the range for each product separately.

This dependency on h seems unavoidable, as is reflected by the lower bounds for the problem. (Lower bounds are discussed later in the introduction.) Yet, somewhat surprisingly, **our first contribution in this chapter is to show that we can replace h by the best fixed price**[4] (that is used in the definition of the benchmark). In particular, we show that the additive bound can

[4]Standard bounds allow regret to depend on the loss of the best action instead of the worst case loss. However, even such bounds still depend linearly on the range of the losses, and thus they would not allow to replace h by the best fixed price.

be made to be $O(\epsilon^{-2} p^* \log h)$, where p^* is the best fixed price in hindsight. This allows us to use a very generous estimate for h; we only lose a $\log h$ factor. The algorithm balances exploration probabilities of different prices carefully and automatically zooms in on the relevant price range. This does not violate known lower bounds, since in those instances p^* is close to h.

For the online auction problem, as mentioned earlier, Bar-Yossef et al. (2002), Blum et al. (2004), and Blum and Hartline (2005) improved the additive term to $O(\epsilon^{-1} h \log(\epsilon^{-1}))$, which is tight. Once again, *we show that h can be replaced with p^**; in particular, we show that the additive term can be made to be $O(\epsilon^{-1} p^* \log(h\epsilon^{-1}))$.

5.2.1 Purely multiplicative bounds and sample complexity

The regret bounds mentioned above can be turned into a purely multiplicative factor in the following way: for any $\epsilon > 0$, the algorithm is guaranteed to get a $1 - O(\epsilon)$ fraction of the best fixed price revenue, provided the number of periods $T \geq E/\epsilon$, where E is the additive term in the regret bounds above. This follows from the observation that a revenue of T is a lower bound on the best fixed price revenue. Call the number of periods required to get a $1 - \epsilon$ multiplicative approximation (as a function of ϵ) as the *convergence rate* of the algorithm.

A $1 - \epsilon$ multiplicative factor is also the target in the recent line of work on the sample complexity of auctions started by Dhangwatnotai et al. (2014); Cole and Roughgarden (2014). (For a more comprehensive discussion of this line of work refer to Section 5.1.2). Here, as mentioned earlier, *i.i.d.* samples of the valuations are given from a *fixed but unknown distribution*, and the goal is to find a price such that its revenue w.r.t. the hidden distribution is a $1-\epsilon$ fraction of the optimum revenue for this distribution. The sample complexity is the minimum number of samples needed to guarantee this (as a function of ϵ).

The sample complexity and the convergence rate (for the full information setting) are closely related to each other. The sample complexity is always smaller than the convergence rate as the former is easier due to the following.

1. The valuations are i.i.d. in case of sample complexity whereas they can be arbitrary (worst case) in case of convergence rate.
2. Sample complexity corresponds to an offline problem: you get all the samples at once. Convergence rate corresponds to an online problem: you need to decide what to do on a given valuation without knowing what valuations arrive in the future.

This is formalized in terms of an *online to offline reduction* [folklore] which shows that a convergence rate upper bound can be automatically translated to a sample complexity upper bound. This lets us convert sample complexity lower bounds into lower bounds on the convergence rate, and in turn into lower bounds on the additive error E in an additive plus multiplicative regret bound. For example, the additive error for the online auction problem (and hence also for the posted pricing problem[5]) cannot be $o(h\epsilon^{-1})$ (Huang et al., 2015). Moreover, it is insightful to compare convergence rates we show with *the best known sample complexity upper bound; proving better convergence rates would mean improving these bounds as well.*

[5] We conjecture that the lower bound for the posted pricing problem should be worse by a factor of ϵ^{-1}, since one needs to explore about ϵ^{-1} different prices.

A natural target convergence rate for a problem is therefore the corresponding sample complexity, but achieving this is not always trivial. An interesting version of the sample complexity bound for auctions did not have an analogous convergence rate bound. This version takes into account *both revenue and market share*, and surprisingly, gets sample complexity bounds that are *scale free*; there is no dependence on h, which means it works for unbounded valuations! For any $\delta \in (0, 1)$, the best fixed price benchmark is relaxed to ignore those prices whose market share (or equivalently probability of sale) is below a δ fraction; as δ increases the benchmark is lower. This is a meaningful benchmark since in many cases revenue is not the only goal, even if you are a monopolist. A more reasonable goal is to maximize revenue subject to the constraint that the market share is above a certain threshold. What is more, this gives a sample complexity of $O(\epsilon^{-2}\delta^{-1}\log(\delta^{-1}\epsilon^{-1}))$ (Huang et al., 2015). In fact δ can be set to h^{-1} without loss of generality, when the values are in $[1, h]$,[6] and the above bound then matches the sample complexity w.r.t. the best fixed price revenue. In addition, this bound gives a precise interpolation: as the target market share δ increase, the number of samples needed decreases almost linearly.

The second contribution of this chapter is to show a convergence rate that almost matches the above sample complexity, for the full information setting. We have a mild dependence on h; the rate is proportional to $\log\log h$. Further, we also show a near optimal convergence rate for the posted pricing problem.[7]

[6] When the values are in $[1, h]$, we can guarantee a revenue of T by posting a price of 1, and to beat this, any other price (and in particular a price of h) would have to sell at least T/h times.

[7] Unfortunately, we cannot yet guarantee that our online algorithm itself gets a market share of δ, although we strongly believe that it does. Showing such bounds on the market share of the algorithm is an important avenue for future research.

Multiple buyers: All of our results in the full information (online auction) setting extend to the multiple buyer model. In this model, in each time period, a new set of n buyers competes for a single item. The seller runs a truthful auction that determines the winning buyer and his payment. The benchmark here is the set of all "Myerson-type" mechanisms. These are mechanisms that are optimal when each period has n buyers of potentially different types, and the value of each buyer is drawn independently from a type dependent distribution. In fact, our convergence rates also imply new sample complexity bounds for these problems (except that they are not computationally efficient).

The various bounds and comparison to previous work are summarized in Tables 5.1 and 5.2. In Table 5.1 note that sample complexity is for the offline case with i.i.d. samples from an unknown distribution. Convergence rate is for the online case with a worst case sequence. Sample complexity is always no larger than the convergence rate. Lower bounds hold for sample complexity too, except for the online posted pricing problem for which there is no sample complexity version. The additive-plus-multiplicative regret bounds are converted to convergence rates by dividing the additive error by ϵ. In the last row, n is the number of buyers. In the last column, p^* denotes the optimal price.

5.2.2 Multi-scale online learning

The main technical ingredients in our results are variants of the classical problems of learning from expert advice and multi-armed bandit. We introduce the multi-scale versions of these problems, where each action has its reward bounded in a different range. **Our third contribution is to give an algorithm for this problem whose regret w.r.t. a certain action scales with the range of**

	Lower bound	Upper bound		
		Best known (Sample complexity)	Best known (Convergence rate)	This chapter (Thm. 5.2.5)
Online single buyer auction	$\Omega\left(\frac{h}{\epsilon^2}\right)$ *	$\tilde{O}\left(\frac{h}{\epsilon^2}\right)$ †	$\tilde{O}\left(\frac{h}{\epsilon^2}\right)$ †	$\tilde{O}\left(\frac{p^*}{\epsilon^2}\right)$
Online posted pricing	$\Omega\left(\max\{\frac{h}{\epsilon^2}, \frac{1}{\epsilon^3}\}\right)$ *§	-	$\tilde{O}\left(\frac{h}{\epsilon^3}\right)$ †	$\tilde{O}\left(\frac{p^*}{\epsilon^3}\right)$
Online multi buyer auction	$\Omega(\frac{h}{\epsilon^2})$ *	$O(\frac{nh}{\epsilon^3})$ ‡	-	$\tilde{O}\left(\frac{nh}{\epsilon^3}\right)$

* Huang et al. (2015); † Blum et al. (2004); ‡ Devanur et al. (2016); Gonczarowski and Nisan (2017); § Kleinberg and Leighton (2003).

Table 5.1: Number of rounds/samples needed to get a $1-\epsilon$ approximation to the best offline price/mechanism.

	Lower bound (Sample complexity)	Upper bound	
		Best known (Sample complexity)	This chapter (Thm. 5.2.6)
Online single buyer auction	$\Omega\left(\frac{1}{\epsilon^2\delta}\right)$ *	$\tilde{O}\left(\frac{1}{\epsilon^2\delta}\right)$ *	$\tilde{O}\left(\frac{1}{\epsilon^2\delta}\right)$
Online posted pricing	$\Omega\left(\max\{\frac{1}{\epsilon^2\delta}, \frac{1}{\epsilon^3}\}\right)$ *†	-	$\tilde{O}\left(\frac{1}{\epsilon^4\delta}\right)$
Online multi buyer auction	$\Omega\left(\frac{1}{\epsilon^2\delta}\right)$ *	-	$\tilde{O}\left(\frac{n}{\epsilon^3\delta}\right)$

* Huang et al. (2015); † Kleinberg and Leighton (2003).

Table 5.2: Sample complexity and convergence rate w.r.t. the optimal mechanism/price with market share $\geq \delta$.

rewards for that particular action. To contrast, the regret bounds in the standard versions scale with the maximum range. We expect such bounds to be of independent interest.

The multi-scale versions of these problems exhibit subtle variations that don't appear in the standard versions. First of all, our applications to auctions and pricing have non-negative rewards, and this actually makes a difference. For both the expert and the bandit versions, the minimax regret bounds for non-negative rewards are *provably better* than those when rewards could be negative.

Further, for the bandit version, we can prove a better bound if we only require the bound to hold w.r.t. the *best* action, rather than *all* actions (for non-negative rewards). The various regret bounds and comparison to standard bounds are summarized in Table 5.3. In this table, note that non-negative rewards refers to when the reward of any action i at any time is in $[0, c_i]$, and symmetric range rewards refers to when the reward of any action i at any time is in $[-c_i, c_i]$. Moreover, suppose T is the time horizon, A is the action set, and k is the number of actions.

	Standard regret bound $O(\cdot)$	Multi-scale bound (this chapter)	
		Upper bound $O(\cdot)$	Lower bound $\Omega(\cdot)$
Experts/non-negative	$c_{\max}\sqrt{T\log(k)}$ *	$c_i\sqrt{T\log(kT)}$	$c_i\sqrt{T\log(k)}$
Bandits/non-negative	$c_{\max}\sqrt{Tk}$ †	$c_i T^{\frac{2}{3}}(k\log(kT))^{\frac{1}{3}}$	$c_i\sqrt{TK}$
		$c_{i^*}\sqrt{Tk\log(k)}$, i^* is the best action	-
Experts/symmetric	$c_{\max}\sqrt{T\log(k)}$ *	$c_i\sqrt{T\log(k\cdot\frac{c_{\max}}{c_{\min}})}$	$c_i\sqrt{T\log(k)}$
Bandits/symmetric	$c_{\max}\sqrt{Tk}$ †	$c_i\sqrt{Tk\cdot\frac{c_{\max}}{c_{\min}}\log(kT\cdot\frac{c_{\max}}{c_{\min}})}$	$c_i\sqrt{Tk\cdot\frac{c_{\max}}{c_{\min}}}$

* Freund and Schapire (1995); † Audibert and Bubeck (2009).

Table 5.3: Pure-additive regret bounds for non-negative rewards and symmetric range adversarial rewards.

We use algorithms based on online (stochastic) mirror descent (OSMD) (Bubeck, 2011), with a weighted negative entropy as the Legendre function. This framework gives regret bounds in terms of a "local norm" as well as an "initial divergence", which we then bound differently for each version of the problem. In the technical sections we highlight how the subtle variations arise as a result of different techniques used to bound these two terms.

5.2.3 Main Results and Multi-scale Style Regret Bounds

Multi-scale regret bounds for online learning

The regret bound w.r.t. action i, i.e., an upper bound on $\mathbf{E}[\text{REGRET}_i]$, depends on the range c_i, as well as any *prior* distribution $\boldsymbol{\pi}$ over the action set A; this way, we can handle countably many actions. Let $c_{\min} = \inf_{i \in A} c_i$ and $c_{\max} = \sup_{i \in A} c_i$ (if applicable) be the minimum and the maximum range. We first state a version of the regret bound which is parameterized by $\epsilon > 0$; such bounds are stronger than $\sqrt{T}$ type bounds which are more standard.

Theorem 5.2.1. *There exists an algorithm for the multi-scale experts problem that takes as input any distribution $\boldsymbol{\pi}$ over A, the ranges c_i, $\forall\, i \in A$, and a parameter $0 < \epsilon \leq 1$, and satisfies:*

$$\forall i \in A: \quad \mathbf{E}[\text{REGRET}_i] \leq \epsilon \cdot G_i + O\left(\frac{1}{\epsilon} \log\left(\frac{1}{\epsilon \pi_i}\right) \cdot c_i\right) \tag{5.3}$$

Compare this to what you get by using the standard analysis for the experts problem (Arora et al., 2012), where the second term in the regret bound is $O\left(\frac{1}{\epsilon}\log(k) \cdot c_{\max}\right)$. Choosing $\boldsymbol{\pi}$ to be the uniform distribution in the above theorem gives $O\left(\frac{1}{\epsilon}\log\left(\frac{k}{\epsilon}\right) \cdot c_i\right)$. Also, one can compare the pure-additive version of this bound with the classic pure-additive regret bound $O\left(c_{\max} \cdot \sqrt{T \log(k)}\right)$ for the experts problem by setting $\epsilon = \sqrt{\frac{\log(kT)}{T}}$ (Corollary 5.2.2).

Corollary 5.2.2. *There exists an algorithm for the multi-scale experts problem that takes as input the ranges c_i, $\forall\, i \in A$, and satisfies:*

$$\forall i \in A: \quad \mathbf{E}[\text{REGRET}_i] \leq O\left(c_i \cdot \sqrt{T \log(kT)}\right) \tag{5.4}$$

For the bandit version, we can get a similar regret guarantee, but only for the *best* action. If we require the regret bound to hold for all actions, then we can only get a weaker bound, where the second term has ϵ^{-2} instead of ϵ^{-1}. The difference between the bounds for the bandit and the full information setting is essentially a factor of k, which is unavoidable.

Theorem 5.2.3. *There exists an algorithm for the online multi-scale bandits problem that takes as input the ranges c_i, $\forall\, i \in A$, and a parameter $0 < \epsilon \le 1$, and satisfies,*

- *for $i^* = \arg\max_{i \in A} G_i$,*

$$\mathbf{E}\left[\text{REGRET}_{i^*}\right] \le \epsilon \cdot G_{i^*} + O\left(\tfrac{1}{\epsilon} k \log\left(\tfrac{k}{\epsilon}\right) \cdot c_{i^*}\right). \tag{5.5}$$

- *for all $i \in A$,*

$$\mathbf{E}\left[\text{REGRET}_{i}\right] \le \epsilon \cdot G_{i} + O\left(\tfrac{1}{\epsilon^2} k \log\left(\tfrac{k}{\epsilon}\right) \cdot c_{i}\right). \tag{5.6}$$

Also, one can compute the pure-additive versions of the bounds in Theorems 5.2.3 by setting $\epsilon = \sqrt{\frac{k\log(kT)}{T}}$ and $\epsilon = (\frac{k\log(kT)}{T})^{\frac{1}{3}}$ resepctively (Corollary 5.2.4), and compare with the pure-additive regret bound $O\left(c_{\max} \cdot \sqrt{Tk}\right)$ for the adversarial multi-armed bandit problem (Audibert and Bubeck, 2009).

Corollary 5.2.4. *There exist algorithms for the online multi-scale bandits problem that satisfies,*

- *For $i^* = \arg\max_{i \in A} G_i$,*

$$\mathbf{E}\left[\text{REGRET}_{i^*}\right] \le O\left(c_{i^*} \cdot \sqrt{Tk\log(kT)}\right) \tag{5.7}$$

- *For all $i \in A$,*

$$\mathbf{E}\left[\text{REGRET}_{i}\right] \le O\left(c_{i} \cdot T^{\frac{2}{3}}(k\log(kT))^{\frac{1}{3}}\right) \tag{5.8}$$

Multi-scale regret bounds for online auctions

We show how to get multiplicative cum additive approximations for these problems with G_{MAX} as the benchmark, à la Blum et al. (2004); Blum and Hartline (2005). The main improvement over these results is that the additive term scales with the best price rather than h. Let p^* be the best fixed price on hindsight, which is the price that achieves G_{MAX}.

Theorem 5.2.5. *There are algorithms for the online single buyer auction, online posted price auction, and the online multi buyer auction problems that take as input a parameter $\epsilon > 0$, and satsify $G_{\text{ALG}} \geq (1-\epsilon)G_{\text{MAX}} - O(E)$, where respectively (for the three problems mentioned above)*

$$E = \frac{p^* \log(\log h/\epsilon)}{\epsilon}, \quad \frac{p^* \log h \log(\log h/\epsilon)}{\epsilon^2}, \quad \textit{and } \frac{hn \log h \log(n \log h/\epsilon)}{\epsilon^2}.$$

Even if h is not known up front, we can still get the similar approximation guarantee for online single buyer auction and online multi buyer auction with:

$$E = \frac{p^* \log(p^*/\epsilon)}{\epsilon}, \quad \textit{and} \quad \frac{hn \log h \log(n \log h/\epsilon)}{\epsilon^2}.$$

Bounds on the *sample complexity* of auctions imply that the first bound in this theorem is tight up to logarithmic factors: the lower bound is $h\epsilon^{-1}$ in an instance where $p^* = h$, and the best upper bound known is $h\epsilon^{-1}\log(1/\epsilon)$. We conjecture that our bound for the online posted pricing problem is tight up to logarithmic factors, and leave resolving this as an open problem. The third bound is not comparable to the best sample complexity for the multi buyer auction problem by Roughgarden and Schrijvers (2016); it is better than theirs for large ϵ (when $1/\epsilon \leq o(nh)$), and is worse for smaller ϵ (when $1/\epsilon \geq \omega(nh)$). Also, compare these to the corresponding upper bounds for the first two problems by Blum et al.

(2004); Blum and Hartline (2005), which are respectively

$$\frac{h\log(1/\epsilon)}{\epsilon}, \quad \text{and } \min\left\{\frac{h\log h\log\log h}{\epsilon^2}, \frac{h\log\log h}{\epsilon^3}\right\}.$$

Competing with δ-guarded benchmarks

For the single buyer auction/pricing problem, we define a δ-guarded benchmark, for any $\delta \in [0,1]$. This benchmark is restricted to only those prices that sell the item in at least a δ fraction of the rounds.

$$G_{\text{MAX}}(\delta) := \max\left\{\sum_{t=1}^{T} g_p(t) : p \in A, \sum_{t=1}^{T} \mathbf{1}(v_t \geq p) \geq \delta T\right\}.$$

As observed in Footnote 6, one can replace δ with $1/h$ and get the corresponding guarantees for G_{MAX} rather than $G_{\text{MAX}}(\delta)$. However, the main point of these results is to show a graceful improvement of the bounds as δ is chosen to be larger.

Multiple buyers: For the multi-buyer online auction problem, we define the δ-guarded benchmark as follows. For any sequence of value vectors $\mathbf{v}(1), \mathbf{v}(2), \ldots, \mathbf{v}(T)$, let $\bar{V}$ denote the largest value such that there are at least δT distinct $\mathbf{v}(t)$'s with $\max_{i\in[n]} v_i(t) \geq \bar{V}$. Define the δ-guarded benchmark to be

$$\texttt{OPT}(\delta) = \max_M \sum_{t=1}^{T} Rev_M\left(\min(\bar{V}\vec{\mathbf{1}}, \mathbf{v}(t))\right),$$

where the min is to be understood to be applied coordinate-wise, and the max is over all Myerson-type mechanisms.

We focus on purely multiplicative approximation factors when competing with $\texttt{OPT}(\delta)$. In particular, for any given $\epsilon > 0$, we are interested in a $1-\epsilon$ approximation. We state our results in terms of the *convergence rate*. We say that

$T(\epsilon, \delta)$ is the convergence rate of an algorithm if for all time horizon $T \geq T(\epsilon, \delta)$, we are guaranteed that $G_{\text{ALG}} \geq (1-\epsilon)\texttt{OPT}(\delta)$. Our main results are as follows.

Theorem 5.2.6. *There are algorithms for the online single buyer auction, online posted pricing, and the online multi buyer auction problems with convergence rates respectively of*

$$O\left(\frac{\log(\log h/\epsilon)}{\epsilon^2\delta}\right), \quad O\left(\frac{\log h}{\epsilon^4\delta}\right), \quad \textit{and } O\left(\frac{n\log(1/\epsilon\delta)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^3\delta} + \frac{\log(\log h/\epsilon)}{\epsilon^2\delta}\right).$$

Even if h is not known up front, we can still get the following similar convergence rates for online single buyer auction and online multi buyer auction respectively:

$$O\left(\frac{\log(p^*/\epsilon)}{\epsilon^2\delta}\right), \quad \textit{and} \quad O\left(\frac{n\log(1/\epsilon\delta)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^3\delta} + \frac{\log(h/\epsilon)}{\epsilon^2\delta}\right).$$

Once again, we compare to the sample compexity bounds: our first is within a $\log\log h$ factor of the best sample complexity upper bound in Huang et al. (2015). The lower bound for the online single buyer auction is $\Omega(\delta^{-1}\epsilon^{-2})$, which is also the best lower bound known for the pricing and the multi-buyer problem.[8] For the online posted pricing problem, we conjecture that the right dependence on ϵ should be ϵ^{-3}. No sample complexity bounds for the multi-buyer problem were known before; in fact we introduce the definition of a δ-guarded benchmark for this problem.

Multi-scale online learning with symmetric range

The standard analysis for the experts and the bandit problems holds even if the range of $g_i(t)$ is $[-c_i, c_i]$, rather than $[0, c_i]$ as we have assumed. In contrast, there

[8] Cole and Roughgarden (2014) show that at least a linear dependence on n is necessary when the values are drawn from a regular distribution, but as is, their lower bound needs unbounded valuations. The lower bound probably holds for "large enough h" but it is not clear if it holds for all h.

are subtle differences on the best acheivable multi-scale regret bounds between the non-negative and the symmetric range. We first show the following upper bound for the full information setting when the range is symmetric. This bound follows the same style of action-specific regret bounds as in Theorem 5.2.1. More detailed discussion on how the choice of initial distribution $\boldsymbol{\pi}$ affects the bound is deferred to the appendix, Section C.1.1.

Theorem 5.2.7. *There exists an algorithm for the multi-scale experts problem with symmetric range that takes as input any distribution $\boldsymbol{\pi}$ over A, the ranges c_i, $\forall\, i \in A$, and a parameter $0 < \epsilon \leq 1$, and satisfies:*

$$\forall i \in A: \quad \mathbf{E}\left[\mathrm{REGRET}_i\right] \leq \epsilon \cdot \mathbf{E}\left[\sum_{t\in[T]} \left|g_t(i)\right|\right] + O\left(\frac{1}{\epsilon} \log\left(\frac{1}{\pi_i} \cdot \frac{c_i}{c_{\min}}\right) \cdot c_i\right). \tag{5.9}$$

Similar to Section 5.2.3, we can compute the pure-additive version of the bound in Theorem 5.2.7 by setting $\epsilon = \sqrt{\frac{\log(k \cdot \frac{c_{\max}}{c_{\min}})}{T}}$, as in Corollary 5.2.2.

Corollary 5.2.8. *There exists an algorithm for the online multi-scale experts problem with symmetric range that takes as input the ranges c_i, $\forall\, i \in A$, and satisfies:*

$$\forall i \in A: \quad \mathbf{E}\left[\mathrm{REGRET}_i\right] \leq O\left(c_i \cdot \sqrt{T \log(k \cdot \frac{c_{\max}}{c_{\min}})}\right) \tag{5.10}$$

If we compare the above regret bound with the standard $O(c_{\max}\sqrt{T \log k})$ regret bound for the experts problem, we see that we replace the dependency on $c_{\max}$ in the standard bound with $c_i \sqrt{\log(\frac{c_{\max}}{c_{\min}})}$. It is natural to ask whether we could get rid of the dependence on $\log(c_i/c_{\min})$ and show a regret bound of $O(c_i \sqrt{T \log k})$, like we did for non-negative rewards. However, the next theorem shows that this dependence on $\log(c_i/c_{\min})$ in the above bound is necessary, in a weak sense: where the constant in the $O(\cdot)$ is universal and does not depend on the ranges c_i. This is because the lower bound only holds for "small" values of the horizon T, which nonetheless grows with the $\{c_i\}$s.[9]

Theorem 5.2.9. *There exists an action set of size k, and ranges $c_i, \forall i \in [k]$, and time horizon T, such that for all algorithms for the online multi-scale experts problem with symmetric range, there is a sequence of T gain vectors such that*

$$\exists i \in A: \quad \mathbf{E}[\text{REGRET}_i] > \tfrac{c_i}{4} \cdot \sqrt{T \log(k \cdot \tfrac{c_{\max}}{c_{\min}})}$$

We then show the following upper bound for the bandit setting when the range is symmetric. This bound also follows the same style of action-specific regret bounds as in Theorem 5.2.3.

Theorem 5.2.10. *There exists an algorithm for the multi-scale bandits problem with symmetric range that takes as input the ranges c_i, $\forall\, i \in A$, and a parameter $0 < \epsilon \le 1/2$, and satisfies:*

$$\forall i \in A: \quad \mathbf{E}[\text{REGRET}_i] \le O\Big(\epsilon T + \tfrac{k}{\epsilon}\tfrac{c_{\max}}{c_{\min}} \log\left(\tfrac{k}{\epsilon}\tfrac{c_{\max}}{c_{\min}}\right)\Big) \cdot c_i. \tag{5.11}$$

Also, similar to Section 5.2.3, we can compute the pure-additive version of the regret bound in Theorem 5.2.10 by setting $\epsilon = \sqrt{\frac{k\frac{c_{\max}}{c_{\min}} \log(kT \cdot \frac{c_{\max}}{c_{\min}})}{T}}$, as in Corollary 5.2.2. This bound is comparable to the standard regret bound of $O(c_{\max}\sqrt{kT})$ (Audibert and Bubeck, 2009) for the adversarial multi-armed bandits problem.

Corollary 5.2.11. *There exists an algorithm for the online multi-scale bandits problem with symmetric range that satisfies:*

$$\forall i \in A: \quad \mathbf{E}[\text{REGRET}_i] \le O\left(c_i \cdot \sqrt{Tk \cdot \tfrac{c_{\max}}{c_{\min}} \log(kT \cdot \tfrac{c_{\max}}{c_{\min}})}\right). \tag{5.12}$$

[9]For this reason we chose not to include this bound in Table 5.3.

Once again, for the bandit problem, the following theorem shows that this bound cannot be improved beyond logarithmic factors (to get a guarantee like that of Theorem 5.2.3, for instance).

Theorem 5.2.12. *There exists an action set of size k, and ranges $c_i, \forall i \in [k]$, such that for all algorithms for the online multi-scale bandit problem with symmetric range, for all sufficiently large time horizon T, there is a sequence of T gain vectors such that*

$$\exists i \in A : \quad \mathbf{E}\left[\text{REGRET}_i\right] > \frac{c_i}{8\sqrt{2}} \cdot \sqrt{Tk \cdot \frac{c_{\max}}{c_{\min}}}.$$

5.2.4 Organization of the Technical Parts

We start in Section 5.3.1 by showing regret upper bounds for the multi-scale experts problem with non-negative rewards (Theorem 5.2.1). The corresponding upper bounds for the bandit version are in section 5.3.2 (Theorem 5.2.3). In Section 5.3.3 we show how the multi-scale regret bounds (Theorems 5.2.1 and 5.2.3) imply the corresponding bounds for the auction/pricing problems (Theorems 5.2.5 and 5.2.6). Finally, the regret (upper and lower) bounds for the symmetric range are discussed in Section 5.3.4 (Theorems 5.2.7, 5.2.9, 5.2.10, and 5.2.12).

5.3 Detailed Results: How to Get Multi-scale Regret Bounds

In this section, we discuss the details and required proofs of our results in this chapter. We first consider the full-information expert setting for multi-scale learning, and then we continue by considering the bandit problem. We then switch gears to the auction and pricing problems, where we use the bounds we developed for multi-scale learning to obtain near-optimal regret bounds for various online auction and pricing problems. We finally discuss the multi-scale online learning framework under the symmetric range assumption.

5.3.1 Multi-Scale Online Learning with Full Information

In this section, we look at the full information multi-scale learning problem, in which different experts have different ranges. We exploit this structure to achieve expert-specific regret bounds.

Here is a map of this section. We propose an algorithm that exploits the aforementioned structure, and later we show how this algorithm is an online mirror descent with weighted negative entropy as the Legendre function. For reward-only instances, we prove the regret bound *without* dependency on $\log(c_i/c_{\max})$.

Multi-Scale Multiplicative-Weight (MSMW) algorithm

We propose the "*Multi-Scale Multiplicative-Weight*" (MSMW) algorithm as a multiplicative-weight update style learning algorithm for our problem. The algorithm is presented in Algorithm 5. The main idea behind this algorithm is taking into account different ranges for different experts, and therefore

1. normalizing the reward of each expert accordingly, i.e. dividing the reward of expert i by c_i;
2. projecting the updated weights accordingly, by performing a *smooth multi-scale projection* into the simplex that will be described later.

Algorithm 5: MSMW

input initial distribution $\boldsymbol{\mu}$ over A, learning rate $0 < \eta \leq 1$.
initialize $\mathbf{p}(1)$ such that $p_i(1) = \mu_i$ for all $i \in A$.
for $t = 1, \ldots, T$ **do**
    Randomly pick an action drawn from $\mathbf{p}(t)$, and observe $\mathbf{g}(t)$.
    $\forall i \in A: \quad w_i(t+1) \leftarrow p_i(t) \cdot \exp(\eta \cdot \frac{g_i(t)}{c_i})$.
    Find λ^* (e.g., binary search) s.t. $\sum_{i \in A} w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) = 1$.
    $\forall i \in A: \quad p_i(t+1) \leftarrow w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i})$.
end for

Equivalence to Mirror Descent (OMD)

While it is possible to analyze the regret of the MSMW algorithm (Algorithm 5) by using first principles (analogous to the proof of Lemma 5.3.3 in the appendix, Section C.1.4), we take a different approach. We show how this algorithm is indeed an instance of the Online Mirror Descent (OMD) algorithm for a particular choice of *Legendre function*.

MSMW algorithm as an OMD. For our application, we focus on a particular choice of Legendre function that captures different learning rates proportional to c_i^{-1} for different experts, as we saw earlier in Algorithm 5. We start by defining the *weighted negative entropy* function.

Definition 5.3.1. Given expert-ranges $\{c_i\}_{i \in A}$, the *weighted negative entropy* is defined by

$$F(x) = \sum_{i \in A} c_i \cdot x_i \ln(x_i) \tag{5.13}$$

Lemma 5.3.1. *$F(x) = \sum_{i\in A} c_i \cdot x_i \ln(x_i)$ is a non-negative Legendre function over $\mathbb{R}^A_+$. Moreover, $\nabla F(x)_i = c_i(1 + \ln(x_i))$ and $D_F(x, y) = \sum_{i\in A} c_i \cdot (x_i \ln(\frac{x_i}{y_i}) - x_i + y_i)$.*

We now have the following lemma that shows Algorithm 5 is indeed an OMD algorithm.

Lemma 5.3.2. *The MSMW algorithm, i.e. Algorithm 5, is equivalent to an OMD algorithm associated with the weighted negative entropy $F(x) = \sum_{i\in A} c_i \cdot x_i \ln(x_i)$ as its Legendre function.*

Proof. Look at the gradient update step of OMD, as in Equation (2.9), with Legendre transform $F(x) = \sum_{i\in A} c_i \cdot x_i \ln(x_i)$. By using Corollary 5.3.1 we have

$$\nabla F(\mathbf{w}(t+1)) = \nabla F(\mathbf{p}(t)) + \eta \cdot \mathbf{g}(t) \Rightarrow c_i(1 + ln(w_i(t+1))) = c_i(1 + ln(p_i(t))) + \eta \cdot g_i(t)\ ,$$

and therefore, $w_i(t+1) = p_i(t) \cdot \exp(\eta \cdot \frac{g_i(t)}{c_i})$. Moreover, for the Bregman projection step we have

$$\begin{aligned}
\mathbf{p}(t+1) &= \operatorname*{argmin}_{\mathbf{p}\in\Delta_A} \left(D_F(\mathbf{p}, \mathbf{w}(t+1))\right) \\
&= \operatorname*{argmin}_{\mathbf{p}\in\Delta_A} \left(\sum_{i\in A} c_i \cdot (p_i \ln(\frac{p_i}{w_i(t+1)}) - p_i + w_i(t+1)) \right)
\end{aligned} \tag{5.14}$$

This optimization problem is indeed a convex minimization over a convex set. To find a closed form solution, we look at the Lagrangian dual function $\mathcal{L}(\mathbf{p}, \lambda) \triangleq \sum_{i\in A} c_i \cdot (p_i \ln(\frac{p_i}{w_i(t+1)}) - p_i + w_i(t+1)) + \lambda(\sum_{i\in A} p_i - 1)$ and the Karush-Kuhn-Tucker (KKT) conditions $\nabla\mathcal{L}(\mathbf{p}^*, \lambda^*) = \mathbf{0}$. We have

$$c_i \cdot \ln(\frac{p_i^*}{w_i(t+1)}) + \lambda^* = 0 \Rightarrow p_i^* = w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) \tag{5.15}$$

As $\sum_{i\in A} p_i^* = 1$, λ^* should be unique number s.t. $\sum_{i\in A} w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) = 1$, and then $p_i(t+1) = w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i})$. So, Algorithm 5 is equivalent to OMD with weighted negative entropy as its Legendre function. □

We now have the following lemma, whose proof is provided in the Appendix C.1.4. Also, this lemma can derived directly by combining Lemma 5.3.2, Lemma 5.3.1 and Lemma 2.4.1.

Lemma 5.3.3. *For any initial distribution $\boldsymbol{\mu}$ over A, and any learning rate parameter $0 < \eta \leq 1$, and any benchmark distribution $\mathbf{q}$ over A, the MSMW algorithm satisfies that:*

$$\sum_{i \in A} q_i \cdot G_i - \mathbf{E}\left[G_{\text{ALG}}\right] \leq \eta \sum_{t \in [T]} \sum_{i \in A} p_i(t) \frac{(g_i(t))^2}{c_i} + \frac{1}{\eta} \cdot \sum_{i \in A} c_i \Big(q_i \ln\left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i \Big).$$

Regret bound for non-negative rewards - proof of Theorem 5.2.1

Proof of Theorem 5.2.1. Suppose $i_{\min}$ is an action with the minimum c_i. Let $\boldsymbol{\mu} = (1-\eta) \cdot \mathbf{1}_{i_{\min}} + \eta \cdot \boldsymbol{\pi}$, and let $\mathbf{q} = (1-\eta) \cdot \mathbf{1}_i + \eta \cdot \boldsymbol{\pi}$ in Lemma 5.3.3. If $i \neq i_{\min}$, we get that (note that $\mu_j = q_j$ for any $j \neq i, i_{\min}$):

$$(1-\eta) \cdot G_i + \eta \cdot \sum_{j \in A} \pi_j \cdot G_j - \mathbf{E}\left[G_{\text{ALG}}\right] \leq \eta \cdot \mathbf{E}\left[G_{\text{ALG}}\right] + \frac{1}{\eta} \cdot c_i \cdot \left(q_i \ln\left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i \right)$$
$$+ \frac{1}{\eta} \cdot c_{i_{\min}} \cdot \left(q_{i_{\min}} \ln\left(\frac{q_{i_{\min}}}{\mu_{i_{\min}}}\right) - q_{i_{\min}} + \mu_{i_{\min}} \right)$$

By $1 \geq q_i > \mu_i \geq \eta\pi_i$, the second term on the RHS is upper bounded as:

$$\frac{1}{\eta} \cdot c_i \cdot \left(q_i \ln\left(\frac{q_i}{\mu_i}\right) - q_i + \mu_i \right) \leq \frac{1}{\eta} \cdot c_i \cdot \ln\left(\frac{1}{\eta\pi_i}\right)$$

Similarly, by $1 \geq \mu_{i_{\min}} > q_{i_{\min}} \geq 0$, the third term on the RHS is upper bounded as

$$\frac{1}{\eta} \cdot c_{i_{\min}} \cdot \left(q_{i_{\min}} \ln\left(\frac{q_{i_{\min}}}{\mu_{i_{\min}}}\right) - q_{i_{\min}} + \mu_{i_{\min}} \right) \leq \frac{1}{\eta} \cdot c_{i_{\min}} \leq \frac{1}{\eta} \cdot c_i$$

Finally, note that $G_j \geq 0$ for all $j \in A$ in reward-only instances. So the LHS is lower bounded by

$$(1-\eta) \cdot G_i - \mathbf{E}\left[G_{\text{ALG}}\right] = (1-\eta) \cdot \text{REGRET}_i - \eta \cdot \mathbf{E}\left[G_{\text{ALG}}\right].$$

Putting together we get that

$$\mathbf{E}\left[\mathrm{REGRET}_i\right] \leq \frac{2\eta}{1-\eta} \cdot \mathbf{E}\left[G_{\mathrm{ALG}}\right] + O\left(\frac{1}{\eta} \ln\left(\frac{1}{\eta\pi_i}\right) \cdot c_i\right) \leq 3\eta \cdot \mathbf{E}\left[G_{\mathrm{ALG}}\right] + O\left(\frac{1}{\eta} \ln\left(\frac{1}{\eta\pi_i}\right) \cdot c_i\right).$$

The theorem then follows by choosing $\eta = \frac{\epsilon}{3}$ and rearranging terms. □

5.3.2 Multi-Scale Online Learning with Bandit Feedback

In this section, we look at the bandit feedback version of multi-scale online learning. Inspired by the online stochastic mirror descent algorithm, we introduce *Bandit-MSMW* algorithm. Our algorithm follows the standard bandit route of using unbiased estimators for the rewards in a full information strategy (in this case MSMW). We also mix the MSMW distribution with an extra uniform exploration, and use a tailored initial distribution for our multi-scale learning setting.

Here is a map of this section. We propose our bandit algorithm and prove its general regret guarantee for non-negative rewards. Then we show how to get a multi-scale style regret guarantee for the best arm c_{i^*}, and a weaker guarantee for all arms $\{c_i\}_{iA}$.

Bandit-MSMW algorithm

We present our Bandit algorithm (Algorithm 6) when the set of actions A is finite (with $|A| = k$). Let η be the learning rate and γ be the exploration probability. We show the following regret bound.

Lemma 5.3.4. *For any exploration probability $0 < \gamma \leq \frac{1}{2}$ and any learning rate parameter $0 < \eta \leq \frac{\gamma}{k}$, the Bandit-MSMW algorithm achieves the following regret bound when*

Algorithm 6: Bandit-MSMW

1: **input** exploration parameter $\gamma > 0$, learning rate $\eta > 0$.

2: **initialize** $\mathbf{p}(1) = (1-\gamma)\mathbf{1}_{i_{\min}} + \frac{\gamma}{k}\mathbf{1}$, where $i_{\min}$ is the arm with min range $c_{i_{\min}}$.

3: **for** $t = 1, \ldots, T$ **do**

4: Let $\tilde{\mathbf{p}}(t) = (1-\gamma)\mathbf{p}(t) + \frac{\gamma}{k}\mathbf{1}$.

5: Randomly pick an expert i_t drawn from $\tilde{\mathbf{p}}(t)$, and observe $g_{i_t}(t)$.

6: Let $\tilde{\mathbf{g}}(t)$ be such that

$$\tilde{g}_i(t) = \begin{cases} \frac{g_i(t)}{\tilde{p}_i(t)} & \text{if } i = i_t; \\ \\ 0 & \text{otherwise.} \end{cases}$$

7: $\forall i \in A: \quad w_i(t+1) \leftarrow p_i(t) \cdot \exp(\frac{\eta}{c_i} \cdot \tilde{g}_i(t))$.

8: Find λ^* (e.g., binary search) s.t. $\sum_{i \in A} w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i}) = 1$.

9: $\forall i \in A: \quad p_i(t+1) \leftarrow w_i(t+1) \cdot \exp(-\frac{\lambda^*}{c_i})$.

10: **end for**

the gains are non-negative :

$$\forall i \in A : \mathbf{E}_{\text{REGRET}_i}[\leq]\, O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right) \cdot c_i + \eta \sum_{j \in A} G_j + \gamma \cdot G_i\right)$$

Proof. We further define:

$$\begin{aligned} \widetilde{G}_{\text{ALG}} &\triangleq \sum_{t \in [T]} g_{i_t}(t) = \sum_{t \in [T]} \tilde{\mathbf{p}}(t) \cdot \tilde{\mathbf{g}}(t) , \\ \widetilde{G}_j &\triangleq \sum_{t \in [T]} \tilde{g}_j(t) . \end{aligned}$$

In expectation over the randomness of the algorithm, we have:

1. $\mathbf{E}[G_{\text{ALG}}] = \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right]$; and

2. $G_j = \mathbf{E}\left[\widetilde{G}_j\right]$ for any $j \in A$.

Hence, to upper bound $\mathbf{E}\left[\mathrm{REGRET}_i\right] = G_i - \mathbf{E}\left[G_{\mathrm{ALG}}\right]$, it suffices to upper bound $\mathbf{E}\left[\widetilde{G}_i - \widetilde{G}_{\mathrm{ALG}}\right]$.

By the definition of the probability that the algorithm picks each arm, i.e., $\tilde{\mathbf{p}}(t)$, we have:

$$\mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] \geq (1-\gamma) \textstyle\sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t) \ .$$

Hence, we have that for any initial distribution $\mathbf{q}$ over A:

$$\begin{aligned}\sum_{j \in A} q_j \cdot \mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] &\leq \ \mathbf{E}\left[\textstyle\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + \tfrac{\gamma}{1-\gamma} \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] \\ &\leq \ \mathbf{E}\left[\textstyle\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t)\right] + 2\gamma \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] \ . \end{aligned} \tag{5.16}$$

Next, we upper bound the 1st term on the RHS. Note that $\mathbf{p}(t)$'s are the probabilities of choosing experts by MSMW when the experts have rewards $\tilde{\mathbf{g}}(t)$'s. By Lemma 5.3.3, we have that for any benchmark distribution $\mathbf{q}$ over S, the Bandit-MSMW algorithm satisfies that:

$$\sum_{j \in A} q_j \cdot \widetilde{G}_j - \sum_{t \in [T]} \mathbf{p}(t) \cdot \tilde{\mathbf{g}}(t) \leq \eta \sum_{t \in [T]} \sum_{j \in A} \frac{p_j(t)}{c_j} \cdot \left(\tilde{g}_j(t)\right)^2 + \frac{1}{\eta} \sum_{j \in A} c_j \left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) . \tag{5.17}$$

For any $t \in [T]$ and any $j \in A$, by the definition of $\tilde{g}_j(t)$, it equals $\frac{g_j(t)}{\tilde{p}_j(t)}$ with probability $\tilde{p}_j(t)$, and equals 0 otherwise. Thus, if we fix the random coin flips in the first $t-1$ rounds and, thus, fix $\tilde{\mathbf{p}}(t)$, and take expectation over the randomness in round t, we have that:

$$\mathbf{E}\left[\frac{p_j(t)}{c_j} \cdot \left(\tilde{g}_j(t)\right)^2\right] = \frac{p_j(t)}{c_j} \cdot \tilde{p}_j(t) \cdot \left(\frac{g_j(t)}{\tilde{p}_j(t)}\right)^2 = \frac{p_j(t)}{\tilde{p}_j(t)} \frac{(g_j(t))^2}{c_j} \ .$$

Further note that since $\tilde{p}_j(t) \geq (1-\gamma)p_j(t)$, and $g_j(t) \leq c_j$, the above is upper bounded by $\frac{1}{1-\gamma} g_j(t) \leq 2g_j(t)$. Putting together with (5.17), we have that for any $0 < \eta \leq \frac{\gamma}{n}$:

$$\begin{aligned}
\mathbf{E}\left[\sum_{j\in A} q_j \cdot \widetilde{G}_j - \sum_{t\in[T]} \mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t)\right] &\leq \eta \sum_{t\in[T]}\sum_{j\in A} 2g_j(t) + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) \\
&= 2\eta \sum_{j\in A} G_j + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right)
\end{aligned}$$

Combining with (5.16), we have:

$$\sum_{j\in A} q_j \cdot \mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] \leq 2\eta\sum_{j\in A} G_j + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) + 2\gamma\mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right]$$

Let $\mathbf{q} = (1-\gamma)\mathbf{1}_i + \frac{\gamma}{k}\mathbf{1}$. Recall that $\mathbf{p}(1) = (1-\gamma)\mathbf{1}_{i_{\min}} + \frac{\gamma}{k}\mathbf{1}$ (recall $i_{\min}$ is the arm with minimum range $c_{i_{\min}}$). Similar to the discussion for the expert problem in Section 5.3.1, the 2nd term on the RHS is upper bounded by $O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i\right)$. Hence, we have:

$$\sum_{j\in A} q_j \cdot \mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] \leq 2\eta\sum_{j\in A} G_j + O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i\right) + 2\gamma\mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right]. \tag{5.18}$$

Further, the LHS is lower bounded as:

$$(1-\gamma)\mathbf{E}\left[\widetilde{G}_i\right] + \frac{\gamma}{k}\sum_{j\in A}\mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] \geq (1-\gamma)\mathbf{E}\left[\widetilde{G}_i\right] - \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right].$$

The lemma then follows by putting it back to (5.18) and rearranging terms.

□

Regret for non-negative rewards - proof of Theorem 5.2.3

Proof of Theorem 5.2.3. Letting $\gamma = \epsilon$ and $\eta = \frac{\gamma}{k} = \frac{\epsilon}{k}$ in Lemma 5.3.4, we get that the expected regret w.r.t. an action $i \in A$ is bounded by:

$$O\left(\epsilon \cdot G_i + \tfrac{\epsilon}{k} \sum_{j\in A} G_j + c_i \cdot \tfrac{k}{\epsilon} \ln\left(\tfrac{k}{\epsilon}\right)\right) .$$

When $i = i^*$ (best arm), regret is bounded by $O\left(\epsilon \cdot G_{i^*} + c_i^* \cdot \frac{k}{\epsilon} \ln\left(\frac{k}{\epsilon}\right)\right)$, as desired.

For the regret w.r.t. an arbitrary action, note that $\mathbf{E}\left[G_{\mathrm{ALG}}\right] \geq \frac{\gamma}{k} \sum_{j\in A} G_j$. Thus, the regret bound w.r.t. an action $i \in A$ in Lemma 5.3.4 is further upper bounded by:

$$O\left(\tfrac{1}{\eta} \log\left(\tfrac{k}{\gamma}\right) \cdot c_i + \left(\tfrac{\eta k}{\gamma} + \gamma\right) \cdot \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right]\right)$$

The theorem then follows by letting $\gamma = \epsilon$ and $\eta = \frac{\gamma^2}{k} = \frac{\epsilon^2}{k}$. □

5.3.3 Auctions and Pricing

Auctions and pricing as multi-scale learning problems

Online single buyer auction and posted pricing Recall that in each round, the algorithm chooses an action, i.e., a price, $p_t \in [1, h]$; the adversary picks a value $v(t) \in [1, h]$; and the algorithm collects reward $g_{p_t}(t) = p_t \cdot \mathbf{1}(v(t) \geq p_t)$. In order to obtain a $1-\epsilon$ approximation of the optimal revenue, it suffices to consider prices of the form $(1+\epsilon)^j$ for $0 \leq j \leq \lfloor \log_{1+\epsilon} h \rfloor = O(\frac{\log h}{\epsilon})$. As a result, we reduce the online single buyer auction problem and the online posted pricing problem to a multi-scale online learning problem with full information and bandit feedback respectively with $k = O(\frac{\log h}{\epsilon})$ actions whose ranges form a geometric sequence $(1+\epsilon)^j$, $0 \le j < k$.

Online multi buyer auction In multi buyer auctions, we consider the set of all discretized Myerson-type auctions as the action space. We start by defining Myerson-type auctions:

Definition 5.3.2 (Myerson-type auctions)**.** A *Myerson-type auction* is defined by n non-decreasing virtual value mappings $\phi_1, \ldots, \phi_n : [1, h] \mapsto [-\infty, h]$. Given a value profile $v_1, \ldots, v_n$, the item is given to the bidder j with the largest non-negative virtual value $\phi_j(v_j)$. Then, bidder j pays the minimum value that would keep him as the the winner.

Myerson (1981) shows that when the bidders' values are drawn from independent (but not necessarily identical) distributions, the revenue-optimal auction is a Myerson-type auction. Devanur et al. (2016, Lemma 5) observe that to obtain a $1-\epsilon$ approximation, it suffices to consider the set of discretized Myerson-type auctions that treat each bidder's value as if it is equal to the closest power of $1+\epsilon$ from below. As a result, it suffices to consider the set of discretized Myerson-type auctions, each of which is defined by the virtual values of $(1+\epsilon)^j$'s, i.e., by $O(n \log h/\epsilon)$ real numbers $\phi_\ell((1+\epsilon)^j)$, for $\ell \in [n]$, and $0 \le j \le \lfloor \log_{1+\epsilon} h \rfloor$. Devanur et al. (2016); Gonczarowski and Nisan (2017) further note that a discretized Myerson-type auction is in fact completely characterized by the total ordering of $\phi_\ell((1+\epsilon)^j)$'s; their actual values do not matter. Indeed, both the allocation rule and the payment rule are determined by the ordering of virtual values. As a result, our action space is a finite set with at most $O((n \log h/\epsilon)!)$ actions. The range of an action, i.e., a discretized Myerson-type auction, is the largest price ever charged by the auction, i.e., the largest value v of the form $(1+\epsilon)^j$ such that there exists $\ell \in [n]$, $\phi_\ell(v) > \phi_\ell((1+\epsilon)^{-1}v)$.

Proof of Theorem 5.2.5

Proof. Online single buyer auction. Recall the above formulation of the problem as an online learning problem with full information. The case when h is known then follows by Theorem 5.2.1, letting $\boldsymbol{\pi}$ be the uniform distribution over the $k = O(\log h/\epsilon)$ actions, i.e., discretized prices.

When h is not known up front, we consider a countably infinite action space comprising of all prices of the form $(1+\epsilon)^j$, for $j \geq 0$. Then, let the prior distribution $\boldsymbol{\pi}$ be such that for any price $p = (1+\epsilon)^j$, $\pi_p = \epsilon(1+\epsilon)^{-j-1} = \frac{\epsilon}{1+\epsilon} \cdot \frac{1}{p}$. The approximation guarantee then follows by Theorem 5.2.1.

Online posted pricing. Recall the above formulation of the problem as an online learning problem with bandit feedback. This part then follows by Theorem 5.2.3 with $k = O(\log h/\epsilon)$ actions.

Online multi buyer auction. Recall the above formulation of the problem as an online learning problem with full information. The case when h is known then follows by Theorem 5.2.1, where we let $\boldsymbol{\pi}$ be the uniform distribution over the $k = O((n \log h/\epsilon)!)$ actions, i.e., Myerson-type auctions.

When h is not known up front, we consider a countably infinite action space A as follows. For any $p = (1+\epsilon)^j$, $j \geq 0$, let the $k_p = O((n \log p/\epsilon)!)$ Myerson-type auctions for values in $[1, p]$ be in A; we assume these auctions treat any values greater than p as if they were p. Further, we choose the prior distribution $\boldsymbol{\pi}$ such that the probability mass of each auction for range $[1, p]$ is equal to $\frac{\epsilon}{1+\epsilon} \cdot \frac{1}{p} \cdot \frac{1}{k_p}$. The approximation guarantee then follows by Theorem 5.2.1. $\square$

Proof of Theorem 5.2.6

Proof. Online single buyer auction. When h is known, by Theorem 5.2.1, letting $\boldsymbol{\pi}$ be the uniform distribution over the $k = O(\log h/\epsilon)$ actions, i.e., discretized prices, we have that for any price p (recall that $c_p = p$):

$$\texttt{ALG} \geq (1-\epsilon) \cdot G_p - O\left(\tfrac{\log(\log h/\epsilon)}{\epsilon} \cdot p\right) .$$

For the δ-guarded optimal price p^* (i.e., subject to selling in at least δT rounds), we have $G_{p^*} \geq \delta T \cdot p^*$. Therefore, when $T \geq O\left(\log(\log h/\epsilon)/\epsilon^2 \delta\right)$, the additive term of the above approximation guarantee is at most $\epsilon \cdot G_{p^*}$. So the theorem holds.

The treatment for the case when h is not known up front is essentially the same as in Theorem 5.2.5. We consider a countably infinite action space comprised of all prices of the form $(1+\epsilon)^j$, for $j \geq 0$. Then, let the prior distribution $\boldsymbol{\pi}$ be such that for any price $p = (1+\epsilon)^j$, $\pi_p = \epsilon(1+\epsilon)^{-j-1} = \frac{\epsilon}{1+\epsilon} \cdot \frac{1}{p}$.

Online posted pricing. Recall the above formulation of the problem as an online learning problem with bandit feedback. By Theorem 5.2.3 with $k = O(\log h/\epsilon)$ actions, we have that for any price p:

$$\texttt{ALG} \geq (1-\epsilon) \cdot G_p - O\left(\tfrac{\log h \log(\log h/\epsilon)}{\epsilon^3} \cdot p\right) .$$

Again, for the δ-guarded optimal price p^* (i.e., subject to selling in at least δT rounds), we have $G_{p^*} \geq \delta T \cdot p^*$. Therefore, when $T \geq O\left(\log h \log\left(\log h/\epsilon\right)/\epsilon^4 \delta\right)$, the additive term of the above approximation guarantee is at most $\epsilon \cdot G_{p^*}$. So the theorem holds.

Online multi buyer auction. Suppose i^* is the δ-guarded best Myerson-type auction. Recall that $\bar{V}$ is the largest value such that there are at least δT distinct $v(t)$'s with $\max_{\ell \in [n]} v_\ell(t) \geq \bar{V}$. So we may assume without loss of generality that i^* does not distinguish values greater than $\bar{V}$. Hence:

$$c_{i^*} \leq \bar{V} \ . \tag{5.19}$$

Further, note that running a second-price auction with anonymous reserve $\bar{V}$ is a Myerson-type auction (e.g., mapping values less than $\bar{V}$ to virtual value $-\infty$ and values greater than or equal to $\bar{V}$ to virtual value $\bar{V}$), and it gets revenue at least $\delta T \cdot \bar{V}$. So we have that:

$$G_{p^*} \geq \delta T \cdot \bar{V} \ . \tag{5.20}$$

Finally, the above implies that to obtain a $1 - \epsilon$ approximation, it suffices to consider prices that are at least $\epsilon\delta\bar{V}$. Hence, it suffices to consider Myerson-type auctions that, for a given $\bar{V}$, do not distinguish among values greater than $\bar{V}$, and do not distinguish among values smaller than $\epsilon\delta\bar{V}$. There are $O(\log h/\epsilon)$ different values of $\bar{V}$. Further, given $\bar{V}$, there are only $O(\log(1/\epsilon\delta)/\epsilon)$ distinct values to be considered and, thus, there are at most $O((n\log(1/\epsilon\delta)/\epsilon)!)$ distinct Myerson-type auctions of this kind. Hence, the total number of distinct Myerson-type actions that we need to consider is at most:

$$k = O\left(\frac{\log h}{\epsilon} \cdot \left(\frac{n\log(1/\epsilon\delta)}{\epsilon}\right)!\right) \ .$$

When h is known, letting $\boldsymbol{\pi}$ be the uniform distribution over the k actions in Theorem 5.2.1, we have that (recall Eqn. (5.19)):

$$\texttt{ALG}_{\geq}(1-\epsilon)\cdot G_{i^*} - O\left(\frac{n\log\left(1/\epsilon\delta\right)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^2} + \frac{\log\left(\log h/\epsilon\right)}{\epsilon}\right)\cdot \bar{V} \ .$$

When $T \geq O\left(\frac{n\log(1/\epsilon\delta)\log(n\log(1/\epsilon\delta)/\epsilon)}{\epsilon^3\delta} + \frac{\log(\log h/\epsilon)}{\epsilon^2\delta}\right)$, the additive term of the above approximation guarantee is at most $\epsilon \cdot G_{i^*}$ due to Eqn. (5.20). So the theorem holds.

Again, the treatment for the case when h is not known up front is similar to that in Theorem 5.2.5. When h is not known up front, we consider a countably infinite action space A as follows. For any $\bar{V} = (1+\epsilon)^j$, $j \geq 0$, let the $k' = O((n\log(1/\epsilon\delta)/\epsilon)!)$ Myerson-type auctions that do not distinguish among values greater than $\bar{V}$, and do not distinguish among values smaller than $\epsilon\delta\bar{V}$ be in A. Further, we choose the prior distribution $\boldsymbol{\pi}$ such that the probability mass of each Myerson-type auction for a given $\bar{V}$ is equal to $\frac{\epsilon}{1+\epsilon} \cdot \frac{1}{\bar{V}} \cdot \frac{1}{k'}$. The approximation guarantee then follows by Theorem 5.2.1 and essentially the same argument as the known h case. $\square$

Remark Devanur et al. (2016) show that when the values are drawn from independent regular distributions, the ϵ-guarded optimal benchmark is a $1-\epsilon$ approximation of the unguarded optimal benchmark. So our convergence rate for the online multi buyer auction problem in Theorem 5.2.1 implies a $\tilde{O}(n\epsilon^{-4})$ sample complexity modulo a mild $\log\log h$ dependency on the range, almost matching the best known sample complexity upper bound for regular distributions.

5.3.4 Multi-scale Online Learning with Symmetric Range

In this section, we consider multi-scale online learning when the rewards are in a symmetric range, i.e. for all $i \in A$ and $t \in [T]$, $g_i(t) \in [-c_i, c_i]$. We look at both full information and bandit settings, and prove action-specific regret upper bounds. We defer the regret lower bound proofs to the appendix, Sections C.1.2 and C.1.3.

Multi-scale expert problem with symmetric range

Recall the proof of Lemma 5.3.3. The proof only requires $g_i(t) \in [-c_i, c_i]$ for all $i \in A, t \in [T]$. Choosing q to be $\mathbf{1}_i$, a vector with a 1-entry in i^{th} coordinate and 0-entries elsewhere for an action $i \in A$, and noting that

$$\sum_{t\in[T]} \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i} \leq \sum_{t\in[T]} \sum_{i\in A} p_i(t) \cdot \left|g_i(t)\right|,$$

we get the following regret bound as a corollary of Lemma 5.3.3.

Corollary 5.3.5. *For any initial distribution μ over A, and any learning rate parameter $0 < \eta \leq 1$, the MSMW algorithm achieves the following regret bound:*

$$\forall i \in A: \quad \mathbf{E}\left[\text{REGRET}_i\right] \leq \eta \cdot \mathbf{E}\left[\sum_{t\in[T]} \left|g_i(t)\right|\right] + \frac{1}{\eta} c_i \cdot \log\left(\frac{1}{\mu_i}\right) + \frac{1}{\eta} \sum_{j\in A} \mu_j c_j \tag{5.21}$$

Now, we can prove the multi-scale regret upper bound in Theorem 5.2.7 using Corollary 5.3.5.

Proof of Theorem 5.2.7. The proof follows by choosing an appropriate initial distribution μ in Corollary 5.3.5. By Corollary 5.3.5, we have:

$$\mathbf{E}\left[\text{REGRET}_i\right] \leq \eta \cdot \mathbf{E}\left[\sum_{t\in[T]} \left|g_i(t)\right|\right] + \frac{1}{\eta} c_i \cdot \log\left(\frac{1}{\mu_i}\right) + \frac{1}{\eta} \sum_{j\in A} \mu_j c_j$$

Let $i_{\min}$ be an action with the minimum range $c_{i_{\min}} = c_{\min}$. Consider an initial distribution $\mu_j = \pi_j \frac{c_{\min}}{c_j}$ for all $j \neq i_{\min}$, and $\mu_{i_{\min}} = 1 - \sum_{j\neq i_{\min}} \mu_j$, i.e., putting all remaining probability mass on action $i_{\min}$. Then, the third term on the RHS is upper bounded by:

$$\sum_{j\in A}\mu_j c_j = \sum_{j\neq i_{\min}}\mu_j c_j + \mu_{i_{\min}} c_{i_{\min}} = \sum_{j\neq i_{\min}}\pi_j c_{\min} + \mu_{i_{\min}} c_{\min} \leq 2c_{\min} \leq 2c_i \ .$$

For $i \neq i_{\min}$, by the definition of μ_i, we have:

$$\begin{aligned}\mathbf{E}\left[\mathrm{REGRET}_i\right] &\leq \eta\cdot\mathbf{E}\left[\sum_{t\in[T]}\left|g_i(t)\right|\right] + \tfrac{1}{\eta}c_i\cdot\log(\tfrac{1}{\pi_i}\cdot\tfrac{c_i}{c_{\min}}) + \tfrac{1}{\eta}\cdot 2c_{\min} \\ &= \eta\cdot\mathbf{E}\left[\sum_{t\in[T]}\left|g_i(t)\right|\right] + O\left(\tfrac{1}{\eta}\log\left(\tfrac{1}{\pi_i}\cdot\tfrac{c_i}{c_{\min}}\right)\cdot c_i\right) \ .\end{aligned}$$

So the theorem follows by choosing $\eta = \epsilon$. For $i = i_{\min}$, note that $\mu_j \leq \pi_j$ for all $j \neq i_{\min}$ and, thus, $\mu_{i_{\min}} = 1 - \sum_{j\neq i_{\min}}\mu_j \geq 1 - \sum_{j\neq i_{\min}}\pi_j = \pi_{i_{\min}} = \pi_{i_{\min}}\frac{c_{\min}}{c_{i_{\min}}}$. The theorem then holds following the same calculation as in the $j \neq i_{\min}$ case. □

5.3.5 Multi-scale Bandit Problem with Symmetric Range

We start by presenting the following regret bound, whose proof is an alteration of that for Lemma 5.3.4 under symmetric range (and is deferred to the appendix, Section C.1.6). Next, we prove Theorem 5.2.10.

Lemma 5.3.6. *For any exploration rate $0 < \gamma \leq \min\{\frac{1}{2}, \frac{c_{\min}}{c_{\max}}\}$ and any learning rate $0 < \eta \leq \frac{\gamma}{k}$, the Bandit-MSMW algorithm (Algorithm 6) achieves the following regret bound:*

$$\forall i \in A : \mathbf{E}\left[\mathrm{REGRET}_i\right] \leq O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i + \gamma T\cdot c_{\max}\right)$$

Proof of Theorem 5.2.10. Let $\gamma = \epsilon\frac{c_{\min}}{c_{\max}}$ and $\eta = \frac{\gamma}{k}$ in Lemma 5.3.6. Theorem follows noting that $\gamma c_{\max} = \epsilon c_{\min} \leq \epsilon c_i$. □

5.4 Conclusion

We conclude the chapter by summarizing our contributions. We considered revenue maximization in online auctions and pricing. A seller sells an identical item in each period to a new buyer, or a new set of buyers. For the online posted pricing problem, we showed regret bounds that scale with the best fixed price, rather than the range of the values. We also showed regret bounds that are almost scale free, and match the offline sample complexity, when comparing to a benchmark that requires a lower bound on the market share. These results were obtained by generalizing the classical learning from experts and multi-armed bandit problems to their multi-scale versions. In this version, the reward of each action is in a different range, and the regret w.r.t. a given action scales with its own range, rather than the maximum range.

APPENDIX A

OMITTED PROOFS AND DISCUSSIONS FROM CHAPTER 3

A.1 Implicit Payment Computation

In this section we describe one standard reduction for computing implicit payments in our general setting, given access to a BIC allocation algorithm $\tilde{\mathcal{A}}$: a multi-parameter counterpart of the single-parameter payment computation procedure used for example by Archer et al. (2004); Hartline and Lucier (2010), which makes $n+1$ calls to $\tilde{\mathcal{A}}$, thus incurring a factor $n+1$ overhead in running time. A different implicit payment computation procedure, described in Babaioff et al. (2013, 2015b), avoids this overhead by calling $\tilde{\mathcal{A}}$ only once in expectation, but incurs a $1-\epsilon$ loss in expected welfare and potentially makes payments of magnitude $\Theta(1/\epsilon)$ *from* the mechanism *to* the agents.

The implicit payment computation procedure assumes that the agents' type spaces $(\mathcal{T}^k)_{k\in[n]}$ are *star-convex at 0*, meaning that for any agent k, any type $t^k \in \mathcal{T}^k$, and any scalar $\lambda \in [0,1]$, there is another type $\lambda t^k \in \mathcal{T}^k$ with the property that $v(\lambda t^k, o) = \lambda v(t^k, o)$ for every $o \in \mathcal{O}$. (The assumption is without loss of generality, as argued in the next paragraph.) The implicit payment computation procedure, applied to type profile $\mathbf{t}$, samples $\lambda \in [0,1]$ uniformly at random and computes outcomes $o^0 \triangleq \tilde{\mathcal{A}}(\mathbf{t})$ as well as $o^k \triangleq \tilde{\mathcal{A}}(\lambda t^k, \mathbf{t}^{-k})$ for all $k \in [n]$. The payment charged to agent k is $v(t^k, o^0) - v(t^k, o^k)$. Note that, in expectation, agent k pays

$$p^k(\mathbf{t}) = v(t^k, \tilde{\mathcal{A}}(\mathbf{t})) - \int_0^1 v(t^k, \tilde{\mathcal{A}}(\lambda t^k, \mathbf{t}^{-k}))\, d\lambda,$$

in accordance with the payment identity for multi-parameter BIC mechanisms when type spaces are star-convex at 0; see Babaioff et al. (2013) for a discussion of this payment identity.

Finally, let us justify the assumption that $\mathcal{T}^k$ is star-convex for all k. This assumption is without loss of generality for the allocation algorithms $\tilde{\mathcal{A}}$ that arise from the RSM reduction, because we can enlarge the type space $\mathcal{T}^k$ if necessary by adjoining types of the form λt^k with $t^k \in \mathcal{T}^k$ and $0 \leq \lambda < 1$. Although the output of the original allocation algorithm $\mathcal{A}$ may be undefined when its input type profile includes one of these artifically-adjoined types, the RSM reduction never inputs such a type into $\mathcal{A}$. It only calls $\mathcal{A}$ on profiles of surrogate types sampled from the type-profile distribution F, whose support excludes the artificially-adjoined types. Thus, even when the input to $\tilde{\mathcal{A}}$ includes an artifically-adjoined type λt^k, it occurs as one of the replicas in the reduction. The behavior of algorithm $\tilde{\mathcal{A}}$ remains well-defined in this case, because replicas are only used as inputs to the valuation function $v(r_i, o_j)$, whose output is well-defined even when $r_i = \lambda t^k$ for $\lambda < 1$.

A.2 Surrogate Selection and BIC Reduction

Lemma A.2.1. *If matching algorithm $M(\mathbf{r}, \mathbf{s})$ produces a perfect k-to-1 matching for the instance in Definition 3.3.5, then its corresponding surrogate selection rule, denoted by Γ^M, is stationary*

Proof. Each surrogate s_j is an i.i.d. sample from F. Moreover, by the principle of deferred decisions the index i^* (the real agent's index in the replica type profile) is a uniform random index in $[mk]$, even after fixing the matching. Since this choice of replica is uniform in $[mk]$ and M is a perfect k-to-1 matching, the selection of surrogate outcome is uniform in $[m]$, and therefore the selection of surrogate type associated with this outcome is also uniform in $[m]$. As a result, the output distribution of the selected surrogate type is F. □

Lemma A.2.2. *If $M(\mathbf{r},\mathbf{s})$ is a feasible replica-surrogate k-to-1 matching and is a truthful allocation rule (in expectation over allocation's random coins) for all replicas (i.e. assuming each replica is a rational agent, no replica has any incentive to misreport), then the composition of Γ^M and interim allocation algorithm $\mathcal{A}(.)$ forms a BIC allocation algorithm for the original mechanism design problem.*

Proof. Each replica-agent $i \in [mk]$ (*including* the real agent i^*) bests off by reporting her true replica type under some proper payments. Now, consider an agent in the original mechanism design problem with true type t. For any given surrogate type profile $\mathbf{s}$, using the Γ^M-reduction the agent receives the same outcome distribution as the one he gets matched to in M in a Bayesian sense, simply because of stationary property of Γ^M (Lemma A.2.1). As allocation M is incentive compatible, this agent doesn't benefit from miss-reporting her true type as long as the value he receives for reporting t' is $v(t, \mathcal{A}(\Gamma^M(t')))$. Therefore conditioning on $\mathbf{s}$ and non-real replicas in $\mathbf{r}$, the final allocation is BIC from the perspective of this agent. The lemma then follows by averaging over the random choice of $\mathbf{s}$ and non-real agent replicas in $\mathbf{r}$. □

A.3 Estimating the Offline Optimal Regularized Matching

To formalize the approximation scheme, first fix the surrogate type profile $\mathbf{s}$. For a given replica profile $\mathbf{r}$ and replica-surrogate edge (i, j), let $v_{i,j}(r_i)$ =

$\mathbf{E}\left[v(r_i, \mathcal{A}(s_j))\right]$ and $\hat{v}_{i,j}(r_i)$ be the empirical mean of N samples of the random variable $v(r_i, \mathcal{A}(s_j)$. Suppose $\mathbf{v}(\mathbf{r})$ and $\hat{\mathbf{v}}(\mathbf{r})$ be the corresponding vectors of expected values and empirical means under replica profile $\mathbf{r}$. Now, draw $\mathbf{r}'$ independently at random from the distribution of $\mathbf{r}$. We now show that $\mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}'))$ is a constant-factor approximation to $\mathrm{OPT}(\mathbf{v}(\mathbf{r}))$ with high probability, and therefore we can use $\mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}'))$ to set γ.

We prove this in two steps. In Lemma A.3.1 we show for a given $\mathbf{r}'$, $\mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}'))$ is a constant-factor approximation to $\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))$ with high probability over the randomness in $\{\mathcal{A}(s_j)\}$. Then, in Lemma A.3.3 we show if $\mathbf{r}$ and $\mathbf{r}'$ are two random independent draws from the replica profile distribution then $\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))$ is a constant-factor approximation to $\mathrm{OPT}(\mathbf{v}(\mathbf{r}))$ with high probability over randomness in $\mathbf{r}$ and $\mathbf{r}'$. These two pieces together prove our claim.

Lemma A.3.1. *If $N \geq \frac{2\log(4m^2 k\cdot\eta^{-1})}{\delta^2(\log m)^2}$, then $1/2 \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r}')) \leq \mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}')) \leq 2\,\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))$ with probability at least $1 - \eta/2$.*

Proof. By using the standard Chernoff-Hoeffding bound together with the union bound, with probability at least $1 - 2m^2 k e^{-\frac{\delta^2(\log m)^2\cdot N}{2}} \geq 1 - \eta/2$ we have

$$\forall (i,j) \in [km] \times [m]: \quad |\hat{v}_{i,j}(r_i') - v_{i,j}(r_i')| \leq 1/2 \cdot \delta \log m$$

Suppose $\mathbf{x}^*$ is the optimal solution of the regularized matching convex program with values $\mathbf{v}(\mathbf{r}')$ and $\mathbf{x}^{**}$ is the optimal solution with values $\hat{\mathbf{v}}(\mathbf{r}')$.

$$\begin{aligned}
\mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}')) &= \sum_i \left(\mathbf{x}_i^{**} \cdot \hat{\mathbf{v}}_i + \delta H(\mathbf{x}_i^{**})\right) \geq \sum_i \left(\mathbf{x}_i^{*} \cdot \hat{\mathbf{v}}_i + \delta H(\mathbf{x}_i^{*})\right) \\
&\geq \sum_i \left(\mathbf{x}_i^{*} \cdot \mathbf{v}_i + \delta H(\mathbf{x}_i^{*})\right) - \frac{\delta k m \log m}{2} \\
&= \mathrm{OPT}(\mathbf{v}(\mathbf{r}')) - \frac{\delta k m \log m}{2} \geq 1/2 \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r}'))
\end{aligned} \tag{A.1}$$

where the last inequality holds as $\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))$ is bounded below by the value of the uniform allocation, i.e. $\mathrm{OPT}(\mathbf{v}(\mathbf{r}')) \geq \delta \cdot mk \log(m)$. Similarly, one can show $\mathrm{OPT}(\mathbf{v}(\mathbf{r}')) \geq 1/2 \cdot \mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}'))$, which completes the proof. □

Before proving the second step, we prove the following lemma, which basically shows that the optimal value of regularized matching $\mathrm{OPT}(\mathbf{v}(\cdot))$ is a 1-Lipschitz multivariate function.

Lemma A.3.2. *For every $i \in [km]$, replica profile $\mathbf{r}$, and replica type r_i' we have:*

$$|\mathrm{OPT}(\mathbf{v}(r_i, r_{-i})) - \mathrm{OPT}(\mathbf{v}(r_i', r_{-i}))| \leq 1$$

Proof. Let variables $\mathbf{x}$ and $\mathbf{x}'$ denote the optimal assignments in $\mathrm{OPT}(\mathbf{v}(r_i, r_{-i}))$ and $\mathrm{OPT}(\mathbf{v}(r_i', r_{-i}))$ respectively. We have

$$\begin{aligned}\mathrm{OPT}(\mathbf{v}(r_i, r_{-i})) &= \sum_l (\mathbf{x}_l \cdot \mathbf{v}_l(r_l) + \delta H(\mathbf{x}_l)) \geq \sum_{l \neq i} (\mathbf{x}_l' \cdot \mathbf{v}_l(r_l) + \delta H(\mathbf{x}_l')) + \mathbf{x}_i' \cdot \mathbf{v}_i(r_i) + \delta H(\mathbf{x}_i') \\ &\geq \sum_{l \neq i} (\mathbf{x}_l' \cdot \mathbf{v}_l(r_l) + \delta H(\mathbf{x}_l')) + \mathbf{x}_i' \cdot \mathbf{v}_i(r_i') + \delta H(\mathbf{x}_i') - 1 = \mathrm{OPT}(\mathbf{v}(r_i', r_{-i})) - 1\end{aligned}$$

where the last inequality holds because $\mathbf{x}_i' \cdot (\mathbf{v}_i(r_i') - \mathbf{v}_i(r_i)) \leq 1$. Similarly, $\mathrm{OPT}(\mathbf{v}(r_i', r_{-i})) \geq \mathrm{OPT}(\mathbf{v}(r_i, r_{-i})) - 1$ by switching the roles of r_i and r_i'. □

Lemma A.3.3. *If $k \geq \frac{32 \log(8\eta^{-1})}{\delta^2 m (\log m)^2}$, then $1/2 \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r})) \leq \mathrm{OPT}(\mathbf{v}(\mathbf{r}')) \leq 3/2 \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r}))$ with probability at least $1 - \eta/2$.*

Proof. We start by defining the following Doob martingale sequence (Motwani and Raghavan, 2010), where (conditional) expectations are taken over the randomness in replica profile $\mathbf{r}$:

$$\begin{aligned} X_0 &= \mathbf{E}[\mathrm{OPT}(\mathbf{v}(\mathbf{r}))] \\ X_n &= \mathbf{E}[\mathrm{OPT}(\mathbf{v}(\mathbf{r})) | r_1, \cdots, r_n], \quad n = 1, 2, \ldots, km \end{aligned}$$

It is easy to check that $\mathbf{E}[X_n | r_1, \ldots, r_{n-1}] = X_{n-1}$, and therefore $\{X_n\}$ forms a martingale sequence with respect to $\{r_n\}$. Moreover, $|X_n - X_{n-1}| \leq 1$ because of Lemma A.3.2. Now, by using Azuma–Hoeffding bound for martingales, we have

$$\Pr\{|X_{km} - X_0| \geq \delta km \log(m)/4\} \leq 2e^{\frac{-km\delta^2(\log m)^2}{32}}$$

and thus with probability at least $1 - 2e^{\frac{-km\delta^2(\log m)^2}{32}}$, $|\mathrm{OPT}(\mathbf{v}(\mathbf{r})) - \mathbf{E}[\mathrm{OPT}(\mathbf{v}(\mathbf{r}))]| \leq \frac{\delta km \log(m)}{4}$. Similarly, with probability at least $1 - 2e^{\frac{-km\delta^2(\log m)^2}{32}}$, we have $|\mathrm{OPT}(\mathbf{v}(\mathbf{r}')) - \mathbf{E}[\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))]| \leq \frac{\delta km \log(m)}{4}$. Note that $\mathrm{OPT}(\mathbf{v}(\mathbf{r}))$ and $\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))$ are identically distributed, and in particular they have the same expectation. Therefore with probability at least $1-4e^{\frac{-km\delta^2(\log m)^2}{32}}$ we have $|\mathrm{OPT}(\mathbf{v}(\mathbf{r}))-\mathrm{OPT}(\mathbf{v}(\mathbf{r}'))| \leq \frac{\delta km \log(m)}{2}$. By using the lower bound of $\delta km \log(m)$ for $\mathrm{OPT}(\mathbf{v}(\mathbf{r}))$ (due to uniform assignment), we conclude that with probability at least $1 - 4e^{\frac{-km\delta^2(\log m)^2}{32}} \geq 1 - \eta/2$ we have the following, as desired:

$$1/2 \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r})) \leq \mathrm{OPT}(\mathbf{v}(\mathbf{r}')) \leq 3/2 \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r})). \qquad \square$$

Corollary A.3.4. *If* $N \geq \frac{2\log(4m^2k\cdot\eta^{-1})}{\delta^2(\log m)^2}$ *and* $k \geq \frac{32\log(8\eta^{-1})}{\delta^2 m(\log m)^2}$, *then* $\gamma = \frac{4}{k}\mathrm{OPT}(\hat{\mathbf{v}}(\mathbf{r}'))$ *satisfies*

$$1/k \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r})) \leq \gamma \leq 12/k \cdot \mathrm{OPT}(\mathbf{v}(\mathbf{r}))\ ,$$

with probability at least $1 - \eta$.

APPENDIX B

OMITTED PROOFS FROM CHAPTER 4

B.1 Missing Proofs from Section 4.3.1

Lemma 4.3.2. *The value of program (P4), denoted by ρ', is an upper bound on the value of program (P3) which is ρ.*

Proof. Let $(\bar{\mathbf{v}}', \bar{\mathbf{q}}')$ be an arbitrary feasible assignment for program (P3) for which $\sum_{i=1}^{n} \bar{v}'_i \bar{q}'_i > 1$. We construct a corresponding assignment $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ which is feasible for program (P4) and yields the same objective value, that is $1 + \sum_{i=2}^{n} \bar{v}_i \bar{q}_i = \sum_{i=1}^{n} \bar{v}'_i \bar{q}'_i$ which then implies that $\text{BOUND}_I \leq \text{BOUND}_{II}$. Without loss of generality assume $\bar{v}'_1 \geq \ldots \geq \bar{v}'_n$. Let j be the smallest index for which $\sum_{i=1}^{j} \bar{v}'_i \bar{q}'_i > 1$. Observe that $2 \leq j \leq n$ because $\bar{v}'_i \bar{q}'_i \leq 1$ for all i. Let $\delta = 1 - \sum_{i=1}^{j-1} \bar{v}'_i \bar{q}'_i$. Observe that $0 \leq \delta < \bar{v}'_j \bar{q}'_j$. We construct a new optimal assignment $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ by setting for each $i \in \{2, \ldots, n\}$:

$$\bar{v}_i = \begin{cases} \bar{v}'_{j+i-2} & 2 \leq i \leq n-j+2 \\ 1 & n-j+3 \leq i \leq n \end{cases} \qquad \bar{q}_i = \begin{cases} \bar{q}'_j - \frac{\delta}{\bar{v}'_j} & i = 2 \\ \bar{q}'_{j+i-2} & 3 \leq i \leq n-j+2 \\ 0 & n-j+3 \leq i \leq n \end{cases} .$$

By the above construction it is easy to see that $1 + \sum_{i=2}^{n} \bar{v}_i \bar{q}_i = \sum_{i=1}^{n} \bar{v}'_i \bar{q}'_i$. So we only need to show that $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ is indeed a feasible assignment. Observe that $\sum_{i=2}^{n} \bar{q}_i \leq \sum_{i=1}^{n} \bar{q}'_i$, so the second constraint holds. So we only need to show that $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ satisfies the constraint (P4.1).

By rearranging (P3.1) we get

$$\prod_{i:\bar{v}'_i \geq p} \left(1 + \frac{\bar{v}'_i \bar{q}'_i}{p \cdot (1 - \bar{q}'_i)}\right) \leq \left(\frac{p}{p-1}\right) \qquad \forall p > 0.$$

We then relax the previous inequality by dropping the term $(1 - \bar{q}'_i)$ from the denominator of the left hand side and take the logarithm of both sides to get

$$\sum_{i:\bar{v}'_i \geq p} \ln\left(1 + \frac{\bar{v}'_i \bar{q}'_i}{p}\right) \leq \ln\left(\frac{p}{p-1}\right) \qquad \forall p > 0. \tag{B.1}$$

On the other hand, by invoking Lemma B.1.1 we can argue that [1]

$$\ln\left(1 + \frac{1}{\bar{v}_k}\right) + \sum_{i=2}^{k} \ln\left(1 + \frac{\bar{v}_i \bar{q}_i}{\bar{v}_k}\right) \leq \sum_{i=1}^{\min(k+j-2,n)} \ln\left(1 + \frac{\bar{v}'_i \bar{q}'_i}{\bar{v}_k}\right) \qquad \forall k \in \{2, \ldots, n\}. \tag{B.2}$$

Observe that for any given k the the right hand side of (B.2) is equal or less than the the left hand side of (B.1) for $p = \bar{v}_k$ which implies

$$\ln\left(1 + \frac{1}{\bar{v}_k}\right) + \sum_{i=2}^{k} \ln\left(1 + \frac{\bar{v}_i \bar{q}_i}{\bar{v}_k}\right) \leq \ln\left(\frac{\bar{v}_k}{\bar{v}_k - 1}\right) \qquad \forall k \in \{2, \ldots, n\}.$$

We can then further relax the above inequality by replacing the terms $\ln(1 + \frac{\bar{v}_i \bar{q}_i}{\bar{v}_k})$ with $\frac{1}{\bar{v}_k} \ln(1 + \bar{v}_i \bar{q}_i)$ and rearranging the terms to get the constraint (P4.1) which implies that $(\bar{\mathbf{v}}, \bar{\mathbf{q}})$ is feasible with respect to that constraint as well. □

Lemma 4.3.4. *The functions $\mathcal{V}(p)$, $\mathcal{Q}(p)$, and $\mathcal{V}(p) - \mathcal{Q}(p)$ are all decreasing in p, for $p > 1$.*

Proof. To prove $\mathcal{V}(p)$ is decreasing, we show its derivative is negative:

$$\mathcal{V}'(p) = \ln\left(1 + \frac{1}{p^2 - 1}\right) - \frac{2}{p^2 - 1} < \frac{1}{p^2 - 1} - \frac{2}{p^2 - 1} < 0.$$

The first inequality uses the fact that $\ln(1 + x) \leq x$. Similarly $\mathcal{Q}(p)$ is also decreasing because $\mathcal{Q}'(p) = \frac{1}{p}\mathcal{V}'(p) < 0$. Finally $\mathcal{V}(p) - \mathcal{Q}(p)$ is also decreasing because

$$(\mathcal{V}(p) - \mathcal{Q}(p))' = \mathcal{V}'(p)\left(1 - \frac{1}{p}\right) < 0.$$

[1] We invoke Lemma B.1.1 by setting $b = \frac{1}{\bar{v}_k}$, $a = \frac{\bar{v}_2 \bar{q}_2}{\bar{v}_k}$, $m = j$ and $z_i = \frac{\bar{v}'_i \bar{q}'_i}{\bar{v}_k}$.

$\square$

Lemma 4.3.5. *$\mathcal{V}(p) - \mathcal{V}(p') < \ln(\frac{p}{p-1}) - \ln(\frac{p'}{p'-1})$ for any $p' > p > 1$.*

Proof. Define $G(p) = \mathcal{V}(p) - \ln(\frac{p}{p-1})$. Observe that proving the inequality in the statement of the lemma is equivalent to proving $G(p) < G(p')$ which we do by showing $G(p)$ has positive derivative and is therefore increasing.

$$G'(p) = \ln\left(1 + \frac{1}{p^2 - 1}\right) - \frac{1}{p^2 + p}.$$

We will show that $G'(p)$ is decreasing which then implies $G'(p) > 0$ because $\lim_{p\to\infty} G'(p) = 0$. Therefore we only need to show that $G''(p) < 0$.

$$G''(p) = \frac{-(3p+1)}{(p-1)p^2(p+1)^2} < 0.$$

$\square$

Lemma B.1.1. *Consider any $a, b, z_1, \ldots, z_m \geq 0$ such that $a + b = \sum_{i=1}^{m} z_i$ and $a \leq z_m \leq b$. Then*

$$\ln(1 + b) + \ln(1 + a) \leq \sum_{i=1}^{m} \ln(1 + z_i).$$

Proof. We can re-write the equation as

$$\ln\left((1 + a)(1 + b)\right) \leq \ln \prod_{i=1}^{m} (1 + z_i) \tag{B.3}$$

Observe that

$$\begin{aligned}
\prod_{i=1}^{m} (1 + z_i) &\geq 1 + \sum_{i=1}^{m} z_i + z_m(\sum_{i=1}^{m-1} z_i) \\
&= (1 + \sum_{i=1}^{m-1} z_i)(1 + z_m) \\
&\geq (1 + a)(1 + b),
\end{aligned}$$

where the first inequality follows by eliminating some terms from the expansion of $\prod_{i=1}^{m}(1 + z_i)$, and the second inequality from the assumption that $(1 + a) \leq (1 + z_m)$ and $(1 + b) \geq (1 + \sum_{i=1}^{m-1} z_i)$. $\square$

APPENDIX C

OMITTED PROOFS AND DISCUSSIONS FROM CHAPTER 5

C.1 Other Deferred Proofs and Discussions

C.1.1 Discussion on choice of $\boldsymbol{\pi}$ for bandit symmetric range

We now describe how the choice of initial distribution $\boldsymbol{\pi}$ affects the bound given in Theorem 5.2.7.

- When the action set is finite, we can choose $\boldsymbol{\pi}$ to be the uniform distribution to get the term
$$O\Big(\frac{1}{\epsilon}\log(kc_i/c_{\min})\cdot c_i\Big)$$
This recovers the standard bound by setting $c_i = c_{\max}$ for all $i \in A$.
- We can choose $\pi_i = \frac{c_i}{\sum_{j\in A} c_j}$ to get $O\Big(\frac{1}{\epsilon}\log(\sum_{j\in A} c_j/c_{\min})\cdot c_i\Big)$. In particular, if the c_i's form an arithmetic progression with a constant difference then this is just $O\Big(\frac{\log k}{\epsilon}\cdot c_i\Big)$.
- If there are infinitely many experts but $\sum_{i\in A} c_i^{-1}$ is convergent, e.g., $c_i = c_{\min}\cdot(1+\epsilon)^{i-1}$, then we can choose $\pi_i = \frac{c_i^{-1}}{\sum_{j\in A} c_j^{-1}}$ for all $i \in A$. This gives $O\left(\frac{1}{\eta}\log\left(\sum_j \frac{c_{\min}}{c_j}\cdot\frac{c_i^2}{c_{\min}^2}\right)\cdot c_i\right)$.

C.1.2 Log factor for symmetric range - proof of Theorem 5.2.9

Proof of Theorem 5.2.9. We first show that for any online learning algorithm, and any sufficiently large $h > 1$, there is an instance that has two experts with $c_1 = 1$

and $c_2 = h$ with $T = \Theta(\log h)$ rounds, such that either

$$\mathbf{E}_{\text{REGRET}_1}[>]\ \tfrac{1}{2}T + \sqrt{h}\ , \qquad \text{or} \qquad \mathbf{E}_{\text{REGRET}_2}[>]\ \tfrac{1}{2}Th + \tfrac{1}{5}h\log_2 h\ .$$

We will construct this instance with $T = \frac{1}{2}\log_2 h - 1$ rounds adaptively that always has gain 0 for action 1 and gain either h or $-h$ for action 2. The proof of the theorem then follows as $c_{\min} = 1$, $c_{\max} = h$, $T = \frac{1}{2}\log_2 h - 1$, and $k = 2$ in this instance. Let q_t denote the probability that the algorithm picks action 2 in round t after having the same rewards 1 and h for the two actions respectively in the first $t-1$ rounds. We will first show that (1) if the algorithm has small regret with respect to action 1, then q_t must be upper bounded since the adversary may let action 2 have cost $-h$ in any round t in which q_t is too large. Then, we will show that (2) since q_t is upper bounded for any $1 \le t \le T$, the algorithm must have large regret with respect to action 2.

We proceed with the upper bounding q_t's. Concretely, we will show the following lemma.

Lemma C.1.1. *Suppose* $\mathbf{E}_{\text{REGRET}_1}[\le]\ \frac{1}{2}T + \sqrt{h}$. *Then, for any* $1 \le t \le T$, *we have* $q_t \le \frac{2^t}{\sqrt{h}}$.

Proof of Lemma C.1.1. We will prove by induction on t. Consider the base case $t = 1$. Suppose for contradiction that $q_1 > \frac{2}{\sqrt{h}}$. Then, consider an instance in which action 2 always has gain. In this case, the expected gain of the algorithm (even if it always correctly picks action 1 in the remaining instance) is at most $q_1 \cdot (-h) < -2\sqrt{h}$. This is a contradiction to the assumption that $\mathbf{E}_{\text{REGRET}_1}[\le]\ \frac{1}{2}T + \sqrt{h} < 2\sqrt{h}$.

Next, suppose the lemma holds for all rounds prior to round t. Then, the expected gain of algorithm in the first $t-1$ rounds if arm 2 has gain H is

$$\sum_{\ell=1}^{t-1} q_\ell \cdot h \leq \sum_{\ell=1}^{t-1} 2^\ell \sqrt{h} = \left(2^t - 2\right)\sqrt{h}\,.$$

Suppose for contradiction that $q_t > \frac{2^t}{\sqrt{h}}$. Then, consider an instance in which action 2 has gain H in the first $t-1$ rounds and $-H$ afterwards. In this case, the expected gain of the algorithm (even if it always correctly picks action 1 after round t) is at most

$$\left(2^t - 2\right)\sqrt{h} + q_t(-h) < \left(2^t - 2\right)\sqrt{h} + 2^t\sqrt{h} < -2\sqrt{h}\,.$$

This is a contradiction to the assumption that $\mathbf{E}_{\text{REGRET}_1}[\leq]\,\frac{1}{2}T + \sqrt{h} < 2\sqrt{h}$. □

Consider an instance in which action 2 always has gain H. Suppose that $\mathbf{E}_{\text{REGRET}_1}[\leq]\,\frac{1}{2}T + \sqrt{h}$. As an immediate implication of the above lemma, the algorithm is that the expected gain of the algorithm is upper bounded by:

$$\sum_{t=1}^{T} q_t h \leq \sum_{t=1}^{T} 2^t\sqrt{h} < 2^{T+1}\sqrt{h} = h\,.$$

Note that in this instance $\mathbf{E}_{G_2}[=]\,T \cdot h$. Thus, the regret w.r.t. action 2 is at least $(T-1)h$, which is greater than $\frac{1}{2} \cdot \mathbf{E}_{G_2}[+]\,\frac{1}{5}h\log_2 h$ for sufficiently large h. □

C.1.3 Regret lower-bound for symmetric range -proof of Theorem 5.2.12

Proof of Theorem 5.2.12. We first show that for any online multi-scale bandits algorithm problem, and there is an instance that has two arms with $c_1 = 1$ and $c_2 = h$ for some sufficiently large h, a sufficiently large T, and $\epsilon = \sqrt{\frac{h}{256T}}$, such that either

$$\mathbf{E}_{\text{REGRET}_1}[>] \, \epsilon T + \frac{1}{256\epsilon}h \,, \qquad \text{or} \qquad \mathbf{E}_{\text{REGRET}_2}[>] \, \epsilon T h + \frac{1}{256\epsilon}h^2$$

We will prove the existence of this instance by looking at the stochastic setting, i.e., the gain vectors $\mathbf{g}(t)$'s are i.i.d. for $1 \le t \le T$. We consider two instances, both of which admit a fixed gain of 0 for action 1. In the first instance, the gain of action 2 is h with probability $\frac{1}{2} - 2\epsilon$, and $-h$ otherwise. Hence, the expected gain of playing action 2 is $-4\epsilon h$ per round in instance 1. In the second instance, the gain of action 2 is h with probability $\frac{1}{2} + 2\epsilon$, and $-h$ otherwise. Hence, the expected gain of playing action two is $4\epsilon h$ per round in instance 2. Note this proves the theorem, as $c_{\min} = 1$, $c_{\max} = h$, $k = 2$ and and $T = \frac{h}{256\epsilon^2}$.

Suppose for contradiction that the algorithm satisfies:

$$\mathbf{E}_{\text{REGRET}_1}[\le] \, \epsilon T + \frac{1}{256\epsilon}h = \frac{1}{128\epsilon}h \quad , \quad \mathbf{E}_{\text{REGRET}_2}[\le] \, \epsilon h T + \frac{1}{256\epsilon}h^2 = \frac{1}{128\epsilon}h^2 \,.$$

Let N_1 denote the expected number of times that the algorithm plays action 2 in instance 1. Then, the expected regret with respect to action 1 in instance 1 is $N_1 \cdot 4\epsilon h$. By the assumption that $\mathbf{E}_{\text{REGRET}_1}[\le] \, \frac{1}{128\epsilon}h$, we have $N_1 \le \frac{1}{512\epsilon^2}$.

Next, by standard calculation, we get that the Kullback-Leibler (KL) divergence of the observed rewards in a single round in the two instances is 0 if action 1 is played and is at most $64\epsilon^2$ (for $0 < \epsilon < 0.1$) if action 2 is played. So the KL divergence of the observed reward sequences in the two instances is at most $64\epsilon^2 \cdot N_1 \le \frac{1}{8}$.

Then, we use a standard inequality about KL divergences. For any measurable function $\psi : X \mapsto \{1, 2\}$, we have $\Pr_{X \sim \rho_1}\Big(\psi(X) = 2\Big) + \Pr_{X \sim \rho_2}\Big(\psi(X) = 1\Big) \ge \frac{1}{2}\exp\Big(- KL(\rho_1, \rho_2)\Big)$. For any $1 \le t \le T$, let ρ_1 and ρ_2 be the distribution of observed rewards up to a round t in the two instances, and let $\psi(X)$ be the action played by the algorithm. By this inequality and the above bound on the KL divergence between the observed rewards in the two instances, we get that in each round, the probability that the algorithm plays action 2 in instance 1, plus the probability that the algorithm plays action 1 in instance 2, is at least $\frac{1}{2}\exp(-\frac{1}{8}) > \frac{2}{5}$ in any round t. Thus, the expected number of times that the algorithm plays action 1 in instance 2 from round 1 to T, denoted as N_2, is at least $N_2 \geq \frac{2}{5} \cdot T - N_1 \geq \frac{1}{3} \cdot T$, where the second inequality holds for sufficiently large h. Therefore, the expected regret w.r.t. action 2 in instance 2 is at least: $4\epsilon h \cdot \frac{1}{3} \cdot T = \frac{4}{3}\epsilon h T > \frac{1}{128\epsilon}h^2$. This is a contradiction to our assumption that $\mathbf{E}_{\text{REGRET}_2}[\leq] \frac{1}{128\epsilon}h^2$. □

C.1.4 Proof of Lemma 5.3.3

Proof of Lemma 5.3.3. We have:

$$\sum_{i \in A} q_i \cdot G_i - \mathbf{E}_{G_{\text{ALG}}}[=] \sum_{t \in [T]} \mathbf{q} \cdot \mathbf{g}(t) - \sum_{t \in [T]} \mathbf{p}(t) \cdot \mathbf{g}(t) = \sum_{t \in [T]} \mathbf{g}(t) \cdot \left(\mathbf{q} - \mathbf{p}(t)\right) \tag{C.1}$$

By applying the regret bound of OMD (Lemma 2.4.1) to upper-bound the RHS, we have

$$\sum_{i \in A} q_i \cdot G_i - \mathbf{E}\left[G_{\text{ALG}}\right] \leq \frac{1}{\eta} \sum_{t \in [T]} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) + \frac{1}{\eta} D_F(\mathbf{q}, \mathbf{p}(1)) \tag{C.2}$$

To bound the first term in regret, a.k.a *local norm*, we have:

$$\begin{aligned} D_F(\mathbf{p}(t), \mathbf{w}(t+1)) &= \sum_{i \in A} c_i \cdot (p_i(t) \ln(\frac{p_i(t)}{w_i(t+1)}) - p_i(t) + w_i(t+1)) \\ &= \sum_{i \in A} c_i \cdot p_i(t)(-\eta \cdot \frac{g_i(t)}{c_i} - 1 + exp(\eta \cdot \frac{g_i(t)}{c_i})) \end{aligned} \tag{C.3}$$

Note that $\eta \cdot \frac{g_i(t)}{c_i} \in [-1, 1]$ because $g_i(t) \in [-c_i, c_i]$ and $0 < \eta \le 1$. By $\exp(x) - x - 1 \le x^2$ for $-1 \le x \le 1$ and that $\eta g_i(t) \in [-c_i, c_i]$, the above is upper bounded by $\eta^2 \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$. We can also rewrite the second term in regret. In fact, if we set $\mathbf{p}(1) = \boldsymbol{\mu}$, then

$$\frac{1}{\eta} \cdot D_F(\mathbf{q}, \mathbf{p}(1)) = \frac{1}{\eta} \cdot \sum_{i\in A} c_i \Big(q_i \ln\Big(\frac{q_i}{\mu_i}\Big) - q_i + \mu_i \Big)$$

By summing the upper-bounds $\eta^2 \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$ on each term of local norm in (C.3) for $t \in [T]$ and putting all the pieces together, we get the desired bound. □

We also provide an elementary proof of this lemma using first principles.

Proof of Lemma 5.3.3 from first principles. By using update rule of Algorithm 5, we have $g_i(t) = \frac{c_i}{\eta} \log(\frac{w_i(t+1)}{p_i(t)})$ for any $i \in A$. Therefore:

$$\begin{aligned}
\mathbf{g}(t) \cdot \big(\mathbf{q} - \mathbf{p}(t)\big) &= \sum_{i\in A} g_i(t)\big(q_i - p_i(t)\big) \\
&= \sum_{i\in A} \frac{c_i}{\eta} \cdot \log\Big(\frac{w_i(t+1)}{p_i(t)}\Big) \cdot \big(q_i - p_i(t)\big) \\
&= \frac{1}{\eta}\Bigg(\sum_{i\in S} c_i \cdot q_i \cdot \log\Big(\frac{w_k(t+1)}{p_k(t)}\Big) + \sum_{i\in A} c_i \cdot p_i(t) \cdot \log\Big(\frac{p_i(t)}{w_i(t+1)}\Big) \Bigg) \\
&= \frac{1}{\eta}\Bigg(\sum_{i\in S} c_i \cdot q_i \cdot \log\Big(\frac{w_k(t+1)}{p_k(t+1)}\Big) + \sum_{i\in S} c_i \cdot q_i \cdot \log\Big(\frac{p_k(t+1)}{p_k(t)}\Big) \\
&\qquad + \sum_{i\in A} c_i \cdot p_i(t) \cdot \log\Big(\frac{p_i(t)}{w_i(t+1)}\Big) \Bigg)
\end{aligned} \tag{C.4}$$

Now, note that due to the normalization step of Algorithm 5, for any $i \in S$ we have:

$$c_i \cdot \log(\frac{w_i(t+1)}{p_i(t+1)}) = \lambda = \sum_{j\in A} c_j \cdot p_j(t+1) \cdot \frac{\lambda}{c_j} = \sum_{j\in A} c_j \cdot p_j(t+1) \cdot \log(\frac{w_j(t+1)}{p_j(t+1)})$$

So the first summation in (C.4) is equal to:

$$\begin{aligned}\sum_{i\in S} c_i \cdot q_i \cdot \log\Big(\frac{w_k(t+1)}{p_k(t+1)}\Big) &= \sum_{i\in S} q_i \cdot \sum_{j\in A} c_j \cdot p_j(t+1) \cdot \log(\frac{w_j(t+1)}{p_j(t+1)}) \\ &= \sum_{j\in A} c_j \cdot p_j(t+1) \cdot \log(\frac{w_j(t+1)}{p_j(t+1)}) \\ &= \sum_{i\in A} c_i \cdot p_i(t+1) \cdot \log(\frac{w_i(t+1)}{p_i(t+1)}) \end{aligned} \tag{C.5}$$

Combining Eqn. (C.4) and (C.5), we have:

$$\begin{aligned}\mathbf{g}(t) \cdot \Big(\mathbf{q} - \mathbf{p}(t)\Big) = \frac{1}{\eta} \sum_{i\in A} c_i \cdot \Big(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) + p_i(t+1) \cdot \log(\frac{w_i(t+1)}{p_i(t+1)})\Big) \\ + \frac{1}{\eta} \sum_{i\in S} c_i \cdot q_i \cdot \log\Big(\frac{p_i(t+1)}{p_i(t)}\Big)\end{aligned}$$

The second part is a telescopic sum when we sum over t. We will upper bound the first part as follows. By $\log(x) \le (x-1)$, we get that:

$$\begin{aligned}&\sum_{i\in A} c_i \cdot \Big(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) + p_i(t+1) \cdot \log(\frac{w_i(t+1)}{p_i(t+1)})\Big) \\ &\le \sum_{i\in A} c_i \cdot \Big(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) - p_i(t+1) + w_i(t+1)\Big) \\ &= \sum_{i\in A} c_i\Big(p_i(t) - p_i(t+1)\Big) + \sum_{i\in A} c_i \Big(p_i(t) \log(\frac{p_i(t)}{w_i(t+1)}) - p_i(t) + w_i(t+1)\Big)\end{aligned}$$

Again, the first part is a telescopic sum when we sum over t. We will further work on the second part. By the relation between $w_i(t+1)$ and $p_i(t)$, we get that:

$$\sum_{i\in A} c_i \cdot \Big(p_i(t) \cdot \log(\frac{p_i(t)}{w_i(t+1)}) - p_i(t) + w_i(t+1)\Big) = \sum_{i\in A} c_i p_i(t) \Big(-\frac{\eta g_i(t)}{c_i} - 1 + \exp(\eta \frac{g_i(t)}{c_i})\Big)$$

Note that $\eta \cdot \frac{g_i(t)}{c_i} \in [-1, 1]$ because $g_i(t) \in [-c_i, c_i]$ and $0 < \eta \le 1$. By $\exp(x) - x - 1 \le x^2$ for $-1 \le x \le 1$ and that $\eta g_i(t) \in [-c_i, c_i]$, the above is upper bounded by $\eta^2 \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$. Putting together, we get that:

$$\mathbf{g}(t) \cdot \Big(\mathbf{q} - \mathbf{p}(t)\Big) \le \frac{1}{\eta} \sum_{i\in S} c_i \cdot \Big(q_i \cdot \log\Big(\frac{p_i(t+1)}{p_i(t)}\Big) + p_i(t) - p_i(t+1)\Big) + \eta \sum_{i\in A} p_i(t) \frac{(g_i(t))^2}{c_i}$$

Summing over t, we have:

$$\mathbf{g}(t)\cdot\left(\mathbf{q}-\mathbf{p}(t)\right) \leq \frac{1}{\eta}\sum_{i\in S} c_i\cdot\left(q_i\cdot\log\left(\frac{p_i(T+1)}{p_i(1)}\right)+p_i(1)-p_i(T+1)\right)+\eta\sum_{t\in[T]}\sum_{i\in A} p_i(t)\frac{(g_i(t))^2}{c_i}$$

Finally, by $\log(x) \leq (x-1)$, we get that $q_i \log\left(\frac{p_i(T+1)}{q_i}\right) \leq p_i(T+1) - q_i$. Hence, we have:

$$\mathbf{g}(t)\cdot\left(\mathbf{q}-\mathbf{p}(t)\right) \leq \frac{1}{\eta}\sum_{i\in S} c_i\cdot\left(q_i\cdot\log\left(\frac{q_i}{p_i(1)}\right)+p_i(1)-q_i\right)+\eta\sum_{t\in[T]}\sum_{i\in A} p_i(t)\frac{(g_i(t))^2}{c_i}$$

The lemma then follows by our choice of the initial distribution. □

C.1.5 Proof of OMD regret bound

In order to prove the OMD regret bound, we need some properties of Bregman divergence.

Lemma C.1.2 (Properties of Bregman divergence (Bubeck, 2011))**.** *Suppose $F(\cdot)$ is a Legendre function and $D_F(\cdot,\cdot)$ is its associated Bregman divergence as defined in Definition 2.4.1. Then:*

- *$D_F(x,y) > 0$ if $x \neq y$ as F is strictly convex, and $D_F(x,x) = 0$.*
- *$D_F(.,y)$ is a convex function for any choice of y.*
- (Pythagorean theorem) *If $\mathcal{A}$ is a convex set, $a \in \mathcal{A}$, $b \notin \mathcal{A}$ and $c =$ $\text{argmin}\, x \in \mathcal{A} D_F(x,b)$, then*

$$D_F(a,c) + D_F(c,b) \leq D_F(a,b)$$

Given Lemma C.1.2, we are now ready to prove Lemma 2.4.1.

Proof of Lemma 2.4.1. To obtain the OMD regret bound, we have:

$$\begin{aligned}
\mathbf{q}\cdot\mathbf{g}(t)-\mathbf{p}(t)\cdot\mathbf{g}(t) &= \frac{1}{\eta}(\mathbf{q}-\mathbf{p}(t))\cdot(\nabla F(\mathbf{w}(t+1))-\nabla F(\mathbf{p}(t)))\\
&= \frac{1}{\eta}(D_F(qb,\mathbf{p}(t))+D_F(\mathbf{p}(t),\mathbf{w}(t+1))-D_F(qb,\mathbf{w}(t+1)))\\
&\overset{(1)}{\leq} \frac{1}{\eta}D_F(\mathbf{p}(t),\mathbf{w}(t+1))+\frac{1}{\eta}\left(D_F(\mathbf{q},\mathbf{p}(t))-D_F(\mathbf{q},\mathbf{p}(t+1))\right) \qquad \text{(C.6)}
\end{aligned}$$

where in inequality (1) we use the facts that $D_F(\mathbf{p}(t+1),\mathbf{w}(t+1)) \geq 0$ and $D_F(\mathbf{q},\mathbf{p}(t+1)) + D_F(\mathbf{p}(t+1),\mathbf{w}(t+1)) \leq D_F(\mathbf{q},\mathbf{w}(t+1))$ due to Pythagorean theorem (Lemma C.1.2). By summing up both hand sides of (C.6) for $t = 1,\cdots,T$ we have:

$$\sum_{t\in[T]}\mathbf{g}(t)\cdot\left(\mathbf{q}-\mathbf{p}(t)\right) \leq \frac{1}{\eta}\sum_{t\in[T]}D_F(\mathbf{p}(t),\mathbf{w}(t+1))+\frac{1}{\eta}D_F(\mathbf{q},\mathbf{p}(1)) \qquad \text{(C.7)}$$

□

C.1.6 Symmetric range bandit regret - proof of Lemma 5.3.6

Proof of Lemma 5.3.6. We further define:

$$\begin{aligned}
\widetilde{G}_{\text{ALG}} &\triangleq \textstyle\sum_{t\in[T]} g_{i_t}(t) = \sum_{t\in[T]}\tilde{\mathbf{p}}(t)\cdot\tilde{\mathbf{g}}(t)\,,\\
\widetilde{G}_j &\triangleq \textstyle\sum_{t\in[T]}\tilde{g}_j(t)\,.
\end{aligned}$$

In expectation over the randomness of the algorithm, we have:

1. $\mathbf{E}\left[G_{\text{ALG}}\right] = \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right]$; and
2. $G_j = \mathbf{E}\left[\widetilde{G}_j\right]$ for any $j \in A$.

Hence, to upper bound $\mathbf{E}\left[\text{REGRET}_i\right] = G_i - \mathbf{E}\left[G_{\text{ALG}}\right]$, it suffices to upper bound $\mathbf{E}\left[\widetilde{G}_i - \widetilde{G}_{\text{ALG}}\right]$.

By the definition of the probability that the algorithm picks each arm, i.e., $\tilde{\mathbf{p}}(t)$, and that reward of each round is at least $-c_{\max}$, we have that:

$$\mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] \geq (1-\gamma)\sum_{t\in[T]} \mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t) - \gamma T c_{\max} .$$

Hence, for any benchmark distribution $\mathbf{q}$ over A, we have that:

$$\begin{aligned}
&\sum_{j\in A} q_j \cdot \mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] \\
&\leq \mathbf{E}\left[\textstyle\sum_{j\in A} q_j\cdot \widetilde{G}_j - \sum_{t\in[T]}\mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t)\right] + \tfrac{\gamma}{1-\gamma}\mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] + \tfrac{\gamma}{1-\gamma}T c_{\max} \\
&\leq \mathbf{E}\left[\textstyle\sum_{j\in A} q_j\cdot \widetilde{G}_j - \sum_{t\in[T]}\mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t)\right] + 2\gamma\mathbf{E}\left[\widetilde{G}_{\text{ALG}}\right] + 2\gamma T c_{\max} \\
&\leq \mathbf{E}\left[\textstyle\sum_{j\in A} q_j\cdot \widetilde{G}_j - \sum_{t\in[T]}\mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t)\right] + 4\gamma T c_{\max} . \qquad \text{(C.8)}
\end{aligned}$$

where the second inequality is due to $\gamma \leq \frac{1}{2}$, and the 3rd inequality follows by that $c_{\max}$ is the largest possible reward per round.

Next, we upper bound the first term on the RHS of (C.8). Note that $\mathbf{p}(t)$'s are the probability of choosing experts by MSMW when the experts have rewards $\tilde{\mathbf{g}}(t)$'s. By Lemma 5.3.3, we have that for any benchmark distribution $\mathbf{q}$ over S, the Bandit-MSMW algorithm satisfies that:

$$\sum_{j\in A} q_j\cdot\widetilde{G}_j - \sum_{t\in[T]}\mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t) \leq \eta\sum_{t\in[T]}\sum_{j\in A}\frac{p_j(t)}{c_j}\cdot\left(\tilde{g}_j(t)\right)^2 + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j\ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) . \qquad \text{(C.9)}$$

For any $t\in[T]$ and any $j\in A$, by the definition of $\tilde{g}_j(t)$, it equals $\frac{g_j(t)}{\tilde{p}_j(t)}$ with probability $\tilde{p}_j(t)$, and equals 0 otherwise. Thus, if we fix the random coin flips in the first $t-1$ rounds and, thus, fix $\tilde{\mathbf{p}}(t)$, and take expectation over the randomness in round t, we have that:

$$\mathbf{E}\left[\frac{p_j(t)}{c_j}\cdot\left(\tilde{g}_j(t)\right)^2\right] = \frac{p_j(t)}{c_j}\cdot\tilde{p}_j(t)\cdot\left(\frac{g_j(t)}{\tilde{p}_j(t)}\right)^2 = \frac{p_j(t)}{\tilde{p}_j(t)}\frac{(g_j(t))^2}{c_j} .$$

Further note that $\tilde{p}_j(t) \geq (1-\gamma)p_j(t)$, and $|g_j(t)| \leq c_j$, the above is upper bounded by $\frac{1}{1-\gamma}|g_j(t)| \leq 2|g_j(t)| \leq 2c_{\max}$. Putting together with (C.9), we have that for any $0 < \eta \leq \frac{\gamma}{n}$:

$$\begin{aligned}
\mathbf{E}\left[\sum_{j\in A} q_j \cdot \widetilde{G}_j - \sum_{t\in[T]} \mathbf{p}(t)\cdot\tilde{\mathbf{g}}(t)\right] &\leq \eta \sum_{t\in[T]}\sum_{j\in A} 2c_{\max} + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) \\
&= 2\eta T k c_{\max} + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right)
\end{aligned}$$

Combining with (C.8), we have (recall that $\eta \leq \frac{\gamma}{k}$):

$$\begin{aligned}
\sum_{j\in A} q_j \cdot \mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] &\leq 2\eta T k c_{\max} + \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) + 4\gamma T c_{\max} \\
&\leq \frac{1}{\eta}\sum_{j\in A} c_j\left(q_j \ln\left(\frac{q_j}{p_j(1)}\right) - q_j + p_j(1)\right) + 6\gamma T c_{\max}
\end{aligned}$$

Let $\mathbf{q} = (1-\gamma)\mathbf{1}_i + \frac{\gamma}{k}\mathbf{1}$. Recall that $\mathbf{p}(1) = (1-\gamma)\mathbf{1}_{i_{\min}} + \frac{\gamma}{k}\mathbf{1}$ (recall $i_{\min}$ is the arm with minimum range $c_{i_{\min}}$). Similar to the discussion for the expert problem in Section 5.3.1, the first term on the RHS is upper bounded by $O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i\right)$. Hence, we have:

$$\sum_{j\in A} q_j \cdot \mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] \leq O\left(\frac{1}{\eta}\log\left(\frac{k}{\gamma}\right)\cdot c_i\right) + 6\gamma T c_{\max}\,. \tag{C.10}$$

Further, the LHS is lower bounded as:

$$(1-\gamma)\mathbf{E}\left[\widetilde{G}_i\right] + \frac{\gamma}{k}\sum_{j\in A}\mathbf{E}\left[\widetilde{G}_j\right] - \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right] \geq (1-\gamma)\mathbf{E}\left[\widetilde{G}_i\right] - \gamma T c_{\max} - \mathbf{E}\left[\widetilde{G}_{\mathrm{ALG}}\right]\,.$$

The lemma then follows by putting it back to (C.10) and rearranging terms.

□

BIBLIOGRAPHY

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006. ACM, 2014.

Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1405–1424. SIAM, 2015.

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *arXiv preprint arXiv:0911.2974*, 2009.

Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, pages 512–521. IEEE, 2011.

Saeed Alaei, Hu Fu, Nima Haghpanah, and Jason Hartline. The simple economics of approximately optimal auctions. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 628–637. IEEE, 2013.

Saeed Alaei, Jason D Hartline, Rad Niazadeh, Emmanouil Pountourakis, and Yang Yuan. Optimal auctions vs. anonymous pricing. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1446–1463. ACM, 2015.

Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Learning prices for repeated auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pages 1169–1177, 2013.

Aaron Archer, Christos Papadimitriou, Kunal Talwar, and Éva Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. *Internet Mathematics*, 1(2):129–150, 2004.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

Moshe Babaioff, Robert Kleinberg, and Aleksandrs Slivkins. Multi-parameter mechanisms with implicit payment computation. In *Proceedings of the 14th ACM Conference on Electronic Commerce*, pages 35–52, 2013.

Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. *ACM Transactions on Economics and Computation*, 3(1):4, 2015a.

Moshe Babaioff, Robert Kleinberg, and Aleksandrs Slivkins. Truthful mechanisms with implicit payment computation. *J. ACM*, 62(2):10:1–10:37, May 2015b. ISSN 0004-5411.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216. IEEE, 2013a.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216. IEEE, 2013b.

Ziv Bar-Yossef, Kirsten Hildrum, and Felix Wu. Incentive-compatible online auctions for digital goods. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 964–970. Society for Industrial and Applied Mathematics, 2002.

Xiaohui Bei and Zhiyi Huang. Bayesian incentive compatibility via fractional assignments. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms*, pages 720–733. SIAM, 2011.

Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57 (6):1407–1420, 2009.

Avrim Blum and Jason Hartline. Near-optimal online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1156–1163. Society for Industrial and Applied Mathematics, 2005.

Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. Online learning in online auctions. *Theoretical Computer Science*, 324(2-3):137–146, 2004.

Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.

Sébastien Bubeck, Nikhil R Devanur, Zhiyi Huang, and Rad Niazadeh. Online auctions and multi-scale online learning design. *arXiv:1705.09700*, 2017.

Jeremy Bulow Bulow and Paul Klemperer. Auctions versus negotiations. *American Economic Review*, 86:180–194, 1996.

Yang Cai and Constantinos Daskalakis. Extreme-value theorems for optimal multidimensional pricing. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, pages 522–531. IEEE, 2011.

Shuchi Chawla, Jason D Hartline, and Robert Kleinberg. Algorithmic pricing via virtual valuations. 2007.

Shuchi Chawla, Jason D Hartline, David Malec, and Balu Sivan. Sequential posted pricing and multi-parameter mechanism design. 2010a.

Shuchi Chawla, David Malec, and Balu Sivan. The power of randomness in bayesian optimal mechanism design. pages 149–158, 2010b.

Shuchi Chawla, Nicole Immorlica, and Brendan Lucier. On the limits of black-box reductions in mechanism design. In *Proceedings of the 44th ACM Symposium on Theory of Computing*, pages 435–448. ACM, 2012.

Ning Chen, Nick Gravin, and Pinyan Lu. Optimal competitive auctions. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 253–262. ACM, 2014.

Xiao Alison Chen and Zizhuo Wang. A near-optimal dynamic learning algorithm for online matching problems with concave returns. *arXiv preprint arXiv:1307.5934*, 2013.

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 243–252, 2014.

Arnoud V den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18, 2015.

Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 29–38. ACM, 2011.

Nikhil R Devanur, Yuval Peres, and Balasubramanian Sivan. Perfect bayesian equilibria in repeated sales. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 983–1002. SIAM, 2014.

Nikhil R Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 426–439. ACM, 2016.

Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. *Games and Economic Behavior*, 2014.

Shaddin Dughmi, Jason D Hartline, Robert Kleinberg, and Rad Niazadeh. Bernoulli factories and black-box reductions in mechanism design. *arXiv:1703.04143*, 2017.

Dylan Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Personal communication. 2017.

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

Andrew Goldberg, Jason D Hartline, Anna Karlin, Michael Saks, and Andrew Wright. Competitive auctions. *Games and Economic Behavior*, 55:242–269, 2006.

Yannai A Gonczarowski and Noam Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the ACM STOC*, 2017.

Nima Haghpanah and Jason Hartline. Reverse mechanism design. *arXiv preprint arXiv:1404.1341*, 2014.

Jason Hartline. Bayesian mechanism design. *Foundations and Trends® in Theoretical Computer Science*, 8(3):143–263, 2013.

Jason D Hartline and Brendan Lucier. Bayesian algorithmic mechanism design. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 301–310. ACM, 2010.

Jason D Hartline and Brendan Lucier. Non-optimal mechanism design. *The American Economic Review*, 105(10):3102–3124, 2015.

Jason D Hartline and Tim Roughgarden. Simple versus optimal mechanisms. 2009.

Jason D Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian incentive compatibility via matchings. *SODA*, 2011.

Jason D Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian incentive compatibility via matchings. *Games and Economic Behavior*, 92(C):401–429, 2015.

Zhiyi Huang and Sampath Kannan. The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*, pages 140–149. IEEE, 2012.

Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. Making the most of your samples. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 45–60. ACM, 2015.

Mark Huber. Optimal linear bernoulli factories for small mean problems. *CoRR*, abs/1507.00843, 2015.

MS Keane and George L O'Brien. A bernoulli factory. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 4(2):213–219, 1994.

Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 303–312. ACM, 2014.

Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 594–605. IEEE, 2003.

Krzysztof Łatuszyński. The bernoulli factory, its extensions and applications. *Proceedings of IWAP 2010*, pages 1–5, 2010.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Jamie H Morgenstern and Tim Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems*, pages 136–144, 2015.

Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.

Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6 (1):58–73, 1981.

Şerban Nacu and Yuval Peres. Fast simulation of new coins from old. *The Annals of Applied Probability*, 15(1A):93–115, 2005.

Noam Nisan. Turing's invisible hand: ec10 and current trends in algorithmic game theory, 2010. URL https://agtb.wordpress.com/2010/06/13/ec10-and-current-trends-in-algorithmic-game-theory/.

Tim Roughgarden and Okke Schrijvers. Ironing in the dark. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 1–18. ACM, 2016.

Ilya Segal. Optimal pricing mechanisms with unknown demand. *The American economic review*, 93(3):509–529, 2003.

Vasilis Syrgkanis. A sample complexity measure with applications to learning optimal auctions. *arXiv preprint arXiv:1704.02598*, 2017.

Kalyan T Talluri and Garrett J Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Science & Business Media, 2006.

Vladimir G Vovk. A game of prediction with expert advice. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 51–60. ACM, 1995.

Qiqi Yan. Mechanism design via correlation gap. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 710–719. SIAM, 2011.