Abstract of "Learning Equilibria of Simulation-Based Games: Applications to Empirical Mechanism Design" by Enrique Areyan Viqueira, Ph.D., Brown University, May 2021.

In this thesis, we first contribute to the empirical-game theoretic analysis (EGTA) literature both from a theoretical and a computational perspective. Theoretically, we present a mathematical framework to precisely describe simulation-based games and analyze their properties. In a simulation-based game, one only gets to observe *samples* of utility functions but never a complete analytical description. We provide results that complement and strengthen previous results on guarantees of the approximate Nash equilibria learned from samples. Computationally, we find and thoroughly evaluate *Probably Approximate Correct* (PAC) learning algorithms, which we show make frugal use of data to provably solve simulation-based games, up to a user's given error tolerance.

Next, we turn our attention to mechanism design. When mechanism design depends on EGTA, it is called *empirical mechanism design* (EMD). Equipped with our EGTA framework, we further present contributions to EMD, in particular to parametric EMD. In parametric EMD, there is an overall (parameterized) mechanism (e.g., a second price auction with reserve prices as parameters). The choice of parameters then determines a mechanism (e.g., the reserve price being \$10 instead of \$100). Our EMD contributions are again two-fold. From a theoretical point of view, we formulate the problem of finding the optimal parameters of a mechanism as a black-box optimization problem. For the special case where the parameter space is finite, we present an algorithm that, with high probability, provably finds an approximate global optimal. For more general cases, we present a Bayesian optimization algorithm and empirically show its effectiveness.

EMD is only as effective as the set of heuristic strategies used to optimize a mechanism's parameters. To demonstrate our methodology's effectiveness, we developed rich bidding heuristics in one specific domain: electronic advertisement auctions. These auctions are an instance of combinatorial auctions, a vastly important auction format used in practice to allocate many goods of interest (e.g., electromagnetic spectra). Our work on designing heuristics for electronic advertisement led us to contribute heuristics for the computation of approximate competitive (or Walrasian) equilibrium, work of interest in its own right.

Learning Equilibria of Simulation-Based Games: Applications to Empirical Mechanism Design

by

Enrique Areyan Viqueira

Sc. M., Brown University, 2017

MAT., Indiana University, 2015

Sc. M., Indiana University, 2013

Licenciado en Computación, Universidad Central de Venezuela, 2010

A dissertation submitted in partial fulfillment of the

requirements for the Degree of Doctor of Philosophy

in the Department of Computer Science at Brown University

Providence, Rhode Island

May 2021

This dissertation by Enrique Areyan Viqueira is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date 3/15/2021 ______________________ Amy Greenwald, Director

Recommended to the Graduate Council

Date 3/15/2021 ______________________ George Konidaris, Reader

Date 3/16/2021 ______________________ Benjamin Lubin, Reader

DocuSigned by:
Serdar Kadioglu
5C768366F41A4A8...

Date 3/18/2021 ______________________ Serdar Kadioglu, Reader

Approved by the Graduate Council

Date ______________________ Andrew G. Campbell
Dean of the Graduate School

Vita

In his home country, Venezuela, Enrique Areyan Viqueira attended Universidad Central de Venezuela, where he received a Licenciado en Computación (Sc.B. in Computer Science). He then moved to the United States and attended graduate school at Indiana University, where he received a Master of Science in Computer Science and a Master of Arts for Teachers with a major in Mathematics. He then attended Brown University, receiving a Masters in Computer Science en route to his Ph.D.

At Brown, Enrique was the sole instructor for CS3: Introduction to Computation for the Humanities and Social Sciences and co-instructor for CS1410: Introduction to Artificial Intelligence. He also actively developed content for CS1951k: Algorithmic Game Theory, including hands-on laboratories where students developed automated trading agents for advertisement exchange models. He received the Paris Kanellakis fellowship for the academic year 2018-2019. In 2017, 2018, and 2019, Enrique was a summer research scientist intern for Amazon.

Together with his advisor, Dr. Amy Greenwald, Enrique spent a year in Tokyo, Japan, working as a researcher at the National Institute of Advanced Industrial Science and Technology (AIST). At AIST, Enrique actively contributed to the Automated Negotiating Agents Competition (ANAC) to further develop automated negotiation. He designed and developed an agent that reached the Supply Chain Management League's finals in 2019 and 2020, which lead to Enrique coauthoring *Supply Chain Management World* [87].

During his Ph.D., Enrique published the following papers: *Principled Autonomous Decision Making for Markets* [3], *On Approximate Welfare-and Revenue-Maximizing Equilibria for Size-Interchangeable Bidders* [4, 5], *Learning Simulation-Based Games from Data* [7, 6], *Improved Algorithms for Learning Equilibria in Simulation-Based Games* [8], *Parametric Mechanism Design under Uncertainty* [125] ,and *Learning Competitive Equilibria in Noisy Combinatorial Markets* [9, 126].

Para Karina, Calena y Pepe

Acknowledgements

This thesis would not have existed without the guidance of my advisor, Dr. Amy Greenwald. There is not enough space here to properly thank her. But I know there will be plenty of opportunities to thank her in the future as we remain close friends and collaborators.

I am blessed to have been part of wonderful institutions where I was lucky enough to meet the following mentors who contributed to my intellectual and personal growth.

During high school, I met Hernan Rosas, who graciously allowed me (a kid at the time) to contribute to the development of VenezuelaTuya.com, which was then a small website to promote tourism in our home country. Hernan later included me in other ventures from which I learned a great deal. I am grateful to Hernan for those opportunities and happy to have him as a friend.

During my senior year at my alma mater, Universidad Central de Venezuela, I knew I wanted to do my undergraduate thesis in AI but didn't know how to begin. Ignacio Calderón, the instructor for the undergraduate introduction to AI, was kind enough to be my thesis advisor. He suggested we studied Ant Colony optimization, and I truly enjoyed it. With my undergraduate thesis, my interest in AI sparked, and so I am genuinely grateful to Ignacio for his guidance.

During a time of turmoil in Venezuela, a keen sense of adventure and some luck landed us at Indiana University to pursue graduate education. Living outside one's home country for the first time is not easy. Indiana University was our home for four beautiful years, and there I met Dr. Esfandiar Haghverdi. He became my mentor and friend while pursuing graduate studies at the Mathematics department. I am grateful for Dr. Haghverdi's continued guidance and friendship.

During my Ph.D. I was thrice a research scientist intern for Amazon. Each of these opportunities was amazing and allowed me to peek at the complexity of modern companies. I also met absurdly talented people, including my mentor Dr. Jayash Koshal. Jayash has tremendous technical talent, but more importantly, he deeply cares for the success of others. Without his guidance, my internships would have been significantly more challenging. I am thankful for his mentoring and friendship.

I would also like to thank my dissertation committee members: Dr. George Konidaris, Dr. Benjamin Lubin, and Dr. Serdar Kadioglu. They each set aside significant portions of their valuable time to advise me during my time at Brown. I learned a great deal from them, and without them, my experience at Brown would not have been as wonderful as it was.

I am eternally in debt to a wonderful group of collaborators with whom I truly enjoyed tackling difficult research questions: Dr. Cyrus Cousins, Dr. Yasser Mohammad, Marilyn George, and Denizalp Goktas. I am especially in debt to Cyrus, who contributed much of the formalism that resulted in several joint publications and eventually in this thesis. Thank you, Cyrus.

The following professors helped me a great deal through grad school. It is not exaggerated to say that without their help, sometimes unbeknownst to them, finishing my Ph.D. would not have been possible or at least significantly more challenging: Dr. Chris Judge, Dr. Geoffrey Brown, Dr. Hernan Awad, Dr. Luis Rocha, and Dr. Melanie Wu.

I would also like to thank friends that made a positive impact in my life and were there for me through happy and challenging times: Dr. Dave Abel, Dr. Archita Argawal, Dr. Kavosh Asadi, Jose Alfredo De Bastos, Joshua Blevins, Lauren Clarke, Luke Camery, Dr. Suyog Chandramouli, Josh Cusano, Hani Dawoud, Hao Li, Dr. Patricia Medina, Takehiro Oyakawa, Patricia Reyes-Cooksey, Dr. Shiva Sinha, Dr. Chris Tanner, Diego Vargas, Jose Villalta, and Dr. Bryce Weaver. I also want to make a special mention to Carlos Castro, whose life journey is a steady source of inspiration.

Without my family any achievement would seriously lack any meaning: Karina, Calena, Pepe, Papa, Mama, Edu, Mariana, La Nata, Kathy, and Enrique (Gruber). My sincere hope is that we get to celebrate this and so many other achievements we have yet to celebrate in a free and prosperous Venezuela.

Finally, much has been said about how hard and challenging it is to complete a Ph.D. I won't repeat any of that here. What I do want to say is that, without my wife Karina, things would have been very different in at least two ways. First, I would have probably had a much harder time in already tiring times as I navigated the ups and downs of the Ph.D. program. Second, and more importantly, Karina kept me grounded but also gave me the freedom to pursue my dreams.

Karina, I look forward to continuing to build a family together that we can both come home to every single day. Te amo.

Glossary

AI	*Artificial intelligence*
AdX	*Advertisement exchange*
BCOwNM	*Budget-constrained optimization with noisy measurements*
BO	*Bayesian optimization*
BRG	*Better response graph*
CE	*Competitive equilibrium* (synonym of *WE*)
CM	*Combinatorial market*
EGTA	*Empirical game-theoretic analysis*
EMD	*Empirical mechanism design*
FAP	*First-price auction*
GP	*Gaussian process*
GS	*Global sampling*
LEFP	*Limited envy-free pricing*
NCM	*Noisy combinatorial markets*
OwNM	*Optimization with noisy measurements*
PAC	*Probably approximate correct*
PSP	*Progressive sampling with Pruning*
pySEGTA	*Python library for statistical EGTA*
SCC	*Strongly connected component*
TAC	*Trading agent competition*
WE	*Walrasian equilibrium* (synonym of *CE*)

Code Repositories [1]

Project	*URL*	*Brief Description*
EGTA, Chapter 3	github.com/eareyan/pysegta	EGTA Python library.
EMD, Chapter 4	github.com/eareyan/emd-adx	EMD Python library.
AdX Simulator, Chapter 5	github.com/eareyan/adxgame	AdX one-day game Java simulator. Capability of running online tournaments.
AdX Bidding, Chapter 5	github.com/eareyan/envy-free-prices	Combinatorial markets Java library.
NCM, Chapter 6	github.com/eareyan/noisyce	Noisy combinatorial markets Python library.

[1] Permanent links to all repos can be found at `www.enriqueareyan.com/phd`

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The analysis of systems inhabited by multiple strategic agents has a long-standing tradition dating back to the 1700s with the discovery of what we know today as a mixed strategy solution for the French card game *le Hère* [18]. In these kinds of systems, multiple agents make choices whose consequences will −*crucially*– affect not only themselves, but other agents as well. The analysis of such systems could take many forms, but our focus here is on a particular kind of analysis embodied by *game theory*. The goal of a game-theoretic analysis is to *solve* multi-agent systems, where solving a system means to find or characterize its *equilibria*, i.e., the steady states where each agent in the system has no incentive to behave otherwise.

More concretely, game theory refers to particular kinds of mathematical models used to analyze multi-agent systems. Its modern treatment is rooted in the seminal work of Von Neumann and Morgenstern [127]. At the heart of this treatment is the theoretical notion of a *game*. In a game, each player (also referred to as an *agent*) chooses a strategy from a set of strategies and earns a utility which, in general, depends on the profile (i.e., vector) of strategies chosen by all the agents. In a traditional game-theoretic analysis, the analyst has access to a complete description of the game of interest, including the number of players, their strategy sets, and their utilities. Moreover, stylized assumptions are often made about the strategic situation so that the ensuing game lends itself to theoretical analysis. For example, in analyzing auctions, one might assume that bidders have quasilinear utilities on prices [51, 73, 135, 137], and then proceed to solve for equilibria, with the hope of arriving at closed-form solutions [51, 89, 127].

1.1 Empirical Game-Theoretic Analysis

More recently, driven by the pervasive use of modern computational devices and electronic networks [135, 137], researchers have developed methodologies to analyze multi-agent systems for which a complete game-theoretic characterization is either too expensive or too difficult to obtain. First, agents usually have access to large strategy spaces. These spaces are often exponential in some natural parameterization of the system, and thus, a naïve game-theoretic analysis becomes quickly intractable. Moreover, even if strategy spaces are tractable, numerous stochastic elements that interact in complex ways impede computing players' payoffs in closed-form. To make matters worst, many multi-agent systems of interest have both exponential strategy spaces and complex stochastic elements. Hence, the need for methodologies to analyze them efficiently.

One such methodological effort is dubbed empirical game-theoretic analysis (EGTA) [134]. An EGTA of a multi-agent system takes as input a system's simulator. Fixing the agents' strategic choices, the simulator yields samples of agents' payoffs by simulating the system's stochastic elements. The goal then is to analyze the equilibria behavior of a system's game-theoretic model. These game models are known in the literature as both *simulation-based games* [129] and *black-box games* [100], precisely because one only gets to observe samples of utility functions but never a complete analytical description of them.

Figure 1.1 illustrates, at a high level, a simulation-based game. On the left, the figure shows a (tabular) normal-form game that defines the number of players (in this case two: the row player and the column player) and the strategies available to them (say m strategies, $S_1^{row}, \ldots, S_m^{row}$, for the row player and n strategies, $S_1^{col}, \ldots, S_n^{col}$, for the column player). Note that the utilities of this game are not directly observable. Instead, on the right, the figure shows a game's simulator, depicted as a black-box, which takes as input a strategy profile (one choice of strategy for each player, S_i^{row}, S_j^{col}) and outputs samples of utilities for that profile. In the figure, each sample is a list with two entries, with each entry corresponding to the observed utility for one of the two players.

EGTA methodology has been applied in a variety of practical settings for which simulators are readily available. Some of these settings include trading agent analyses in supply chains [66, 130, 136], ad auctions [65], ad exchanges [116, 125], and energy markets [70]; designing network routing protocols [138]; adversarial planning [113]; strategy selection in real-time games [120]; and the dynamics of RL algorithms, like AlphaGo [121].

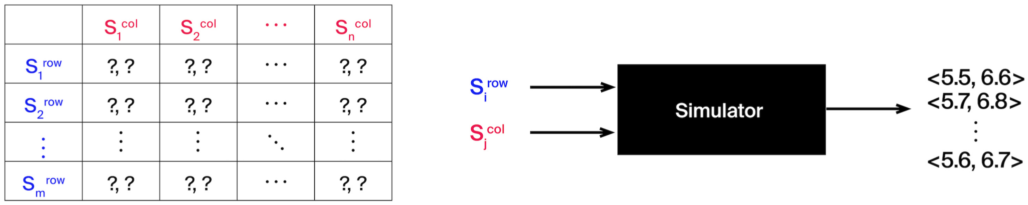


Figure 1.1: A simulation-based game consists of a set of players and strategies available to players (two-player game, left), and a game's simulator depicted here as a black-box (right).

In this thesis, we first contribute to the EGTA literature both from a theoretical and a computational point of view [6, 7, 8]. Theoretically, we present a mathematical framework[1] to precisely describe simulation-based games and analyze their properties. We provide results that complement and strengthen previous results on guarantees of the approximate Nash equilibria learned from samples. We also present novel approximation results for a relaxation of sink equilibria that considers only the strongly connected components of a game's best-response graph. Computationally, we find and thoroughly evaluate *Probably Approximate Correct* (PAC) learning algorithms, which we show make frugal use of data to provably solve simulation-based games, up to a user's given error tolerance with high probability.

Producing a robust empirical evaluation of any EGTA methodology is, at first, a daunting task, as the space of possible games is vast. Moreover, computing equilibria is computationally intractable [39] even for games in closed form, so one can only employ heuristics without run-time guarantees [54, 55, 71, 102]. Fortunately, researchers have devised tools to address these issues. The first, GAMUT, is a state-of-the-art suite of game generators capable of producing a myriad of games with rich strategic structure [93]. The second, Gambit, is a state-of-the-art solver to compute Nash equilibria [86] where possible. We use both GAMUT and Gambit to evaluate the performance of our algorithms. Furthermore, to allow other researchers to build on our work, we developed a Python library called pySEGTA, for *statistical* EGTA[2]. Our library, pySEGTA, interfaces with both GAMUT and Gambit, exposing simple interfaces by which users can generate games (GAMUT), learn them (via our learning algorithms, for example), and solve them (Gambit).

[1]This mathematical framework was developed jointly with fellow Ph.D. student Cyrus Cousins.
[2]pySEGTA is publicly accessible at http://github.com/eareyan/pysegta.

1.2 Empirical Mechanism Design

Game-theoretic models of multi-agent systems assume that the rules of interaction among agents are fixed and known in advance to all agents[3]. The goal then is to solve a multi-agent system by describing the equilibria of a game that models it. The equilibria of a game is a solution in the sense that it serves as a prediction(s) of the state(s) one can reasonably hope the system to arrive at after agents have had a chance to interact. A natural consideration at this point is whether one can *design* a game whose ensuing equilibria are desirable according to some well-defined metric. The design of such games is the object of study in *mechanism design* [26].

Examples of mechanism design problems abound. Here, we will mention just a few. Perhaps the best-known example of a mechanism design problem from the microeconomics literature is the problem of auction design [73]. When designing an auction, an auctioneer might want to maximize the welfare of all participants [33, 57, 124] or might want to maximize only its own revenue [88]. Another example from the microeconomics literature is the problem of designing negotiation protocols [111] by which self-interested parties will interact to reach an agreement, usually prescribing the division or exchange of goods or services. A broad class of problems known as matching problems also falls into the category of mechanism design. In these problems, a set of scarce resources must be allocated among agents, each of which has different (and often conflicting) preferences over the resources [107, 108]. Concrete examples include the design of college admission systems [47], the assignment of medical students to hospitals [106], etc.

Like in traditional game-theoretic analysis, economists have traditionally approached mechanism design problems by making significant simplifying assumptions about the environment for which they are designing the mechanism, and then proceeding to study its analytical properties. While said simplifying assumptions serve an essential role in our understanding of multi-agent systems, it can be the case that the resulting analysis does not match reality, resulting in frail (i.e., non-robust) mechanisms [37, 97].

In an effort to find more robust mechanisms, researchers have turned their attention to statistical and algorithmic methodologies and tools to design mechanisms. One such effort is dubbed *empirical mechanism design* (EMD) [98, 99, 130][4]. Like traditional mechanism design, in EMD, a designer

[3]And that agents know that other agents know all that they know. In turn, other agents know this, and so on. This assumption is known as the *common knowledge* assumption [141].

[4]Not to be confused with *automated mechanism design*, see related work in Section 2.2

wishes to design systems where the behavior of participants leads to desirable outcomes, as measure by some well-defined metric. Unlike traditional mechanism design, in EMD, the designer relies solely on gameplay observations. Thus, EMD itself relies on EGTA methodologies. In particular, the EMD methodology we present in this thesis relies on our EGTA methodology (Section 1.1). Still, it could be extended to incorporate other available EGTA methodologies, albeit perhaps not with the same guarantees. Figure 1.2 shows a schematic view of EMD.

The design of mechanisms via EMD is challenging as the space of potential mechanisms is vast. To gain some traction, we concentrate on parametric EMD [125, 130]. In parametric EMD, there is an overall (parameterized) mechanism, e.g., a second price auction with a reserve price as a parameter. The choice of parameters then determines a particular instance of this mechanism, which we refer to simply as a mechanism, e.g., a second price auction with a reserve price of $10. This simplification allows us to contribute to the literature on EMD first, from a theoretical point of view, by formulating the problem of finding the optimal parameters of a mechanism as a black-box optimization problem. While for many mechanisms of interest, the space of parameters remains vast, so that it would seem like we have not made not much progress, our second contribution is a Bayesian optimization algorithm that can search this space efficiently in practice.

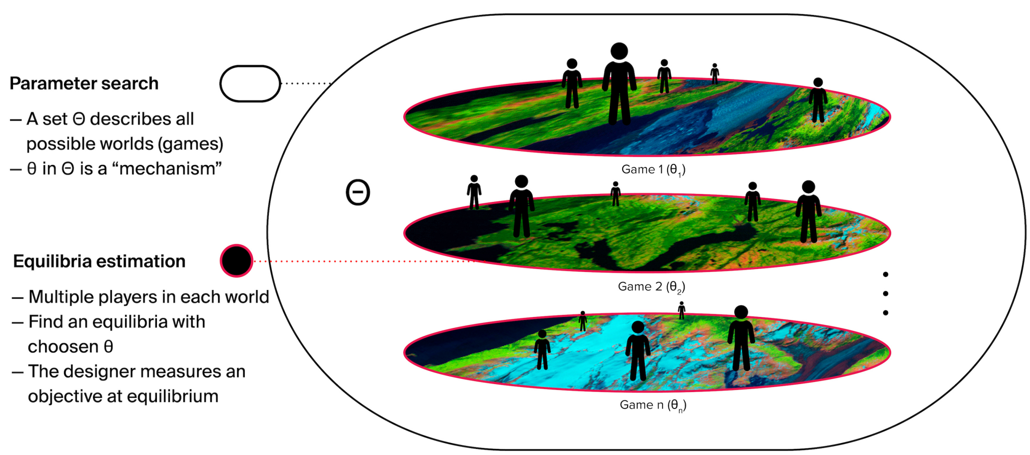


Figure 1.2: An Schematic view of empirical mechanism design. A mechanism designer has access to Θ, a set of available mechanisms where each element $\theta \in \Theta$ (e.g., different immigration policies) defines a mechanism (an immigration policy). Associated with each mechanism θ, there is a game, Game(θ), whose utilities are accessible only through (possibly noisy) observations. The designer's goal is then to select θ^* that maximizes some objective function $f : \Theta \to \mathbb{R}$ (e.g., f might be the population's welfare). The designer assumes that players reach an (approximate) equilibrium for any Game(θ) (e.g., after observing θ, each member of the population decides where to immigrate).

We thoroughly evaluate our Bayesian optimization algorithm both in simple settings where the optimal mechanism can be derived analytically and in richer settings closer to systems deployed in practice for which an optimal mechanism is not readily available. We first show that our algorithm recovers optimal mechanisms when they are known more efficiently than existing baselines. We also show that our algorithm can produce higher-quality mechanisms more efficiently than standard baselines in richer settings.

To evaluate our EMD methodology, which itself uses our EGTA methodology, we study a rich model of electronic advertisement auctions. Concretely, we investigate the problem of finding revenue-maximizing reserve prices as an example of a task a mechanism designer might want to undertake. But, observe that the mechanisms derived from any EMD methodology are only as effective as the set of heuristic strategies used by the participating agents when optimizing the mechanisms' parameters. Consequently, to demonstrate the effectiveness of our methods, we also present work leading to the development of bidding heuristics[5] for electronic advertisement auctions.

1.3 Heuristic Bidding for Electronic Ad Auctions

Digital advertising earnings in the U.S. keep reaching new highs, a recent one being $57.9 billion during the first six months of 2019, the highest earnings in history for the first semester of the year [63]. Central to the functioning of this massive market are advertisement (ad) exchanges and on-line ad networks. An *ad exchange* is a centralized platform that promotes the buying and selling of on-line ads, usually through the use of auctions. An on-line *ad network* is a company that serves as an intermediary between advertisers and websites that publish ads.

Digital ads can serve a variety of purposes, e.g., they can persuade customers to buy a product directly [2], or they can create and maintain brand awareness for future sales [1]. The majority of ad revenue stems from ads displayed alongside web content, aimed at creating and maintaining brand awareness [83]. In this thesis, we consider the challenge faced by ad networks as they attempt to fulfill brand awareness advertisement campaigns through ad exchanges.

An *advertising campaign* is a contract between an advertiser and an ad network. In this contract, the ad network commits to displaying at least a certain number of ads on behalf of the advertiser to users of specific demographics in exchange for a fixed budget that is set beforehand.

[5] Our EMD algorithms and heuristics are publicly available in a Python library at https://github.com/eareyan/emd-adx.

Given a campaign, the challenge for an ad network is to procure enough display ad *impression opportunities* to fulfill the campaign at the lowest cost possible in the face of competition from other ad networks. We call this problem the *ad network problem*.

We present two bidding heuristics for the ad network problem: (1) a market equilibrium-based approach, and (2) an auction-simulation approach. We used our heuristics based on approaches (1) and (2) in an extensive set of experiments to show that our EMD methodology can find (near) revenue-maximizing reserve prices. In these experiments (section 4.5), applying our EMD methodology means that, for a fixed set of reserve prices, we first use our EGTA methodology to find (near) equilibria in a game where players are allowed to switch between our two heuristics. For each such game, the designer's (i.e., auctioneer's) objective function selects the worst (i.e., lowest) revenue among all (near) equilibria states. Now, regarding the auctioneer's objective function as a black-box that maps a set of reserve prices to the worst-revenue at (near) equilibria, the goal is to find a set of revenue-maximizing prices. For this purpose, we enhance standard search routines for black-box optimization problems to include piecewise constant noise (section 4.3.2), precisely the kind of noise that characterizes PAC learners, which are fundamental to our EGTA methodology.

Ad exchanges are examples of *combinatorial markets* since impression opportunities cannot be divided among multiple agents. More generally, in a combinatorial market, goods cannot be divided, but instead, each must be either wholly allocated to a single consumer or not allocated at all. As the final contribution of this thesis, we extend our EGTA methodology to the problem of learning equilibria in combinatorial markets, given limited access to consumers' valuations for goods.

1.4 Noisy Combinatorial Markets

Combinatorial Markets (CMs) are a class of markets in which buyers are interested in acquiring bundles of goods for which their values can be arbitrary. Real-world examples of CMs include: spectrum auctions [36] allocation of landing and take-off slots at airports [15]; internet ad placement [41]; and procurement of bus routes [29]. An outcome of a CM is an assignment of bundles to buyers together with prices for the goods. A competitive equilibrium (CE) is an outcome of particular interest in CMs and other well-studied economic models [21, 132]. In a CE, buyers are utility-maximizing (i.e., they maximize their utilities among all feasible allocations at the posted prices) and the seller maximizes its revenue (again, over all allocations at the posted prices).

While CEs are a static equilibrium concept, they can sometimes arise as the outcome of a dynamic price adjustment process (e.g., [32]). In such a process, prices might be adjusted by an imaginary auctioneer, who poses demand queries to buyers: i.e., asks them their demands at given prices. Similarly, we imagine that prices in a CM are set by a market maker, who poses value queries to buyers: i.e., asks them their values on select bundles.

One of the defining features of CMs is that they afford buyers the flexibility to express complex preferences, which in turn has the potential to increase market efficiency. However, the extensive expressivity of these markets presents challenges for both the market maker and the buyers. With an exponential number of bundles in general, it is infeasible for a buyer to evaluate them all. We thus present a model of noisy buyer valuations: e.g., buyers might use approximate or heuristic methods to obtain value estimates [45]. In turn, the market maker chooses an outcome in the face of uncertainty about the buyers' valuations. We call these markets *noisy combinatorial markets* (NCM) to emphasize that buyers do not have direct access to their values for bundles, but instead can only noisily estimate them.

In this thesis, we formulate a mathematical model of NCMs. Our goal is then to design learning algorithms with rigorous finite-sample guarantees that approximate the competitive equilibria of NCMs. First, we present tight lower- and upper-bounds on the set of CE, given uniform approximations of buyers' valuations. We then present two learning algorithms. The first one—Elicitation Algorithm; EA—serves as a baseline. It uses Hoeffding's inequality [62] to produce said uniform approximations. Our second algorithm—Elicitation Algorithm with Pruning; EAP—leverages the first welfare theorem of economics to adaptively prune value queries when it determines that they are provably not part of a CE.

After establishing the correctness of our algorithms, we evaluate their empirical performance using both synthetic unit-demand valuations and two spectrum auction value models. The former are a class of valuations central to the literature on economics and computation [79], for which there are efficient algorithms to compute CE [58]. In the spectrum auction value models, the buyers' valuations are characterized by complements, which complicate the questions of existence and computability of CE. In all three models, we measure the average quality of learned CE via our algorithms, compared to the CE of the corresponding certain market (i.e., here, "certain" means lacking uncertainty), as a function of the number of samples. We find that EAP often yields better error guarantees than EA using far fewer samples, because it successfully prunes buyers' valuations (i.e., it ceases querying for buyers' values on bundles of goods that a CE provably does not comprise), even without any *a priori* knowledge of the market's combinatorial structure.

An interesting tradeoff arises between computational and sample efficiency. To prune a value query and still retain rigorous guarantees on the quality of the learned CE, we must solve a welfare-maximizing problem whose complexity grows with the size of the market. Consequently, at each iteration of EAP, for each value query, we are faced with a choice. Either solve said welfare-maximizing problem and potentially prune the value query (thereby saving on future samples), or defer attempts to prune the value query, until more is known about the market. To combat this problem, we show that an upper bound on the optimal welfare's value (rather than the precise value) suffices to obtain rigorous guarantees on the learned CE's quality. Such upper bounds can be found easily, by solving a relaxation of the welfare-maximization problem. Reminiscent of designing admissible heuristics in classical search problems, this methodology applies to any combinatorial market, but at the same time allows for the application of domain-dependent knowledge to compute these upper bounds, when available. Empirically, we show that a computationally cheap relaxation of the welfare-maximization problem yields substantial sample and computational savings. We experiment with a large market that models a scenario in which buyers increase their values provided they obtain bundles that are spacially close together, a common situation in spectrum auctions.

Thesis roadmap. The rest of this thesis is outlined as follows. In the next chapter (chapter 2), we review prior work. In chapter 3, we present our contributions to EGTA, and in chapter 4, our contributions to EMD. To evaluate our EMD methodology, we developed bidding heuristics for the ad network problem, which we detail in chapter 5. Chapter 6 presents our contributions to the statistical learning of competitive equilibria in combinatorial markets. We conclude and point to future research directions in chapter 7.

Chapter 2

Prior Work

In this chapter, we first review prior work on empirical game-theoretic analysis, followed by prior work on empirical mechanism design. Then, we review prior work on electronic advertisement markets, which serves as our primary application throughout. We conclude with a summary of literature concerning the learning of competitive equilibria in combinatorial markets.

2.1 Prior Work: Empirical Game-Theoretic Analysis

The EGTA literature, while relatively young, is growing rapidly, with researchers actively contributing methods for myriad game models. Some of these methods are designed for normal-form games [120, 129], and others, for extensive-form games [49, 84, 142]. Most of these methods apply to games with finite strategy spaces, but some apply to games with infinite strategy spaces [84, 131, 140]. Our methodology applies to normal-form games with either finite or infinite strategy spaces.

We take [134] as our starting point from a methodological point of view. The author proposes a high-level, architectural view of EGTA made up of three components: (1) parameterization of strategy spaces, (2) estimation of empirical games, and (3) analysis of empirical games. To illustrate the methodology, he deploys it in the Supply Chain Management game of the Trading Agent Competition (TAC/SCM) game [66]. In this thesis, we present contributions to (2) and (3). An exciting future research direction is to extend our methodologies to address problems related to the parameterization of strategy spaces.

To analyze simulation-based games (component (3)), one first defines a notion of *approximate* equilibrium amenable to statistical estimation. Most prior work centers around ε-Nash equilibrium [66, 67, 120, 121, 122, 130], an approximation of Nash equilibrium up to an additive error of ε, following the long-standing tradition of using Nash equilibria as the *de facto* solution concept to analyze non-cooperative games [51]. We follow this tradition.

Having defined the notion of equilibria of interest, the *search* problem is now a combination of components (2) and (3), where an algorithm both estimates and analyzes an empirical game *simultaneously*. The search space in EGTA refers to the space of strategy profiles. Jordan et al. [67] make the interesting distinction between two models of search: (a) the *revealed-utility* model, where each search step determines the exact utility for a designated pure-strategy profile; and (b) the *noisy-utility* model, where each search step draws a stochastic sample corresponding to such a utility. Our work falls into model (b).

Most EGTA methodologies share the same goal, namely, to estimate an equilibrium of the game (except for a few notable exceptions [121, 128, 139]). In this thesis' EGTA methodology, we aim to bound the regret over all equilibria of a simulation-based instead of one equilibrium. Hence, strictly speaking, we do not directly address the search problem as previously stated. Nonetheless, we tackle an analogous search problem for our setting. Specifically, our problem is to minimize the number of queries to the game's simulator while maintaining robust guarantees over all the game's equilibria. Towards this end, we contribute two learning algorithms: global sampling (GS), and *progressive sampling with pruning* (PSP). Global sampling uniformly samples all utilities of a simulation-based game to obtain, with high probability, a desired degree of accuracy. Progressive sampling with pruning dynamically allocates samples to minimize the number of queries to the game's simulator.

In contexts other than EGTA, prior art [42, 104, 105] has used progressive sampling to obtain a desired accuracy given a failure probability by guessing an initial sample size, computing (statistical) bounds, and repeating with larger sample sizes until said accuracy is attained. Our work applies this idea to EGTA and complements it with *pruning*: at each iteration, utilities of strategy profiles that have already been sufficiently well estimated for the task at hand (here, equilibrium estimation) are pruned as subsequent iterations of PSP do not refine their bounds. Pruning represents both a statistical and computational improvement over earlier progressive sampling techniques.

Finally, is worth mentioning that EGTA methodologies have been applied in a variety of practical settings. Some of these include trading agent analyses in supply chains [66, 130, 136], ad auctions [65], ad exchanges [116, 125], and energy markets [70]; designing network routing protocols [138]; adversarial planning [113]; strategy selection in real-time games [120]; and the dynamics of RL algorithms, like AlphaGo [121].

2.2 Prior Work: Empirical Mechanism Design

Mechanism design is concerned with the design of games in which the strategic behavior of participants leads to desired outcomes. Traditional mechanism design lies within the realm of game theory, itself an area of economics. Traditionally, a mechanism designer would first manually design a game, guided by domain-specific expertise, and would then try to prove that the ensuing outcomes satisfy certain criteria [88]. However, with the advent of strategic autonomous agents, particularly in e-commerce settings, there has been a surge of interest in *computational* mechanism design within the computer science community [38, 91, 96, 115].

In the artificial intelligence community, at least two approaches to mechanism design have emerged. The first is *automated mechanism design* [35, 59, 115], where a mechanism is automatically created by an algorithm that searches through a space of mechanisms constrained by standard mechanism design criteria, such as individual rationality and incentive compatibility. The second is *empirical mechanism design* (EMD) [130], where the designer is interested in optimizing a mechanism's parameters (e.g., reserve prices in auctions) relative to some objective function of interest, (e.g., revenue), under the assumption of (e.g., Nash) equilibrium play. Crucially, and like in EGTA, in EMD, we assume no direct access to players' utilities under different choices of mechanism; we instead assume access to a simulator capable of producing samples of utilities. One distinguishing feature of our EMD work *vis à vis* the existing literature is that we formulate the search for a mechanism's optimal parameters as a black-box optimization problem, and then leverage Bayesian optimization techniques to carry out the search.

2.3 Prior Work: Bidding for Electronic Ad Auctions

Our work on bidding heuristics for electronic advertisement markets was initially inspired by the ad exchange (AdX) game [116], one of many games played as part of the Trading Agent Competition (TAC) [137], a tournament in which teams of programmers from around the world build autonomous agents that trade in market simulations. Application domains for these games have ranged from travel [56] to supply chain management [112] to ad auctions [65] to energy markets [70].

In the TAC AdX game, agents representing ad networks bid to fulfill ad campaigns, which they do by acquiring opportunities to show impressions to users from different demographics. Bidding repeats over a series of simulated days, and agents' reputations for successfully fulfilling past campaigns impact their ability to procure future campaigns. Our focus is on the ad network problem only: i.e., how to bid to most profitably fulfill one campaign. Hence, we define a one-day AdX game, consisting of only a single simulated day. Nonetheless, ours is a complex game. In the simulation, auctions are held repeatedly over the course of the day, as users visit web sites. Each such user/impression opportunity is allocated via a second-price auction with reserve. But agents submit their bids upfront, and cannot change them over the course of the day. At decision time, agents know only the distribution and the expected number of users of each demographic, not the particular realizations. The richness of this game implies that it is unlikely to be solved analytically. Hence, we devise bidding heuristics for it.

The market induced by the one-day AdX game description has indivisible goods since impression opportunities are either entirely allocated to an agent or not. One of our bidding heuristics consists of computing a competitive equilibrium of said market, using it as a prediction for allocation and prices, and bidding accordingly. The study of competitive equilibria has a long standing tradition, starting from the seminal work of French economist Léon Walras [132]. In his work, Walras considered divisible goods, i.e., goods that can be allocated to consumers in fractional amounts. More recently, authors have extended Walras' work to markets with multiple, indivisible goods [21, 46, 68]. These markets are a class of markets where buyers are interested in acquiring packages or bundles of goods. These markets underlie a variety of mechanisms that are fundamental to modern economies. Some examples of such mechanisms include: spectrum auctions, of which the 2014 Canadian 700 MHZ raised upwards of $5 billion [133], allocation of landing and take-off slots at airports [15], Internet ad placement [41], procurement of bus routes [29], among others [40, 95].

2.4 Prior Work: Noisy Combinatorial Markets

The inspiration for the model of noisy combinatorial markets introduced in this thesis stemmed from the work on abstraction in Fisher markets by Kroer et al. [74]. There, the authors tackle the problem of computing equilibria in large markets by creating an abstraction of the market, computing equilibria in the abstraction, and lifting those equilibria back to the original market. Likewise, we develop a pruning criterion which in effect builds an abstraction of any combinatorial market. Given this abstraction, we then compute a competitive equilibrium (CE) of it, which we show is provably also an approximate CE in the original market.

Jha and Zick [64] have also tackled the problem of learning CE in combinatorial markets[1]. Whereas our approach is to accurately learn only those components of the buyers' valuations that determine a CE (up to PAC guarantees), their approach bypasses the learning of agent preferences altogether, going straight for learning a solution concept, such as a CE. It is an open question as to whether one approach dominates the other, in the context of noisy combinatorial markets.

Another related line of research is concerned with learning valuation functions from data [13, 14, 78]. In contrast, our work is concerned with learning buyers' valuations only in so much as it facilitates learning CE. Indeed, our main conclusion is that CE often can be learned from just a subset of the buyers' valuations.

There is also a long line of work on preference elicitation in combinatorial auctions (e.g., [34]), where an auctioneer aims to pose value queries in an intelligent order so as to minimize the computational burden on the bidders, while still clearing the auction.

Whereas intuitively, a basic pruning criterion for games is arguably more straightforward—simply prune dominated strategies—the challenge in this case was to discover a pruning criterion that would likewise prune valuations that are provably not part of a CE. Our pruning criterion relies on a novel application of the first welfare theorem of economics. While prior work has connected economic theory with algorithmic complexity [110], this work connects economic theory with statistical learning theory.

[1] The market structure they investigate is not identical to the structure studied here. Thus, at present, our results are not directly comparable.

Chapter 3

Empirical Game-Theoretic Analysis

In this chapter, we present our contributions to the empirical game-theoretic analysis literature. The contents of this chapter are an extended version of the following published papers: *Learning Simulation-Based Games from Data* [7] (arXiv version *Learning Equilibria of Simulation-Based Games* [6]), and *Improved Algorithms for Learning Equilibria in Simulation-Based Games* [8].

3.1 Game Theory Background

We first offer some necessary background on game theory. Central to our work is the fundamental notion of a normal-form game, which we formally define next.

Definition 1 (Normal-form game)

A *normal-form game* $\Gamma \doteq \langle P, \{S_p \mid p \in P\}, \boldsymbol{u}(\cdot)\rangle$ consists of a finite set of agents P, with *pure strategy set* S_p available to agent $p \in P$. We define $\boldsymbol{S} \doteq S_1 \times \cdots \times S_{|P|}$ to be the pure strategy profile space of Γ, and then $\boldsymbol{u} : \boldsymbol{S} \to \mathbb{R}^{|P|}$ is a vector-valued utility function (equivalently, a vector of $|P|$ scalar utility functions $\boldsymbol{u}_p$).

Given a normal-form game Γ, we let $S_p^\diamond$ denote the set of distributions over S_p; this set contains p's *mixed strategies*. We define $\boldsymbol{S}^\diamond = S_1^\diamond \times \cdots \times S_{|P|}^\diamond$, and then, overloading notation, we write $\boldsymbol{u}(\boldsymbol{s})$

to denote the expected utility of a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^{\diamond}$.

A solution to a normal-form game is a prediction of how strategic agents will play the game. One solution concept that has received a great deal of attention in the literature is Nash equilibrium [90], a (pure or mixed) strategy profile at which each agent selects a utility-maximizing strategy, fixing all other agents' strategies. In our work, we are concerned with ε-Nash equilibrium, an approximation of Nash equilibrium that is amenable to statistical estimation.

Given a game Γ, fix an agent p and a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^{\diamond}$. Let $\boldsymbol{T}^{\diamond}_{p,\boldsymbol{s}}$ be the set of all mixed strategy profiles in which the strategies of all agents $q \neq p$ are fixed at $\boldsymbol{s}_q$. Mathematically, $\boldsymbol{T}^{\diamond}_{p,\boldsymbol{s}} \doteq \{\boldsymbol{t} \in \boldsymbol{S}^{\diamond} \mid \boldsymbol{t}_q = \boldsymbol{s}_q, \forall q \neq p\}$.

Definition 2 (Regret)

Given a game Γ with utility $\boldsymbol{u}$, fix an agent p and a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^{\diamond}$. We define $\mathrm{Adj}^{\diamond}_{p,\boldsymbol{s}} \doteq \{\boldsymbol{t} \in \boldsymbol{S}^{\diamond} \mid \boldsymbol{t}_q = \boldsymbol{s}_q, \forall q \neq p\}$. In words, $\mathrm{Adj}^{\diamond}_{p,\boldsymbol{s}}$ is the set of adjacent mixed strategy profiles: i.e., those in which the strategies of all agents $q \neq p$ are fixed at $\boldsymbol{s}_q$. Agent p's *regret* at $\boldsymbol{s}$ is defined as: $\mathrm{Reg}^{\diamond}_p(\boldsymbol{u}, \boldsymbol{s}) \doteq \sup_{\boldsymbol{s}' \in \mathrm{Adj}^{\diamond}_{p,\boldsymbol{s}}} \boldsymbol{u}_p(\boldsymbol{s}') - \boldsymbol{u}_p(\boldsymbol{s})$. By restricting $\boldsymbol{s}$ and $\mathrm{Adj}^{\diamond}_{p,\boldsymbol{s}}$ to pure strategy profiles, agent p's *pure regret* $\mathrm{Reg}_p(\boldsymbol{u}, \boldsymbol{s})$ can be defined similarly, with respect to $\mathrm{Adj}_{p,\boldsymbol{s}}$.

Note that $\mathrm{Reg}^{\diamond}_p(\boldsymbol{u}, \boldsymbol{s}), \mathrm{Reg}_p(\boldsymbol{u}, \boldsymbol{s}) \geq 0$, since agent p can deviate to any strategy $s' \in S_p$, including $\boldsymbol{s}_p$ itself. A strategy profile $\boldsymbol{s}$ that has regret at most $\varepsilon \geq 0$, for all $p \in P$, is an ε-Nash equilibrium:

Definition 3 (ε-Nash equilibrium)

Given $\varepsilon \geq 0$, a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^{\diamond}$ in a game Γ is an ε-*Nash equilibrium* if, for all $p \in P$, $\mathrm{Reg}^{\diamond}_p(\boldsymbol{u}, \boldsymbol{s}) \leq \varepsilon$. At a pure strategy ε-Nash equilibrium $\boldsymbol{s} \in \boldsymbol{S}$, for all $p \in P$, $\mathrm{Reg}_p(\boldsymbol{u}, \boldsymbol{s}) \leq \varepsilon$. We denote by $\mathrm{E}^{\diamond}_{\varepsilon}(\boldsymbol{u})$ the set of mixed ε-Nash equilibria, and by $\mathrm{E}_{\varepsilon}(\boldsymbol{u})$, the set of pure ε-Nash equilibria. Note that $\mathrm{E}_{\varepsilon}(\boldsymbol{u}) \subseteq \mathrm{E}^{\diamond}_{\varepsilon}(\boldsymbol{u})$.

3.2 A Framework for EGTA

Our first contribution towards a complete framework for EGTA is to show that the equilibria of a game can be approximated with bounded error, given only a uniform approximation of the game's utility functions. Specifically, our theorem establishes perfect recall[1]: the approximate game contains all true positives: i.e., all (exact) equilibria of the original game. It also establishes approximately perfect precision: all false positives in the approximate game are approximate equilibria in the original game.

Building towards this result, we first define the metric by which we will measure the error between games, i.e., the ℓ_∞*-norm*. The error measurement holds only between games with the same agents sets P and strategy profile spaces, but whose utilities functions are not necessarily the same. We call such games *compatible games*. Later on, we will see that associated with a ground-truth game (whose utility functions we do not get to observe), there are empirical games (whose utility functions we get to observe via samples). An empirical game is compatible with its ground-truth counterpart, and thus, it makes sense to measure the error between them.

Definition 4 (ℓ_∞*-norm* of games)

We define the ℓ_∞*-norm* between two compatible games, with the same agents sets P and strategy profile spaces $\boldsymbol{S}$, and with utility functions $\boldsymbol{u}$, $\boldsymbol{u}'$, respectively, as follows:

$$\|\Gamma - \Gamma'\|_\infty \doteq \|\boldsymbol{u}(\cdot) - \boldsymbol{u}'(\cdot)\|_\infty \doteq \sup_{p \in P, \boldsymbol{s} \in \boldsymbol{S}} |\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})|$$

While the ℓ_∞-norm as defined applies only to pure normal-form games, it is in fact sufficient to use this metric even to show that the utilities of mixed strategy profiles approximate one another. We formalize this claim in the next lemma.

Lemma 1 (Approximations in Mixed Strategies)**.** *If* Γ, Γ' *are compatible games, i.e., they differ only in their utility functions,* $\boldsymbol{u}, \boldsymbol{u}'$*, then*

$$\sup_{p \in P, \boldsymbol{s} \in \boldsymbol{S}^\diamond} |\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})| = \|\Gamma - \Gamma'\|_\infty$$

Proof. For any agent p and mixed strategy profile $\boldsymbol{t} \in \boldsymbol{S}^\diamond$, $\boldsymbol{u}_p(\boldsymbol{t}) = \sum_{\boldsymbol{s} \in \boldsymbol{S}} \boldsymbol{t}(\boldsymbol{s}) \boldsymbol{u}_p(\boldsymbol{s})$, where $\boldsymbol{t}(\boldsymbol{s}) = \prod_{p' \in P} \boldsymbol{t}_{p'}(\boldsymbol{s}_{p'})$. So, $\boldsymbol{u}_p(\boldsymbol{t}) - \boldsymbol{u}'_p(\boldsymbol{t}) = \sum_{\boldsymbol{s} \in \boldsymbol{S}} \boldsymbol{t}(\boldsymbol{s})(\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})) \leq \sup_{\boldsymbol{s} \in \boldsymbol{S}} |\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})|$, by Hölder's inequality. Hence, $\sup_{\boldsymbol{t} \in \boldsymbol{S}^\diamond} |\boldsymbol{u}_p(\boldsymbol{t}) - \boldsymbol{u}'_p(\boldsymbol{t})| \leq \sup_{\boldsymbol{s} \in \boldsymbol{S}} |\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})|$, from which it follows that

[1] We use the term *recall* in the information retrieval sense. The juxtaposition of the two words "perfect" and "recall" is not a reference to extensive-form games.

$\sup_{p \in P, \boldsymbol{t} \in \boldsymbol{S}^\diamond} |\boldsymbol{u}_p(\boldsymbol{t}) - \boldsymbol{u}'_p(\boldsymbol{t})| \leq \|\Gamma - \Gamma'\|_\infty$. Equality holds for any p and $\boldsymbol{s}$ that realize the supremum in $\|\Gamma - \Gamma'\|_\infty$, as any such pure strategy profile is also mixed. □

Central to our work is the notion of uniform ε-approximation between games. We make this notion precise in the next definition.

Definition 5 (ε-Uniform approximation of games)

Γ' is said to be a *uniform ε-approximation* of Γ when $\|\Gamma - \Gamma'\|_\infty \leq \varepsilon$.

In an ε-uniform approximation of Γ' by Γ, the bound between utility deviations in Γ' and Γ (and, symmetrically, in Γ and Γ') holds *uniformly* over *all* players and strategy profiles. Hence the name *uniform approximation*. We can now present our theorem that characterizes the approximation guarantees on the set of Nash equilibria of a game given an ε-uniform approximation of it. Looking ahead, we rely on this theorem to show good approximations between simulation-based games (definition 7) and their empirical counterparts (definition 8).

Theorem 1 (Approximate Equilibria)**.** *Given normal-form games Γ, Γ', with utility functions $\boldsymbol{u}$ and $\boldsymbol{u}'$, respectively, such that $\|\Gamma - \Gamma'\|_\infty \leq \varepsilon$, the following hold:*

1. $\mathrm{E}(\boldsymbol{u}) \subseteq \mathrm{E}_{2\varepsilon}(\boldsymbol{u}') \subseteq \mathrm{E}_{4\varepsilon}(\boldsymbol{u})$, *and*
2. $\mathrm{E}^\diamond(\boldsymbol{u}) \subseteq \mathrm{E}^\diamond_{2\varepsilon}(\boldsymbol{u}') \subseteq \mathrm{E}^\diamond_{4\varepsilon}(\boldsymbol{u})$.

Proof. First note the following: if $A \subseteq B$, then $C \cap A \subseteq C \cap B$. Hence, since any pure Nash equilibrium is also a mixed Nash equilibrium, taking C to be the set of all pure strategy profiles, we need only show 2. We do so by showing $\mathrm{E}^\diamond_\gamma(\boldsymbol{u}) \subseteq \mathrm{E}^\diamond_{2\varepsilon+\gamma}(\boldsymbol{u}')$, for $\gamma \geq 0$, which implies both containments, taking $\gamma = 0$ for the lesser, and $\gamma = 2\varepsilon$ for the greater.

Suppose $\boldsymbol{s} \in \mathrm{E}^\diamond_\gamma(\boldsymbol{u})$, for all $p \in P$. We will show that $\boldsymbol{s}$ is $(2\varepsilon + \gamma)$-optimal in Γ', for all $p \in P$. Fix an agent p, and recall $\mathrm{Adj}^\diamond_{p,\boldsymbol{s}} \doteq \{\boldsymbol{t} \in \boldsymbol{S}^\diamond \mid \boldsymbol{t}_q = \boldsymbol{s}_q, \forall q \neq p\}$. Now take $\boldsymbol{s}^* \in \arg\max_{\boldsymbol{t} \in \mathrm{Adj}^\diamond_{p,\boldsymbol{s}}} \boldsymbol{u}_p(\boldsymbol{t})$ and $\boldsymbol{s}'^* \in \arg\max_{\boldsymbol{t} \in \mathrm{Adj}^\diamond_{p,\boldsymbol{s}}} \boldsymbol{u}'_p(\boldsymbol{t})$. Then:

$$
\begin{aligned}
\mathrm{Reg}^\diamond_p(\boldsymbol{u}', \boldsymbol{s}) &= \boldsymbol{u}'_p(\boldsymbol{s}'^*) - \boldsymbol{u}'_p(\boldsymbol{s}) \\
&\leq (\boldsymbol{u}_p(\boldsymbol{s}'^*) + \varepsilon) - (\boldsymbol{u}_p(\boldsymbol{s}) - \varepsilon) \\
&\leq (\boldsymbol{u}_p(\boldsymbol{s}^*) + \varepsilon) - (\boldsymbol{u}_p(\boldsymbol{s}) - \varepsilon) \\
&\leq (\boldsymbol{u}_p(\boldsymbol{s}^*) + \varepsilon) - (\boldsymbol{u}_p(\boldsymbol{s}^*) - \varepsilon - \gamma) = 2\varepsilon + \gamma
\end{aligned}
$$

The first line follows by definition. The second holds by Lemma 1 and the fact that Γ' is a uniform ε-approximation of Γ, the third as $\boldsymbol{s}^*$ is optimal for p in Γ, and the fourth as $\boldsymbol{s}$ is a γ-Nash in Γ. □

We note that part of this theorem was concurrently proved by other researchers [121], albeit in a different formalism than ours. We now briefly explain their results in the language of our work. They showed the set containment $\mathrm{E}(\boldsymbol{u}) \subseteq \mathrm{E}_{2\varepsilon}(\boldsymbol{u}')$. We complement this result by further observing that $\mathrm{E}_{2\varepsilon}(\boldsymbol{u}') \subseteq \mathrm{E}_{4\varepsilon}(\boldsymbol{u})$. We also provide a machine learning interpretation in the language perfect recall and approximately perfect precision.

3.2.1 Learning Framework

We move on from approximating equilibria in games to learning them. We first present a learning framework where we define our model of simulation-based games (definitions 6 and 7). We then define empirical games (definition 8) which are the games we ultimately get to observe by sampling a game's simulator. In the next section, we present algorithms that learn so-called empirical games, which comprise estimates of the expected utilities of simulation-based games. We further derive uniform convergence bounds, PAC-style guarantees proving that our algorithms output empirical games that uniformly approximate their expected counterparts, with high probability. By Theorem 1, the equilibria of these empirical games thus approximate those of the corresponding simulation-based games, with high probability.

Definition 6 (Conditional Normal-Form Game)

A *conditional normal-form game* $\Gamma_{\mathcal{X}} \doteq \langle \mathcal{X}, P, \{S_p \mid p \in P\}, \boldsymbol{u}(\cdot) \rangle$ consists of a set of conditions $\mathcal{X}$, a set of agents P, with pure strategy set S_p available to agent p, and a vector-valued conditional utility function $\boldsymbol{u} : \boldsymbol{S} \times \mathcal{X} \to \mathbb{R}^{|P|}$.

In the definition above, it is convenient to imagine a condition $x \in \mathcal{X}$ as pertaining to the set $P \times \boldsymbol{S}$. Given such a condition, $\boldsymbol{u}(\cdot; x)$ yields a standard utility function of the form $\boldsymbol{S} \to \mathbb{R}^{|P|}$, which evaluates as usual, returning the vector $\boldsymbol{u}(\boldsymbol{s}; x)$ when given a strategy profile $\boldsymbol{s} \in \boldsymbol{S}$.

Associated with a conditional normal-form game, we define, for a fixed distribution $\mathscr{D}$ over condition set $\mathcal{X}$, its corresponding expected normal-form game as follows. Finally, we define the empirical normal-form games, which are games obtained by drawing samples from $\mathcal{X}$.

Definition 7 (Expected Normal-Form Game)

Given a conditional normal-form game $\Gamma_{\mathcal{X}}$ together with distribution $\mathscr{D}$, we also define the *expected utility function* $\boldsymbol{u}(\boldsymbol{s};\mathscr{D}) = \mathbb{E}_{x\sim\mathscr{D}}\left[\boldsymbol{u}(\boldsymbol{s};x)\right]$, and the *expected normal-form game* as $\Gamma_{\mathscr{D}} \doteq \langle P, \{S_p \mid p \in P\}, \boldsymbol{u}(\cdot;\mathscr{D})\rangle$.

Definition 8 (Empirical Normal-Form Game)

Given a conditional normal-form game $\Gamma_{\mathcal{X}}$ together with a distribution $\mathscr{D}$ from which we can draw sample conditions $\boldsymbol{X} = (x_1, \ldots, x_m) \sim \mathscr{D}^m$, we define the *empirical utility function* $\hat{\boldsymbol{u}}(\boldsymbol{s};\boldsymbol{X}) \doteq \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{u}(\boldsymbol{s};x_j)$.

The corresponding *empirical normal-form game* is then $\hat{\Gamma}_{\boldsymbol{X}} \doteq \langle P, \{S_p \mid p \in P\}, \hat{\boldsymbol{u}}(\cdot;\boldsymbol{X})\rangle$.

Together, conditional and expected normal-form games constitute our mathematical model of simulation-based games. In a simulation-based game, players' utilities depend not only on the strategies at play but also on unobserved exogenous elements, often stochastic. Abstractly, we model these elements condition set $\mathcal{X}$ that, together with the chosen strategies[2], defines the corresponding conditional normal-form game. By repeatedly querying a simulator, we observe samples of utilities that then form an empirical normal-form game. Our goal is to approximate, from empirical normal-form games, the equilibria of the corresponding expected normal-form game for fixed distribution $\mathscr{D}$ over condition set $\mathcal{X}$. The expected normal-form game resolves all uncertainty and thus, serves as our ground-truth game for strategic analysis.

The following simple example illustrates our mathematical model of simulation-based games.

Example 1 (Conditional, Expected, and Empirical Normal-Form Game)

Consider the following simplified version of the *diner's dilemma* game[a]. You and your friend decide to go out to eat together. Since you are friends (and maybe a bit careless!), you decide, before ordering, to split the bill in half. To keep this example simple, suppose that your restaurant of choice offers only two options: a cheap dish that costs \$2 and an expensive dish that costs \$8. You and your friend grew up together, so naturally, you have similar tastes.

[2]In this thesis, we assume a fixed set of strategies is given to us prior to any strategic analysis. This condition holds in meta-games like Starcraft [121], for example, where agents choices comprise a few high-level heuristic strategies, not intractably many low-level game-theoretic strategies.

In particular, you both share the same (monetary) values of \$5 and \$10 for the cheap and expensive dish, respectively. In this example, the utility associated with a dish is the value of the dish minus the payment, i.e., half the bill for the entire meal.

Given that both you and your friends are utility-maximizing agents, the central strategic question now is: what dish should you (your friend) choose considering what your friend (you) might choose? To help us answer this question, we can write the normal-form game that models this strategic situation. Let C denote the choice of a cheap dish and E the choice of an expensive dish. If both you and your friend chose a cheap dish, then each receives utility 3, $\boldsymbol{u}(\mathrm{C}, \mathrm{C}) = [5 - (2 + 2) / 2, 5 - (2 + 2) / 2] = [3, 3]$. If you both chose expensive meals, then each receives 2, $\boldsymbol{u}(\mathrm{E}, \mathrm{E}) = [10 - (8 + 8) / 2, 10 - (8 + 8) / 2] = [2, 2]$. Finally, if the choices are not the same, the cost of half the bill is $(2 + 8) / 2 = 5$, which means that whoever chose the cheap dish receives 0 utility and the other 5, $\boldsymbol{u}(\mathrm{C}, \mathrm{E}) = [5 - 5, 10 - 5] = [0, 5]$. The following table summarizes the utility of all possible choices of meals in this game.

	C	E
C	3, 3	0, 5
E	5, 0	2, 2

Table 3.1: Bimatrix for the diner's dilemma game.

Note that the table above completely characterizes the game you and your friend face, given that both of you have perfect information about all essential elements to make a decision. The reader can verify that the game's unique 0-Nash equilibrium is when both players chose to order the expensive dish. In this case, both make a utility of 2. However, a higher-utility state for both agents would be when both chose the cheap dish, but alas, the incentives are so that choosing the expensive deal is a better strategic choice. Hence, each player faces a dilemma between choosing the cheap dish and hoping their friend does the same, thereby maximizing both of their utilities; or following the strictly rational choice and selecting the expensive dish to maximize its own utility regardless of its friend choice.

But now, suppose that the situation complicates slightly. It is still the case that you (and your friend) value the expensive dish at \$10, but only when the restaurant's cook is in a good mood. In this case, the cook puts all his efforts into making the expensive dish, resulting in a delicious meal. Otherwise, if the cook is not in a good mood, he spends less effort preparing it[b], and the result is not so delicious. In this case, your (and your friend's) value for the dish drops to \$8. Since the cheap dish is much easier to prepare, your value for the cheap dish remains at \$5, regardless of the cook's mood. The cost of both dishes is the same as before[c]. We can model the conditional normal-form associated with this situation as follows.

Define set of conditions $\mathscr{X} = \{\textsc{Good Mood}, \textsc{Bad Mood}\}$. The conditional utility function is then given by $\boldsymbol{u}(\mathrm{C}, \mathrm{C}; \textsc{Good Mood}) = \boldsymbol{u}(\mathrm{C}, \mathrm{C}; \textsc{Bad Mood}) = \boldsymbol{u}(\mathrm{C}, \mathrm{C}) = [3,\ 3]$, in case both chose the cheap dish. In case they both choose the expensive dish, $\boldsymbol{u}(\mathrm{E}, \mathrm{E}; \textsc{Good Mood}) = \boldsymbol{u}(\mathrm{E}, \mathrm{E}) = [2,\ 2]$, and $\boldsymbol{u}(\mathrm{E}, \mathrm{E}; \textsc{Bad Mood}) = [8\ -\ (8\ +\ 8)/\ 2,\ 8\ -\ (8\ +\ 8)/\ 2] = [0,\ 0]$. Note that, in this case, the players' utilities depend not only on the players' choices but also on the (possibly unobserved) cook's mood. Finally, if you go with the cheap dish while your friend goes with the expensive cheap, then $\boldsymbol{u}(\mathrm{C}, \mathrm{E}; \textsc{Good Mood}) = \boldsymbol{u}(\mathrm{C}, \mathrm{E}) = [0,\ 5]$ and $\boldsymbol{u}(\mathrm{C}, \mathrm{E}; \textsc{Bad Mood}) = [5\ -\ 5,\ 8\ -\ 5] = [0,\ 3]$. This completely defines the conditional normal-form game.

How should you decide what to do in this more complicated situation? Suppose that your decision complicates by the fact that you can't tell precisely the mood of the cook. However, you might have observed (say, from past visits to the restaurant) that the cook is in a good mood most of the time. Your goal is to make a good strategic decision most of the time (say, on future visits with your friend to the restaurant). In that case, we can further model your lack of certain knowledge about the cook's mood by positing the existence of distribution, $\mathscr{D}$, over the cook's mood. For example, suppose that your observations indicate that $Pr_{\mathscr{D}}(\textsc{Good Mood}) = 3/4$ and $Pr_{\mathscr{D}}(\textsc{Bad Mood}) = 1/4$. Given this information, we can model the expected normal-form game as the game whose utilities are given by taking expectations of the corresponding conditional normal-form game's utilities over $\mathscr{D}$. For example, let us compute the utility of the expected normal-form game in case you and your friend chose the expensive dish:

$$
\begin{aligned}
\boldsymbol{u}(E, E; \mathscr{D}) &= \mathop{\mathbb{E}}_{x \sim \mathscr{D}} [\boldsymbol{u}(E, E; x)] \\
&= {}^{3}\!/\!_{4}\,\boldsymbol{u}(E, E; \textsc{Good Mood}) + {}^{1}\!/\!_{4}\,\boldsymbol{u}(E, E; \textsc{Bad Mood}) \\
&= {}^{3}\!/\!_{4}[2\ , 2] + {}^{1}\!/\!_{4}[0, 0] \\
&= [{}^{3}\!/\!_{2}\ , {}^{3}\!/\!_{2}]
\end{aligned} \tag{3.1}
$$

The reader can verify that computing utilities for all strategies profiles yields the following expected-normal form game,

	C	E
C	3, 3	0, $^9/_2$
E	$^9/_2$, 0	$^3/_2$, $^3/_2$

Table 3.2: Bimatrix for the expected normal-form diner's dilemma game

Estimating the equilibria of the above game is our EGTA methodology's goal. This goal would reduce to a purely computational problem if we had access to the bimatrix above. Note, however, that per our setup, we never get to observe said bimatrix. In fact, in most cases of practical importance, we never get to observe neither $\mathcal{X}$ nor $\mathscr{D}$. Instead, we only get to observe utility samples that we can collect into empirical normal-form games. In the case of your choice of meal, samples might come from multiple visits to the restaurant where both you and your friend select a dish, pay the bill, and measure your utilities. In practice, samples come from a game's simulator.

[a] A diner's dilemma game can also be understood as a n-player prisoner's dilemma game [94].
[b] Being angry requires some effort!
[c] The restaurant would be wise then to keep the cook in a good mood!

Having established our learning framework, our present goal, then, is to "uniformly learn" empirical games (i.e., obtain uniform convergence guarantees) from finitely many samples. This learning problem is non-trivial because it involves multiple comparisons. Tuyls et al. [121] use Hoeffding's inequality to estimate a single utility value, and then apply a Šidák correction to estimate all utility values simultaneously, assuming independence among agents' utilities. Similarly, one can apply a Bonferroni correction (i.e., a union bound) to Hoeffding's inequality, which does not require independence, but yields a slightly looser bound.

Theorem 2 (Finite-Sample Bounds via Hoeffding's Inequality)**.** *Consider finite, conditional normal-form game* $\Gamma_{\mathcal{X}}$ *together with distribution* $\mathscr{D}$ *and index set* $\boldsymbol{I} \subseteq P \times \boldsymbol{S}$ *such that for all* $x \in \mathcal{X}$ *and* $(p, \boldsymbol{s}) \in \boldsymbol{I}$*, it holds that* $\boldsymbol{u}_p(\boldsymbol{s}; x) \in [-c/2, c/2]$*, where* $c \in \mathbb{R}$*. Then, with probability at least* $1 - \delta$ *over* $\boldsymbol{X} \sim \mathscr{D}^m$*, it holds that*

$$\sup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \leq c\sqrt{\frac{\ln(\frac{2|\boldsymbol{I}|}{\delta})}{2m}}$$

Proof. By Hoeffding's inequality [62],

$$Pr(|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \geq \varepsilon) \leq 2e^{-2m\left(\frac{\varepsilon}{c}\right)^2} \tag{3.2}$$

Now, applying union bound over all events $|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \geq \varepsilon$ where $(p, \boldsymbol{s}) \in \boldsymbol{I}$,

$$Pr\left(\bigcup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \geq \varepsilon\right) \leq \sum_{(p,\boldsymbol{s}) \in \boldsymbol{I}} Pr\left(|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \geq \varepsilon\right) \tag{3.3}$$

Using bound (3.2) in the right-hand side of (3.3),

$$Pr\left(\bigcup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \geq \varepsilon\right) \leq \sum_{(p,\boldsymbol{s}) \in \boldsymbol{I}} 2e^{-2m(\frac{\varepsilon}{c})^2} = 2|\boldsymbol{I}|e^{-2m\left(\frac{\varepsilon}{c}\right)^2} \tag{3.4}$$

Where the last equality follows because the summands on the right-hand size of eq. (3.4) do not depend on the summation index. Now, note that eq. (3.4) implies a lower bound for the event that complements $\bigcup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \geq \varepsilon$,

$$Pr\left(\bigcup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \leq \varepsilon\right) \geq 1 - 2|\boldsymbol{I}|e^{-2m\left(\frac{\varepsilon}{c}\right)^2} \tag{3.5}$$

The event $\bigcap_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \leq \varepsilon$ is equivalent to the event $\max_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})| \leq \varepsilon$. Setting $\delta = 2|\boldsymbol{I}|e^{-2m(\frac{\varepsilon}{c})^2}$ and solving for ε yields $\varepsilon = c\sqrt{\ln(2|\boldsymbol{I}|/\delta)/2m}$.

The results follows by substituting ε in eq. (3.5).

□

Remark Given a game, we state all theorems and algorithms for an arbitrary index set $\boldsymbol{I}$. Taking $\boldsymbol{I} = P \times \boldsymbol{S}$, we bound $\|\boldsymbol{u}(\cdot; \mathscr{D}) - \hat{\boldsymbol{u}}(\cdot; \boldsymbol{X})\|_\infty$.

Consider $X_{1:m}$ independent and identically distributed random variables, and their mean $\bar{X}$. (In our case, $X_j = \boldsymbol{u}(\boldsymbol{s}; x_j)$.) Hoeffding's inequality for bounded random variables can be used to obtain tail bounds on the probability that an empirical mean differs greatly from its expectation. One way to characterize such bounds is to compare them to well-studied cases of common random variables. We focus on the case of upper tails, as we apply only symmetric two-tail bounds in this work.

Random variables that obey the Gaussian Chernoff bound can be characterized as $\sigma^2_{\mathcal{N}}$-*sub-Gaussian*: i.e.,

$$Pr\left(\bar{X} \geq \mathbb{E}[\bar{X}] + \varepsilon\right) \leq \exp\left(\frac{-m\varepsilon^2}{2\sigma^2_{\mathcal{N}}}\right) \iff Pr\left(\bar{X} \geq \mathbb{E}[\bar{X}] + \sqrt{\frac{2\sigma^2_{\mathcal{N}} \ln(\frac{1}{\delta})}{m}}\right) \leq \delta$$

Here, $\sigma^2_{\mathcal{N}}$ is deemed a *variance proxy*. Using this characterization, Hoeffding's inequality reads "If X_i has range c, then X_i is $c^2/4$-sub-Gaussian," because, by Popoviciu's inequality [101], $\mathbb{V}[X_i] \leq c^2/4$. Thus, Hoeffding's inequality yields sub-Gaussian tail bounds.

This result is not entirely satisfying, however, as it is stated in terms of the *worst-case* variance; when $\mathbb{V}[X_i] \ll c^2/4$, tighter bounds should be possible. We might hope that knowledge of the variance σ^2 would imply σ^2-sub-Gaussian, but it does not; taking the range c to ∞ allows X_i to exhibit arbitrary tail behaviors. A (σ^2_{Γ}, c)-*sub-gamma* [27] random variable obeys

$$Pr\left(\bar{X} \geq \mathbb{E}[\bar{X}] + \frac{c\ln(\frac{1}{\delta})}{3m} + \sqrt{\frac{2\sigma^2_{\Gamma} \ln(\frac{1}{\delta})}{m}}\right) \leq \delta$$

While the form of such bounds is more complicated than in the sub-Gaussian case, there is an intuitive interpretation of the tail behavior as sample size increases. The key is to observe that the additive error consists of a *hyperbolic* $c\ln(\frac{1}{\delta})/3m$ term and a *root-hyperbolic* $\sqrt{2\sigma^2_{\mathcal{N}} \ln(\frac{1}{\delta})/m}$ term, which in learning theory are often called *fast* and *slow* terms, respectively. Sub-gamma random variables then yield *mixed convergence rates*, which initially decay quickly while the c-term dominates, before slowing to the root-hyperbolic rate when the $\sigma^2_{\mathcal{N}}$ term comes to dominate.

Bennett's inequality 1962, while usually stated as a sub-Poisson bound, immediately implies that if X_i has range c and variance σ^2, then X_i is (σ^2, c)-sub-gamma. Bennett's inequality, however, assumes the range and variance are *known*. Various *empirical* Bennett bounds have been shown [10, 11, 85], which all essentially operate by bounding the variance of a random variable in terms of its empirical variance and range, and then applying Bennett's inequality. Simply put, Bennett's inequality gives us Gaussian-like tail bounds, where the scale-dependent term acts as an *asymptotically negligible* correction for working with non-Gaussian distributions and *empirical estimates* of variance. Asymptotic central-limit-theorem bounds behave similarly, but lack corresponding finite-sample guarantees.

Our work differs from previous applications in that we require confidence intervals of *uniform width*, and thus our bounds are limited by the *maximum* variance over all parameters being estimated. The maximum variance over a set of random variables is known as the *wimpy variance* [27]. We apply a sub-gamma bound to the wimpy variance in terms of an empirical estimate, and then apply Bennett's inequality to the upper and lower tails of each utility function using the wimpy variance bound, as this, by definition, upper bounds the variance of each utility. This strategy yields uniform-width confidence intervals over all utilities, and requires a union bound over m upper tails, m lower tails, and the wimpy variance, whereas bounding each variance separately would require a union bound over m upper, m lower, and m variance bounds. Thus, our method outperforms such strategies, yielding tighter confidence intervals.

We now present a novel concentration inequality that combines both Hoeffding's and Bennett's inequalities[3].

Theorem 3 (Bennett-Type Variance-Sensitive Finite-Sample Bounds)**.** *Suppose as in Theorem 2. Take*

$$\hat{v} \doteq \sup_{(p,\boldsymbol{s})\in\boldsymbol{I}} \frac{1}{m-1}\sum_{j=1}^{m}\left(\boldsymbol{u}_p(\boldsymbol{s};x_j) - \hat{\boldsymbol{u}}_p(\boldsymbol{s};\boldsymbol{X})\right)^2 \quad ;$$

$$\varepsilon_v \doteq \frac{c\ln(\frac{3}{\delta})}{m-1} + \sqrt{\left(\frac{c\ln(\frac{3}{\delta})}{m-1}\right)^2 + \frac{2\hat{v}\ln(\frac{3}{\delta})}{m-1}} \quad ; \quad \textit{and}$$

$$\varepsilon_\mu \doteq \min\Bigg(\underbrace{c\sqrt{\frac{\ln(\frac{3|\boldsymbol{I}|}{\delta})}{2m}}}_{\textsc{Hoeffding}}, \underbrace{\frac{c\ln(\frac{3|\boldsymbol{I}|}{\delta})}{3m} + \sqrt{\frac{2(\hat{v}+\varepsilon_v)\ln(\frac{3|\boldsymbol{I}|}{\delta})}{m}}}_{\textsc{Empirical Bennett}}\Bigg) \quad .$$

Then, with probability at least $1-\delta$ over $\boldsymbol{X}\sim\mathscr{D}^m$, it holds that

$$\sup_{(p,\boldsymbol{s})\in\boldsymbol{I}} \left|\boldsymbol{u}_p(\boldsymbol{s};\mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s};\boldsymbol{X})\right| \leq \varepsilon_\mu \quad .$$

[3]This inequality is the sole contribution of fellow Ph.D. student Cyrus Cousins and its proof can be found in https://github.com/eareyan/pysegta.

From the definition of ε_μ, we see that when $\hat{v}$ is $\approx \frac{c^2}{4}$ (near-maximal), the Hoeffding term applies, so this bound matches Theorem 2 to within constant factors (in particular, $\ln(\frac{3|\boldsymbol{I}|}{\delta})$ instead of $\ln(\frac{2|\boldsymbol{I}|}{\delta})$). On the other hand, when $\hat{v}$ is small, Theorem 3 is much sharper than Theorem 2. A few simplifying inequalities yield

$$\varepsilon_\mu \leq \frac{7c\ln(\frac{3|\boldsymbol{I}|}{\delta})}{3(m-1)} + \sqrt{\frac{2\hat{v}\ln(\frac{3|\boldsymbol{I}|}{\delta})}{m}}$$

which matches the standard sub-gamma Bennett's inequality up to constant factors, with dependence on $\hat{v}$ instead of v. In the extreme, when $\hat{v} \approx 0$ (i.e., the game is *near-deterministic*), then Theorem 3 improves asymptotically over Theorem 2 by a $\Theta(\sqrt{\ln(\frac{|\boldsymbol{I}|}{\delta})/m})$ factor.

3.2.2 Learning Algorithms

We are now ready to present our algorithms. Specifically, we discuss two Monte-Carlo sampling-based algorithms that can be used to uniformly learn empirical games, and hence ensure that the equilibria of the games they are learning are accurately approximated with high probability. Note that our algorithms apply only to finite games, as they require an enumeration of the index set $\boldsymbol{I}$.

A conditional normal form game $\Gamma_{\mathcal{X}}$, together with distribution $\mathscr{D}$, serves as our mathematical model of a simulator from which the utilities of a simulation-based game can be sampled. Given strategy profile $\boldsymbol{s}$, we assume the simulator outputs a sample $\boldsymbol{u}_p(\boldsymbol{s}, x)$, for *all* agents $p \in P$, after drawing a *single* condition value $x \sim \mathscr{D}$.

Our first algorithm, *global sampling* (GS), is a straightforward application of Thms. 2 and 3. The second, *progressive sampling with pruning* (PSP), is a progressive-sampling-based approach [104, 105], which iteratively prunes strategies, and thereby has the potential to expedite learning by obtaining tighter bounds than GS, given the same number of samples. We explore PSP's potential savings in our experiments, Section 3.3.

GS (Algorithm 1), samples all utilities of interest, given a sample size m and a failure probability δ, and returns the ensuing empirical game together with an $\hat{\varepsilon}$ determined by either Thm. 2 or 3 that guarantees an $\hat{\varepsilon}$-uniform approximation.

More specifically, GS takes in a conditional game $\Gamma_{\mathcal{X}}$, a black box from which we can sample distribution $\mathscr{D}$, an index set $\boldsymbol{I} \subseteq P \times \boldsymbol{S}$, a sample size m, a utility range c such that utilities are required to lie in $[-c/2, c/2]$, and a bound type Bd, and then draws m samples to produce an empirical

Algorithm 1 Global Sampling

procedure $\mathrm{GS}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{I}, m, \delta, c, \textsc{Bd}) \to (\tilde{\boldsymbol{u}}, \hat{\varepsilon})$

  input:

  conditional game $\Gamma_{\mathcal{X}}$
  black box from which we can sample distribution $\mathscr{D}$
  index set $\boldsymbol{I}$
  sample size m
  failure probability δ
  utility range c
  bound type $\textsc{Bd}$.

  output:

  empirical utilities $\tilde{\boldsymbol{u}}, \forall (p, \boldsymbol{s}) \in \boldsymbol{I}$
  additive error $\hat{\varepsilon}$

  $\boldsymbol{X} \sim \mathscr{D}^m$ ▷ Draw m samples from distribution $\mathscr{D}$

  $\forall (p, \boldsymbol{s}) \in \boldsymbol{I} : \ \tilde{\boldsymbol{u}}_p(\boldsymbol{s}) \leftarrow \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})$

  if $\textsc{Bd} = \mathrm{H}$ **then** ▷ See Thm. 2 (Hoeffding)

    $\hat{\varepsilon} \leftarrow c\sqrt{\frac{\ln(\frac{2|\boldsymbol{I}|}{\delta})}{2m}}$

  else if $\textsc{Bd} = \mathrm{B}$ **then** ▷ See Thm. 3 (Empirical Bennett)

    $\hat{v} \leftarrow \sup_{(p, \boldsymbol{s}) \in \boldsymbol{I}} \frac{1}{m-1} \sum_{j=1}^{m} \left(\boldsymbol{u}_p(\boldsymbol{s}; x_j) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})\right)^2$

    $\varepsilon_v \leftarrow \frac{c \ln(\frac{3}{\delta})}{m-1} + \sqrt{\left(\frac{c \ln(\frac{3}{\delta})}{m-1}\right)^2 + \frac{2\hat{v} \ln(\frac{3}{\delta})}{m-1}}$

    $\hat{\varepsilon} \leftarrow \min\left(c\sqrt{\frac{\ln(\frac{3|\boldsymbol{I}|}{\delta})}{2m}}, \ \frac{c \ln(\frac{3|\boldsymbol{I}|}{\delta})}{3m} + \sqrt{\frac{2(\hat{v}+\varepsilon_v) \ln(\frac{3|\boldsymbol{I}|}{\delta})}{m}} \right)$

  end if

  return $(\tilde{\boldsymbol{u}}, \hat{\varepsilon})$

end procedure

game $\hat{\Gamma}_{\boldsymbol{X}}$, represented by $\tilde{\boldsymbol{u}}(\cdot)$, as well as an additive error $\hat{\varepsilon}$, with the following guarantee:

Theorem 4 (Approximation Guarantees of Global Sampling)**.** *Consider conditional game $\Gamma_{\mathcal{X}}$ together with distribution $\mathscr{D}$ and take index set $\boldsymbol{I} \subseteq P \times \boldsymbol{S}$ such that for all $x \in \mathcal{X}$ and $(p, \boldsymbol{s}) \in \boldsymbol{I}$, $\boldsymbol{u}_p(\boldsymbol{s}; x) \in [-c/2, c/2]$, for some $c \in \mathbb{R}$.*

If $\mathrm{GS}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{I}, m, \delta, c, \textsc{Bd})$ outputs pair $(\tilde{\boldsymbol{u}}, \hat{\varepsilon})$, then with probability at least $1 - \delta$, it holds that

$$\sup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})| \leq \hat{\varepsilon} \ .$$

Proof of Theorem 4. Use Theorem 2 when Bound = Hoeffding and Theorem 3 when Bound = Bennett. □

Next, we present PSP (Algorithm 2), which, using GS as a subroutine, draws progressively larger samples, refining the empirical game at each iteration, and stopping when the equilibria are approximated to the desired accuracy, or when the sampling budget is exhausted. Although performance ultimately depends on a game's structure, PSP can potentially learn equilibria using vastly fewer resources than GS.

As the name suggests, PSP is a pruning algorithm. The key idea is to prune (i.e., cease estimating the utilities of) strategy profiles that (w.h.p.) are provably not equilibria. Recall that $\boldsymbol{s} \in \mathrm{E}_\varepsilon(\boldsymbol{u})$ if and only if $\mathrm{Reg}_p(\boldsymbol{u}, \boldsymbol{s}) \leq \varepsilon$, for all $p \in P$. Thus, if there exists $p \in P$ such that $\mathrm{Reg}_p(\boldsymbol{u}, \boldsymbol{s}) > \varepsilon$, then $\boldsymbol{s} \notin \mathrm{E}_\varepsilon(\boldsymbol{u})$. In the search for pure equilibria, such strategy profiles can be pruned.

A strategy $s \in S_p$ is said to *ε-dominate* another strategy $s' \in S_p$ if for all $\boldsymbol{s} \in \boldsymbol{S}$, taking $\boldsymbol{s}' = (s_1, \ldots, s_{p-1}, s'_p, s_{p+1}, \ldots, s_{|P|})$, it holds that $\boldsymbol{u}_p(\boldsymbol{s}) - \varepsilon \geq \boldsymbol{u}_p(\boldsymbol{s}')$. Given a game Γ with utility function $\boldsymbol{u}$, the *ε-rationalizable* strategies $\mathrm{Rat}_\varepsilon(\boldsymbol{u})$ are those that remain after iteratively removing all ε-dominated strategies. This set can easily be computed via the *iterative elimination of ε-dominated strategies*. Only strategies in $\mathrm{Rat}_\varepsilon(\boldsymbol{u})$ can have nonzero weight in a mixed ε-Nash equilibrium [51]; thus eliminating strategies not in $\mathrm{Rat}_\varepsilon(\boldsymbol{u})$ is a natural pruning criterion for mixed equilibria.

If a strategy $s \in S_p$ is ε-dominated by another strategy $s' \in S_p$, then p always regrets playing strategy s, regardless of other agents' strategies. Consequently, the mixed pruning criterion is more conservative than the pure, which means more pruning occurs when learning pure equilibria.

Like GS, PSP takes in a conditional game $\Gamma_{\mathcal{X}}$, a black box from which we can sample distribution $\mathscr{D}$, a utility range c, and a bound type Bd. Instead of a single sample size, however, it takes in a *sampling schedule* $\boldsymbol{M}$ in the form of a (possibly infinite) strictly increasing sequence of integers; and instead of a single failure probability, it takes in a *failure probability schedule* $\boldsymbol{\delta}$, with each $\boldsymbol{\delta}_t$ in this sequence and their sum in $(0, 1)$. These two schedules dictate the number of samples to draw and the failure probability to use at each iteration. PSP also takes in a boolean Pure that indicates whether the equilibria of interest are pure or mixed, and an *error threshold* ε, which enables early termination as soon as equilibria of the desired sort are estimated to within the additive factor ε.

Algorithm 2 Progressive Sampling with Pruning

procedure $\mathrm{PSP}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{M}, \boldsymbol{\delta}, c, \textsc{Bd}, \textsc{Pure}, \varepsilon) \to ((\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}), (\hat{E}, \hat{\varepsilon}), \hat{\delta})$

  input:

    conditional game $\Gamma_{\mathcal{X}}$
    black box from which we can sample distribution $\mathscr{D}$
    sampling schedule $\boldsymbol{M}$
    failure probability schedule $\boldsymbol{\delta}$
    utility range c
    bound type Bd
    equilibrium type Pure
    error threshold ε

  output:

    Empirical utilities $\tilde{\boldsymbol{u}}$, $\forall (p, \boldsymbol{s}) \in P \times \boldsymbol{S}$
    utility error $\tilde{\boldsymbol{\varepsilon}}$
    empirical equilibria $\hat{E}$
    equilibria error $\hat{\varepsilon}$
    failure probability $\hat{\delta}$

  $\boldsymbol{I} \leftarrow P \times \boldsymbol{S}$ ▷ Initialize index set
  $\forall (p, \boldsymbol{s}) \in \boldsymbol{I} :\ (\tilde{\boldsymbol{u}}_p(\boldsymbol{s}), \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s})) \leftarrow (0, c/2)$, ▷ Initialize outputs
  for $t \in 1, \ldots, |\boldsymbol{M}|$ **do** ▷ Progressive sampling iterations
    $(\tilde{\boldsymbol{u}}, \hat{\varepsilon}) \leftarrow \mathrm{GS}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{I}, \boldsymbol{M}_t, \boldsymbol{\delta}_t, c, \textsc{Bd})$ ▷ Improve utility estimates
    $\forall (p, \boldsymbol{s}) \in \boldsymbol{I} :\ \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s}) \leftarrow \hat{\varepsilon}$ ▷ Update confidence intervals
    if $\hat{\varepsilon} \leq \varepsilon$ or $t = |\boldsymbol{M}|$ **then** ▷ Termination condition
      $\hat{E} \leftarrow \begin{cases} \textsc{Pure} & : \ \mathrm{E}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \\ \neg\textsc{Pure} & : \ \mathrm{E}^{\diamond}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \end{cases}$
      return $((\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}), (\hat{E}, \hat{\varepsilon}), \sum_{i=1}^{t} \boldsymbol{\delta}_i)$
    end if
    $\boldsymbol{I} \leftarrow \begin{cases} \textsc{Pure} : & \{(p, \boldsymbol{s}) \in \boldsymbol{I} \mid \mathrm{Reg}_p(\tilde{\boldsymbol{u}}, \boldsymbol{s}) \leq 2\hat{\varepsilon}\} \\ \neg\textsc{Pure} : & \{(p, \boldsymbol{s}) \in \boldsymbol{I} \mid \forall q \in P : \boldsymbol{s}_q \in \mathrm{Rat}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}})\} \end{cases}$
  end for
end procedure

Theorem 5 (Approximation Guarantees of Progressive Sampling with Pruning)**.** *Suppose conditional game* $\Gamma_{\mathcal{X}}$ *and distribution* $\mathscr{D}$ *such that for all* $x \in \mathcal{X}$ *and* $(p, \boldsymbol{s}) \in P \times \boldsymbol{S}$*,* $\boldsymbol{u}_p(\boldsymbol{s}; x) \in [-c/2, c/2]$ *for some* $c \in \mathbb{R}$*. If* $\mathrm{PSP}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{M}, \boldsymbol{\delta}, c, \textsc{Bd}, \textsc{Pure}, \varepsilon)$ *outputs* $\big((\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}), (\hat{E}, \hat{\varepsilon}), \hat{\delta}\big)$*, it holds that:*

1. $\hat{\delta} \leq \sum_{\boldsymbol{\delta}_t \in \boldsymbol{\delta}} \boldsymbol{\delta}_t, \hat{\delta} \in (0, 1)$
2. *If* $\lim_{t \to \infty} \ln(1/\boldsymbol{\delta}_t)/\boldsymbol{M}_t = 0$*, then* $\hat{\varepsilon} \leq \varepsilon$*.*

Furthermore, if PSP *terminates, then with probability at least* $1 - \hat{\delta}$*, the following hold simultaneously:*

3. $|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})| \leq \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s})$*, for all* $(p, \boldsymbol{s}) \in P \times \boldsymbol{S}$
4. *If* $\textsc{Pure}$*, then* $\mathrm{E}(\boldsymbol{u}) \subseteq \mathrm{E}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \subseteq \mathrm{E}_{4\hat{\varepsilon}}(\boldsymbol{u})$
5. *If* $\neg\textsc{Pure}$*, then* $\mathrm{E}^{\diamond}(\boldsymbol{u}) \subseteq \mathrm{E}^{\diamond}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \subseteq \mathrm{E}^{\diamond}_{4\hat{\varepsilon}}(\boldsymbol{u})$*.*

Proof of Theorem 5. To see 1, note that $\hat{\delta}$ is computed on line 11 as a partial sum of $\boldsymbol{\delta}$, each addend and the sum of which are all by assumption on $(0, 1)$; thus the result holds.

To see 2, note that if $\lim_{t \to \infty} \ln(1/\boldsymbol{\delta}_t)/\boldsymbol{M}_t = 0$, then both the Hoeffding's and Bennett's bounds employed by GS tend to 0, as both decay asymptotically (in expectation) as $\mathcal{O}(\sqrt{\ln(1/\delta_t)/\boldsymbol{M}_t})$ (see Theorems 2 and 3). For infinite sampling schedules, the termination condition of line 9 ($\hat{\varepsilon} \leq \varepsilon$) is eventually met, as $\hat{\varepsilon}$ is the output of GS, and thus 2 holds.

To establish 3, we show the following: assuming termination occurs at timestep n, with probability at least $1 - \hat{\delta}$, at every t in $\{1, \dots, n\}$, it holds that $\sup_{(p, \boldsymbol{s}) \in P \times \boldsymbol{S}} |\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})| \leq \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s})$. This property follows from the GS guarantees of Theorem 4, as at each timestep t, the guarantee holds with probability at least $1 - \boldsymbol{\delta}_t$; thus by a union bound, the guarantees hold simultaneously at all time steps with probability at least $1 - \sum_{i=1}^{n} \boldsymbol{\delta}_t = 1 - \hat{\delta}$. That the GS guarantees hold for unpruned indices should be clear; for pruned indices, since only error bounds for indices updated on line 7 are tightened on line 8, it holds by the GS guarantees of previous iterations.

Without pruning, 4 and 5 would follow directly from 3 via Theorem 1, but with pruning, the situation is a bit more involved. To see 4, observe that at each time step, only indices (p, $\boldsymbol{s}$) such that $\mathrm{Reg}_p\big(\tilde{\boldsymbol{u}}(\cdot), \boldsymbol{s}\big) > 2\hat{\varepsilon}$ are pruned (line 13), thus we may guarantee that with probability at least $1 - \hat{\delta}$, $\mathrm{Reg}_p\big(\boldsymbol{u}(\cdot; \mathscr{D}), \boldsymbol{s}\big) > 0$. Increasing the accuracy of the estimates of these strategy profiles is thus not necessary, as they do not comprise pure equilibria (w.h.p.), and they will never be required to refute equilibria, as these will never be a best response for any agent from any strategy profile.

5 follows similarly, except that nonzero regret implies that a pure strategy profile is not a pure Nash equilibrium, but it does *not* imply that it is *not* part of any mixed Nash equilibrium. Consequently, we use the more conservative pruning criterion of strategic dominance[4], requiring 2ε-dominance in $\tilde{\boldsymbol{u}}$, as this implies nonzero dominance in $\boldsymbol{u}$. □

Finally, we propose two possible sampling and failure probability schedules for PSP, $\boldsymbol{M}$ and $\boldsymbol{\delta}$, depending on whether the sampling budget is finite or infinite. Given a finite sampling budget $m < \infty$, a neutral choice is to take $\boldsymbol{M}$ to be a doubling sequence such that $\sum_{M_i \in \boldsymbol{M}} M_i \leq m$, with $\boldsymbol{M}_1$ sufficiently large so as to possibly permit pruning after the first iteration (iterations that neither prune nor achieve ε-accuracy are effectively wasted), and to take $\boldsymbol{\delta}_t = {}^{\delta}/_{|\boldsymbol{M}|}$, where δ is some maximum tolerable failure probability. This strategy always respects the sampling budget, but may fail to produce the desired ε-approximation, as it may exhaust the sampling budget first. To guarantee a particular ε-δ-approximation, then we can take $\boldsymbol{M}$ to be an infinite doubling sequence, and $\boldsymbol{\delta}$ to be a geometrically decreasing sequence such that $\sum_{t=1}^{\infty} \boldsymbol{\delta}_t = \delta$, for which the conditions of item 2 of Theorem 5 hold.

3.3 Experiments

We now set out to evaluate the strength of our methodology to learn simulation-based games and their equilibria from samples. The empirical performance of an algorithm can vary dramatically under different distributions of inputs; in particular, the success of game-theoretic solvers can vary dramatically even within the same class of games [80, 93]. Consequently, we employ GAMUT [93], a state-of-the-art suite of game generators that is capable of producing a wide variety of interesting game inputs of varying scales with rich strategic structure, thereby affording us an opportunity to conduct a robust evaluation of our methodology. Furthermore, we employ Gambit [86], a state-of-the-art equilibrium solver. We bundled both of these packages together with our statistical learning algorithms in a python library for empirical game-theoretic analysis, *pySEGTA*, to make it easier for other EGTA researchers to benchmark their algorithms against ours.

[4]A strategy is dominated if it is not rationalizable.

3.3.1 Simulation-Based Game Design

In all our experiments, we use GAMUT to generate what we call *ground-truth games*. Ground-truth games are ordinarily inaccessible; however, we rely on them here to measure the loss experienced by our algorithms: i.e., the regrets in a learned game as compared those in the corresponding ground-truth game. To simulate a simulation-based game, we simply add noise drawn from a zero-centered distribution to the utilities of a ground-truth game. We detail this construction presently.

Let Γ be a realization of a ground-truth game drawn from GAMUT, and let $\boldsymbol{u}_p(\boldsymbol{s})$ be the utility of player p at profile $\boldsymbol{s}$ in Γ. Fix a condition set $\mathcal{X} = [a, b]$, where $a < b$. In the conditional game $\Gamma_{\mathcal{X}}$, $\boldsymbol{u}_p(\boldsymbol{s}; x_{p,\boldsymbol{s}}) = \boldsymbol{u}_p(\boldsymbol{s}) + x_{p,\boldsymbol{s}}$, for $x_{p,\boldsymbol{s}} \in \mathcal{X}$. Conditional game $\Gamma_{\mathcal{X}}$ together with distribution $\mathscr{D}$ on $\mathcal{X}$ is then our model for a simulation-based game. For simplicity, all noise $x_{p,\boldsymbol{s}} \sim \mathscr{D}$ is drawn i.i.d..

We only consider noise distributions where $\mathscr{D}$ is zero-centered. Consequently, the expected-normal form game $\Gamma_{\mathscr{D}}$, which is the object of our algorithms' estimation, exactly coincides with Γ: i.e., it holds that for every p and $\boldsymbol{s}$:

$$\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) = \mathop{\mathbb{E}}_{x_{p,\boldsymbol{s}} \sim \mathscr{D}} [\boldsymbol{u}_p(\boldsymbol{s}; x_{p,\boldsymbol{s}})] = \mathop{\mathbb{E}}_{x_{p,\boldsymbol{s}} \sim \mathscr{D}} [\boldsymbol{u}_p(\boldsymbol{s}) + x_{p,\boldsymbol{s}}] = \boldsymbol{u}_p(\boldsymbol{s}) + \mathop{\mathbb{E}}_{x_{p,\boldsymbol{s}} \sim \mathscr{D}} [x_{p,\boldsymbol{s}}] = \boldsymbol{u}_p(\boldsymbol{s})$$

where the last equality follows because $\boldsymbol{u}_p(\boldsymbol{s})$ is constant and $\mathscr{D}$ is zero-centered.

3.3.2 Experimental Setup

We normalize the utilities generated by GAMUT to lie in the range [-10, 10]. We experiment with three different noise regimes, high, medium, and low variance. Letting $U[a, b]$ be a uniform distribution over $[a, b]$, we model high, medium, and low variance noise by distributions $U[-2.5, 2.5], U[-.5, .5]$, and $U[-.1, .1]$, respectively.

We test both GS, Algorithm 1, and PSP, Algorithm 2. These algorithms take as input a flag BD $\in$ {H, B}, indicating which bound, Hoeffding's (Theorem 2) or empirical Bennett-type (Theorem 3), to use. Henceforth, to refer to an algorithm that uses bound BD, we write GS(BD) and PSP(BD). Throughout our experiments, we fix $\delta = 0.05$.

3.3.3 Sample Efficiency of GS

In this experiment, we investigate the sampling efficiency of our algorithms; that is, the quality of the games learned, as measured by ε, as a function of the number of samples needed to achieve said guarantee. We tested ten different classes of GAMUT games, all of them two-player, with varying numbers of strategies, either indicated in parentheses next to the game's name, or two by default.

For each class of games, we draw 60 random ground-truth games from GAMUT, and for each such draw, we run GS 20 times for each of sample sizes $m \in \{10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120\}$, measuring ε for all possible combinations of these parameters. We then average the measured values of ε, fixing the number of samples. Figure 3.1 plots, in a log-log scale, these averages, comparing the performance of GS given Hoeffding's bound and our empirical Bennett-type bound, for the cases of high and low variance. In all cases, we found that our empirical Bennett-type bound produces better estimates for the same number of samples, as measured by ε. Note the initial $1/m$ decay rate, later slowing to $1/\sqrt{m}$, in the Bennett bounds, reflecting the fast c term and slow σ^2 terms of the sub-gamma bounds.

3.3.4 Empirical Regret of GS

In this experiment, we investigate the quality of the equilibria learned by our algorithms. To compute the equilibria of a game, we use Gambit [86], a state-of-the-art equilibrium solver. The goal of these experiments is to provide empirical evidence for Theorem 1, namely that our algorithms are capable of learning games that approximately preserve the Nash equilibria of simulation-based games. The goal is not to test the quality of different equilibrium solvers—we refer the reader to [93] for an evaluation along those lines. Hence, we fix one such solver throughout, namely, Gambit's GNM solver, a global Newton method that computes Nash equilibria [54].

To measure the quality of learned equilibria, given a game Γ with utility function $\boldsymbol{u}$ and a subset of the strategy profile space $\boldsymbol{S}' \subseteq \boldsymbol{S}^\diamond$, we define the metric

$$\textsc{Max-Regret}(\boldsymbol{u}, \boldsymbol{S}') = \sup_{\boldsymbol{s} \in \boldsymbol{S}'} \max_{p \in P} \mathrm{Reg}_p^\diamond(\boldsymbol{u}, \boldsymbol{s})$$

i.e, the maximum regret of any player p at any profile $\boldsymbol{s}$ in $\boldsymbol{S}'$. Note that, given two compatible games Γ and Γ', and $\boldsymbol{S}'$, we can measure Max-Regret of *either* game, since the strategy profile space is shared by compatible games. This is useful because, given a ground-truth game Γ and

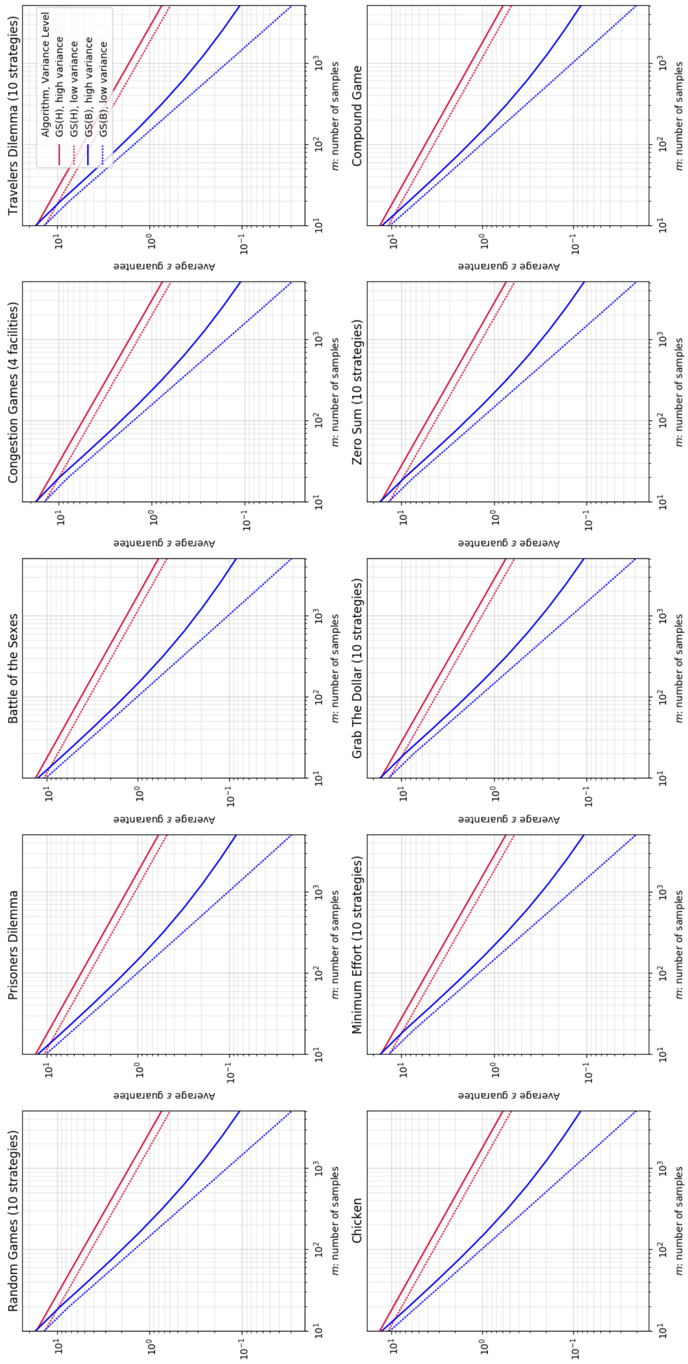


Figure 3.1: Quality of Learned Games.

a corresponding empirical estimate $\hat{\Gamma}_{\boldsymbol{X}}$, we can measure the maximum regret of a set of Nash equilibrium profiles in Γ, say $\boldsymbol{S}^*$, in its empirical estimate $\hat{\Gamma}_{\boldsymbol{X}}$. Theorem 1 implies that, given an ε-uniform approximation $\hat{\Gamma}_{\boldsymbol{X}}$ of Γ with utility function $\hat{\boldsymbol{u}}$, we should observe $\textsc{Max-Regret}(\hat{\boldsymbol{u}}, \boldsymbol{S}^*) \leq 2\varepsilon$. Theorem 4 then implies that if said ε-uniform approximation holds with probability $1 - \delta$, then we should likewise observe $\textsc{Max-Regret}(\hat{\boldsymbol{u}}, \boldsymbol{S}^*) \leq 2\varepsilon$ with probability $1 - \delta$.

We empirically measure Max-Regret, where equilibria are computed using Gambit, for the same ten different classes of games as in the previous experiment, again over 60 draws for each class, where for each such draw, we run GS 10 times for each of sample sizes $m \in \{10, 20, 40, 80, 160\}$, measuring Max-Regret for all possible combinations of these parameters. We then average the measured values of Max-Regret, fixing the number of samples. Figure 3.2 plots, in a log-log scale, both these averages (the markers) and the theoretical guarantees (the lines). This plot complements our theory, establishing experimentally that our algorithms are capable of preserving equilibria of simulation-based games. This learning is robust to various classes of games, for example, for dominance-solvable games (such as Prisoners' Dilemma), games with guaranteed pure-strategy equilibria (such as congestion games), as well as other games with no guarantee on their equilibria other than existence (such as random and zero-sum games). This learning is also robust to different levels of noise, with our algorithms consistently achieving higher accuracy in practice than in theory, all across the board.

3.3.5 Sample Efficiency of PSP

In this experiment, we investigate the sample efficiency of PSP as compared to GS. We say that algorithm A has better *sample efficiency* than algorithm B if A requires fewer samples than B to achieve a desired accuracy ε.

Our experimental design is as follows. Fixing a game, and the following values of $\varepsilon \in \{0.125, 0.25, 0.5, 1.0\}$, we compute the number of samples $m(\varepsilon)$ that would be required for GS(H) to achieve accuracy ε. We then run both GS(H) and GS(B) with $m(\varepsilon)$ samples.

For PSP, we use the doubling strategy, $\boldsymbol{M}(m(\varepsilon)) = [m(\varepsilon)/4, m(\varepsilon)/2, m(\varepsilon), 2m(\varepsilon)]$, as a sampling schedule, rounding to the nearest integer as necessary. For $\boldsymbol{\delta}$, we use a uniform schedule such that $\sum_{\boldsymbol{\delta}_t \in \boldsymbol{\delta}} \boldsymbol{\delta}_t = \delta$: i.e., $\boldsymbol{\delta} = [0.0125, 0.0125, 0.0125, 0.0125]$. Using these schedules, we run both PSP(H) and PSP(B) until completion by setting the desired accuracy to zero. We prune using the mixed-strategy criterion, namely the set of rationalizable strategies.

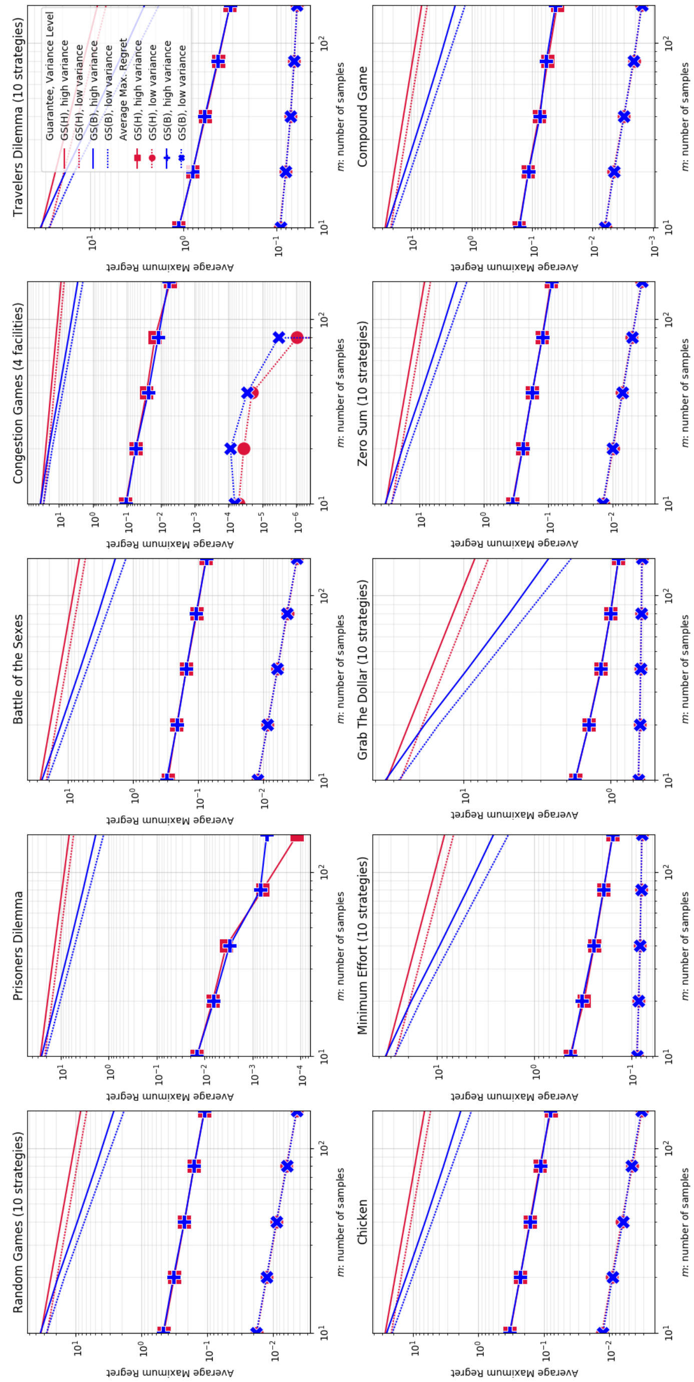


Figure 3.2: Average Maximum Regret.

We ran this experiment on three different classes of games: congestion games (2 players, and 2, 3, 4, and 5 facilities; game sizes 18, 98, 450, and 1,922 respectively), random games (2 players, and 5, 10, 20, and 30 strategies each; game sizes 50, 200, 800, and 1,800 respectively), and zero-sum games (2 players, and 5, 10, 20, and 30 strategies; game sizes 50, 200, 800, and 1,800 respectively). As with our other experiments, we draw multiple games for each class of games (in this case 30) and multiple runs (in this case 10 for each draw of each game). We consider medium variance only.

For all algorithms, we measure the total number of samples across all players and strategy profiles. If and when it prunes, PSP requires progressively fewer and fewer samples, with the number decreasing each iteration to the size of the unpruned game.

Table 3.3 summarizes the results of these experiments for select games. In all cases, we simply report the total number of samples, averaged across all experiments. The theory tells us that given this number of samples, GS must achieve at least the desired $\varepsilon \in \{0.125, 0.25, 0.5, 1.0\}$. Although there is no such guarantee for PSP, in these experiments PSP always achieved a strictly greater accuracy than GS (these accuracies are also reported in Table 3.3, under the columns labeled $\varepsilon_{\mathrm{PSP}}$). Moreover, PSP tends to exhibit significantly better sample efficiency than GS; notable exceptions include cases where either the games are small or the ε guarantee is loose (e.g., $\varepsilon \leq 1.0$). These results demonstrate the promise of PSP as an algorithm for learning black-box games, as its sample efficiency generally exceeds that of GS.

3.3.6 Limitations of PSP

While our experiments demonstrate that PSP can yield substantial savings when learning games in many different classes, in some GAMUT games, our simple doubling schedule yielded no such gains. We found Grab The Dollar to be a particularly difficult game. In Grab The Dollar, there is a prize (or "dollar") that two players are free to grab at any time, and there are two utility values, one high and one low. If both players grab for the dollar at the same time, it will rip; so the players earn the low utility. If one grabs the dollar before the other, then that player wins the dollar (and thus high utility), while the other player earns utility somewhere between the high and the low values.

The utility structure of this game is such that the player's utilities are the same across many different strategy profiles—in particular, whenever one player "Grabs The Dollar" before their opponent. As a result, there are few ε-dominated strategies, which in turn makes pruning ineffective. PSP is most effective in cases where utilities between neighboring strategy profiles (i.e., where only

	$\varepsilon \leq 0.125$		$\varepsilon \leq 0.25$		$\varepsilon \leq 0.5$		$\varepsilon \leq 1.0$	
Bound	Hoeffding	Emp. Bennett	Hoeffding	Emp. Bennett	Hoeffding	Emp. Bennett	Hoeffding	Emp. Bennett
Game/Algorithm	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}	GS; PSP; ε_{PSP}
Congestion Games (5 facilities)	3,051; **1,654**; 0.08	3,051; **1,449**; 0.00	762; **464**; 0.17	762; **364**; 0.01	190; **146**; 0.34	190; **93**; 0.01	**47**; 58; 0.70	47; **25**; 0.04
Zero-Sum Games (30 strategies)	2,841; **1,691**; 0.08	2,841; **1,383**; 0.00	710; **502**; 0.17	710; **349**; 0.01	177; **166**; 0.35	177; **90**; 0.01	**44**; 62; 0.71	44; **25**; 0.04
Random Games (30 strategies)	2,841; **1,666**; 0.08	2,841; **1,375**; 0.00	710; **491**; 0.17	710; **347**; 0.01	177; **159**; 0.35	177; **90**; 0.01	**44**; 58; 0.71	44; **25**; 0.04
Congestion Games (4 facilities)	622; **492**; 0.09	622; **438**; 0.00	156; **138**; 0.17	156; **110**; 0.01	**39**; 41; 0.35	39; **28**; 0.01	**10**; 15; 0.71	10; **8**; 0.04
Zero-Sum Games (20 strategies)	1,171; **829**; 0.09	1,171; **708**; 0.00	293; **240**; 0.17	293; **179**; 0.01	**73**; 77; 0.35	73; **46**; 0.01	**18**; 28; 0.71	18; **13**; 0.04
Random Games (20 strategies)	1,171; **809**; 0.09	1,171; **698**; 0.00	293; **232**; 0.17	293; **176**; 0.01	**73**; **73**; 0.35	73; **45**; 0.01	**18**; 25; 0.71	18; **12**; 0.04
Congestion Games (3 facilities)	**114**; 145; 0.09	**114**; 135; 0.00	**29**; 40; 0.18	**29**; 34; 0.01	**7**; 12; 0.36	**7**; 9; 0.02	**2**; 4; 0.73	**2**; **2**; 0.05
Zero-Sum Games (10 strategies)	**254**; 268; 0.09	254; **242**; 0.00	**63**; 73; 0.18	63; **61**; 0.01	**16**; 22;0.36	16; **15**;0.02	**4**; 7; 0.73	**4**; **4**; 0.05
Random Games (10 strategies)	**254**; **254**; 0.09	254; **233**; 0.00	**63**; 69; 0.18	63; **59**; 0.01	**16**; 21;0.36	16; **15**; 0.02	**4**; 7; 0.72	**4**; **4**; 0.05
Congestion Games (2 facilities)	**17**; 37; 0.09	**17**; 37; 0.00	**4**; 10; 0.19	**4**; 9; 0.01	**1**; 3; 0.38	**1**; 2; 0.02	**1**; **1**; 0.76	**1**; **1**; 0.05
Zero-Sum Games (5 strategies)	**54**; 94; 0.09	**54**; 89; 0.00	**13**; 25; 0.18	**13**; 22; 0.01	**3**; 7; 0.37	**3**; 6; 0.02	**1**; 2; 0.75	**1**; **1**; 0.05
Random Games (5 strategies)	**54**; 83; 0.09	**54**; 90; 0.00	**13**; 22; 0.18	**13**; 20; 0.01	**3**; 6; 0.37	**3**; 5; 0.02	**1**; 2; 0.74	**1**; **1**; 0.05

Table 3.3: PSP's sample efficiency. Numbers of samples are reported in tens of thousands. The values in bold are smaller than their counterparts; as ε is fixed, they indicate the more sample efficient algorithms.

one player's strategy differs) are distinct enough that pruning is possible. Arguably, this kind of structure is common in practice, where, fixing all other players' strategies, one player's strategy (like defect in the Prisoners' Dilemma) can yield very different utilities than neighboring strategies (like cooperate). Finally, our PSP algorithm does not compare favorably to the baseline in games where there are few strategies, and hence few opportunities to prune.

3.3.7 pySEGTA

We carried out our experiments in a Python library we developed and named pySEGTA, for *statistical* EGTA. [5] pySEGTA interfaces with both GAMUT and Gambit, exposing simple interfaces by which users can generate games (GAMUT), learn them (via our learning algorithms, for example), and solve them (Gambit). As the logic concerning game implementation is entirely separate from game learning and/or solving, pySEGTA can be used to analyze arbitrarily complex simulation-based games with arbitrarily complex strategies. pySEGTA already affords access to most GAMUT games, and is designed to be easily extensible to interface with other game generators. To do so only requires describing a game's structure (number of players, and per-player numbers of strategies), and implementing one query method, which takes as input a strategy profile and returns sample utilities for all players at the given strategy profile. pySEGTA also includes parameterizable implementations of both GS and PSP, and was designed with extensibility in mind, so that other users can incorporate their learning algorithms as they are developed. Our intent is that pySEGTA ease the work of benchmarking empirical game-theoretic learning algorithms.

[5] pySEGTA is publicly accessible at `http://github.com/eareyan/pysegta`.

3.4 Chapter Summary

In this chapter, we presented and evaluated a methodology for learning games that cannot be expressed analytically. On the contrary, it is assumed that a black-box simulator is available that can be queried to obtain noisy samples of utilities. In many simulation-based games of interest, a running assumption is that queries to the simulator are exceedingly expensive, so that the time and effort required to obtain sufficiently accurate utility estimates dwarves that of any other relevant computations, including equilibrium computations. This condition holds in meta-games like Starcraft [121], for example, where agents choices comprise a few high-level heuristic strategies, not intractably many low-level game-theoretic strategies. Thus, our primary concern is to limit the need for sampling, while still guaranteeing that we are estimating a game well.

We developed an algorithm that progressively samples a game, all the while pruning strategy profiles: i.e., ceasing to estimate those strategy profiles that provably (with high probability) do not comprise any (approximate) equilibria. In extensive experimentation over a broad swath of games, we show that this algorithm makes frugal use of samples, often requiring far fewer to learn to the same—or even a better—degree of accuracy than a variant of the sampling algorithm in Tuyls et al. [121]'s, which serves as a baseline. Finally, we develop pySEGTA, a Python library that interfaces with state-of-the-art game-theory software, and which can serve as a standard benchmarking environment in which to test empirical-game theoretic algorithms.

While we consider games with no (or prohibitively expensive) analytical description in this chapter, we still assume that their rules remain fixed in our analysis. In other words, the rules of a simulation-based game do not change between calls to their simulation. Our goal then was to estimate their equilibria. A natural question then arises: can we design games such that agents' equilibria behavior leads to desirable outcomes, even if there is no analytical description for either the game or the equilibria? We tackle this question in the next chapter, where we detail our empirical mechanism design methodology.

Chapter 4

Empirical Mechanism Design

In this chapter, we present our contributions to the empirical mechanism design literature. The contents of this chapter are an extended version of the published paper *Parametric Mechanism Design under Uncertainty* [125].

4.1 Preliminaries

In chapter 3, we developed an EGTA framework for approximating the equilibria of simulation-based games. In this chapter, we extend our EGTA framework into a methodology for parametric empirical mechanism design (EMD). In parametric EMD, there is an overall (parameterized) mechanism (e.g., a second price auction with reserve prices as parameters). The choice of parameters then determines a mechanism (e.g., the reserve price being $10 instead of $100). Given a mechanism, the equilibria of the ensuing game serve as a prediction of the state one can reasonably hope the system to arrive at after agents have had a chance to interact. The designer can then evaluate how desirable an equilibrium is according to some well-defined metric (e.g., the welfare accrued at equilibrium in a second price auction with some reserve price). In what follows, we refer to this metric as the designer's objective function.

At a high level, the above description of parametric EMD can be viewed as bilevel search problem. First, the problem of searching for the optimal mechanism's parameter. Second, for each candidate parameter, the search for equilibria of the ensuing game. Abstractly, letting θ_i denote a mechanism's parameter and f the designer's objective function, Figure 4.1 shows an schematic view of parametric

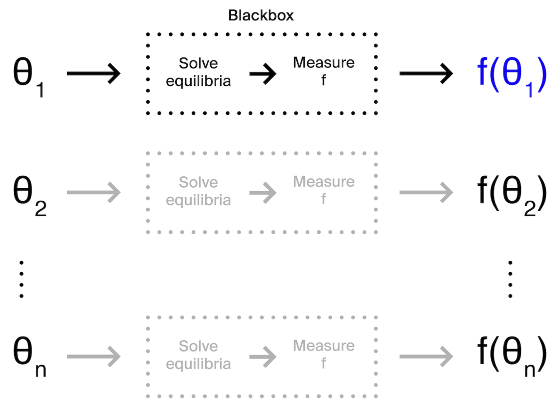


Figure 4.1: Parametric EMD as a bilevel search problem. First, the problem of searching through the mechanism's parameter space Θ to maximize some objective function f. Second, for each parameter $\theta \in \Theta$, one must solve for the game's equilibria, itself a search problem.

EMD. Note that for each candidate mechanism parameter, θ_i, we depict the search for equilibria and measurement of $f(\theta_i)$ as the result of querying a black-box. More specifically, said black-box's output could be given by the methodology in chapter 3, where we focused on approximating Nash equilibria of simulation-based games.

Nash equilibria, however, are not guaranteed to exist, except in mixed strategies (i.e., by allowing for the possibility randomization), and mixed strategy equilibria are notoriously difficult to compute [39]. Since equilibria computation is an integral part of parametric EMD where equilibria might need to be computed a large number of times, we next explore alternative solutions concepts that are both amenable to ε-uniform approximations and computationally tractable.

Solution concept	*Existence?*	*Computationally Tractable?*	*Statistically Tractable?*
Nash	Always	No	Yes[1]
Pure Nash	Sometimes	Yes	Yes
Sink	Always	Yes	No[2]
Strongly Connected Components	Always	Yes	Yes[3]

Table 4.1: Classification of some solution concepts. A solution concept is computationally tractable if it can be found in time polynomial in the input game's size. A solution concept exists for a class of games if every game in the class exhibits it. Statistical tractability varies by solution concept, see theorem 1 and theorem 6.

[1]See theorem 1 which also covers the case of pure Nash equilibria.

[2]See example 3.

[3]In a sense to be made precise in theorem 6.

4.2 Best-Response Graphs and Strongly Connected Components

To conduct mechanism design (empirical or not), one must decide the class of solution concept used to predict player's equilibria behavior. If we hope to scale *empirical* mechanism design to real-world applications, we must pay attention not only to a solution concept's existence but also to its computational and statistical tractability. Table 4.1 summarizes key properties of some solution concepts of interest in this thesis. Concretely, the table shows whether the given solution concept exists and its computational and statistical tractability. Note that we deem a solution concept computationally tractable if it can be found in time polynomial in the input game's size[4], but different solution concepts might have different requirements to be considered statistically tractable[5].

As shown in Table 4.1, an alternative to (mixed) Nash that is easier to compute and always exist are *sink* equilibria [52]. The sink equilibria are the sinks (i.e., strongly connected components without any outgoing edges) of what is called the game's *better-response graph* (BRG). This is a directed graph whose nodes are strategy profiles (one strategy per agent), and where each edge indicates that an agent would deviate from that node to the one to which it points. It turns out that sink equilibria, while efficiently computable, are not readily amenable to ε-uniform approximations, as shown in example 3. Nonetheless, we next show that if we take as our solution concept the larger set of all *strongly connected components* (SCCs) of a game's BRG, we find a solution concept that is both approximable and computationally tractable (Theorem 6).

Remark. Our main motivation for using SCC was to demonstrate our rich methodology end-to-end without diving too deeply into any one solution concept's intricacies. For this purpose, we wanted a solution concept that was both approximable and relatively easy to compute. Hence, we devised SCC, a generalization of sink equilibrium, which is both approximable (sink is not) and amenable to fast computation (Nash is not). We want to stress that our general methodology extends to other solution concepts, but so far, we have found that one would either have to give up either approximability (which is undesirable from a statistical point of view) or efficient computation (which is undesirable if we wish to scale EMD to real-world scenarios).

[4] Consistent with our methodology in chapter 3, we assume a fixed set of strategies, and thus, computational tractability is with respect to the size of the game with this fixed strategy set.

[5] Nash equilibria are statistically tractable in the sense of theorem 6. Sink equilibria are not statistically tractable as shown in example 3, but strongly connected components (SCC) are, in the sense of theorem 6.

We begin by presenting basic definitions building to Theorem 6.

Definition 9 (ε-Better Response)

An ε-better response for agent p at strategy profile $\boldsymbol{s}$ is a strategy profile $\boldsymbol{s}^* = (s_1, \ldots, s_p^*, \ldots, s_{|P|})$ where agent p plays $s_p^* \in S_p$, and all agents other than p play s_j, such that $\boldsymbol{u}_p(\boldsymbol{s}^*) + \varepsilon \geq \boldsymbol{u}_p(\boldsymbol{s})$.

Definition 10 (ε-Better Response Graph)

An ε-*better response graph*, $B_\varepsilon(\Gamma) = (\mathcal{V}, \mathcal{E}_\varepsilon)$, is a directed graph with a node for each strategy profile $\boldsymbol{s} \in \boldsymbol{S}$, and an edge $(\boldsymbol{s}, \boldsymbol{s}')$ iff $\boldsymbol{s}'$ is an ε-better response for some agent at $\boldsymbol{s}$.

Example 2 (Better-Response Graph)

The Prisoner's Dilemma game and its corresponding better-response graph are shown in Figure 4.2. Nodes are labeled by strategy profiles (e.g., CC), and edges are labeled by color, with red (blue) corresponding to the row (column) player. Strictly speaking, in a graph, there are no multiple edges. We add them anyways as visual aids.

	C	D
C	3, 3	0, 5
D	5, 0	1, 1

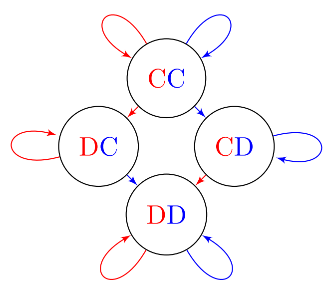
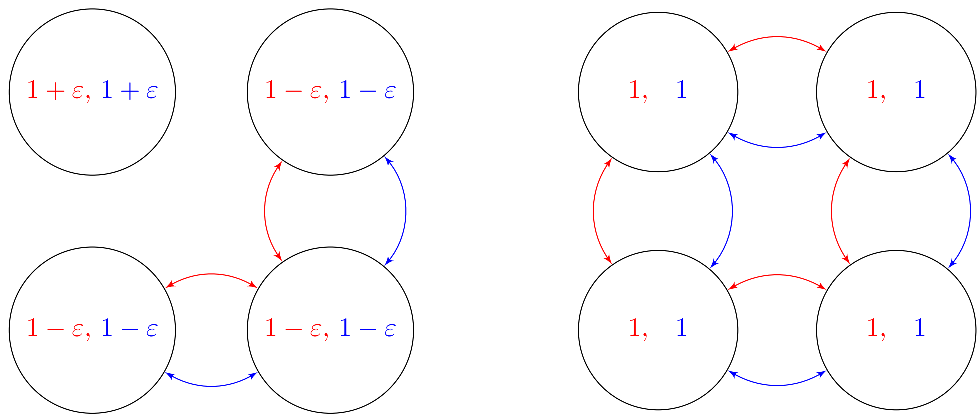


Figure 4.2: Prisoners' Dilemma's better-response graph.

Definition 11 (Strongly Connected Component)

Let $G = (V, E)$ be a directed graph. A *strongly connected component* (SCC) of G is a set of nodes $U \subseteq V$ s.t. for all $u, v \in U$, there exists a path from u to v, and one from v to u.



(a) BRG of game Γ_a with 2 players and 2 strategies (b) BRG of game Γ_b with 2 players and 2 strategies

Figure 4.3: An example showing that sink equilibria are inapproximable. To avoid clutter, we do not draw self-loops in the best-response graphs.

Example 3 (Inapproximability of sink equilibria)

Recall that a game's sink equilibrium is a strongly connected component of its corresponding better-response graph with no outgoing edges, i.e., edges that connect to other components. Figure 4.3 shows two games, Γ_a and Γ_b. Game Γ_a (subfigure (a)) is compatible with game Γ_b (subfigure (b)), and they are uniform approximations of each other, i.e., $\|\Gamma_a - \Gamma_b\|_\infty \leq \varepsilon$. Note that Γ_a has two sink equilibria: one consisting of the top-left profile and another consisting of the remaining three profiles. The entire BRG of Γ_b is the unique sink equilibrium of Γ_b.

It is not hard to see that this example can extend arbitrarily so that the top-left strategy profile of Γ_a remains its unique sink equilibrium. Coupled with a game analogous to Γ_b where all utilities but those in top-left are $1-\varepsilon$, we see that a sink equilibrium can degrade arbitrarily. More precisely, we see that, even if a game is close to another, the sink equilibria given by some approximation can contain an arbitrary number of profiles not in the original sink.

In example 3, we see that the requirement of having no outgoing edges can be easily lost in an approximated game, even if the approximation is very close. In fact, the example shows that this is the case for any $\varepsilon > 0$. Instead, we next consider *ε-SCC equilibrium* as a weaker notion of sink equilibria that lifts the requirements of components having no outgoing edges. Formally, an *ε-SCC equilibrium* is a SCC of an ε-better response graph. We denote by $\text{SCC}_\varepsilon(\Gamma)$ the set of all ε-SCC equilibria of Γ. Similarly, an *ε-sink equilibrium* is a SCC of an ε-better response graph.

Theorem 6 (Approximate Equilibria)**.** *Let* $\varepsilon > 0$*. If* Γ' *is a uniform* ε*-approximation of* Γ*, then* $SCC_0(\Gamma) \rightsquigarrow SCC_{2\varepsilon}(\Gamma') \rightsquigarrow SCC_{4\varepsilon}(\Gamma)$*, where* $\mathcal{A} \rightsquigarrow \mathcal{B}$ *means that for all* $A \in \mathcal{A}$*, there exist* $B \in \mathcal{B}$*, such that* $A \subseteq B$*.*

The keen reader might find similarities between Theorem 6 (approximation guarantees for SCC) and Theorem 1 (approximation guarantees for Nash equilibria). We show Theorem 6 via a lemma concerning the edge set of the better-responses graphs of uniform approximations.

Lemma 2 (ε-BRG edges containment.)**.** *Let* $\varepsilon > 0$*. If* Γ' *is a uniform approximation of* Γ*, then* $\mathcal{E}_0(\Gamma) \subseteq \mathcal{E}_{2\varepsilon}(\Gamma') \subseteq \mathcal{E}_{4\varepsilon}(\Gamma)$*.*

Proof. If $(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{E}_0(\Gamma)$, then there exists p such that $\boldsymbol{u}_p(\boldsymbol{t}) \geq \boldsymbol{u}_p(\boldsymbol{s})$. The following chain of reasoning then holds: $\boldsymbol{u}'_p(\boldsymbol{t}) + \varepsilon \geq \boldsymbol{u}_p(\boldsymbol{t}) \geq \boldsymbol{u}_p(\boldsymbol{s}) \geq \boldsymbol{u}'_p(\boldsymbol{s}) - \varepsilon$, where the first and last inequalities follow from the uniform approximation assumption. Hence, $\boldsymbol{u}'_p(\boldsymbol{t}) \geq \boldsymbol{u}'_p(\boldsymbol{s}) - 2\varepsilon$, and thus $(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{E}_{2\varepsilon}(\tilde{\Gamma})$. Now, starting from the assumption that $(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{E}_{2\varepsilon}(\tilde{\Gamma})$, the following chain of reasoning also holds: $\boldsymbol{u}_p(\boldsymbol{t}) + \varepsilon \geq \boldsymbol{u}'_p(\boldsymbol{t}) \geq \boldsymbol{u}'_p(\boldsymbol{s}) - 2\varepsilon \geq (\boldsymbol{u}_p(\boldsymbol{s}) - \varepsilon) - 2\varepsilon = \boldsymbol{u}_p(\boldsymbol{s}) - 3\varepsilon$. Hence, $\boldsymbol{u}_p(\boldsymbol{t}) \geq \boldsymbol{u}_p(\boldsymbol{s}) - 4\varepsilon$, and thus $(\boldsymbol{s}, \boldsymbol{t}) \in \mathcal{E}_{4\varepsilon}(\Gamma)$. □

Proof of Theorem 6. We now show that Theorem 6 follows from Lemma 2. We must show that any SCC of $B_0(\Gamma)$ remains strongly connected in $B_{2\varepsilon}(\Gamma')$. Consider $Z \in B_0(\Gamma)$. Since all the edges in $\mathcal{E}_0(\Gamma)$ are also present in $\mathcal{E}_{2\varepsilon}(\Gamma')$, it follows that any path connecting two nodes in Z is preserved in $\mathcal{E}_{2\varepsilon}(\Gamma')$. Consequently, Z remains strongly connected in $B_{2\varepsilon}(\Gamma')$. Similarly, any SCC of $B_{2\varepsilon}(\Gamma')$ remains strongly connected in $B_{4\varepsilon}(\Gamma)$. □

Having shown that SCCs are statistically tractable, we now tackle the problem of optimizing the designer's objective function when viewed as a black-box optimization problem. The designer function maps a strategy profile to a number that measures the designer's satisfaction if the game reaches said strategy profile at equilibrium. In the next section, we set aside equilibria estimation and focus on optimizing the designer's function for noisy measurements. Later, in section 4.4, we present our final EMD methodology, which combines both our EGTA methodology, using SCC as our solution concept, with the methods of the next section to find near-optimal mechanisms.

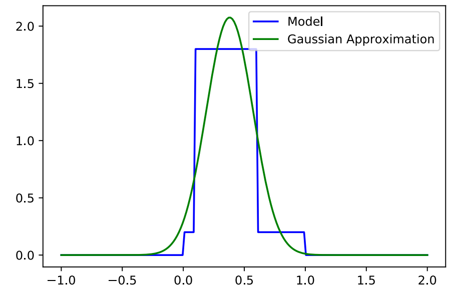


Figure 4.4: The Gaussian approximation of a 90% confidence interval on $[0.1, 0.6]$ where F ranges over $[0, 1]$.

4.3 Black-Box Optimization with Noisy Measurements

In this section, we define two generic black-box optimization problems, and two corresponding algorithmic solutions. The first algorithm is primarily of theoretical interest; the second is heuristic, but more practical. Later, using the fact that empirical mechanism design is an instance of black-box optimization, we apply our heuristic approach to two EMD applications.

Definition 12 (Optimization with noisy measurements (OwNM))

Given a design space Θ, an objective function $F : \Theta \to \mathbb{R}$, a noise model $\mathscr{D}$, and a measurement operator $M : \Theta \to P_F$, where P_F is the space of all possible probability distributions over the range of F, find $\theta^* \in \arg\max_{\theta \in \Theta} F(\theta)$.

We are concerned with a specific form of measurements, which produce *piecewise constant uniform* noise. This noise model is that which results from probably approximately correct (PAC)-style guarantees of the form, "accuracy is achieved up to additive error ε with probability $1 - \delta$" [123].

More formally, we assume the measurement operator M returns $\hat{F}(\theta)$ along with an additive error bound ε that holds with probability $1 - \delta$. In other words, the algorithm outputs a $1 - \delta$ confidence interval $[c_1, c_2]$ of width 2ε centered at $\hat{F}(\theta)$. Now assuming the range of $F(\theta)$ is $[c_-, c_+]$, and letting $\Delta \doteq c_+ - c_-$, we take as a sample measurement of the pdf $p_F(x)$ the following:

$$\hat{p}_F(x) = \begin{cases} \frac{\delta}{\Delta - 2\varepsilon} & c_- \leq x < c_1 \\ \frac{1-\delta}{2\varepsilon} & c_1 \leq x \leq c_2 \\ \frac{\delta}{\Delta - 2\varepsilon} & c_2 < x \leq c_+ \\ 0 & \text{otherwise} \end{cases}, \tag{4.1}$$

Figure 4.4 depicts an example of such a distribution. Intuitively, this distribution captures our complete ignorance about the value of the objective function, except that it lies somewhere in the interval $[c_1, c_2]$ with probability $1 - \delta$, and elsewhere with probability δ. This model is only valid if both lower and upper bounds of the objective function are known and finite, but we use this same information to achieve our PAC guarantees anyway.

Algorithm 3 PAC-OwNM

---

procedure PAC-OwNM$(\Theta, F, \mathscr{D}, \varepsilon, \delta, \Delta) \to (\hat{\theta}^*, \hat{F}(\hat{\theta}^*))$
  input:
    finite design space Θ
    designer's parameterized black-box objective function F
    distribution $\mathscr{D}$
    error tolerance ε
    failure probability δ
    range Δ
  output:
    maximizing parameter $\hat{\theta}^*$
    designer's approximate black-box objective value $\hat{F}(\hat{\theta}^*)$.
  $m \leftarrow \left\lceil (\Delta/\varepsilon)^2 \ln(2|\Theta|/\delta)/2 \right\rceil$ ▷ Hoeffding bound
  $\boldsymbol{X} \leftarrow \mathscr{D}^m$ ▷ Draw m samples from $\mathscr{D}$
  for $\theta \in \Theta$ **do**
    $\hat{F}[\theta] \leftarrow$ Measure$(\theta, F, \boldsymbol{X})$
  end for
  $\hat{\theta}^* \leftarrow \arg\max_{\theta \in \Theta} \hat{F}[\theta]$
  return $(\hat{\theta}^*, \hat{F}[\hat{\theta}^*])$
end procedure

4.3.1 Exhaustive Search

In the special case where the parameter space is finite and is searched exhaustively, it is straightforward to extend PAC guarantees on multiple independent measurements to a global guarantee across the search space. Algorithm 3 presents such an exhaustive search, and Theorem 7, which again invokes Hoeffding's inequality, describes the guarantee it achieves.

Consider a design space Θ and an objective function $F : \Theta \mapsto [c_-, c_+]$, with $\Delta \doteq c_+ - c_-$. Let $\hat{F}_1(\theta), \ldots, \hat{F}_m(\theta)$ be a sequence of m i.i.d. samples of $F(\theta)$ drawn from distribution $\mathscr{D}$.

Hoeffding's inequality [62] upper bounds the probability that the absolute difference between the empirical mean and its expected value exceeds ε as

$$Pr_{\boldsymbol{X}\sim\mathscr{D}}\left(\left|\mathop{\mathbb{E}}_{\mathscr{D}}[F(\theta)] - \frac{1}{m}\sum_{j=1}^{m}\hat{F}_j(\theta)\right| \geq \varepsilon\right) \leq 2e^{-2\varepsilon^2 m/\Delta^2}$$

Theorem 7 (PAC-Exhaustive)**.** *Consider an OwNM problem s.t.* $\theta^* \in \arg\max_{\theta\in\Theta} F(\theta)$*, and assume the measurement noise is uniform piecewise constant. Algorithm 3 applied to such a problem outputs parameter* $\hat{\theta}^*$ *and value* $\hat{F}[\hat{\theta}^*]$ *such that* $\left|F(\theta^*) - \hat{F}[\hat{\theta}^*]\right| \leq \varepsilon$ *with probability at least* $1-\delta$*, where* $\delta = \sum_{\theta\in\Theta} 2e^{-2\varepsilon^2 m/\Delta^2}$.

Proof. Algorithm 3 explores the entire parameter space. By applying a union bound to Hoeffding's inequality, it follows that all confidence intervals (CI) hold simultaneously, with probability $1-\delta$.

The algorithm then returns the maximum measurement, namely $\hat{F}[\hat{\theta}^*]$. We can bound the difference between this output and the optimal value $F(\theta)^*$ as follows:

$$-\varepsilon \leq F(\theta^*) - (F(\hat{\theta}^*) + \varepsilon) \leq F(\theta^*) - \hat{F}[\hat{\theta}^*] \leq (\hat{F}[\theta^*] + \varepsilon) - \hat{F}[\hat{\theta}^*] \leq \varepsilon$$

On the left, we used the CI around $\hat{F}[\hat{\theta}^*]$ and the fact that $F(\theta^*)$ is optimal; on the right, we used the CI around $\hat{F}[\theta^*]$ and the fact that $\hat{F}[\hat{\theta}^*] \geq \hat{F}[\theta^*]$. Therefore, $\left|F(\theta^*) - \hat{F}[\hat{\theta}^*]\right| \leq \varepsilon$. □

To summarize, given ε, δ, and Δ, Algorithm 3 calculates the requisite number of samples m to ensure that the output is accurate up to additive error ε with probability $1-\delta$.

4.3.2 Heuristic Search

Algorithm 3 *exhaustively* searches the whole design space. But this is impossible for uncountable and continuous spaces, and becomes computationally prohibitive very fast even for countable, discrete finite spaces. Hence, we seek a methodology that can find a good approximation of θ^* using limited computational resources. That is the search problem we address presently. The problem can be stated in general terms as follows:

Definition 13 (Budget-constrained optimization with noisy measurements (BCOwNM))

Given a design space Θ, an objective function $F : \Theta \to \mathbb{R}$, a noise model $\mathscr{D}$, and a measurement operator $M : \Theta \to P_F$, where P_F is the space of all possible probability distributions over the range of F, approximate

$$\theta^* \in \arg\max_{\theta \in \Theta} F(\theta)$$

as well as possible, invoking M no more than some budget $B \in \mathbb{N}$ times.

Bayesian optimization (BO) is a common tool used to solve BCOwNM problems. BO works by constructing and maintaining a probabilistic model of the objective function. After sampling, the probabilistic model is updated accordingly. This model is used to decide where to take the next measurement, until the budget is exhausted.

Most implementations of BO employ a *Gaussian Process* (GP) to model the uncertainty surrounding the objective function. Technically, a GP is a collection of random variables, any finite number of which are jointly distributed by a Gaussian [103]. BO with a GP model has been shown to outperform state-of-the-art global optimization methods in a number of benchmark problems (e.g., [118]).

Standard GPs can handle measurements with i.i.d. Gaussian noise by adding a diagonal term to the covariance matrix [103]. But there is no easy way to incorporate general noise models into GPs. We incorporate piecewise constant uniform noise into a GP, heuristically, using the Gaussian that best approximates $\hat{p}_F(x)$, by minimizing the Kullback-Leibler divergence, D_{KL}.

Definition 14 (Best Approximating Gaussian)

Given a continuous (discrete) random variable x with pdf (pmf) p, the best approximating Gaussian q^* is one s.t.

$$q^*(x) \in \arg\min_{q(x)} D_{\mathrm{KL}}\left[p(x), q(x)\right]$$

The following proposition is straightforward.

Proposition 1. *Given any distribution $\hat{p}_F(x)$ in the form of Equation 4.1, the best approximating Gaussian has mean $\mu^* = c_1$ and variance $\sigma^* = 0$ when $c_1 = c_2$, otherwise they are given by:*

$$\mu^* = \frac{\alpha}{2}\left(c_1^2 + c_+^2 - c_-^2 - c_2^2\right) + \frac{\beta}{2}\left(c_2^2 - c_1^2\right)$$
$$\sigma^* = \sqrt{\frac{\alpha}{2}\left(c_1^3 + c_+^3 - c_-^3 - c_3^2\right) + \frac{\beta}{3}\left(c_2^3 - c_1^3\right) - \mu^{*2}}$$

where $\alpha \doteq \frac{\delta}{\Delta - 2\hat{\varepsilon}}$ and $\beta \doteq \frac{1-\delta}{2\hat{\varepsilon}}$. It is easy to show that μ^* and σ^* are precisely the mean and variance of $\hat{p}_F(x)$.

For any valid confidence interval (i.e., one that lies completely within the range of the function F), the square root operand is necessarily positive, which means that a real-valued solution always exists. The singular case where $\hat{\varepsilon} = 0$ (i.e., $c_1 = c_2$) occurs only if $c_1 = c_+$ or $c_2 = c_-$.

There are (at least) two ways to utilize Proposition 1 within BO. The first is to assume Gaussian noise with mean μ^* and variance σ^*, i.e., take μ^* as the measured value and σ^* as the noise in the measurement process. This is the best possible white noise approximation. We refer to this approach as $\mathcal{GP}$-$\mathcal{N}$.

One shortcoming of $\mathcal{GP}$-$\mathcal{N}$ is that repeated measurements at the same value of θ are likely to produce very similar confidence intervals that do not significantly improve the accuracy of the probabilistic model. Nonetheless, we consider $\mathcal{GP}$-$\mathcal{N}$ as a baseline. As an alternative heuristic approach, we assume Gaussian noise with mean μ^* and variance 0, thereby avoiding repeated measurements. We refer to this approach as $\mathcal{GP}$-$\mathcal{M}$.

$\mathcal{GP}$-$\mathcal{M}$ may seem incorrect as compared to $\mathcal{GP}$-$\mathcal{N}$, but the width of the confidence interval, 2ε, is independent of θ and known, given the mean of the GP, so it is possible to recover the *correct* posterior at any point by adding to the posterior at that point a diagonal matrix that encodes the disregarded variance. But as we use the GP only to guide the search, this correction is not required.

The main advantage of $\mathcal{GP}$-$\mathcal{M}$ over the more straightforward $\mathcal{GP}$-$\mathcal{N}$ approach is improved exploration. As a second heuristic approach in this same spirit, we consider a third variant, which we call $\mathcal{GP}$, in which the variance is again zero, but the mean is set to the mean of the confidence interval, namely $\hat{F}(\theta)$. This approach, while simple and intuitive, is not the best possible Gaussian approximation, since μ^* need not equal $\hat{F}(\theta)$.

4.4 EMD: Putting it all together

Next, we describe how EMD can be viewed as an instance of black-box optimization. We then proceed to apply the aforementioned BO heuristics to two EMD applications.

Let Θ be an abstract design space over which a mechanism designer is free to choose parameters $\theta \in \Theta$. Conditioned on θ, we denote by Γ_θ the ensuing θ-parameterized game where, in the definition of a normal-form game (def. 1), we augment all strategies and utilities to depend on θ. We define $\Gamma_\theta \doteq \langle P, \boldsymbol{S}_\theta, \boldsymbol{u}_\theta(\cdot) \rangle$, where $\boldsymbol{S}_\theta$ and $\boldsymbol{u}_\theta(\cdot)$ denote the dependency of strategies and utilities on θ.

Overloading notation, we write $f(\boldsymbol{s}(\theta); \Gamma_\theta)$ to denote the value of the designer's objective in game Γ_θ at strategy profile $\boldsymbol{s}(\theta)$. In optimizing f, the designer assumes that players will play at (or near) equilibrium. So, as θ varies, the value of f likewise varies, as $\boldsymbol{s}(\theta)$ potentially moves from one equilibrium to another.

While solution concepts are meant to be predictive—that is, to predict the outcome of a game—most yield sets of equilibria, rather than unique predictions. Accordingly, a mechanism designer often faces a choice. In this work, we assume they choose a worst-case outcome, minimizing the value of f over the set $\mathrm{E}(\Gamma_\theta)$ of equilibria of Γ_θ. We denote this worst-case objective by $F^{\mathrm{E}}(\theta; \Gamma_\theta) = \min_{\boldsymbol{s} \in \mathrm{E}(\Gamma_\theta)} f(\boldsymbol{s}; \Gamma_\theta)$. More generally, for $\varepsilon \geq 0$, letting $\mathrm{E}_\varepsilon(\Gamma_\theta)$ denote the set of equilibria of Γ_θ up to ε, we define $F^{\mathrm{E}}_\varepsilon(\theta; \Gamma_\theta) = \min_{Z \in \mathrm{E}_\varepsilon(\Gamma_\theta)} \min_{\boldsymbol{s} \in Z} f(\boldsymbol{s}; \Gamma_\theta)$. (N.B. We usually write $\boldsymbol{s} \in \boldsymbol{S}_\theta$, suppressing the dependency of strategies on θ, because strategies always depend on θ, and θ is usually clear from context.)

The *worst-case EMD problem* is to optimize the black-box, worst-case objective function $F^{\mathrm{E}}(\theta; \Gamma_\theta)$. Our approach to this problem is to learn the designer's objective f up to some additive error ε, and then optimize the corresponding objective $F^{\mathrm{E}}_\varepsilon(\theta; \Gamma_\theta)$. We now argue that this approach is reasonable, assuming SCCs as the equilibria.

Definition 15 ((θ, ε)-approximable objective function)

Let Γ_θ and Γ'_θ be two θ-parameterized games, and let f be a designer's objective. If $\max_{\theta \in \Theta, \boldsymbol{s} \in \boldsymbol{S}_\theta} |f(\boldsymbol{s}; \Gamma_\theta) - f(\boldsymbol{s}; \Gamma'_\theta)| \leq \varepsilon$, then we say that the objective f is ε-*approximable.*

The next theorem states that, for $\varepsilon > 0$, when f is ε-approximable, a solution to the ε-worst-case EMD problem is an ε-approximate solution to the exact problem, assuming SCC as the equilibrium.

Theorem 8. *Let θ^* optimize $F^{SCC}(\theta; \Gamma_\theta)$, and let θ^*_ε optimize $F^{SCC}_\varepsilon(\theta; \Gamma_\theta)$, for $\varepsilon > 0$. If f is ε-approximable, then $\left|F^{SCC}(\theta^*; \Gamma_\theta) - F^{SCC}_\varepsilon(\theta^*_\varepsilon; \Gamma_\theta)\right| \leq \varepsilon$.*

We show Theorem 8 via an intermediate lemma, which shows that when f is ε-approximable, $F^{\mathrm{E}}_\varepsilon(\theta; \Gamma)$ well approximates $F^{\mathrm{E}}(\theta; \Gamma)$, assuming E = SCC: i.e.,

$$F^{\mathrm{E}}(\Gamma) \doteq F^{\mathrm{SCC}}(\theta; \Gamma) = \min_{Z \in \mathrm{E}(\theta; \Gamma)} \min_{\boldsymbol{s} \in Z} f(\boldsymbol{s}; \Gamma)$$

Lemma 3. *If f is ε-approximable, then for all $\theta \in \Theta$, $\left|F^{SCC}(\theta; \Gamma_\theta) - F^{SCC}_\varepsilon(\theta; \Gamma'_\theta)\right| \leq \varepsilon$.*

Proof. First, note that the strongly connected components of a graph partition its vertices. Therefore, for all $\varepsilon \geq 0$, it holds that,

$$F^{\mathrm{SCC}}_\varepsilon(\theta; \Gamma_\theta) = \min_{Z \in \mathrm{SCC}_\varepsilon(\Gamma_\theta)} \min_{\boldsymbol{s} \in Z} f(\boldsymbol{s}; \Gamma_\theta) = \min_{\boldsymbol{s} \in \boldsymbol{S}_\theta} f(\boldsymbol{s}; \Gamma_\theta)$$

Now,

$$\begin{aligned} F^{\mathrm{SCC}}(\theta; \Gamma_\theta) - F^{\mathrm{SCC}}_\varepsilon(\theta; \Gamma'_\theta) &= \min_{\boldsymbol{s} \in \boldsymbol{S}_\theta} f(\boldsymbol{s}; \Gamma_\theta) - \min_{\boldsymbol{s} \in \boldsymbol{S}_\theta} f(\boldsymbol{s}; \Gamma'_\theta) \\ &\leq \min_{\boldsymbol{s} \in \boldsymbol{S}_\theta} f(\boldsymbol{s}; \Gamma_\theta) - \min_{\boldsymbol{s} \in \boldsymbol{S}_\theta} \left(f(\boldsymbol{s}; \Gamma_\theta) - \varepsilon\right) \\ &= \varepsilon. \end{aligned}$$

A similar argument shows that $F^{\mathrm{SCC}}(\theta; \Gamma_\theta) - F^{\mathrm{SCC}}_\varepsilon(\theta; \Gamma'_\theta) \geq -\varepsilon$, which completes the proof. ☐

(Proof of Theorem 8). First, observe the following: $F^{\mathrm{SCC}}(\theta^*; \Gamma_\theta) \geq F^{\mathrm{SCC}}(\theta'; \Gamma_\theta) \geq F^{\mathrm{SCC}}_\varepsilon(\theta'; \Gamma'_\theta) - \varepsilon$. The first inequality follows from the fact that θ^* maximizes F, and the second, from Lemma 3. Via analogous reasoning, $F^{\mathrm{SCC}}_\varepsilon(\theta'; \Gamma'_\theta) \geq F^{\mathrm{SCC}}_\varepsilon(\theta^*; \Gamma'_\theta) \geq F^{\mathrm{SCC}}(\theta^*; \Gamma_\theta) - \varepsilon$, assuming θ' maximizes $F^{\mathrm{SCC}}_\varepsilon$. Together, these two inequalities imply the result. ☐

Note that Theorem 8 also holds for the best-case EMD problem, assuming SCC as the equilibrium, which is to optimize $F^{\mathrm{SCC}} = \max_{Z \in \mathrm{SCC}_\varepsilon(\Gamma_\theta)} \max_{\boldsymbol{s} \in Z} f(\boldsymbol{s}; \Gamma_\theta)$.

Algorithm 4 EMD_Measure

procedure EMD_MEASURE($\Gamma_\theta, f, \boldsymbol{X}$) $\rightarrow \hat{F}(\theta)$

  input:

    parameterized simulation-based game Γ_θ
    designer's objective function f
    m samples $\boldsymbol{X}$

  output:

    Designer's approximate black-box, worst-case objective value $\hat{F}(\theta)$

  for $p \in P$ and $\boldsymbol{s} \in \boldsymbol{S}_\theta$ **do**
    $\hat{\boldsymbol{u}}_p(\boldsymbol{s}) \leftarrow \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{u}_p(x_j, \boldsymbol{s})$
  end for
  $\hat{\Gamma}_\theta \leftarrow \langle P, \boldsymbol{S}_\theta, \hat{\boldsymbol{u}}(\cdot) \rangle$ ▷ Empirical parameterized game
  $\mathrm{SCC}_\varepsilon(\hat{\Gamma}_\theta) \leftarrow$ FINDSCCS($\hat{\Gamma}_\theta, \varepsilon$) ▷ Find SCCs of empirical game
  $\hat{F}(\theta) \leftarrow \min_{Z \in \mathrm{SCC}_\varepsilon(\hat{\Gamma}_\theta)} \min_{\boldsymbol{s} \in Z} f(\boldsymbol{s}; \hat{\Gamma}_\theta)$
  return $\hat{F}(\theta)$
end procedure

In sum, it suffices to approximate f up to ε to obtain an ε-approximation of $F^{\mathrm{SCC}}(\theta; \Gamma)$. Furthermore, it suffices to optimize $F_\varepsilon^{\mathrm{SCC}}(\theta; \Gamma_\theta)$ up to ε to obtain an ε-approximate solution to the worst-case EMD problem. Such estimates can be computed with high probability via Algorithm 3, using the measurement procedure defined in Algorithm 4. But to obtain the usual guarantees, the number of samples m calculated in Line 4 of Algorithm 3 must be updated to $\left\lceil (\Delta/\varepsilon)^2 \ln(2(|\Theta|+|\Gamma|)/\delta)/2 \right\rceil$ to account for a union bound over both the mechanism's and the game's parameters.

4.5 Experiments

In this section, we present experimental results using our equilibrium estimation and BO search methodology. We note that all BO algorithms use a Mattern Kernel [103] in the underlying Gaussian Process and the expected improvement [118] acquisition function. We use $\mathcal{GP}$-$\mathcal{N}$, $\mathcal{GP}$, and uniform sampling over the range $[0.5, 1.5]^8$ as our baselines. Note that uniform sampling is a competitive search strategy for hyper-parameter optimization [20].

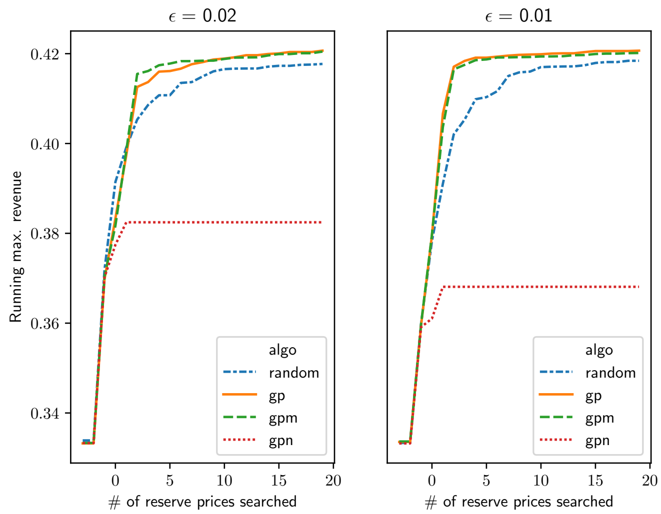


Figure 4.5: FPA search for optimal reserve price, $\delta = 0.1$

We experiment with our methodology in two settings. The first is in first-price auctions where analytical solutions are known, and the condition of Definition 15 holds, as the revenue curve is a known to be a continuous function of the reserve price. The second is in a setting with no known analytical solutions, and with no guarantees as to whether condition of Definition 15 holds or not. Nonetheless, we report anecdotally that an empirical analog of the condition is often met in our experiments.

4.5.1 First-price Auctions

As our first application domain, we consider *first-price auctions* in the standard independent, private value model [73]. There is one good up for auction, and n bidders, with bidders' values drawn from some joint distribution G. Bidders submit bids $b_1, \ldots, b_n$, and the highest bidder wins and pays his bid. As a proof of concept, we apply our BO search heuristics in this domain, in attempt to maximize revenue as a function of a *reserve price*, r, below which the good will not be sold.

In this controlled setting, analytical solutions are known in certain special cases. For example, when bidders' valuations are drawn i.i.d. uniformly on $[0, 1]$, the (unique) symmetric Bayes-Nash strategy [73] is given by:

$$s(v) \doteq \mathbb{1}(v \geq r) \left[{r^n}/{v^{n-1}} + {(n-1)(v^n - r^n)}/{n v^{n-1}} \right]$$

Assuming $n = 2$ bidders play this equilibrium, the optimal reserve price is $1/2$, which yields a revenue of $5/12$.

Figure 4.5 depicts a comparison of $\mathcal{GP}$-$\mathcal{M}$ and $\mathcal{GP}$ against the baselines, $\mathcal{GP}$-$\mathcal{N}$ and uniform sampling, for $\varepsilon \in \{0.02, 0.01\}$. We note that a similar behavior was observed for $\varepsilon = 0.03$. Specifically, we plot a running maximum of revenue as a function of the number of reserve prices searched. Each point is an average over 30 trials, where a single trial consists of exploring 20 different reserve prices with different initial random points. All the BO heuristics are initialized with the same three initial random points.

Our BO heuristic $\mathcal{GP}$-$\mathcal{M}$ consistently outperforms uniform sampling and $\mathcal{GP}$-$\mathcal{N}$ (whose predicted anomalous behavior was explained in Section 4.3), taking fewer measurements to achieve near-optimal values of revenue. The close performance between $\mathcal{GP}$-$\mathcal{M}$ and $\mathcal{GP}$ can be explained by the fact that δ is so small that the mean of a measurement's confidence interval is close to μ^*.

4.5.2 Advertisement Exchange

In this section, we illustrate our EMD methodology in a one-shot advertisement exchange game,[6] a game rich enough that analytic solutions are not readily available, but nonetheless amenable to probabilistic analysis through sampling and simulation. The game is designed to model a scenario common in electronic advertisement platforms, such as Google's AdWords© and Amazon Sponsored Brands©.

We begin by describing, at a high-level,[7] the elements and dynamics of the game, and the strategies used by agents. We conclude with experiments showing that our algorithms more quickly accrue higher revenue than a baseline in an 8-dimensional parameter space.

AdX game in a nutshell.

In the AdX game, agents play the role of advertising networks competing in an exchange for opportunities to show Internet users impressions (i.e., advertisements) needed to fulfill advertising campaigns. These impression opportunities (henceforth, impressions) are sold in real-time sequential auctions, as users arrive, but agents submit their bids in advance, placing potentially different bids for different classes of users.

[6] A simplification of TAC AdX [116].
[7] A complete mathematical formalization of this game is given in chapter 5.

Users are characterized by *attributes*: e.g. GENDER, INCOME, AGE, etc., each of which is characterized by a set of *attribute values*: e.g., {YOUNG, OLD}. Each user belongs to a *market segment*, which is a list of values for (not necessarily all) attributes: e.g., ⟨FEMALE, YOUNG⟩.

A market segment $\mathbb{M}$ *matches* another market segment $\mathbb{M}'$ if all the attribute values in $\mathbb{M}'$ are present in $\mathbb{M}$. For example, the market segments ⟨FEMALE, YOUNG, HIGH INCOME⟩ and ⟨FEMALE, YOUNG, LOW INCOME⟩ both match ⟨FEMALE, YOUNG⟩; however, ⟨FEMALE⟩ does not match ⟨FEMALE, YOUNG⟩, since attribute value YOUNG is missing from the former.

In the AdX game, agents target (i.e., aim to display advertisements to) some market segments, but not others, as described by their (single) advertising campaign. Each advertising campaign $C_j = \langle I_j, \mathbb{M}_j, R_j \rangle$ demands $I_j \in \mathbb{N}$ impressions in total, procured from users belonging to any market segment $\mathbb{M}'$ that matches the campaign's desired market segment $\mathbb{M}_j$. A campaign's budget $R_j \in \mathbb{R}_+$ is the maximum amount the advertiser is willing to spend on those impressions. From the agent's point of view, the budget maps to its potential revenue.

To a first approximation then, agent j's goal, is to procure at least I_j impressions matching market segment $\mathbb{M}_j$ to fulfill C_j's so that it can earn revenue, which depends on R_j. (Note that impressions in market segments other than $\mathbb{M}_j$ yield no value.) More specifically, the value of a number of procured impressions z is determined via a sigmoidal function that maps z to a percentage of the budget: i.e., small values of z yield a small percentage of R_j, while values close to I_j yield values close to R_j. The non-linearity inherent in this function models complementarities, [8] because it incentivizes agents to focus either on completely satisfying a campaign's demand, or not to bother satisfying it at all.

The AdX game is a one-shot game. To play, agents submit bids and spending limits for each market segment. Then, simulated users arrive at random, from the various market segments. For each user that arrives from market segment $\mathbb{M}$, a second-price auction with a publicly known reserve price $r_{\mathbb{M}} \in \mathbb{R}_+$ is held among all agents whose bids match $\mathbb{M}$, and who have not yet reached their spending limit in $\mathbb{M}$. Agent j's utility is computed as the difference between its revenue and its expenditure for all auctions.

[8] A good is a complement of another if the value of acquiring both is strictly greater than the value of acquiring either one of them but not the other: e.g., the value of acquiring a pair of shoes (left and right) is usually greater than the value of only the left or the right one.

Strategies.

Designing bidding strategies for electronic ad exchanges (and ad auctions, more generally) is an active research area (e.g. [30, 43, 50]). The goal of these experiments is not to investigate the performance of state-of-the art bidding strategies, though this is certainly an interesting future research direction, but rather to test the methodology developed in this thesis. Toward this end, we devised two heuristics which, we call Walrasian Equilibrium (WE) and Waterfall (WF). Detailed descriptions of these heuristics are provided in chapter 5. Here, we present only their main ideas.

At a high level, both heuristics work by building an off-line model of the market induced by the AdX game, which is then used to compute an allocation (assignment of impressions to campaigns) and prices (for impressions), based on which bids and limits are determined. The strategies differ in how this outcome—the allocation and prices—are computed. In a nutshell, the WE strategy searches for an outcome which forms a near-Walrasian equilibrium, i.e., an outcome whose prices provide little incentive for agents to relinquish their allocation. The study of equilibria and near-equilibrium computation in combinatorial markets (ad exchanges, being one example) is an active research area (e.g., [25, 44, 48]) with promising applications (e.g., [4, 79, 82]). The WF strategy works by simulating the arrival of impression opportunities in a fixed order that is endogenously determined by the campaigns present in the market. We call this strategy Waterfall because impressions are allocated *to* campaigns in descending order of budget-per-impression, and *from* market segments in ascending order of second-highest budget-per-impression. Hence, the final bid prices can be visualized as a descending waterfall-like structure.

4.5.3 Experimental Setup

We assumed three user attributes: gender, age, and income level, each with two values. We then simulated a fixed number of impression opportunities, namely $K = 500$, distributed over 8 market segments, corresponding to all the possible combinations of attribute values: $\{\textsc{Male}, \textsc{Female}\} \times \{\textsc{Young}, \textsc{Old}\} \times \{\textsc{Low Income}, \textsc{High Income}\}$. The distribution π over these impression opportunities was constructed from statistical data at www.alexa.com [116].

Each agent's campaign $C_j = \langle I_j, \mathbb{M}_j, R_j \rangle$ is determined as follows: A market segment $\mathbb{M}_j$ is drawn uniformly over all 20 possible market segments corresponding to combinations of user attributes of size 2 (e.g., $\langle \textsc{Male}, \textsc{Young} \rangle$) and 3 (e.g., $\langle \textsc{Male}, \textsc{Young}, \textsc{Low Income} \rangle$). Given $\mathbb{M}_j$, the demand

$I_j := {}^{K\pi_{\mathbb{M}_j}}/_N$, where $K\pi_{\mathbb{M}_j}$ is the expected size of market segment $\mathbb{M}_j$, and N is the number of agents in the game. Given I_j, the budget R_j is a noisy signal of the demand modeled by a beta distribution: $R_j \sim I_j(\mathcal{B}(\alpha = 10, \beta = 10) + 0.5)$.

The task is to find an 8-dimensional vector of reserve prices $\boldsymbol{r} \in \mathbb{R}^8$, consisting of one reserve price per market segment, that maximizes the auctioneer's revenue, up to a desired accuracy ε and specified failure probability δ. We experiment with $N = 4$ agents, and two possible strategies; in particular, for all p, $S_p = \{\text{WE, WF}\}$. Since by construction these strategies never bid higher than a campaign's budget-per-impression, ${}^{R_j}/_{I_j} = \mathcal{B}(\alpha = 10, \beta = 10) + 0.5 \in [0.5, 1.5]$, we bound the search for each reserve price to this same range, and hence search the space $[0.5, 1.5]^8$.

Even in this bounded region, the complexity of this task is substantial: for each candidate vector of reserve prices, we must first learn the corresponding game it induces, and then solve for the equilibria of this game. In this thesis, we use Algorithm 4 for this purpose, which finds the set SCC_ε using Tarjan's algorithm [119], and then returns the minimum revenue among all SCC_ε equilibria. This is a computationally intensive task. Indeed, it took approximately 5 days to obtain these results using 4 machines, each with an E5-2690v4 (2.60GHz, 28Core) Xeon processor, and 1,536 GB of memory.

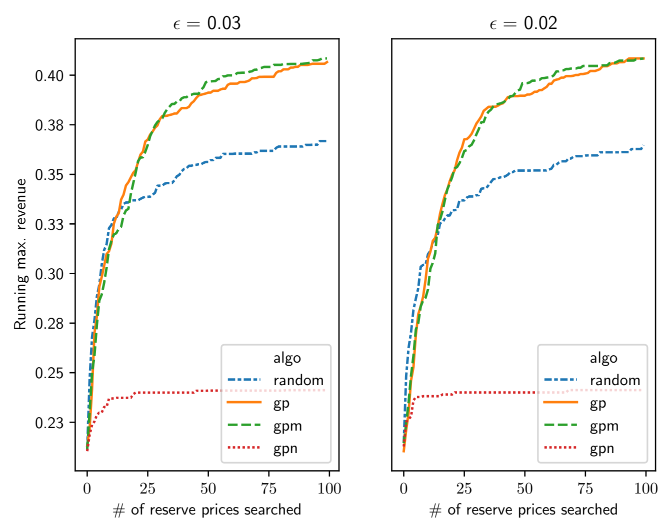


Figure 4.6: AdX search for optimal reserve prices, $\delta = 0.1$

Algorithm 5 Experimental Procedure

procedure EXPERIMENTALPROCEDURE($\Gamma_{\boldsymbol{r}}, f, \boldsymbol{X}, n_t, n_i, n_s$) $\rightarrow \{\mu_s\}_{s=1}^{n_s}$

  input:

   simulation-based game parameterized by reserve prices $\Gamma_{\boldsymbol{r}}$
   designer's objective function f
   samples $\boldsymbol{X}$
   number of trials n_t
   number of initial reserves n_i
   number of search steps n_s

  output:

   running maximum, $\{\mu_s\}_{s=1}^{n_s}$, for n_s search steps, averaged across n_t trials.

  for $t = 1, \ldots, n_t$ **do** ▷ n_t trials.
   $\Theta = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{n_i}\} \leftarrow$ GETINITIALRESERVEPRICES(t) ▷ Get n_i initial reserve price vectors.
   N.B. $\boldsymbol{r}_j = \left\langle r_j^1, \ldots, r_j^8 \right\rangle \in \mathcal{R}$, where $r_j^{\mathbb{M}}$ is the reserve price in market segment $\mathbb{M}$.
   for $\boldsymbol{r} \in \mathcal{R}$ **do**
    $\hat{F}_t(\boldsymbol{r}) \leftarrow$ EMD_MEASURE($\Gamma_{\boldsymbol{r}}$, REVENUE, $\boldsymbol{X}$) ▷ Call to Algorithm 4
   end for
   $\mathcal{G}_t \leftarrow$ INITIALIZEGAUSSIANPROCESS($\{\boldsymbol{r}, \hat{F}_t(\boldsymbol{r})\}_{\boldsymbol{r} \in \mathcal{R}}$)
   $\nu_t = []$ ▷ Store trial t's running maximum in a list
   for $s = 1, \ldots, n_s$ **do** ▷ n_s search steps.
    $\boldsymbol{r}_s^t \leftarrow$ ACQUISITIONFUNCTION($\mathcal{G}_t$) ▷ We use Expected Improvement
    $\hat{F}_t(\boldsymbol{r}_s^t) \leftarrow$ EMD_MEASURE($\Gamma_{\boldsymbol{r}_s^t}$, REVENUE, $\boldsymbol{X}$) ▷ Call to Algorithm 4
    $\nu_t[s] = \max_{k \leq s} \hat{F}_t(\boldsymbol{r}_k^t)$ ▷ Compute current running maximum
    $\mathcal{G}_t \leftarrow$ UPDATEGAUSSIANPROCESS($\mathcal{G}_t, \{\boldsymbol{r}_s^t, \hat{F}_t(\boldsymbol{r}_s^t)\}$)
   end for
  end for
  for $s = 1, \ldots, n_s$ **do**
   $\mu_s \leftarrow \frac{1}{n_t} \sum_{t=1}^{n_t} \nu_t[s]$ ▷ Averaging across trials
  end for
  return $\{\mu_s\}_{s=1}^{n_s}$
end procedure

4.5.4 Experimental Results

Experimental results in Figure 4.6 were obtained by running the procedure described in Algorithm 5, for each of $\mathcal{GP}$, $\mathcal{GP}$-$\mathcal{M}$, and $\mathcal{GP}$-$\mathcal{N}$. This procedure takes as input a simulator for parameterized game Γ_θ, a designer's objective function f (in this case, revenue), samples $\boldsymbol{X}$, a number of trials n_t, a number of initial reserve prices n_i, and a number of search steps n_s. It then outputs a sequence of length n_s, $\{\mu_s\}_{s=1}^{n_s}$, where each μ_s is the running maximum of the revenue obtained by to step s, averaged across n_t independent trials. More specifically, letting t index trials and s index steps,

$\mu_s = \frac{1}{n_t} \sum_{t=1}^{n_t} \max_{k \leq s} \hat{F}_t(\boldsymbol{r}_k^t)$, where $\hat{F}_t(\boldsymbol{r}_k^t)$ is the revenue assuming reserve prices $\boldsymbol{r}_k^t$, which denotes the reserve prices explored during the kth step of the tth trial.

Figure 4.6 summarizes our AdX results. Specifically, we plot a running maximum of revenue as a function of the number reserve prices evaluated so far. Each plot is an average over 30 trials, where a single trial consists of exploring 100 different vectors of reserve prices, initialized at random. To ensure a fair comparison, all search algorithms are fed the same 10 initial points during each trial. These results are compared to uniform sampling. As in the first-price auction experiment, our BO heuristic $\mathcal{GP}$-$\mathcal{M}$ outperforms uniform sampling and $\mathcal{GP}$-$\mathcal{N}$, taking fewer measurements to achieve higher values of revenue. $\mathcal{GP}$-$\mathcal{M}$'s performance is on par with that of $\mathcal{GP}$, which once again can be explained by a sufficiently small value of δ. Note, however, the apparent difficulty in optimizing revenue in this game; $\mathcal{GP}$-$\mathcal{M}$ and $\mathcal{GP}$ take significantly more measurements to outperform the baselines. Nonetheless, these results show that, even under a fairly constrained budget, our BO heuristics are effective as compared to our baselines.

4.6 Chapter Summary

Our EMD methodology tackles the fundamental problem of optimizing a designer's objective function in parametric systems inhabited by strategic agents. The fundamental assumptions are: 1) players play equilibria among an *a priori* known set of strategies, which in general depend on the parameters of the system; and 2) there is no analytical description available of either the strategies or the system, but only a simulator (i.e., a procedural description) capable of producing data about game play. This framework captures modern, computationally intensive systems, such as electronic advertisement exchanges. The challenge then is two-fold: first, one must *learn equilibria* of a game for a fixed setting of parameters, and second, one must *search* the space of parameters for those that maximize the designer's function.

Our main contribution is a PAC-style framework to solve the former, while for the latter we enhance standard search routines for black-box optimization problems to include piecewise constant noise, precisely the kind of noise that characterizes PAC learners. We prove theoretical guarantees on the quality of the learned parameters when the parameter space is finite. We also demonstrate the practical feasibility of our methodology, first in a setting with known analytical solutions, and then in a stylized but rich model of advertisement exchanges with no known analytical solutions—precisely the kind of setting for which we devised our methodology—and show that we can find solutions of higher quality than standard baselines.

We emphasize that evaluating the quality of a mechanism produced by our EMD methodology is itself a challenging task. In addition to developing an ad exchange model, we created two bidding heuristic strategies that operate in it. Beyond using them in our experimental evaluations in this chapter, this model and heuristics are intricate and of interest on their own. Thus, we detailed them in the next chapter.

Chapter 5

Heuristic Bidding for Electronic Ad Auctions

In this chapter, we present our contributions to the design of heuristic bidding strategies for electronic advertisement auctions. The contents of this chapter are complementary to the following published papers: *Principled Autonomous Decision Making for Markets* [3], and *On Approximate Welfare-and Revenue-Maximizing Equilibria for Size-Interchangeable Bidders* [4] (arXiv version [5]).

5.1 Preliminaries

This chapter presents a detailed mathematical description of the *one-day electronic advertisement game* (one-day AdX) that we devised and used to evaluate our EMD methodology in section 4.5.2. We call this simplified game the one-day AdX to remark that our game models a single day (or any other arbitrary unit of time) of the more challenging TAC AdX game [116]. Still, the challenge faced by agents (or ad networks) in our one-day AdX game is substantial. Namely, given a campaign, the challenge for an ad network is to procure enough display ad impression opportunities to fulfill the campaign at the lowest cost possible in the face of competition from other ad networks. We call this problem the *ad network problem.* The one-day AdX game is a simplified version of the TAC AdX that allows us to focus more clearly on the ad network problem.

Towards developing bidding heuristics for the ad network problem, we first model the market induced by ad exchanges as a combinatorial market [12, 16, 31, 81]. In our formulation, we model impression opportunities as the supply side of the market and ad networks as the demand side. Since demographics characterize impression opportunities, we model each demographic as a single type of good for which there are potentially multiple units available to consumers.

Having modeled the ad exchange market, we then turn our attention to designing bidding heuristics for the ad network problem. At a high level, our heuristics first compute a prediction of the market outcome, which consists of an allocation of impression opportunities to ad networks together with prices for each impression opportunity. An agent that uses our heuristic bidding strategies submits bids and budget limits to the ad exchange based on said predictions.

We present two different approaches to compute predictions of market outcomes, (1) a market-equilibrium-based approach (WE heuristic), and (2) an auction-simulation approach (WF heuristic). In approach (1), our goal is to compute a (near) competitive equilibrium of the market as defined classically for markets with divisible goods by French economist Léon Walras [132] and more recently extended to markets with indivisible goods (known in the literature as *combinatorial markets* [21]). In approach (2), we simulate the workings of an ad exchange by computing an allocation and bids that simulate the second-price auctions used to place an advertisement on the internet[1]. For tractability, we assume users arrive grouped by their market segments, but the order of arrival is endogenous, meaning determined by the algorithm.

5.2 Game Elements

An advertising **campaign** $C = \langle I, \mathbb{M}, R \rangle$ demands $I \in \mathbb{N}$ impressions in total, procured from users belonging to any market segment $\mathbb{M}' \in M$ that *matches* the campaign's desired market segment $\mathbb{M} \in M$. (The match function, and market segments, are defined precisely below.) A campaign's budget $R \in \mathbb{R}_+$ is the maximum amount the advertiser is willing to spend on those impressions. The **one-shot AdX game** is played by a set A of agents, where each $j \in A$ is endowed with a campaign $C_j = \langle I_j, \mathbb{M}_j, R_j \rangle$.

The AdX game is a game of incomplete information. In general, agents do not know one another's campaigns. Consequently, agent j's strategy set consists of all functions mapping agent j's campaign

[1]For example, Google's® AdSense https://www.google.com/adsense/.

C_j (i.e., its private information) to tuples of the form $\langle \boldsymbol{b}^j, \boldsymbol{l}^j \rangle$. The first component of this tuple is a bid vector $\boldsymbol{b}^j = \left\langle b_1^j, b_2^j, \cdots, b_{|M|}^j \right\rangle$, where $b_{\mathbb{M}}^j \in \mathbb{R}_+$ is agent j's bid for impressions that match market segment $\mathbb{M}$. The second component is a limit vector $\boldsymbol{l}_j = \left\langle l_1^j, l_2^j, \cdots, l_{|M|}^j \right\rangle$, where $l_{\mathbb{M}}^j \in \mathbb{R}_+$ is agent j's spending limit on impressions that match $\mathbb{M}$.

Example 4 (A two-player, one-shot AdX game)

The following is an example of a two-player ($A = \{1, 2\}$), one-shot AdX game. Let $M = \{\mathbb{M}_1, \mathbb{M}_2, \mathbb{M}_3\}$ be a set of three market segments. The first agent's campaign is defined as $C_1 = \langle 10, \mathbb{M}_1, 100 \rangle$. In other words, given a budget of \$100, the first agent's goal is to procure at least ten different impression opportunities, i.e., ten opportunities to show advertisements to users. Crucially, each advertisement must be shown to a user that matches market segment $\mathbb{M}_1$. The second agent's campaign is defined as $C_2 = \langle 5, \mathbb{M}_3, 25 \rangle$. Likewise, agent 2 has a budget of \$25, and its goal is to show advertisement impressions to at least five users, each of which matches market segment $\mathbb{M}_3$.

Agent's $j \in A = \{1, 2\}$ strategy in this game is a function, $C_j \mapsto \left\langle \left\langle b_1^j, b_2^j, b_3^j \right\rangle, \left\langle l_1^j, l_2^j, l_3^j \right\rangle \right\rangle$, from its campaign to bids and limits. For example, suppose that agent 1 wishes to bid, for each impression it demands, an amount proportional to its budget. Suppose also that the agent is only interested in procuring impressions that match $\mathbb{M}_1$ (for simplicity, suppose in this example that $\mathbb{M}_k$ matches only with itself for $k = 1, 2, 3$). Additionally, the agent is unwilling to spend any amount above its budget for impressions that match its market segment. For impressions that do not match its market segment, the agent is unwilling to spend any positive amount of its budget. The agent can realize this strategy with the following mapping,

$$\langle 10, \mathbb{M}_1, 100 \rangle \mapsto \langle \langle {}^{\$100}/_{10}, 0, 0 \rangle, \langle \$100, 0, 0 \rangle \rangle$$

Note that the following mapping implements the same kind of strategy for agent 2,

$$\langle 5, \mathbb{M}_3, 25 \rangle \mapsto \langle \langle 0, 0, {}^{\$25}/_{5} \rangle, \langle 0, 0, \$25 \rangle \rangle$$

We now proceed to define utilities. We start by more precisely defining what it means for a user's impression to match an advertiser's campaign. Let $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ be a set of k *attributes*. Each attribute is defined as a discrete set of *attribute values*. A *market segment* $\mathbb{M}$ is defined as a vector of attribute values, $\mathbb{M} = \langle a_1, \ldots, a_k \rangle$, where $a_i \in \mathcal{A}_i$.

Example 5 (Attributes, attribute values, and market segments)

Consider the following set of attributes $\mathcal{A} = \{\textsc{Gender}, \textsc{Income}, \textsc{Age}\}$.

Define, for each attribute, values as follows,

$\textsc{Gender} = \{\textsc{Male}, \textsc{Female}\}$, $\textsc{Income} = \{\textsc{High}, \textsc{Low}\}$, and $\textsc{Age} = \{\textsc{Old}, \textsc{Young}\}$.

In this case, there are $2^3 = 8$ possible market segments, namely, $M =$

$\{\langle\textsc{Male}, \textsc{High}, \textsc{Old}\rangle, \langle\textsc{Male}, \textsc{High}, \textsc{Young}\rangle, \langle\textsc{Male}, \textsc{Low}, \textsc{Old}\rangle, \langle\textsc{Male}, \textsc{Low}, \textsc{Young}\rangle,$

$\langle\textsc{Female}, \textsc{High}, \textsc{Old}\rangle, \langle\textsc{Female}, \textsc{High}, \textsc{Young}\rangle, \langle\textsc{Female}, \textsc{Low}, \textsc{Old}\rangle, \langle\textsc{Female}, \textsc{Low}, \textsc{Young}\rangle\}$

We augment each attribute $\mathcal{A}_i$ with a special symbol $*$ to create the *augmented attribute* $\mathcal{A}_i^* = \mathcal{A}_i \cup \{*\}$. An *augmented market segment* $\mathbb{M}^*$ is a vector of augmented attributes' values, $\mathbb{M}^* = \langle a_1^*, \ldots, a_k^*\rangle$, where $a_i^* \in \mathcal{A}_i^*$. We denote by M the set of all possible augmented market segments.

Given two augmented market segments $\mathbb{M}$ and $\mathbb{M}'$, we say that $\mathbb{M}$ *matches* $\mathbb{M}'$, and write $\mathbb{M} \preceq \mathbb{M}'$, if and only if for $i = 1, \ldots, k$, if $a_i' \neq *$, then $a_i' = a_i$. In words, $\mathbb{M} \preceq \mathbb{M}'$ if and only if

1. In case the i-th attribute value of $\mathbb{M}'$ is the special symbol $*$, the i-th attribute value of $\mathbb{M}$ is allowed to be any value.

2. In case the i-th attribute value of $\mathbb{M}'$ is not $*$, the i-th attribute value of $\mathbb{M}'$ is the same as the i-th attribute value of $\mathbb{M}$.

Example 6 (Matching market segments)

Using attributes as defined in example 5, the following are matches:

1. $\langle\textsc{Female}, \textsc{High}, \textsc{Young}\rangle \preceq \langle *, *, *\rangle$.
2. $\langle\textsc{Female}, \textsc{High}, \textsc{Young}\rangle \preceq \langle *, \textsc{High}, \textsc{Young}\rangle$ and $\langle\textsc{Male}, \textsc{High}, \textsc{Young}\rangle \preceq \langle *, \textsc{High}, \textsc{Young}\rangle$.
3. $\langle\textsc{Female}, \textsc{Low}, \textsc{Young}\rangle \preceq \langle\textsc{Female}, *, \textsc{Young}\rangle$ and $\langle\textsc{Female}, \textsc{High}, \textsc{Young}\rangle \preceq \langle\textsc{Female}, *, \textsc{Young}\rangle$.

The following are not matches:

1. $\langle\textsc{Female}, \textsc{High}, \textsc{Young}\rangle \npreceq \langle\textsc{Male}, *, *\rangle$, since $\textsc{Female} \neq \textsc{Male}$.
2. $\langle\textsc{Male}, \textsc{High}, \textsc{Young}\rangle \npreceq \langle\textsc{Male}, *, \textsc{Old}\rangle$, since $\textsc{Young} \neq \textsc{Old}$.

The above definitions are intended to be used to describe which users' impressions match which advertisers' campaigns. Users belong to market segments (and thus, augmented market segments), and campaigns can be specified in the language of augmented market segments,[2] so these definitions are sufficient for matching impressions with campaigns. Hereafter, we drop the qualifier "augmented," as we only ever consider augmented markets segments.

Denote by $\boldsymbol{y} = \langle y_1, y_2, \ldots, y_{|M|} \rangle$ a bundle of impressions, where $y_{\mathbb{M}} \in \mathbb{N}$ denotes the number of impressions from market segment $\mathbb{M}$ in bundle $\boldsymbol{y}$. The **utility** u_j of agent j, as a function of bundle $\boldsymbol{y}$, is given by:

$$u_j(\boldsymbol{y}, C_j) = \rho(\mu(\boldsymbol{y}, C_j), C_j) - p(\boldsymbol{y}) \tag{5.1}$$

Here $p(\boldsymbol{y})$ is the total **cost** of bundle $\boldsymbol{y}$, and $\mu(\boldsymbol{y}, C_j) = \sum_{\mathbb{M} \in M} y_{\mathbb{M}} \mathbb{1}\{\mathbb{M} \preceq \mathbb{M}_j\}$ is a filtering function, which, given a bundle of impressions and a campaign, calculates the number of impressions in the bundle that match the campaign. Finally,

$$\rho(z, C_j) = \left(\frac{2R_j}{b}\right) \left(\arctan\left(\frac{bz}{I_j} - a\right) - \arctan(a)\right) \quad ,$$

where, for any nonzero $k \in \mathbb{R}$, $a = b + k$ and b is the unique solution to the equation

$$\frac{\arctan(k) - \arctan(-b)}{1 + b} = \frac{1}{1 + k^2}$$

Following Schain and Mansour [116], we use $k = 1$, which implies $a \approx -3.08577$ and $b \approx 4.08577$. Intuitively, $\rho(z, C_j)$ maps a number of impressions z to a percentage of R_j in a sigmoidal fashion: i.e., small values of z yield a small percentage of R_j, while values close to I_j yield values close to R_j. The non-linearity inherent in this function models *complementarities*,[3] because it incentivizes agents to focus either on completely satisfying a campaign's demand, or not to bother satisfying it at all. Figure 5.1 depicts a sample sigmoidal, for a campaign that demands 200 impressions. Note that from an agent's point of view, its campaign's budget maps to its potential revenue.

This concludes our description of agents' types (i.e., their private information), their strategies, and their utilities in the one-shot AdX game.

[2] Augmented market segments are also a natural way to describe users, since not all user attributes are revealed, in general.

[3] A good is a complement of another if the value of acquiring both is strictly greater than the value of acquiring either one of them but not the other: e.g., the value of acquiring a pair of shoes (left and right) is usually greater than the value of only the left or the right one.

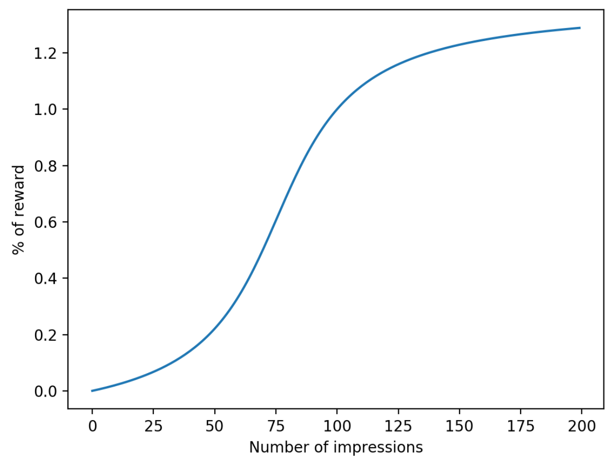


Figure 5.1: Sigmoidal for a campaign that demands 200 impressions.

5.3 Game Dynamics

As the one-shot AdX game is a one-shot game, bids and spending limits are submitted to the ad exchange only once. Still, the game unfolds in four stages.

Stage 0: The agents learn their campaigns. In the current experiments (Section 4.5.2), *we assume agents are well versed in their competition, and thus they also learn one another's campaigns.* As such, we model the agents playing a complete-, rather than an incomplete-, information game.

Stage 1: The ad exchange announces a vector of reserve prices $\boldsymbol{r} \in \mathbb{R}_+^M$, i.e., prices above which agents must bid to participate in the various auctions, where $r_{\mathbb{M}} \in \mathbb{R}_+$ is a price above which agents must bid to participate in auctions for impressions belonging to market segment $\mathbb{M} \in M$.

Stage 2: All agents submit their bids and their spending limits for all auctions: i.e., agent j submits tuple $\langle \boldsymbol{b}^j, \boldsymbol{l}^j \rangle$. As the name suggests, this limit is the *maximum* the agent is willing to spend on impressions that match $\mathbb{M}$. In case there are multiple matches, the minimum among spending limits is enforced.

Stage 3: Some random number of impressions $K \sim U[K_l, K_u]$ arrive in a random order $\left\langle \vec{\mathbb{M}} \right\rangle = \left\langle \mathbb{M}^1, \mathbb{M}^2, \ldots, \mathbb{M}^K \right\rangle$, where $\mathbb{M}^i \sim \boldsymbol{\pi} = \left\langle \pi_1, \pi_2, \ldots, \pi_{|M|} \right\rangle$, and $\pi_{\mathbb{M}}$ is the probability of $\mathbb{M}$. For each impression that arrives from market segment $\mathbb{M}$, a second-price auction with reserve price $r_{\mathbb{M}} \in \mathbb{R}_+$ is held among all agents whose bids are matched by $\mathbb{M}$, and who have not yet reached their spending limit in $\mathbb{M}$. If an agent submits multiple matching bids, it does not compete against itself; the maximum of its matching bids is entered into the auction.

Stage 4: The output of stage 3 is, for each agent j, a bundle of impressions $\boldsymbol{y}^j = \left\langle y_1^j, y_2^j, \ldots, y_{|M|}^j \right\rangle$, which is used to compute agent j's utility according to Equation (5.1). The auctioneer's revenue is defined as $\sum_j p(\boldsymbol{y}^j)$.

5.4 Strategies

We devised two heuristic strategies for the one-day AdX game. The first heuristic is a market equilibrium-based approach which we call Walrasian Equilibrium (WE). The second is an auction-simulation approach which we call Waterfall (WF)[4]. At a high level, both heuristics work by collapsing the dynamics of the one-day AdX game into a static market model, which is then used to compute an allocation (assignment of impressions to campaigns) and prices (for impressions), based on which bids and limits are determined.

5.4.1 Static Market Model

The static market model employed by both heuristics is an augmented bipartite graph $\mathcal{M} = \langle M, C, E, \vec{N}, \vec{I}, \vec{R} \rangle$, with a set of $n \in \mathbb{N}$ market segments M; a set of $c \in \mathbb{N}$ campaigns C; a set of edges E from market segments to campaigns indicating which segments match to which campaigns; supply vector $\vec{N} = \langle N_1, \ldots, N_n \rangle$, where $N_i \in \mathbb{N}$ is the number of available impressions from market segment $i \in M$; demand vector $\vec{I} = \langle I_1, \ldots, I_c \rangle$, where $I_j \in \mathbb{N}$ is the number of impressions demanded by campaign $j \in C$; and reward vector $\vec{R} = \langle R_1, \ldots, R_c \rangle$, where $R_j \in \mathbb{R}_+$ is campaign j's reward.

Let's look at an example of this construction.

[4] Not to be confused with *waterfalling*, the process by which publishers try to sell remnant inventory; see https://clearcode.cc/blog/what-is-waterfalling/.

Example 7 (Static market as a bipartite graph)

As in example 4, suppose that there are two campaigns defined as $C_1 = \langle 10, \mathbb{M}_1, 100\rangle$ and $C_2 = \langle 5, \mathbb{M}_3, 25\rangle$. Further suppose that $M = \{\mathbb{M}_1, \mathbb{M}_2, \mathbb{M}_3\}$ and that $\mathbb{M}_k$ matches only with itself for $k = 1, 2, 3$. To fully specified a market $\mathcal{M}$ from this input, we still have to define the number of available impressions for each market segment. Suppose that there are seven impressions of $\mathbb{M}_1$, three of $\mathbb{M}_2$, and six of $\mathbb{M}_3$. We can now fully specify $\mathcal{M}$.

Concretely, $\mathcal{M} = \langle M, C, E, \vec{N}, \vec{I}, \vec{R}\rangle$ where

$$M = \{\mathbb{M}_1, \mathbb{M}_2, \mathbb{M}_3\},\ C = \{C_1, C_2\},\ E = \{\{C_1, \mathbb{M}_1\}, \{C_2, \mathbb{M}_3\}\},$$
$$\vec{N} = \langle 7, 3, 6\rangle,\ \vec{I} = \langle 10, 5\rangle,\ \vec{R} = \langle \$100, \$25\rangle$$

Given a market $\mathcal{M}$, our goal is to compute both an allocation of impression opportunities to campaigns and prices that serve as predictions for bidding strategies. We denote by $\boldsymbol{X} \in \mathbb{Z}_+^n \times \mathbb{Z}_+^c$ an allocation matrix, where entry x_{ij} is the allocation (i.e., number of impressions) from market segment i assigned to campaign j. We also denote by $\boldsymbol{p} \in \mathbb{R}_+^n$ a price vector, with price p_i per impression from market segment i. Pair $(\boldsymbol{X}, \boldsymbol{p})$ is an outcome of market $\mathcal{M}$ in the sense that it resolves the utility that each agent receives in the game. As defined, this outcome is anonymously-priced because agents face the same price per impression, p_i, for market segment i. We also consider the case of personalized-priced outcomes where agents might face different prices per impression in the same market segment. Formally, let $\boldsymbol{P} \in \mathbb{R}_+^n \times \mathbb{R}_+^c$ be a price matrix, with price p_{ij} per impression from market segment i to campaign j. Then, pair $(\boldsymbol{X}, \boldsymbol{P})$ is a personalized-priced outcomes.

5.4.2 Walrasian Equilibrium Bidding Strategy

An anonymously-priced[5] Walrasian (or competitive) equilibrium (WE or CE) is a class of market outcomes of particular interest in mathematical economic theory, with a rich history that extends as far back as Walras' work. [132]. In a CE, buyers are utility-maximizing (i.e., they maximize their utilities among all feasible allocations at the posted prices) and the seller maximizes its revenue (again, over all allocations at the posted prices). A competitive equilibrium serves as an idealized prediction of a market outcome[6], and thus, our motivation to use it as a predictive model for our WE heuristic.

[5] If there are multiple copies of some homogeneous good, all copies must have the same price.
[6] Under a number of assumptions whose validity for the AdX market need to be investigated in future work.

Unfortunately, an anonymously-priced WE always exist only for a small class of combinatorial markets [69] and, consequently, there is a rich line of work dedicated to devising CE relaxations that retain some of their nice properties [28, 44, 60]. In previous work [4], we noted that a WE outcome is not guaranteed to exist in a static market model of the one-day AdX game. Still, motivated by the desired to compute outcomes close to competitive equilibria, we proposed a novel relaxation of CE, which we called *limited envy-free pricing* [4, 5].[7]. We further derived a polynomial-time[8] heuristic that computes an approximate revenue-maximizing, limited envy-free pricing. A (slight) modification of that algorithm forms the core of the WE bidding heuristic.

Next, we summarize the WE bidding heuristic we devised for the ad network problem. We defer a more in-depth treatment of competitive equilibrium theory in combinatorial markets[9], and our contributions from a statistical learning perspective, to the next chapter.

The WE bidding heuristic works as follow. First, we solve for an approximate welfare-maximizing allocation (Algorithm 6). A welfare-maximizing allocation is an allocation that maximizes the sum of agents' values. Fixing the allocation, we solve for a revenue-maximizing limited envy-free pricing, abbreviated LEFP (Algorithm 7)[10]. A LEFP relaxes the requirement that *all* agents maximize their utilities at the posted prices and instead insists that only agents that received a *non-empty bundle* maximize their utilities. The agent then bids the LEFP price of any market segment for which it has been allocated and places a limit proportional to its total spending in the segment. In summary, an agent employing the WE bidding heuristic[11] runs Algorithms 6 and 7; then, given $(\boldsymbol{X_r}, \boldsymbol{p})$, it bids p_i in all market segments i for which $x_{ij} > 0$, and places limits $x_{ij}p_i$ in all market segments i.

We provided justification for the WE heuristic in previous work [4]. Concretely, we showed in extensive experimentation using real-world web usage data obtained from `www.alexa.com` that the above two-step procedure results in market outcomes that are very close to WE. More specifically, in a setup akin to the one-day AdX game's static model, we experimentally showed that the constraints that define WE are violated with small additive errors, on average, over multiple runs.

[7] Our relaxation, in turn, builds on a previously proposed relaxation known as an envy-free pricing [60]. We also note that after we proposed our relaxation, researchers have further studied it under the name of buyer preselection [23, 24].

[8] Since electronic advertisement markets operate under tight time requirements, computational tractability is of paramount importance.

[9] In a combinatorial market, goods cannot be divided, but instead, each must be either wholly allocated to a single consumer or not allocated at all. The one-day AdX static market $\mathcal{M}$ is an example of a combinatorial market since impression opportunities cannot be divided among multiple agents.

[10] Note that Algorithms 6 and 7 slightly generalize those presented in [4], in that they handle a vector of reserve prices, one per market segment.

[11] As currently described, supply vector $\vec{N}$ remains undefined. In our experiments, we use the expected number of available impressions for each market segment. This number ignores the order of arrival of impressions and instead assumes that there are enough of them so that the exact order is not of importance but their total quantity.

Algorithm 6 Greedy allocation algorithm

procedure GreedyAllocation($\mathcal{M}, \boldsymbol{r}$)$\rightarrow \boldsymbol{X_r}$
  input:
    market $\mathcal{M}$
    reserve prices $\boldsymbol{r} \in \mathbb{R}^n_+$
  output:
    Allocation $\boldsymbol{X_r}$
  for all i, j, initialize $x_{ij} = 0$
  for $j \in C$ **do** ▷ Loop through j in descending order of bang-per-buck, i.e., R_j/I_j.
    Let $E_j = \{i \in M \mid \{i, j\} \in E \text{ and } \sum_{j=1}^{c} x_{ij} < N_i\}$
    if $\sum_{i \in E_j} N_i - \sum_{j=1}^{c} x_{ij} \geq I_j$ **then** ▷ "enough supply remains to satisfy j"
      for $i \in E_j$ **do** ▷ Loop through E_j in descending order of supply.
        $x_{ij} = \min\{I_j - \sum_{i=1}^{n} x_{ij}, N_i - \sum_{j=1}^{c} x_{ij}\}$
      end for
      if $\sum_{i=1}^{n} r_i x_{ij} > R_j$ **then** ▷ "campaign j cannot afford the assigned bundle"
        for all i: $x_{ij} = 0$ ▷ Unallocate j.
      end if
    end if
  end for
  return $\boldsymbol{X_r}$
end procedure

Algorithm 7 Linear program that computes (near) revenue-maximizing LEFP

procedure LEFP($\mathcal{M}, \boldsymbol{r}, \boldsymbol{X_r}$)$\rightarrow \boldsymbol{p}$
  input:
    Market $\mathcal{M}$
    reserve prices $\boldsymbol{r} \in \mathbb{R}^n_+$
    allocation $\boldsymbol{X}_r$
  output:
    A pricing $\boldsymbol{p}$
  maximize$_{\boldsymbol{p},\boldsymbol{\alpha}}$ $\sum_{j \in C, i \in M} x_{ij} p_i - \sum_{j \in C, (i,k) \in M \times M} \alpha_{jik}$, subject to:
  (1) $\forall j \in C$: If $\sum_{i=1}^{n} x_{ij} > 0$, then $\sum_{i=1}^{n} x_{ij} p_i \leq R_j$
  (2) $\forall i \in M, \forall j \in C$:
    If $x_{ij} > 0$ then $\forall k \in M$: If $\{k, j\} \in E$ and $x_{kj} < N_k$ then $p_i \leq p_k + \alpha_{jik}$
  (3) $\forall i \in M : p_i \geq r_i$
  (4) $\forall j \in C, \forall (i,k) \in M \times M : \alpha_{jik} \geq 0$
  return $\boldsymbol{p}$
end procedure

5.4.3 Waterfall Bidding Strategy

The Waterfall heuristic (WF) is designed to simulate the dynamics of the one-shot AdX game. In the game, the highest bid in a market segment stands for some period of time, namely until the spending limit of the winning campaign is reached, or the campaign is satisfied. Hence, prices in each market segment remain constant for some period of time at the current highest bid. As the game unfolds, more and more impressions are allocated, consuming the budgets of the winning campaigns in each market segment, until the winners drop out, and the second highest bidder is promoted. The bid of the campaign that drops out is by definition higher than the new winning bid, which again stands for some further period of time. The resulting prices thus resemble a waterfall: i.e., decreasing, and constant between decreases.

An agent employing the WF heuristic computes an allocation $\boldsymbol{X}$ and a price matrix $\boldsymbol{P}$ via Algorithm 8, given a market $\mathcal{M}$ and reserve prices $\boldsymbol{r}$. The algorithm works by simulating n second-price auctions assuming agents bid R_j/I_j, and then allocating impressions to winning campaigns from market segments sorted in ascending order by their second highest bids (breaking ties randomly),[12] and promoting campaigns as previous winners' budgets are exhausted. Given $(\boldsymbol{X}, \boldsymbol{P})$, an agent then bids p_{ij} in all market segments i for which $x_{ij} > 0$, and places limits $x_{ij}p_{ij}$ in all market segments i.

Example 8 (An example run of the WF algorithm (algorithm 8))

Consider market $\mathcal{M}$ defined as follows,

$$\mathcal{M} = \langle\{\mathbb{M}_1, \mathbb{M}_2\}, \{C_1, C_2\}, \{\{\mathbb{M}_1, C_1\}, \{\mathbb{M}_2, C_1\}, \{\mathbb{M}_1, C_2\}, \{\mathbb{M}_2, C_2\}\}, [8, 7], [10, 5], [100, 25]\rangle$$

Note that $\mathcal{M}$ consists of two market segments and two campaigns, and all market segments match all campaigns' market segments. Further consider reserve prices $\boldsymbol{r} = [1/200, 1/400]$.

When run on input $(\mathcal{M}, \boldsymbol{r})$, algorithm 8 outputs $\boldsymbol{X}, \boldsymbol{P}$ such that,

$$\boldsymbol{X} = \begin{pmatrix} 3 & 5 \\ 2 & 0 \end{pmatrix}, \boldsymbol{P} = \begin{pmatrix} 1/200 & 1/10 \\ 1/400 & 0 \end{pmatrix}$$

The algorithm first selected C_2 since its reward-to-demand ratio, $5/25$, is higher than that of C_1's, $10/100$. Campaign C_2 competes with $C_!$ on both market segments, so the algorithm (arbitrarily) selects to allocate all of C_2's demand from $\mathbb{M}_1$. Finally, the algorithm allocates to C_1.

[12] This ordering is arbitrary; another possibility is to sort market segments in *descending* order by their second highest bids.

Algorithm 8 Waterfall Algorithm

procedure WATERFALL($\mathcal{M}, \boldsymbol{r}$) $\rightarrow \boldsymbol{X}, \boldsymbol{P}$
  input: A market $\mathcal{M}$ and reserve prices $\boldsymbol{r} \in \mathbb{R}_+^n$
  output: An allocation matrix $\boldsymbol{X_r}$ and a pricing matrix $\boldsymbol{P_r}$
  $\mathscr{N} = N$ ▷ a copy of the supply vector
  $\mathscr{C} = C$ ▷ a copy of the campaigns
  $\forall i, j : x_{ij} = 0$ ▷ initialize an empty allocation matrix
  $\forall i, j : p_{ij} = \infty$ ▷ initialize an infinite price matrix
  while $\mathscr{C}$ is not empty **do**
    for $j \in \mathscr{C}$ **do**
      if $\sum_{\{i|(i,j)\in E\}} \mathscr{N}_i < I_j$ **then** ▷ "j cannot be fulfilled with remaining supply"
        $\mathscr{C} = \mathscr{C} \setminus \{j\}$ ▷ remove j from the set of candidate campaigns
      end if
    end for
    Choose $j^* \in \arg\max_{j\in\mathscr{C}}\{R_j/I_j\}$
    for $i \in \{i \mid (i, j^*) \in E$ and $\mathscr{N}_i > 0\}$ **do**
      Initialize bid vector $\mathbf{b}_i$.
      for $j \in \mathscr{C}$ **do**
        if $(i, j) \in E$ **then**
          Insert bid R_j/I_j in $\mathbf{b}_i$
        end if
      end for
      Insert the 2nd-highest bid in $\mathbf{b}_i$ into vector `2nd_HIGHEST_BIDS`.
      Insert the reserve r_i into vector `2nd_HIGHEST_BIDS`.
    end for
    $i^* \in \arg\min_i\{\texttt{2nd\_HIGHEST\_BIDS}[i]\}$ ▷ select a segment with the lowest 2nd highest bid
    $p_{i^*j^*} = \texttt{2nd\_HIGHEST\_BIDS}[i^*]$
    $x_{i^*j^*} = \min\{\mathscr{N}_{i^*}, I_{j^*} - \sum_{i=1}^n x_{ij^*}\}$ ▷ to satisfy the campaign, allocate the lessor of the impressions as needed or the remaining supply
    $\mathscr{N}_{i^*} = \mathscr{N}_{i^*} - x_{i^*j^*}$ ▷ decrement the supply
    if j^* is satisfied (i.e., $\sum_{i=1}^n x_{ij} = I_j$) **then**
      $\mathscr{C} = \mathscr{C} \setminus \{j\}$
    end if
  end while
  return $\boldsymbol{X}, \boldsymbol{P}$
end procedure

5.5 Chapter Summary

An ad exchange is a centralized platform that matches web users to advertisers who wish to promote their brand to the users. The sheer number of web users means that there is an immense opportunity to make a substantial economic impact, provided ad exchanges operate effectively. An ad exchange is effective if it matches brands to users likely to engage with it, at the lowest cost for the advertisers, while making as much profit as possible. We contend that devising precise mathematical and algorithmic models of ad exchanges' inner workings is crucial to improve them.

In this chapter, we contributed to the mathematical and algorithmic modeling of ad exchanges by proposing a stylized-but-rich model of an ad exchange's dynamics. We further contribute two heuristic bidding strategies for our ad exchange model. Our bidding heuristics rely on an additional static model of an ad exchange which agents then use to compute expected price predictions that guide their bidding behavior. We leverage insides from the literature on competitive equilibria, along with our own previous work, to design our first bidding heuristic. This heuristic computes (near) competitive equilibria of the static market model as price predictions. For our second heuristic, we take a different approach where we simulate the ad exchange dynamics to obtain price predictions. Crucially, both of our heuristics run in polynomial-time on the input market's size, a requirement of predominant importance for their applicability to real-world ad exchanges which operate under tight time requirements.

Ad exchanges are just one example of a combinatorial market where goods are indivisible. In the next chapter, we dive deeper into competitive equilibrium theory in combinatorial markets and present our contributions to it from a statistical learning perspective.

Chapter 6

Noisy Combinatorial Markets

In this chapter, we present our contributions to the statistical learning literature in the context of combinatorial markets. The contents of this chapter are an extended version of the published paper *Learning Competitive Equilibria in Noisy Combinatorial Markets* [9] (arXiv version [126]).

6.1 Preliminaries

In this chapter, we formulate a mathematical model of noisy combinatorial markets (NCM). Our goal is then to design learning algorithms with rigorous finite-sample guarantees that approximate the competitive equilibria of NCMs. First, we present tight lower- and upper-bounds on the set of CE, given uniform approximations of buyers' valuations. We then present two learning algorithms. The first one—Elicitation Algorithm; EA—serves as a baseline. It uses Hoeffding's inequality [62] to produce said uniform approximations. Our second algorithm—Elicitation Algorithm with Pruning; EAP—leverages the first welfare theorem of economics to adaptively prune value queries when it determines that they are provably not part of a CE. Finally, after establishing our algorithms' correctness, we evaluate their empirical performance using both synthetic unit-demand valuations and two spectrum auction value models.

6.2 Model

We write $\mathbb{X}_+$ to denote the set of positive values in a numerical set $\mathbb{X}$ including zero. Given an integer $k \in \mathbb{Z}$, we write $[k]$ to denote the first k integers, inclusive: i.e., $[k] = \{1, 2, \ldots, k\}$. Given a finite set of integers $Z \subset \mathbb{Z}$, we write 2^Z to denote the power set of Z.

A *combinatorial market* is defined by a set of goods and a set of buyers. We denote the set of goods by $G = [N]$, and the set of buyers by $N = [n]$. We index an arbitrary good by $j \in G$, and an arbitrary buyer by $i \in N$. A *bundle* of goods is a set of goods $S \subseteq G$. Each buyer i is characterized by their preferences over bundles, represented as a valuation function $v_i : 2^G \to \mathbb{R}_+$, where $v_i(S) \in \mathbb{R}_+$ is buyer i's value for bundle S. We assume valuations are normalized so that $v_i(\emptyset) = 0$, for all $i \in N$. Using this notation, a combinatorial market—market, hereafter—is a tuple $\mathcal{M} = (G, N, \{v_i\}_{i \in N})$.

Given a market $\mathcal{M}$, an *allocation* $\mathcal{S} = (S_1, \ldots, S_n)$ denotes an assignment of goods to buyers, where $S_i \subseteq G$ is the bundle assigned to buyer i. We consider only feasible allocations. An allocation $\mathcal{S}$ is *feasible* if $S_i \cap S_k = \emptyset$ for all $i, k \in N$ such that $i \neq k$. We denote the set of all feasible allocations of market $\mathcal{M}$ by $\mathcal{F}(\mathcal{M})$. The *welfare* of allocation $\mathcal{S}$ is defined as $w(\mathcal{S}) = \sum_{i \in N} v_i(S_i)$. A welfare-maximizing allocation $\mathcal{S}^*$ is a feasible allocation that yields maximum welfare among all feasible allocations, i.e., $\mathcal{S}^* \in \arg\max_{\mathcal{S} \in \mathcal{F}(\mathcal{M})} w(\mathcal{M})$. We denote by $w^*(\mathcal{M})$ the welfare of any welfare-maximizing allocation $\mathcal{S}^*$, i.e., $w^*(\mathcal{M}) = w(\mathcal{S}^*) = \sum_{i \in N} v_i(S_i^*)$.

A pricing profile $\boldsymbol{p} = (P_1, \ldots, P_n)$ is a vector of n pricing functions, one function $P_i : 2^G \to \mathbb{R}_+$ for each buyer, each mapping bundles to prices, $P_i(S) \in \mathbb{R}_+$. The seller's revenue of allocation $\mathcal{S}$ given a pricing $\boldsymbol{p}$ is $\sum_{i \in N} P_i(S_i)$. We refer to pair $(\mathcal{S}, \boldsymbol{p})$ as a *market outcome*—outcome, for short. Given an outcome, buyer i's utility is difference between its attained value and its payment, $v_i(S_i) - P_i(S_i)$, and the seller's utility is equal to its revenue.

In this chapter, we are interested in approximations of one market by another. We now define a mathematical framework in which to formalize such approximations. In what follows, whenever we decorate a market $\mathcal{M}$, e.g., $\mathcal{M}'$, what we mean is that we decorate each of its components: i.e., $\mathcal{M}' = (G', N', \{v_i'\}_{i \in N'})$.

It will be convenient to refer to a subset of buyer–bundle pairs. We use the notation $\mathcal{I} \subseteq N \times 2^G$ for this purpose.

Definition 16 (Compatible Markets)

Markets $\mathcal{M}$ and $\mathcal{M}'$ are *compatible* if $G = G'$ and $N = N'$.

Note that whenever a market $\mathcal{M}$ is compatible with a market $\mathcal{M}'$, an outcome of $\mathcal{M}$ is also an outcome of $\mathcal{M}'$.

Definition 17 (Market Norm)

Given two compatible markets $\mathcal{M}$ and $\mathcal{M}'$, we measure the difference between them at $\mathcal{I}$ as $\|\mathcal{M} - \mathcal{M}'\|_{\mathcal{I}} = \max_{(i,S)\in\mathcal{I}} |v_i(S) - v_i'(S)|$. When $\mathcal{I} = N \times 2^G$, this difference is precisely the infinity norm.

Equipped with a notion of distance between markets, we can now precisely define what it means for a market to be close to another.

Definition 18 (ε-Approximate Markets)

Given $\varepsilon > 0$, $\mathcal{M}$ and $\mathcal{M}'$ are called *ε-approximations* of one another if the difference between them is small: i.e., $\|\mathcal{M} - \mathcal{M}'\|_{\infty} \leq \varepsilon$.

The solution concept of interest in this chapter is competitive equilibrium[1]. A competitive equilibrium consists of two conditions: the utility-maximization (UM) condition and the revenue-maximization (RM) condition. UM ensures that the allocation maximizes buyers' utilities given the pricings, while RM ensures that the seller maximizes its utility. Together, both conditions constitute an equilibrium of the market, i.e., an outcome where no agent has an incentive to deviate by, for example, relinquishing its allocation. We now formalize this solution concept, followed by its relaxation, central when working with approximate markets.

Definition 19 (Competitive Equilibrium)

Given a market $\mathcal{M}$, an outcome $(\mathcal{S}, \boldsymbol{p})$ is a *competitive equilibrium* (CE) if:

(UM) $\forall i \in N, T \subseteq G : v_i(S_i) - P_i(S_i) \geq v_i(T) - P_i(T)$

(RM) $\forall \mathcal{S}' \in \mathcal{F}(\mathcal{M}) : \sum_{i\in N} P_i(S_i) \geq \sum_{i\in N} P_i(S_i')$

[1]For arbitrary pricing functions, a competitive equilibrium is always guaranteed to exist [22].

In a situation reminiscent of simulation-based games and Nash equilibria (chapter 3), we now need to relax competitive equilibria so that it is amenable to perturbation in a sense to be precise in theorem 9. A correct selection of approximate equilibria is crucial, as not all equilibria concepts are preserved even when introducing a small ($\varepsilon > 0$) additive noise (see chapter 4's example 3).

Definition 20 (Approximate Competitive Equilibria)

Let $\varepsilon > 0$. An outcome $(\mathcal{S}, \boldsymbol{p})$ is a ε-competitive equilibrium (ε-CE) if it is a CE in which UM holds up to ε:

ε-(UM) $\forall i \in N, T \subseteq G : v_i(S_i) - P_i(S_i) + \varepsilon \geq v_i(T) - P_i(T)$

For $\alpha \geq 0$, we denote by $\mathcal{CE}_\alpha(\mathcal{M})$ the set of all α-approximate CE of $\mathcal{M}$, i.e., $\mathcal{CE}_\alpha(\mathcal{M}) = \{(\mathcal{S}, \boldsymbol{p}) : (\mathcal{S}, \boldsymbol{p}) \text{ is a } \alpha\text{-approximate CE of } \mathcal{M}\}$. Note that $\mathcal{CE}_0(\mathcal{M})$ is the set of (exact) CE of market $\mathcal{M}$, which we denote $\mathcal{CE}(\mathcal{M})$.

Theorem 9 (Competitive Equilibrium Approximation)**.** *Let $\varepsilon > 0$. If $\mathcal{M}$ and $\mathcal{M}'$ are compatible markets such that $\|\mathcal{M} - \mathcal{M}'\|_\infty \leq \varepsilon$, then $\mathcal{CE}(\mathcal{M}) \subseteq \mathcal{CE}_{2\varepsilon}(\mathcal{M}') \subseteq \mathcal{CE}_{4\varepsilon}(\mathcal{M})$.*

Proof. We prove that: $\mathcal{CE}_\alpha(\mathcal{M}) \subseteq \mathcal{CE}_{\alpha+2\varepsilon}(\mathcal{M}')$, for $\alpha \geq 0$. This result then implies $\mathcal{CE}(\mathcal{M}) \subseteq \mathcal{CE}_{2\varepsilon}(\mathcal{M}')$ when $\alpha = 0$; likewise, it (symmetrically) implies $\mathcal{CE}_{2\varepsilon}(\mathcal{M}') \subseteq \mathcal{CE}_{4\varepsilon}(\mathcal{M})$ when $\alpha = 2\varepsilon$.

Let $\mathcal{M}$ and $\mathcal{M}'$ be compatible markets s.t. $\|\mathcal{M} - \mathcal{M}'\|_\infty \leq \varepsilon$. Suppose $(\mathcal{S}, \boldsymbol{p})$ is a α-competitive equilibrium of $\mathcal{M}$. Our task is to show that $(\mathcal{S}, \boldsymbol{p})$, interpreted as an outcome of $\mathcal{M}'$, is a $(\alpha + 2\varepsilon)$-competitive equilibrium of $\mathcal{M}'$.

First, note that the RM condition is immediately satisfied, because $\mathcal{S}$ and $\boldsymbol{p}$ do not change when interpreting $(\mathcal{S}, \boldsymbol{p})$ as an outcome of $\mathcal{M}'$. Thus, we need only show that the approximation holds for the UM condition:

$$
\begin{aligned}
v_i'(S_i) - P_i(S_i) &\geq v_i(S_i) - P_i(S_i) - \varepsilon, && \forall i, S_i && (6.1)\\
&\geq v_i(T) - P_i(T) - \alpha - \varepsilon, && \forall T \subseteq G && (6.2)\\
&\geq v_i'(T) - P_i(T) - \alpha - 2\varepsilon, && \forall T \subseteq G && (6.3)
\end{aligned}
$$

where (6.1) and (6.3) follow because $\|\mathcal{M} - \mathcal{M}'\|_\infty \leq \varepsilon$, and (6.2) follows because $(\mathcal{S}, \boldsymbol{p})$ is a α-approximate CE of $\mathcal{M}$. $\square$

6.3 Learning Methodology

Extending our formalism for learning simulation-based games (chapter 3), we now present a formalism in which to model noisy combinatorial markets. Intuitively, a noisy market is one in which buyers' valuations over bundles are not known precisely; rather, only noisy samples are available. Note that our setup generalizes the standard model of combinatorial markets by allowing buyers' values to be drawn from (possibly unknown) probability distributions. In the standard model, these distributions would be known and degenerate.

Definition 21 (Conditional Combinatorial Markets)

A *conditional combinatorial market* $\mathcal{M}_{\mathcal{X}} = (\mathcal{X}, G, N, \{v_i\}_{i \in N})$ consists of a set of conditions $\mathcal{X}$, a set of goods G, a set of buyers N, and a set of conditional valuation functions $\{v_i\}_{i \in N}$, where $v_i : 2^G \times \mathcal{X} \to \mathbb{R}_+$.

Given a condition $x \in \mathcal{X}$, the value $v_i(S, x)$ is i's value for bundle $S \subseteq G$.

Definition 22 (Expected Combinatorial Market)

Let $\mathcal{M}_{\mathcal{X}} = (\mathcal{X}, G, N, \{v_i\}_{i \in N})$ be a conditional combinatorial market and let $\mathscr{D}$ be a distribution over $\mathcal{X}$.

For all $i \in N$, define the expected valuation function $v_i : 2^G \to \mathbb{R}_+$ by $v_i(S, \mathscr{D}) = \mathbb{E}_{x \sim \mathscr{D}}[v_i(S, x)]$, and the corresponding expected combinatorial market as $\mathcal{M}_{\mathscr{D}} = (G, N, \{v_i\}_{i \in N})$.

The goal of this chapter is to design algorithms that learn the approximate CE of expected combinatorial markets. We will learn their equilibria given access only to their empirical counterparts, which we define next.

Definition 23 (Empirical Combinatorial Market)

Let $\mathcal{M}_{\mathcal{X}} = (\mathcal{X}, G, N, \{v_i\}_{i \in N})$ be a conditional combinatorial market and let $\mathscr{D}$ be a distribution over $\mathcal{X}$. Denote by $\boldsymbol{x} = (x_1, \ldots, x_m) \sim \mathscr{D}$ a vector of m samples drawn from $\mathcal{X}$ according to distribution $\mathscr{D}$.

For all $i \in N$, we define the empirical valuation function $\hat{v}_i : 2^G \to \mathbb{R}_+$ by $\hat{v}_i(S) =$

$\frac{1}{m}\sum_{j=1}^{m} v_i(S, x_j)$, and the corresponding empirical combinatorial market $\hat{\mathcal{M}}_{\boldsymbol{x}} = (G, N, \{\hat{v}_i\}_{i \in N})$.

Learnability Let $\mathcal{M}_{\mathcal{X}}$ be a conditional combinatorial market and let $\mathscr{D}$ be a distribution over $\mathcal{X}$. Let $\mathcal{M}_{\mathscr{D}}$ and $\hat{\mathcal{M}}_{\boldsymbol{x}}$ be the corresponding expected and empirical combinatorial markets. If, for some $\varepsilon, \delta > 0$, it holds that $Pr\left(\left\|\mathcal{M}_{\mathscr{D}} - \hat{\mathcal{M}}_{\boldsymbol{x}}\right\| \leq \varepsilon\right) \geq 1 - \delta$, then the competitive equilibria of $\mathcal{M}_{\mathscr{D}}$ are learnable: i.e, any competitive equilibrium of $\mathcal{M}_{\mathscr{D}}$ is a 2ε-competitive equilibrium of $\hat{\mathcal{M}}_{\boldsymbol{x}}$ with probability at least $1 - \delta$.

Theorem 9 implies that CE are approximable to within any desired $\varepsilon > 0$ guarantee. The following lemma shows we only need a finitely many samples to learn them to within any $\delta > 0$ probability.

Lemma 4 (Finite-Sample Bounds for Expected Combinatorial Markets via Hoeffding's Inequality)**.** *Let $\mathcal{M}_{\mathcal{X}}$ be a conditional combinatorial market, $\mathscr{D}$ a distribution over $\mathcal{X}$, and $\mathcal{I} \subseteq N \times 2^G$ an index set. Suppose that for all $x \in \mathcal{X}$ and $(i, S) \in \mathcal{I}$, it holds that $v_i(S, x) \in [0, c]$ where $c \in \mathbb{R}_+$. Then, with probability at least $1 - \delta$ over samples $\boldsymbol{x} = (x_1, \ldots, x_m) \sim \mathscr{D}$, it holds that $\left\|\mathcal{M}_{\mathscr{D}} - \hat{\mathcal{M}}_{\boldsymbol{x}}\right\|_{\mathcal{I}} \leq c\sqrt{\ln(2|\mathcal{I}|/\delta)/2m}$, where $\delta > 0$.*

Proof. The keen reader will find that this proof closely resembles the proof of theorem 2.

Let $\mathcal{M}_{\mathcal{X}}$ be a conditional combinatorial market, $\mathscr{D}$ a distribution over $\mathcal{X}$, and $\mathcal{I} \subseteq N \times 2^G$ an index set. Let $\boldsymbol{x} = (x_1, \ldots, x_m) \sim \mathscr{D}$ be a vector of m samples drawn from $\mathscr{D}$. Suppose that for all $x \in \mathcal{X}$ and $(i, S) \in \mathcal{I}$, it holds that $v_i(S, x) \in [0, c]$ where $c \in \mathbb{R}_+$. Let $\delta > 0$ and $\varepsilon > 0$. Then, by Hoeffding's inequality [62],

$$Pr(|v_i(S) - \hat{v}_i(S)| \geq \varepsilon) \leq 2e^{-2m(\frac{\varepsilon}{c})^2} \tag{6.4}$$

Now, applying union bound over all events $|v_i(S) - \hat{v}_i(S)| \geq \varepsilon$ where $(i, S) \in \mathcal{I}$,

$$Pr\left(\bigcup_{(i,S)\in\mathcal{I}} |v_i(S) - \hat{v}_i(S)| \geq \varepsilon\right) \leq \sum_{(i,S)\in\mathcal{I}} Pr\left(|v_i(S) - \hat{v}_i(S)| \geq \varepsilon\right) \tag{6.5}$$

Using bound (3.2) in the right-hand side of (3.3),

$$Pr\left(\bigcup_{(i,S)\in\mathcal{I}} |v_i(S) - \hat{v}_i(S)| \geq \varepsilon\right) \leq \sum_{(i,S)\in\mathcal{I}} 2e^{-2m(\frac{\varepsilon}{c})^2} = 2|\mathcal{I}|e^{-2m(\frac{\varepsilon}{c})^2} \tag{6.6}$$

Where the last equality follows because the summands on the right-hand size of eq. (3.4) do not depend on the summation index. Now, note that eq. (3.4) implies a lower bound for the event that complements $\bigcup_{(i,S)\in\mathcal{I}} |v_i(S) - \hat{v}_i(S)| \geq \varepsilon$,

$$Pr\left(\bigcap_{(i,S)\in\mathcal{I}} |v_i(S) - \hat{v}_i(S)| \leq \varepsilon\right) \geq 1 - 2|\mathcal{I}|e^{-2m(\frac{\varepsilon}{c})^2} \tag{6.7}$$

The event $\bigcap_{(i,S)\in\mathcal{I}} |v_i(S) - \hat{v}_i(S)| \leq \varepsilon$ is equivalent to the event $\max_{(i,S)\in\mathcal{I}} |v_i(S) - \hat{v}_i(S)| \leq \varepsilon$. Setting $\delta = 2|\mathcal{I}|e^{-2m(\frac{\varepsilon}{c})^2}$ and solving for ε yields $\varepsilon = c\sqrt{\ln(2|\mathcal{I}|/\delta)/2m}$.

The results follows by substituting ε in eq. (3.5). ☐

Hoeffding's inequality is a convenient and simple bound, where only knowledge of the range of values is required. However, the union bound can be inefficient in large combinatorial markets. This shortcoming can be addressed via uniform convergence bounds and Rademacher averages [7, 17, 72]. Furthermore, we showed in chapter 3 that sharper empirical variance sensitive bounds can improve the sample complexity in learning the Nash equilibria of simulation-based games. In particular, to obtain a confidence interval of radius ε in a combinatorial market with index set $\mathcal{I} = N \times 2^G$, Hoeffding's inequality requires $t \in \mathcal{O}(c^2|G|/\varepsilon^2 \ln |N|/\delta)$ samples. Uniform convergence bounds can improve the $|G|$ term arising from the *union bound*, and variance-sensitive bounds can largely replace dependence on c^2 with *variances*. Nonetheless, even without these augmentations, our methods are statistically efficient in $|G|$, requiring only *polynomial* sample complexity to learn *exponentially* large combinatorial markets.

6.3.1 Baseline Algorithm

EA (Algorithm 9) is a preference elicitation algorithm for combinatorial markets. The algorithm places value queries, but is only assumed to elicit noisy values for bundles. The following guarantee follows immediately from lemma 4.

Theorem 10 (Elicitation Algorithm Guarantees)**.** *Let $\mathcal{M}_{\mathcal{X}}$ be a conditional market, $\mathscr{D}$ be a distribution over $\mathcal{X}$, $\mathcal{I}$ an index set, $m \in \mathbb{N}_{>0}$ a number of samples, $\delta > 0$, and $c \in \mathbb{R}_+$. Suppose that for all $x \in \mathcal{X}$ and $(i,S) \in \mathcal{I}$, it holds that $v_i(S,x) \in [0,c]$. If* EA *outputs $(\{\hat{v}_i\}_{(i,S)\in\mathcal{I}}, \hat{\varepsilon})$ on input $(\mathcal{M}_{\mathcal{X}}, \mathscr{D}, \mathcal{I}, m, \delta, c)$, then, with probability at least $1-\delta$, it holds that $\left\|\mathcal{M}_{\mathscr{D}} - \hat{\mathcal{M}}_{\boldsymbol{x}}\right\|_{\mathcal{I}} \leq c\sqrt{\ln(2|\mathcal{I}|/\delta)/2m}$.*

Proof. The result follows from lemma 4. ☐

Algorithm 9 Elicitation Algorithm (EA)

procedure EA($\mathcal{M}_{\mathcal{X}}, \mathscr{D}, \mathcal{I}, m, \delta, c.$) $\to (\hat{v}_i(S), \hat{\varepsilon})$, for all $(i, S) \in \mathcal{I}$.
  input:
    conditional combinatorial market $\mathcal{M}_{\mathcal{X}}$,
    distribution $\mathscr{D}$ over $\mathcal{X}$,
    an index set $\mathcal{I}$,
    sample size m,
    failure probability δ,
    valuation range c.
  output:
    valuation estimates $\hat{v}_i(S)$, for all $(i, S) \in \mathcal{I}$,
    and an approximation error $\hat{\varepsilon}$.
  $(x_1, \ldots, x_m) \sim \mathscr{D}$ ▷ Draw m samples from $\mathscr{D}$
  for $(i, S) \in \mathcal{I}$ **do**
    $\hat{v}_i(S) \leftarrow \frac{1}{m} \sum_{j=1}^{m} v_i(S, x_j)$
  end for
  $\hat{\varepsilon} \leftarrow c\sqrt{\ln(2|\mathcal{I}|/\delta)/2m}$ ▷ Compute error
  return $(\{\hat{v}_i\}_{(i,S)\in\mathcal{I}}, \hat{\varepsilon})$
end procedure

6.3.2 Pruning Algorithm

EA elicits buyers' valuations for all bundles, but in certain situations, some buyer valuations are not relevant for computing a CE—although bounds on all of them are necessary to guarantee strong bounds on the set of CE (Theorem 9). For example, in a first-price auction for one good, it is enough to accurately learn the highest bid, but is not necessary to accurately learn all other bids, if it is known that they are lower than the highest. Since our goal is to learn CE, we present EAP (Algorithm 10), an algorithm that does not sample uniformly, but instead adaptively decides which value queries to prune so that, with provable guarantees, EAP's estimated market satisfies the conditions of Theorem 9.

EAP (Algorithm 10) takes as input a sampling schedule $\boldsymbol{M}$, a failure probability schedule $\boldsymbol{\delta}$, and a pruning budget schedule $\boldsymbol{\pi}$. The sampling schedule $\boldsymbol{M}$ is a sequence of $|\boldsymbol{M}|$ strictly decreasing integers $m_1 > m_2 > \cdots > m_{|\boldsymbol{M}|}$, where m_k is the total number of samples to take for each (i, S) pair during EAP's k-th iteration. The failure probability schedule $\boldsymbol{\delta}$ is a sequence of the same length as $\boldsymbol{M}$, where $\delta_k \in (0, 1)$ is the k-th iteration's failure probability and $\sum_k \delta_k \in (0, 1)$ is the total failure probability. The pruning budget schedule $\boldsymbol{\pi}$ is a sequence of integers also of the same length as $\boldsymbol{M}$, where π_k is the maximum number of (i, S) pruning candidate pairs. The algorithm progressively elicits buyers' valuations via repeated calls to EA. However, between calls to EA, EAP searches for value queries that are provably not part of a CE; the size of this search is dictated by the pruning schedule. All such queries (i.e., buyer–bundle pairs) then cease to be part of the index set with which EA is called in future iterations.

In what follows, we prove several intermediate results, which enable us to prove the main result of this section, Theorem 12, which establishes EAP's correctness. Specifically, the market learned by EAP—with potentially different numbers of samples for different (i, S) pairs—is enough to provably recover any CE of the underlying market.

Lemma 5 (Optimal Welfare Approximations)**.** *Let $\mathcal{M}$ and $\mathcal{M}'$ be compatible markets such that they ε-approximate one another. Then $|w^*(\mathcal{M}) - w^*(\mathcal{M}')| \leq \varepsilon n$.*

Proof. Let $\mathcal{S}^*$ be a welfare-maximizing allocation for $\mathcal{M}$ and $\mathcal{U}^*$ be a welfare-maximizing allocation for $\mathcal{M}'$. Let $w^*(\mathcal{M})$ be the maximum achievable welfare in market $\mathcal{M}$. Then, The first inequality follows from the optimality of $\mathcal{S}^*$ in $\mathcal{M}$, and the second from the ε-approximation assumption, i.e., $\|\mathcal{M} - \mathcal{M}'\|_\infty \leq \varepsilon$. Likewise, $w^*(\mathcal{M}') \geq w^*(\mathcal{M}) - \varepsilon n$, so the result holds. □

The key to this work was the discovery of a pruning criterion that removes (i, S) pairs from consideration if they are provably not part of any CE. Our check relies on computing the welfare of the market without the pair: i.e., in submarkets.

Definition 24 (Submarket)

Given a market $\mathcal{M}$ and buyer–bundle pair (i, S), the (i, S)-*submarket* of $\mathcal{M}$, denoted by $\mathcal{M}_{-(i,S)}$, is the market obtained by removing all goods in S and buyer i from market $\mathcal{M}$. That is, $\mathcal{M}_{-(i,S)} = (G \setminus S, N \setminus \{i\}, \{v_k\}_{k \in N \setminus \{i\}})$.

Algorithm 10 Elicitation Algorithm with Pruning (EAP)

procedure EAP($\mathcal{M}_{\mathcal{X}}, \mathscr{D}, \boldsymbol{M}, \boldsymbol{\delta}, c, \varepsilon.$) $\rightarrow (\hat{v}_i(S), \hat{\varepsilon}_{i,S})$, for all $(i, S) \in \mathcal{I}$, and $(\hat{\delta}, \hat{\varepsilon})$.

  input:

   conditional combinatorial market $\mathcal{M}_{\mathcal{X}}$,
   distribution $\mathscr{D}$ over $\mathcal{X}$,
   a sampling schedule $\boldsymbol{M}$,
   a failure probability schedule $\boldsymbol{\delta}$,
   a pruning budget schedule $\boldsymbol{\pi}$,
   valuation range c, and
   target approximation error ε.

  output:

   valuation estimates $\hat{v}_i(S)$, for all $(i, S) \in \mathcal{I}$,
   approximation errors $\hat{\varepsilon}_{i,S}$,
   failure probability $\hat{\delta}$, and
   CE error $\hat{\varepsilon}$.

  $\mathcal{I} \leftarrow N \times 2^G$ ▷ Initialize index set
  $(\hat{v}_i(S), \hat{\varepsilon}_{i,S}) \leftarrow (0, c/2), \forall (i, S) \in \mathcal{I}$ ▷ Initialize outputs
  for $k \in 1, \ldots, |\boldsymbol{M}|$ **do**
    $(\{\hat{v}_i\}_{(i,S)\in\mathcal{I}}, \hat{\varepsilon}) \leftarrow \mathrm{EA}(\mathcal{M}_{\mathcal{X}}, \mathscr{D}, \mathcal{I}, m_k, \delta_k, c)$ ▷ Call algorithm 9
    $\hat{\varepsilon}_{i,S} \leftarrow \hat{\varepsilon}, \forall (i, S) \in \mathcal{I}$ ▷ Update error rates
    if $\hat{\varepsilon} \leq \varepsilon$ or $k = |\boldsymbol{M}|$ or $\mathcal{I} = \emptyset$ **then** ▷ Check termination conditions
      return $(\{\hat{v}_i\}_{i \in N}, \{\hat{\varepsilon}_{i,S}\}_{(i,S)\in N \times 2^G}, \sum_{l=1}^{k} \delta_l, \hat{\varepsilon})$
    end if
    Let $\hat{\mathcal{M}}$ be the market with valuations $\{\hat{v}_i\}_{(i,S)\in\mathcal{I}}$
    $\mathcal{I}_{\text{PRUNE}} \leftarrow \emptyset$ ▷ Initialize set of indices to prune
    $\mathcal{I}_{\text{CANDIDATES}} \leftarrow$ a subset of $\mathcal{I}$ of size at most π_k ▷ Select some active pairs as pruning candidates
    for $(i, S) \in \mathcal{I}_{\text{CANDIDATES}}$ **do**
      Let $\hat{\mathcal{M}}_{-(i,S)}$ be the (i, S)-submarket of $\hat{\mathcal{M}}$.
      Let $w^{\diamond}_{(i,S)}$ an upper bound of $w^*(\hat{\mathcal{M}}_{-(i,S)})$.
      if $\hat{v}_i(S) + w^{\diamond}_{(i,S)} + 2\hat{\varepsilon}n < w^*(\hat{\mathcal{M}})$ **then**
        $\mathcal{I}_{\text{PRUNE}} \leftarrow \mathcal{I}_{\text{PRUNE}} \cup (i, S)$
      end if
    end for
    $\mathcal{I} \leftarrow \mathcal{I} \setminus \mathcal{I}_{\text{PRUNE}}$
  end for
end procedure

Lemma 6 (Pruning Criteria)**.** *Let $\mathcal{M}$ and $\mathcal{M}'$ be compatible markets such that $\|\mathcal{M} - \mathcal{M}'\|_\infty \leq \varepsilon$. In addition, let (i, S) be a buyer, bundle pair, and $\mathcal{M}'_{-(i,S)}$ be the (i, S)-submarket of $\mathcal{M}'$. Finally, let $w^\diamond_{(i,S)} \in \mathbb{R}_+$ upper bound $w^*(\mathcal{M}'_{-(i,S)})$, i.e., $w^*(\mathcal{M}'_{-(i,S)}) \leq w^\diamond_{(i,S)}$. If the following pruning criterion holds, then S is not allocated to i in any welfare-maximizing allocation of $\mathcal{M}$:*

$$v'_i(S) + w^\diamond_{(i,S)} + 2\varepsilon n < w^*(\mathcal{M}') \ . \tag{6.8}$$

Proof. Let $\mathcal{S}^*$,$\mathcal{U}^*$, and $\mathcal{U}^*_{-(i,S)}$ be welfare-maximizing allocations of markets $\mathcal{M}, \mathcal{M}'$, and $\mathcal{M}'_{-(i,S)}$, respectively. Then,

$$\begin{aligned}
w^*(\mathcal{M}) &\geq w^*(\mathcal{M}') - \varepsilon n && (6.9)\\
&> v'_i(S) + w^\diamond_{(i,S)} + \varepsilon n && (6.10)\\
&\geq v'_i(S) + w^*(\mathcal{M}'_{-(i,S)}) + \varepsilon n && (6.11)\\
&\geq v_i(S) - \varepsilon + w^*(\mathcal{M}_{-(i,S)}) - \varepsilon(n-1) + \varepsilon n && (6.12)\\
&= v_i(S) + w^*(\mathcal{M}_{-(i,S)}) && (6.13)
\end{aligned}$$

The first inequality follows from Lemma 5. The second follows from Equation (6.8) and the third because $w^\diamond_{(i,S)}$ is an upper bound of $w^*(\mathcal{M}'_{-(i,S)})$. The fourth inequality follows from the assumption that $\|\mathcal{M} - \mathcal{M}'\|_\infty \leq \varepsilon$, and by Lemma 5 applied to submarket $\mathcal{M}_{-(i,S)}$. Therefore, the allocation where i gets S cannot be welfare-maximizing in market $\mathcal{M}$. □

Lemma 6 provides a family of pruning criteria parameterized by the upper bound $w^\diamond_{(i,S)}$. The closer $w^\diamond_{(i,S)}$ is to $w^*(\mathcal{M}'_{-(i,S)})$, the sharper the pruning criterion, with the best pruning criterion being $w^\diamond_{(i,S)} = w^*(\mathcal{M}'_{-(i,S)})$. However, solving for $w^*(\mathcal{M}'_{-(i,S)})$ exactly can easily become a bottleneck, as the pruning loop requires a solution to *many* such instances, one per (i, S) pair (Line 15 of Algorithm 10). Alternatively, one could compute looser upper bounds, and thereby trade off computation time for opportunities to prune more (i, S) pairs, when the upper bound is not tight enough. In our experiments, we show that even relatively loose but cheap-to-compute upper bounds result in significant pruning and, thus, savings along both dimensions—computational and sample complexity.

To conclude this section, we establish the correctness of EAP. For our proof we rely on the following generalization of the first welfare theorem of economics, which handles additive errors.

Theorem 11 (First Welfare Theorem [109])**.** *For $\varepsilon > 0$, let $(\mathcal{S}, \boldsymbol{p})$ be an ε-competitive equilibrium of $\mathcal{M}$. Then, $\mathcal{S}$ is a welfare-maximizing allocation of $\mathcal{M}$, up to additive error εn.*

Theorem 12 (Elicitation Algorithm with Pruning Guarantees)**.** *Let $\mathcal{M}_{\mathcal{X}}$ be a conditional market, let $\mathscr{D}$ be a distribution over $\mathcal{X}$, and let $c \in \mathbb{R}_+$. Suppose that for all $x \in \mathcal{X}$ and $(i, S) \in \mathcal{I}$, it holds that $v_i(S, x) \in [0, c]$, where $c \in \mathbb{R}$. Let $\boldsymbol{M}$ be a sequence of strictly increasing integers, and $\boldsymbol{\delta}$ a sequence of the same length as $\boldsymbol{M}$ such that $\delta_k \in (0, 1)$ and $\sum_k \delta_k \in (0, 1)$.*

If EAP *outputs $(\{\hat{v}_i\}_{i \in N}, \{\hat{\varepsilon}_{i,S}\}_{(i,S) \in N \times 2^G}, 1 - \sum_k \delta_k, \hat{\varepsilon})$ on input $(\mathcal{M}_{\mathcal{X}}, \mathscr{D}, \boldsymbol{M}, \boldsymbol{\delta}, c, \varepsilon)$, then the following holds with probability at least $1 - \sum_k \delta_k$:*

1. $\left\| \mathcal{M}_{\mathscr{D}} - \hat{\mathcal{M}} \right\|_{\mathcal{I}} \leq \hat{\varepsilon}_{i,S}$
2. $\mathcal{CE}(\mathcal{M}_{\mathscr{D}}) \subseteq \mathcal{CE}_{2\hat{\varepsilon}}(\hat{\mathcal{M}}) \subseteq \mathcal{CE}_{4\hat{\varepsilon}}(\mathcal{M}_{\mathscr{D}})$

Here $\hat{\mathcal{M}}$ is the empirical market obtained via EAP*, i.e., the market with valuation functions given by $\{\hat{v}_i\}_{i \in N}$.*

Proof. To show part 1, note that at each iteration k of EAP, Line 8 updates the error estimates for each (i, S) after a call to EA (Line 7 of EAP) with input failure probability δ_k. Theorem 10 implies that each call to EA returns estimated values that are within $\hat{\varepsilon}$ of their expected value with probability at least $1 - \delta_k$. By union bounding all calls to EA within EAP, part 1 then holds with probability at least $1 - \sum_k \delta_k$.

To show part 2, note that only pairs (i, S) for which Equation (6.8) holds are removed from index set $\mathcal{I}$ (Line 17 of EAP). By Lemma 6, no such pair can be part of any approximate welfare-maximizing allocation of the expected market, $\mathcal{M}_{\mathscr{D}}$. By Theorem 11, no such pair can be a part of any CE. Consequently, $\hat{\mathcal{M}}$ contains accurate enough estimates (up to ε) of all (i, S) pairs that may participate in any CE. Part 2 then follows from Theorem 9. □

6.4 Experiments

The goal of our experiments is to robustly evaluate the empirical performance of our algorithms. To this end, we experiment with a variety of qualitatively different inputs. In particular, we evaluate our algorithms on both unit-demand valuations, the Global Synergy Value Model (GSVM) [53], and the Local Synergy Value Model (LSVM) [117]. Unit-demand valuations are a class of valuations central to the literature on economics and computation [79] for which efficient algorithms exist to compute CE [58]. GSVM and LSVM model situations in which buyers' valuations encode complements; CE are not known be efficiently computable, or even representable, in these markets.

While CE are always guaranteed to exist (e.g., [22]), in the worst case, they might require personalized bundle prices. These prices are computationally complex, not to mention out of favor [61]. A pricing $\boldsymbol{p} = (P_1, \ldots, P_n)$ is *anonymous* if it charges every buyer the same price, i.e., $P_i = P_k = P$ for all $i \neq k \in N$. Moreover, an anonymous pricing is *linear* if there exists a set of prices $\{p_1, \ldots, p_N\}$, where p_j is good j's price, such that $P(S) = \sum_{j \in S} p_j$. In what follows, we refer to linear, anonymous pricings as linear prices.

Where possible, it is preferable to work with linear prices, as they are simpler, e.g., when bidding in an auction [76]. In our present study—one of the first empirical studies on learning CE—we thus focus on linear prices, leaving as future research the empirical [2] effect of more complex pricings.[3]

To our knowledge, there have been no analogous attempts at learning CE; hence, we do not reference any baseline algorithms from the literature. Rather, we compare the performance of EAP, our pruning algorithm, to EA, investigating the quality of the CE learned by both, as well as their sample efficiencies.

6.4.1 Experimental Setup.

We first explain our experimental setup, and then present results. We let $U[a, b]$ denote the continuous uniform distribution over range $[a, b]$, and $U\{k, l\}$, the discrete uniform distribution over set $\{k, k+1, \ldots, l\}$, for $k \leq l \in \mathbb{N}$.

Simulation of Noisy Combinatorial Markets. We start by drawing markets from experimental market distributions. Then, fixing a market, we simulate noisy value elicitation by adding noise drawn from experimental noise distributions to buyers' valuations in the market. We refer to a market realization $\mathcal{M}$ drawn from an experimental market distribution as the *ground-truth* market. Our experiments then measure how well we can approximate the CE of a ground-truth market $\mathcal{M}$ given access only to noisy samples of it.

Fix a market $\mathcal{M}$ and a condition set $\mathcal{X} = [a, b]$, where $a < b$. Define the conditional market $\mathcal{M}_{\mathcal{X}}$, where $v_i(S, x_{iS}) = v_i(S) + x_{iS}$, for $x_{iS} \in \mathcal{X}$. In words, when eliciting i's valuation for S, we assume

[2] Note that all our theoretical results hold for any pricing profile.
[3] Lahaie and Lubin [77], for example, search for prices in between linear and bundle.

additive noise, namely x_{iS}. The market $\mathcal{M}_{\mathcal{X}}$ together with distribution $\mathscr{D}$ over $\mathcal{X}$ is the model from which our algorithms elicit noisy valuations from buyers. Then, given samples $\boldsymbol{x}$ of $\mathcal{M}_{\mathcal{X}}$, the empirical market $\hat{\mathcal{M}}_{\boldsymbol{x}}$ is the market estimated from the samples. Note that $\hat{\mathcal{M}}_{\boldsymbol{x}}$ is the only market we get to observe in practice.

We consider only zero-centered noise distributions. In this case, the expected combinatorial market $\mathcal{M}_{\mathscr{D}}$ is the same as the ground-truth market $\mathcal{M}$ since, for every $i, S \in N \times 2^G$ it holds that $v_i(S, \mathscr{D}) = \mathbb{E}_{\mathscr{D}}[v_i(S, x_{iS})] = \mathbb{E}_{\mathscr{D}}[v_i(S) + x_{iS}] = v_i(S)$. While this noise structure is admittedly simple, we robustly evaluate our algorithms along another dimension, as we study several rich market structures (unit-demand, GSVM, and LSVM). An interesting future direction would be to also study richer noise structures, e.g., letting noise vary with a bundle's size, or other market characteristics.

Utility-Maximization (UM) Loss To measure the quality of a CE $(y', \boldsymbol{p}')$ computed for a market $\mathcal{M}'$ in another market $\mathcal{M}$, we first define the per-buyer metric $\textit{UM-Loss}_{\mathcal{M},i}$ as follows,

$$\textit{UM-Loss}_{\mathcal{M},i}(y', \boldsymbol{p}') = \max_{S \subseteq G}(v_i(S) - P'(S)) - (v_i(S'_i) - P'(S'_i)),$$

i.e., the difference between the maximum utility i could have attained at prices $\boldsymbol{p}'$ and the utility i attains at the outcome $(y', \boldsymbol{p}')$. Our metric of interest is then $\textit{UM-Loss}_{\mathcal{M}}$ defined as,

$$\textit{UM-Loss}_{\mathcal{M}}(y', \boldsymbol{p}') = \max_{i \in N} \textit{UM-Loss}_{\mathcal{M},i}(y', \boldsymbol{p}'),$$

which is a worst-case measure of utility loss over all buyers in the market. Note that it is not useful to incorporate the SR condition into a loss metric, because it is always satisfied.

In our experiments, we measure the UM loss that a CE of an empirical market obtains, evaluated in the corresponding ground-truth market. Thus, given an empirical estimate $\hat{\mathcal{M}}_{\boldsymbol{x}}$ of $\mathcal{M}$, and a CE $(\hat{y}, \hat{\boldsymbol{p}})$ in $\hat{\mathcal{M}}_{\boldsymbol{x}}$, we measure $\textit{UM-Loss}_{\mathcal{M}}(\hat{y}, \hat{\boldsymbol{p}})$, i.e., the loss in $\mathcal{M}$ at prices $\hat{\boldsymbol{p}}$ of CE $(\hat{y}, \hat{\boldsymbol{p}})$. Theorem 9 implies that if $\hat{\mathcal{M}}_{\boldsymbol{x}}$ is an ε-approximation of $\mathcal{M}$, then $\textit{UM-Loss}_{\mathcal{M}}(\hat{y}, \hat{\boldsymbol{p}}) \leq 2\varepsilon$. Moreover, Theorem 10 yields the same guarantees, but with probability at least $1 - \delta$, provided the ε-approximation holds with probability at least $1 - \delta$.

Sample Efficiency of EAP**.** We say that algorithm A has better *sample efficiency* than algorithm B if A requires fewer samples than B to achieve at least the same ε accuracy.

Fixing a condition set $\mathcal{X}$, a distribution $\mathscr{D}$ over $\mathcal{X}$, and a conditional market $\mathcal{M}_{\mathcal{X}}$, we use the following experimental design to evaluate EAP's sample efficiency relative to that of EA. Given a desired error guarantee $\varepsilon > 0$, we compute the number of samples $m(\varepsilon)$ that would be required for EA to achieve accuracy ε. We then use the following doubling strategy as a sampling schedule for EAP, $\boldsymbol{M}(m(\varepsilon)) = [m(\varepsilon)/4, m(\varepsilon)/2, m(\varepsilon), 2m(\varepsilon)]$, rounding to the nearest integer as necessary, and the following failure probability schedule $\boldsymbol{\delta} = [0.025, 0.025, 0.025, 0.025]$, which sums to 0.1.

Finally, the exact pruning budget schedules will vary depending on the value model (unit demand, GSVM, or LSVM). But in all cases, we denote an *unconstrained* pruning budget schedule by $\boldsymbol{\pi} = [\infty, \infty, \infty, \infty]$, which by convention means that at every iteration, all active pairs are candidates for pruning. Using these schedules, we run EAP with a desired accuracy of zero. We denote by $\varepsilon_{\text{EAP}}(\varepsilon)$ the approximation guarantee achieved by EAP upon termination.

6.4.2 Unit-demand Experiments

A buyer i is endowed with unit-demand valuations if, for all $S \subseteq G$, $v_i(S) = \max_{j \in S} v_i(\{j\})$. In a unit-demand market, all buyers have unit-demand valuations. A unit-demand market can be compactly represented by matrix $\mathbf{V}$, where entry $v_{ij} \in \mathbb{R}_+$ is i's value for j, i.e., $v_{ij} = v_i(\{j\})$. In what follows, we denote by $\mathcal{V}$ a random variable over unit-demand valuations.

We construct four different distributions over unit-demand markets: Uniform, Preferred-Good, Preferred-Good-Distinct, and Preferred-Subset. All distributions are parameterized by n and N, the number of buyers and goods, respectively. A uniform unit-demand market $\mathcal{V} \sim$ Uniform is such that for all i, j, $v_{ij} \sim U[0, 10]$. When $\mathcal{V} \sim$ Preferred-Good, each buyer i has a preferred good j_i, with $j_i \sim U\{1, \ldots, N\}$ and $v_{ij_i} \sim U[0, 10]$. Conditioned on v_{ij_i}, i's value for good $k \neq j_i$ is given by $v_{ik} = v_{ij_i}/2^k$. Distribution Preferred-Good-Distinct is similar to Preferred-Good, except that no two buyers have the same preferred good. (Note that the Preferred-Good-Distinct distribution is only well defined when $n \leq N$.) Finally, when $\mathcal{V} \sim$ Preferred-Subset, each buyer i is interested in a subset of goods $i_G \subseteq G$, where i_G is drawn uniformly at random from the set of all bundles. Then, the value i has for j is given by $v_{ij} \sim U[0, 10]$, if $j \in i_G$; and 0, otherwise.

In unit-demand markets, we experiment with three noise models, low, medium, and high, by adding noise drawn from $U[-.5, .5]$, $U[-1, 1]$, and $U[-2, 2]$, respectively. We choose $n, N \in \{5, 10, 15, 20\}^2$.

Distribution	$\varepsilon = 0.05$ $\hat{\boldsymbol{p}}_{\text{MIN}}$	$\varepsilon = 0.05$ $\hat{\boldsymbol{p}}_{\text{MAX}}$	$\varepsilon = 0.2$ $\hat{\boldsymbol{p}}_{\text{MIN}}$	$\varepsilon = 0.2$ $\hat{\boldsymbol{p}}_{\text{MAX}}$
Uniform	0.0018	0.0020	0.0074	0.0082
Preferred-Good	0.0019	0.0023	0.0080	0.0094
Preferred-Good-Distinct	0.0000	0.0020	0.0000	0.0086
Preferred-Subset	0.0019	0.0022	0.0076	0.0090

Table 6.1: Average *UM-Loss* for $\varepsilon \in \{0.05, 0.2\}$.

Unit-demand Empirical UM Loss of EA**.** As a learned CE is a CE of a learned market, we require a means of computing the CE of a market—specifically, a unit-demand market $\mathbf{V}$. To do so, we first solve for the[4] welfare-maximizing allocation $y^*_{\mathbf{V}}$ of $\mathbf{V}$, by solving for the maximum weight matching using Hungarian algorithm [75] in the bipartite graph whose weight matrix is given by $\mathbf{V}$. Fixing $y^*_{\mathbf{V}}$, we then solve for prices via linear programming [22]. In general, there might be many prices that couple with $y^*_{\mathbf{V}}$ to form a CE of $\mathbf{V}$. For simplicity, we solve for two pricings given $y^*_{\mathbf{V}}$, the revenue-maximizing $\boldsymbol{p}_{\text{MAX}}$ and revenue-minimizing $\boldsymbol{p}_{\text{MIN}}$, where revenue is defined as the sum of the prices.

For each distribution, we draw 50 markets, and for each such market $\mathbf{V}$, we run EA four times, each time to achieve guarantee $\varepsilon \in \{0.05, 0.1, 0.15, 0.2\}$. EA then outputs an empirical estimate $\hat{\mathbf{V}}$ for each $\mathbf{V}$. We compute outcomes $(y^*_{\hat{\mathbf{V}}}, \hat{\boldsymbol{p}}_{\text{MAX}})$ and $(y^*_{\hat{\mathbf{V}}}, \hat{\boldsymbol{p}}_{\text{MIN}})$, and measure $\textit{UM-Loss}_{\mathbf{V}}(y^*_{\hat{\mathbf{V}}}, \hat{\boldsymbol{p}}_{\text{MAX}})$ and $\textit{UM-Loss}_{\mathbf{V}}(y^*_{\hat{\mathbf{V}}}, \hat{\boldsymbol{p}}_{\text{MIN}})$. We then average across all market draws, for both the minimum and the maximum pricings. Table 6.1 summarizes a subset of these results. The error guarantees are consistently met across the board, indeed by one or two orders of magnitude, and they degrade as expected: i.e., with higher values of ε. We note that the quality of the learned CE is roughly the same for all distributions, except in the case of $\hat{\boldsymbol{p}}_{\text{MIN}}$ and Preferred-Good-Distinct, where learning is more accurate. For this distribution, it is enough to learn the preferred good of each buyer. Then, one possible CE is to allocate each buyer its preferred good and price all goods at zero which yields near no *UM-Loss*. Note that, in general, pricing all goods at zero is not a CE, unless the market has some special structure, like the markets drawn from Preferred-Good-Distinct.

[4]Since in all our experiments, we draw values from continuous distributions, we assume that the set of markets with multiple welfare-maximizing allocations is of negligible size. Therefore, we can ignore ties.

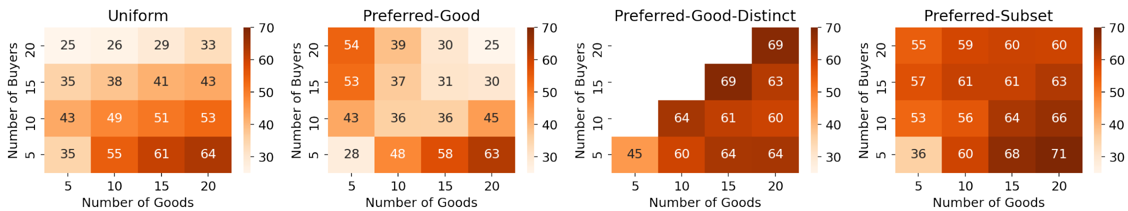


Figure 6.1: Mean EAP sample efficiency relative to EA, $\varepsilon = 0.05$. Each (i, j) pair is annotated with the corresponding % saving.

Unit-demand Sample Efficiency We use pruning schedule $\boldsymbol{\pi} = [\infty, \infty, \infty, \infty]$ and for each (i, j) pair, we use the Hungarian algorithm [75] to compute the optimal welfare of the market without (i, j). In other words, in each iteration, we consider all active (i, j) pairs as pruning candidates (Algorithm 10, Line 14), and for each we compute the optimal welfare (Algorithm 10, Line 17).

For each market distribution, we compute the average of the number of samples used by EAP across 50 independent market draws. We report samples used by EAP as a percentage of the number of samples used by EA to achieve the same guarantee, namely, $\varepsilon_{\text{EAP}}(\varepsilon)$, for each initial value of ε. Figure 6.1 depicts the results of these experiments as heat maps, for all distributions and for $\varepsilon = 0.05$, where darker colors indicate more savings, and thus better EAP sample efficiency.

A few trends arise, which we note are similar for other values of ε. For a fixed number of buyers, EAP's sample efficiency improves as the number of goods increases, because fewer goods can be allocated, which means that there are more candidate values to prune, resulting in more savings. On the other hand, the sample efficiency usually decreases as the number of buyers increases; this is to be expected, as the pruning criterion degrades with the number of buyers (Lemma 6). While savings exceed 30% across the board, we note that Uniform, the market with the least structure, achieves the least savings, while Preferred-Subset and Preferred-Good-Distinct achieve the most. This finding shows that EAP is capable of exploiting the structure present in these distributions, despite not knowing anything about them *a priori*.

Finally, we note that sample efficiency quickly degrades for higher values of ε. In fact, for high enough values of ε (in our experiments, $\varepsilon = 0.2$), EAP might, on average, require more samples than EA to produce the same guarantee. Most of the savings achieved are the result of pruning enough (i, j) pairs early enough: i.e., during the first few iterations of EAP. When ε is large, however, our sampling schedule does not allocate enough samples early on. When designing sampling schedules for EAP, one must allocate enough (but not too many) samples at the beginning of the schedule.

Precisely how to determine this schedule is an empirical question, likely dependent on the particular application at hand.

6.4.3 Value Models

In this next set of experiments, we test the empirical performance of our algorithms in more complex markets, where buyers valuations contain synergies. Synergies are a common feature of many high-stakes combinatorial markets. For example, telecommunication service providers might value different bundles of radio spectrum licenses differently, depending on whether the licenses in the bundle complement one another. For example, a bundle including New Jersey and Connecticut might not be very valuable unless it also contains New York City.

Specifically, we study the Global Synergy Value Model (GSVM) [53] and the Local Synergy Value Model (LSVM) [117]. These models or markets capture buyers' synergies as a function of buyers' types and their (abstract) geographical locations. In both GSVM and LSVM, there are 18 licenses, with buyers of two types: national or regional. A national buyer is interested in larger packages than regional buyers, whose interests are limited to certain regions. GSVM has six regional bidders and one national bidder and models geographical regions as two circles. LSVM has five regional bidders and one national bidder and uses a rectangular model. The models differ in the exact ways buyers' values are drawn, but in any case, synergies are modeled by suitable distance metrics. We refer the interested reader to [117] for a detailed explanation of the models. In our experiments, we draw instances of both GSVM and LSVM using SATS, a universal spectrum auction test suite developed by researchers to test algorithms for combinatorial markets [133].

Experimental Setup. On average, the value a buyer has for an arbitrary bundle in either GSVM or LSVM markets is approximately 80. We introduce noise i.i.d. noise from distribution $U[-1, 1]$ whose range is 2, or 2.5% of the expected buyer's value for a bundle. As GSVM's buyers' values are at most 400, and LSVM's are at most 500, we use valuation ranges $c = 402$ and $c = 502$ for GSVM and LSVM, respectively. We note that a larger noise range yields qualitatively similar results with errors scaling accordingly.

For the GSVM markets, we use the pruning budget schedule $\boldsymbol{\pi} = [\infty, \infty, \infty, \infty]$. For each (i, S) pair, we solve the welfare maximization problem using an off-the-shelf solver. [5] In an LSVM market,

[5] We include ILP formulations and further technical details in the section 6.5.1.

the national bidder demands all 18 licenses. The welfare optimization problem in an LSVM market is solvable in a few seconds. [6] Still, the many submarkets (in the hundreds of thousands) call for a finite pruning budget schedule and a cheaper-to-compute welfare upper bound. In fact, to address LSVM's size complexity, we slightly modify EAP, as explained next.

A two-pass strategy for LSVM Because of the complexity of LSVM markets, we developed a heuristic pruning strategy, in which we perform two pruning passes during each iteration of EAP. The idea is to compute a computationally cheap upper bound on welfare with pruning budget schedule $\boldsymbol{\pi} = [\infty, \infty, \infty, \infty]$ in the first pass, use this bound instead of the optimal for each active (i, S). We compute this bound using the classic relaxation technique to create admissible heuristics. Concretely, given a candidate (i, S) pair, we compute the maximum welfare in the absence of pair (i, S), ignoring feasibility constraints:

$$w^{\diamond}_{(i,S)} = \sum\nolimits_{k \in N \setminus \{i\}} \max\{v_k(T) \mid T \in 2^G \text{ and } S \cap T = \emptyset\}$$

After this first pass, we undertake a second pass over all remaining active pairs. For each active pair, we compute the optimal welfare without pair (i, S), but using the following *finite* pruning budget schedule $\boldsymbol{\pi} = [180, 90, 60, 45]$. In other words, we carry out this computation for just a few of the remaining candidate pairs. We chose this pruning budget schedule so that one iteration of EAP would take approximately two hours.

One choice remains undefined for the second pruning pass: which (i, S) candidate pairs to select out of those not pruned in the first pass? For each iteration k, we sort the (i, S) in descending order according to the upper bound on welfare computed in the first pass, and then we select the bottom $\boldsymbol{\pi}_k$ pairs (180 during the first iteration, 90 during the second, etc.). The intuition for this choice is that pairs with lower upper bounds might be more likely to satisfy Lemma 6's pruning criteria than pairs with higher upper bounds. Note that the way candidate pairs are selected for the second pruning pass uses no information about the underlying market, and is thus widely applicable. We will have more to say about the lack *a priori* information used by EAP in what follows.

[6] Approximately 20 seconds in our experiments, details appear in section 6.5.1.

	GSVM				LSVM			
ε	EA	EAP	ε_{EAP}	UM Loss	EA	EAP	ε_{EAP}	UM Loss
1.25	$2,642$	$\mathbf{720 \pm 10}$	0.73 ± 0.01	0.0022 ± 0.0002	$330,497 \pm 386$	$\mathbf{270,754 \pm 14,154}$	0.89 ± 0.00	0.0011 ± 0.0003
2.50	660	$\mathbf{226 \pm 10}$	1.57 ± 0.02	0.0041 ± 0.0005	$82,624 \pm 96$	$\mathbf{73,733 \pm 3,629}$	1.78 ± 0.00	0.0018 ± 0.0003
5.00	165	$\mathbf{117 \pm 11}$	3.41 ± 0.03	0.0063 ± 0.0008	$\mathbf{20,656 \pm 24}$	$22,054 \pm 933$	3.59 ± 0.01	0.0037 ± 0.0005
10.0	$\mathbf{41}$	69 ± 4	7.36 ± 0.04	0.0107 ± 0.0010	$\mathbf{5,164 \pm 6}$	$7,580 \pm 211$	7.27 ± 0.01	0.0072 ± 0.0011

Table 6.2: GSVM (left group) and LSVM (right group) results. Each group reports sample efficiency and UM loss. Each row of the table reports results for a fixed value of ε. Results are 95% confidence intervals over 40 GSVM market draws and 50 LSVM market draws, except for EA's number of samples in the case of GSVM which is a deterministic quantity (a GSVM market is of size 4,480). The values in **bold** indicate the more sample efficient algorithm. Numbers of samples are reported in millions.

Results. Table 6.2 summarizes the results of our experiments with GSVM and LSVM markets. The table shows 95% confidence intervals around the mean number of samples needed by EA and EAP to achieve the indicated accuracy (ε) guarantee for each row of the table. The table also shows confidence intervals around the mean ε guarantees achieved by EAP, denoted ε_{EAP}, and confidence intervals over the UM loss metric. Several observations follow.

Although ultimately a heuristic method, on average EAP uses far fewer samples than EA and produces significantly better ε guarantees. We emphasize that EAP is capable of producing these results without any *a priori* knowledge about the underlying market. Instead, EAP *autonomously* samples those quantities that can provably be part of an optimal solution. The EAP guarantees are slightly worse in the LSVM market than for GSVM, where we prune *all* eligible (i, S) pairs. In general, there is a tradeoff between computational and sample efficiency: at the cost of more computation, to find more pairs to prune up front, one can save on future samples. Still, even with a rather restricted pruning budget $\boldsymbol{\pi} = [180, 90, 60, 45]$ (compared to hundreds of thousands potentially active (i, S) pairs), EAP achieves substantial savings compared to EA in the LSVM market.

Finally, the UM loss metric follows a trend similar to those observed for unit-demand markets, i.e., the error guarantees are consistently met and degrade as expected (worst guarantees for higher values of ε). Note that in our experiments, all 40 GSVM market instances have equilibria with linear and anonymous prices. In contrast, only 18 out of 32 LSVM market do, so the table reports UM loss over this set. For the remaining 32 markets, we report here a UM loss of approximately 12 ± 4 *regardless* of the value of ε. This high UM loss is due to the lack of CE in linear pricings which dominates any UM loss attributable to the estimation of values.

6.5 Chapter Summary

In this chapter, we propose a simple extension of the standard model of combinatorial markets that allows buyers' values to be drawn from (possibly unknown) probability distributions. Even though valuations are not known with complete certainty, we assume that noisy samples can be obtained, for example, by using approximate methods, heuristics, or truncating the run-time of a complete algorithm. For this model, we tackle the problem of learning CE. We first show tight lower- and upper-bounds on the buyers' utility loss, and hence the set of CE, given a uniform approximation of one market by another. We then develop learning algorithms that, with high probability, learn said uniform approximations using only finitely many samples.

Leveraging the first welfare theorem of economics, we define a pruning criterion under which an algorithm can provably stop learning about buyers' valuations for bundles, without affecting the quality of the set of learned CE. We embed these conditions in an algorithm that we show experimentally is capable of learning CE with far fewer samples than a baseline. Crucially, the algorithm need not know anything about this structure *a priori*; our algorithm is general enough to work in any combinatorial market. Moreover, we expect substantial improvement with sharper sample complexity bounds; in particular, variance-sensitive bounds can be vastly more efficient when the variance is small, whereas Hoeffding's inequality essentially assumes the worst-case variance.

6.5.1 Experimental Technical Details

For our experiments, we solve for CE in linear prices. To compute CE in linear prices, we first solve for a welfare-maximizing allocation y^* and then, fixing y^*, we solve for CE linear prices. Note that, if a CE in linear prices exists, then it is supported by any welfare-maximizing allocation [110]. Moreover, since valuations in our experiments are drawn from continuous distributions, we assume that the set of welfare-maximizing allocations for a given market is of negligible size.

Mathematical Programs Next, we present the mathematical programs we used to compute welfare-maximizing allocations and find linear prices. Given a combinatorial market $\mathcal{M}$, integer linear program (6.14) [92] computes a welfare-maximizing allocation y^*.

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in N, S \subseteq G} v_i(S) x_{iS} \\ \text{subject to} \quad & \sum_{i \in N, S | j \in S} x_{iS} \leq 1, \quad j = 1, \ldots, N \\ & \sum_{S \subseteq G} x_{iS} \leq 1, \quad i = 1, \ldots, n \\ & x_{iS} \in \{0, 1\}, \quad i \in N, S \subseteq G \end{aligned} \tag{6.14}$$

Given a market $\mathcal{M}$ and a solution y^* to (6.14), the following set of linear inequalities, (6.15), define all linear prices that couple with allocation y^* to form a CE in $\mathcal{M}$. The inequalities are defined over variables $P_1, \ldots, P_N$ where P_j is good j's price. The price of bundle S is then $\sum_{j \in S} P_j$.

$$\begin{aligned} v_i(S) - \textstyle\sum_{j \in S} P_j \leq v_i(S_i^*) - \textstyle\sum_{j \in S_i^*} P_j, \quad & i \in N, S \subseteq G \\ \text{If } j \notin \cup_{i \in N} S_i^*, \text{ then } P_j = 0, \quad & j = 1, \ldots, N \\ P_j \geq 0, \quad & j \in G \end{aligned} \tag{6.15}$$

The first set of inequalities of (6.15) enforce the UM conditions. The second set of inequalities states that the price of goods not allocated to any buyer in y^* must be zero. In the case of linear pricing, this condition is equivalent to the RM condition. In practice, a market might not have CE in linear pricings, i.e., the set of feasible solutions of (6.15) might be empty. In our experiments, we solve linear program (6.16), a relaxation of (6.15) where we introduce slack variables α_{iS} to relax the UM constraints. We define as objective function the sum of all slack variables, $\sum_{i \in N, S \subseteq G} \alpha_{iS}$, which we wish to minimize.

$$
\begin{aligned}
\text{minimize} \sum_{i \in N, S \subseteq G} \alpha_{iS} \\
\text{subject to} \\
v_i(S) - \textstyle\sum_{j \in S} P_j - \alpha_{iS} \leq v_i(S_i^*) - \textstyle\sum_{j \in S_i^*} P_j, &\quad i \in N, S \subseteq G \\
\text{If } j \notin \cup_{i \in N} S_i^*, \text{ then } P_j = 0, &\quad j = 1, \ldots, N \\
P_j \geq 0, &\quad j \in G \\
\alpha_{iS} \geq 0, &\quad i \in N, S \subseteq G
\end{aligned} \tag{6.16}
$$

As reported in section 6.4.3, for each GSVM market we found that the optimal solution of (6.16) was such that $\sum_{i \in N, S \subseteq G} \alpha_{iS} = 0$, which means that an exact CE in linear prices was found. In contrast, for LSVM markets only 18 out of 50 markets had linear prices ($\sum_{i \in N, S \subseteq G} \alpha_{iS} = 0$) whereas 32 did not ($\sum_{i \in N, S \subseteq G} \alpha_{iS} > 0$).

Further technical details We used the COIN-OR [114] library, through Python's PuLP [7] interface, to solve all mathematical programs. We wrote all our experiments in Python, and all code is available at https://github.com/eareyan/noisyce. We ran our experiments in a cluster of two Google's GCloud *c2-standard-4* machines. Unit-demand experiments took approximately two days to complete, GSVM experiments approximately four days, and LSVM experiments approximately eight days.

[7] https://pypi.org/project/PuLP/

Chapter 7

Conclusions and Future Directions

Our work contributed computational and statistical learning tools for analyzing and designing systems inhabited by multiple strategic agents. We first focused on developing and thoroughly evaluating methods that permit game-theoretic analysis of multi-agent systems whose complete description is prohibitively expensive. Game-theoretic models of such systems are known as simulation-based games because we only get to observe samples of utility functions by running the game's simulator but never a complete analytical description of them. We first proved tight bounds on the equilibria error of a game's uniform approximation and proposed algorithms that in practice efficiently learn said uniform approximations for simulation-based games. In extensive experimentation, we showed that our algorithms produce high-quality uniform approximations that preserve the game's equilibria making frugal use of data or calls of the game's simulator – a major computational bottleneck when analyzing simulation-based games in practice.

Equipped with algorithms capable of learning the equilibria of simulation-based games, we introduced a methodology to optimize the performance of multi-agent systems given a designer's objective function – a well-defined success metric mapping a state of the system to a number. Our methods formalize the design of mechanisms as a search problem within the parameter space of an overall parameterized mechanism. For example, when auctioning online advertisements, an advertisement exchange's goal is to maximize its revenue by setting reserve prices, i.e., prices below which it will not allocate ads. The challenge then is to find a set of reserve prices such that, at equilibrium, the ad exchange maximizes its revenue. In addition to facing a massive search space, this task is challenging because the ad exchange does not have access to agents' strategies but only to samples of their behavior. Having access only to samples of agent's behavior is a feature of modern multi-agent systems, and we believe our methodological contributions are a step towards their efficient analysis.

To evaluate our empirical mechanism design methodology, we developed a stylized-but-rich model of an electronic advertisement exchange's dynamics, paying particular attention to the exchange's ability to set different reserve prices based on user demographic data. Since the quality of the reserve prices found via our empirical mechanism design methods depends on the quality of the heuristic strategies used to find them in the first place, we further contributed two heuristic bidding strategies for our ad exchange model. Crucially, these heuristics are responsive to different reserve price choices by the ad exchange, allowing us to simulate sophisticated bidders' behavior. We showed that our empirical mechanism design methodology recovers optimal mechanisms when they are known more efficiently than existing baselines. For our ad exchange model, we experimentally demonstrated that our methods produce reserve prices that, at equilibrium, yield higher revenue more efficiently (with fewer samples) than standard baselines.

Ad exchanges are one example of a broader class of markets known as combinatorial markets. In a combinatorial market, goods are indivisible, meaning that a seller must allocate each good entirely to a single buyer. These markets are pervasive and fundamental to modern economies; examples include the allocation of online advertisement, the allocation of rights to use radio spectrum, etc. The combinatorial nature of buyers' preferences complicates their analysis, particularly buyers' ability to completely characterize their preferences in the presence of exponentially many bundles of goods. Our final contribution is a simple but powerful model that allows buyers to express their uncertainty over bundles of goods and learning algorithms that, with high probability, preserve the market's approximate competitive equilibria where the approximation error is a function of buyer's uncertainty over bundles of goods. Extensive experimentation on spectrum auction value models shows the efficacy of our learning algorithms. Moreover, our experiments illustrate the various trade-offs a market maker faces between an outcome's quality and the required amount of computational resources when implementing a market for which bidders have uncertain values.

Much work remains to make the methods developed in this thesis fully usable in practice. We hope that this thesis effectively introduced key theoretical and experimental insights that researchers can build upon to make further contributions of their own and eventually products and services that improve quality of life everywhere.

We conclude by pointing out future research directions we believe will lead to fruitful results.

Future Work

Our work opens new and exciting future research directions. We mention some of them next.

Equilibria Subset. In our work, we paid particular attention to algorithms that rigorously learn, up to some user-desired approximation error, a simulation-based game's equilibria set, or a noisy combinatorial market's competitive equilibria set. A consequence of this choice is that, in practice, our methods sometimes require a prohibitively large number of samples. One possibility to lessen this high-sampling requirement would be to adapt our methods to learn only some equilibria subset. The challenge would be to identify relevant parts of the equilibria space and discard the rest. Another possibility would be to relax the method's rigorousness and instead settle for weaker guarantees achievable with fewer samples and computational resources.

Heuristic concentration inequalities. Empirically, we found that concentration inequalities such as Hoeffding's or Bennett's are often too pessimistic in the number of samples they require. In practice, over the set of games we tested, the number of samples needed for these inequalities to retain strict guarantees on learned games or mechanisms' quality is far superior to the actual number of samples necessary. While this phenomenon might be due to our sometimes admittedly simple noise models, an interesting future direction would be to investigate heuristic relaxations of these worst-case inequalities and evaluate them in practice. Said practical evaluation would require designing more sophisticated noise models or collecting samples directly from real-world applications, both of which we believe are fruitful research directions.

Simulation-based game-theoretic solution concepts. Our current empirical mechanism design uses a parameterized simulation-based game's strongly connected components to predict agents' behavior. While the strongly connected components are statistically and computationally tractable, work remains to evaluate their predictive power in simulation-based games. Furthermore, the strongly connected components are a relaxation of sink equilibria itself statistically intractable but computationally tractable. The question remains whether we can find other solution concepts "in-between" the strongly connected components and sink equilibria that are both computationally and statistically tractable. It would also be worthwhile to investigate different solution concepts, e.g., correlated equilibria, and their applicability in our empirical mechanism design methodology.

Empirical mechanism design by competitive equilibria. Finally, when designing combinatorial markets, an alternative to learning game-theoretic solution concepts for agent's predictive behavior would be to learn competitive equilibria as demonstrated in chapter 6.

Bibliography

[1] amazon.com. Amazon display ads. https://advertising.amazon.com/products/display-ads, 2020. Accessed: 2020-04-05.

[2] amazon.com. Amazon sponsored products. https://advertising.amazon.com/products/sponsored-products, 2020. Accessed: 2020-04-05.

[3] E. Areyan Viqueira. Principled autonomous decision making for markets. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1869–1870, 2017.

[4] E. Areyan Viqueira, A. Greenwald, and V. Naroditskiy. On approximate welfare-and revenue-maximizing equilibria for size-interchangeable bidders. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1466–1468. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[5] E. Areyan Viqueira, A. Greenwald, and V. Naroditskiy. On approximate welfare-and revenue-maximizing equilibria for size-interchangeable bidders. *arXiv preprint arXiv:1708.03097*, 2017.

[6] E. Areyan Viqueira, C. Cousins, E. Upfal, and A. Greenwald. Learning equilibria of simulation-based games. *arXiv preprint arXiv:1905.13379*, 2019.

[7] E. Areyan Viqueira, A. Greenwald, C. Cousins, and E. Upfal. Learning simulation-based games from data. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1778–1780. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[8] E. Areyan Viqueira, C. Cousins, and A. Greenwald. Improved algorithms for learning equilibria in simulation-based games. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 79–87, 2020.

[9] E. Areyan Viqueira, C. Cousins, and A. Greenwald. Learning competitive equilibria in noisy combinatorial markets. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 2021.

[10] J.-Y. Audibert, R. Munos, and C. Szepesvári. Tuning bandit algorithms in stochastic environments. In *International conference on algorithmic learning theory*, pages 150–165. Springer, 2007.

[11] J.-Y. Audibert, R. Munos, and C. Szepesvári. Variance estimates and exploration function in multi-armed bandit. In *CERTIS Research Report 07–31*. Citeseer, 2007.

[12] M. Babaioff, N. Nisan, and R. Paes Leme. Price competition in online combinatorial markets. In *Proceedings of the 23rd international conference on World wide web*, pages 711–722, 2014.

[13] M. Balcan and N. J. A. Harvey. Learning submodular functions. In L. Fortnow and S. P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 793–802. ACM, 2011. doi: 10.1145/1993636.1993741. URL https://doi.org/10.1145/1993636.1993741.

[14] M. Balcan, F. Constantin, S. Iwata, and L. Wang. Learning valuation functions. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, volume 23 of *JMLR Proceedings*, pages 4.1–4.24. JMLR.org, 2012. URL http://proceedings.mlr.press/v23/balcan12b/balcan12b.pdf.

[15] M. Ball, G. Donohue, and K. Hoffman. Auctions for the safe, efficient, and equitable allocation of airspace system resources. *Combinatorial auctions*, 1, 2006.

[16] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, pages 72–87, 2003.

[17] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[18] D. R. Bellhouse, N. Fillion, et al. Le her and other problems in probability discussed by bernoulli, montmort and waldegrave. *Statistical Science*, 30(1):26–39, 2015.

[19] G. Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45, 1962.

[20] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[21] S. Bikhchandani and J. W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of economic theory*, 74(2):385–413, 1997.

[22] S. Bikhchandani, J. M. Ostroy, et al. The package assignment model. *Journal of Economic theory*, 107(2):377–406, 2002.

[23] V. Bilò, M. Flammini, G. Monaco, and L. Moscardelli. On the impact of buyers preselection in pricing problems. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1871–1873. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[24] V. Bilò, M. Flammini, G. Monaco, and L. Moscardelli. Pricing problems with buyer preselection. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[25] L. Blumrosen and N. Nisan. Combinatorial auctions. *Algorithmic game theory*, 267:300, 2007.

[26] T. Börgers. *An introduction to the theory of mechanism design*. Oxford University Press, USA, 2015.

[27] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[28] E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[29] E. Cantillon and M. Pesendorfer. Auctioning bus routes: The london experience. ., 2006.

[30] M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, A. R. Karlin, C. Mathieu, and M. Schwarz. Greedy bidding strategies for keyword auctions. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 262–271. ACM, 2007.

[31] Y. Chen, S. Goel, and D. M. Pennock. Pricing combinatorial markets for tournaments. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 305–314, 2008.

[32] Y. K. Cheung, R. Cole, and N. R. Devanur. Tatonnement beyond gross substitutes? gradient descent to the rescue. *Games and Economic Behavior*, 123:295–326, 2020.

[33] E. H. Clarke. Multipart pricing of public goods. *Public choice*, pages 17–33, 1971.

[34] W. Conen and T. Sandholm. Preference elicitation in combinatorial auctions. In M. P. Wellman and Y. Shoham, editors, *Proceedings 3rd ACM Conference on Electronic Commerce (EC-2001), Tampa, Florida, USA, October 14-17, 2001*, pages 256–259. ACM, 2001. doi: 10.1145/501158.501191. URL https://doi.org/10.1145/501158.501191.

[35] V. Conitzer and T. Sandholm. An algorithm for automatically designing deterministic mechanisms without payments. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 128–135. IEEE, 2004.

[36] P. Cramton et al. Spectrum auctions. *Handbook of telecommunications economics*, 1:605–639, 2002.

[37] R. W. Crandall. New zealand spectrum policy: A model for the united states? *The Journal of Law and Economics*, 41(S2):821–840, 1998.

[38] R. K. Dash, N. R. Jennings, and D. C. Parkes. Computational-mechanism design: A call to arms. *IEEE intelligent systems*, 18(6):40–47, 2003.

[39] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[40] S. De Vries and R. V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on computing*, 15(3):284–309, 2003.

[41] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1): 242–259, 2007.

[42] T. Elomaa and M. Kääriäinen. Progressive Rademacher sampling. In *AAAI/IAAI*, pages 140–145, 2002.

[43] E. Even Dar, V. S. Mirrokni, S. Muthukrishnan, Y. Mansour, and U. Nadav. Bid optimization for broad match ad auctions. In *Proceedings of the 18th international conference on World wide web*, pages 231–240. ACM, 2009.

[44] M. Feldman, N. Gravin, and B. Lucier. Combinatorial walrasian equilibrium. *SIAM Journal on Computing*, 45(1):29–48, 2016.

[45] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 548–553. Morgan Kaufmann, 1999. URL http://ijcai.org/Proceedings/99-1/Papers/079.pdf.

[46] D. Gale. Equilibrium in a discrete exchange economy with money. *International Journal of Game Theory*, 13(1):61–64, 1984.

[47] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[48] R. Garg, S. Kapoor, and V. Vazirani. An auction-based market equilibrium algorithm for the separable gross substitutability case. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 128–138. Springer, 2004.

[49] N. Gatti and M. Restelli. Equilibrium approximation in simulation-based extensive-form games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, page 199–206, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 0982657153.

[50] S. Gerakaris and S. Ramamoorthy. Learning best response strategies for agents in ad exchanges. In *European Conference on Multi-Agent Systems*, pages 77–93. Springer, 2018.

[51] R. Gibbons. *Game theory for applied economists*. Princeton University Press, 1992.

[52] M. Goemans, V. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 142–151. IEEE, 2005.

[53] J. K. Goeree and C. A. Holt. Hierarchical package bidding: A paper & pencil combinatorial auction. *Games and Economic Behavior*, 70(1):146–169, 2010.

[54] S. Govindan and R. Wilson. A global newton method to compute nash equilibria. *Journal of Economic Theory*, 110(1):65–86, 2003.

[55] S. Govindan and R. Wilson. Computing nash equilibria by iterated polymatrix approximation. *Journal of Economic Dynamics and Control*, 28(7):1229–1241, 2004.

[56] A. Greenwald, V. Naroditskiy, and S. J. Lee. Bidding heuristics for simultaneous auctions: Lessons from tac travel. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, pages 131–146. Springer, 2010.

[57] T. Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[58] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic theory*, 87(1):95–124, 1999.

[59] M. Guo and V. Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[60] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1164–1173. Society for Industrial and Applied Mathematics, 2005.

[61] O. Hinz, I.-H. Hann, and M. Spann. Price discrimination in e-commerce? an examination of dynamic pricing in name-your-own price markets. *Mis quarterly*, pages 81–98, 2011.

[62] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[63] iab.com. U.s. digital ad revenue climbs to $57.9 billion in first half 2019, up 17% yoy, according to iab internet advertising revenue report. https://www.iab.com/news/u-s-digital-ad-revenue-climbs-to-57-9-billion-in-first-half-2019/, 2020. Accessed: 2020-05-09.

[64] T. Jha and Y. Zick. A learning framework for distribution-based game-theoretic solution concepts. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 355–377, 2020.

[65] P. R. Jordan and M. P. Wellman. Designing an ad auctions game for the trading agent competition. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer, 2010.

[66] P. R. Jordan, C. Kiekintveld, and M. P. Wellman. Empirical game-theoretic analysis of the tac supply chain game. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 193. ACM, 2007.

[67] P. R. Jordan, Y. Vorobeychik, and M. P. Wellman. Searching for approximate equilibria in empirical games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 1063–1070. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[68] M. Kaneko and Y. Yamamoto. The existence and computation of competitive equilibria in markets with an indivisible commodity. *Journal of Economic Theory*, 38(1):118–136, 1986.

[69] A. S. Kelso Jr and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, pages 1483–1504, 1982.

[70] W. Ketter, M. Peters, and J. Collins. Autonomous agents in future energy markets: The 2012 power trading agent competition. In *AAAI*, 2013.

[71] D. Koller, N. Megiddo, and B. Von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and economic behavior*, 14(2):247–259, 1996.

[72] V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[73] V. Krishna. *Auction theory.* Academic press, 2009.

[74] C. Kroer, A. Peysakhovich, E. Sodomka, and N. E. S. Moses. Computing large market equilibria using abstractions. In A. Karlin, N. Immorlica, and R. Johari, editors, *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 745–746. ACM, 2019. doi: 10.1145/3328526.3329553. URL https://doi.org/10.1145/3328526.3329553.

[75] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[76] A. M. Kwasnica, J. O. Ledyard, D. Porter, and C. DeMartini. A new and improved design for multiobject iterative auctions. *Management science*, 51(3):419–434, 2005.

[77] S. Lahaie and B. Lubin. Adaptive-price combinatorial auctions. In A. Karlin, N. Immorlica, and R. Johari, editors, *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 749–750. ACM, 2019. doi: 10.1145/3328526.3329615. URL https://doi.org/10.1145/3328526.3329615.

[78] S. Lahaie and D. C. Parkes. Applying learning algorithms to preference elicitation. In J. S. Breese, J. Feigenbaum, and M. I. Seltzer, editors, *Proceedings 5th ACM Conference on Electronic Commerce (EC-2004), New York, NY, USA, May 17-20, 2004*, pages 180–188. ACM, 2004. doi: 10.1145/988772.988800. URL https://doi.org/10.1145/988772.988800.

[79] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[80] K. Leyton-Brown, E. Nudelman, and Y. Shoham. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In *International Conference on Principles and Practice of Constraint Programming*, pages 556–572. Springer, 2002.

[81] B. Lubin and D. C. Parkes. *Combinatorial markets in theory and practice: Mitigating incentives and facilitating elicitation.* Citeseer, 2010.

[82] J. K. MacKie-Mason, A. Osepayshvili, D. M. Reeves, and M. P. Wellman. Price prediction strategies for market-based scheduling. *AAAI*, 2004.

[83] magnaglobal.com. Global advertising forecast. https://www.magnaglobal.com/wp-content/uploads/2016/12/MAGNA-December-Global-Forecast-Update-Press-Release.pdf, 2017. Accessed: 2017-11-15.

[84] A. Marchesi, F. Trovò, and N. Gatti. *Learning Probably Approximately Correct Maximin Strategies in Simulation-Based Games with Infinite Strategy Spaces*, page 834–842. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2020. ISBN 9781450375184.

[85] A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[86] R. McKelvey, A. McLennan, and T. Turocy. Gambit: Software tools for game theory, version 16.0.1., 2019. URL http://www.gambit-project.org.

[87] Y. Mohammad, E. A. Viqueira, N. A. Ayerza, A. Greenwald, S. Nakadai, and S. Morinaga. Supply chain management world. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 153–169. Springer, 2019.

[88] R. B. Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[89] R. B. Myerson. *Game theory*. Harvard university press, 2013.

[90] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 1950.

[91] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic behavior*, 35 (1-2):166–196, 2001.

[92] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. Algorithmic game theory, 2007. *Google Scholar Google Scholar Digital Library Digital Library*, 2007.

[93] E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 880–887. IEEE Computer Society, 2004.

[94] J. O'Donovan, R. Jones, L. R. Marusich, Y. Teng, C. Gonzalez, and T. Höllerer. A model-based evaluation of trust and situation awareness in the diner's dilemma game. In *Proceedings of the 22th Behavior Representation in Modeling & Simulation (BRIMS) Conference*, pages 11–14, 2013.

[95] D. C. Parkes. *Iterative combinatorial auctions*. MIT press, 2006.

[96] D. C. Parkes. Computational mechanism design. In *Lecture Notes of Tutorials Presented at the 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-05)*. Institute of Mathematical Sciences, University of Singapore, 2008.

[97] D. C. Parkes. When analysis fails: Heuristic mechanism design via self-correcting procedures. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 62–66. Springer, 2009.

[98] S. Phelps, S. Parsons, P. McBurney, and E. Sklar. Co-evolution of auction mechanisms and trading strategies: Towards a novel approach to microeconomic design. In *In GECCO-02 Workshop on Evolutionary Computation in Multi-Agent Systems*. Citeseer, 2002.

[99] S. Phelps, S. Parsons, E. Sklar, and P. McBurney. Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the workshop on Agents for Electronic Commerce*, 2003.

[100] V. Picheny, M. Binois, and A. Habbal. A bayesian optimization approach to find nash equilibria. *arXiv preprint arXiv:1611.02440*, 2016.

[101] T. Popoviciu. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9: 129–145, 1935.

[102] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008.

[103] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[104] M. Riondato and E. Upfal. Mining frequent itemsets through progressive sampling with Rademacher averages. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM, 2015.

[105] M. Riondato and E. Upfal. Abra: Approximating betweenness centrality in static and dynamic graphs with Rademacher averages. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):61, 2018.

[106] A. E. Roth. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica: Journal of the Econometric Society*, pages 425–427, 1986.

[107] A. E. Roth. Deferred acceptance algorithms: History, theory, practice, and open questions. *international Journal of game Theory*, 36(3-4):537–569, 2008.

[108] A. E. Roth and M. Sotomayor. Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541, 1992.

[109] T. Roughgarden. Algorithmic game theory. *Commun. ACM*, 53(7):78–86, 2010. doi: 10.1145/1785414.1785439. URL https://doi.org/10.1145/1785414.1785439.

[110] T. Roughgarden and I. Talgam-Cohen. Why prices need algorithms. In T. Roughgarden, M. Feldman, and M. Schwarz, editors, *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 19–36. ACM, 2015. doi: 10.1145/2764468.2764515. URL https://doi.org/10.1145/2764468.2764515.

[111] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109, 1982.

[112] N. Sadeh, R. Arunachalam, J. Eriksson, N. Finne, and S. Janson. Tac-03–a supply-chain trading competition. *AI magazine*, 24(1):92, 2003.

[113] F. Sailer, M. Buro, and M. Lanctot. Adversarial planning through strategy simulation. In *2007 IEEE symposium on computational intelligence and games*, pages 80–87. IEEE, 2007.

[114] M. J. Saltzman. Coin-or: an open-source library for optimization. In *Programming languages and systems in computational economics and finance*, pages 3–32. Springer, 2002.

[115] T. Sandholm. Automated mechanism design: A new application area for search algorithms. In *International Conference on Principles and Practice of Constraint Programming*, pages 19–36. Springer, 2003.

[116] M. Schain and Y. Mansour. Ad exchange–proposal for a new trading agent competition game. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pages 133–145. Springer, 2013.

[117] T. Scheffel, G. Ziegler, and M. Bichler. On the impact of package selection in combinatorial auctions: an experimental study in the context of spectrum auction design. *Experimental Economics*, 15(4):667–692, 2012.

[118] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[119] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2): 146–160, 1972.

[120] A. Tavares, H. Azpurua, A. Santos, and L. Chaimowicz. Rock, paper, starcraft: Strategy selection in real-time strategy games. In *The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)*, 2016.

[121] K. Tuyls, J. Perolat, M. Lanctot, J. Z. Leibo, and T. Graepel. A generalised method for empirical game theoretic analysis. *arXiv preprint arXiv:1803.06376*, 2018.

[122] K. Tuyls, J. Perolat, M. Lanctot, E. Hughes, R. Everett, J. Z. Leibo, C. Szepesvári, and T. Graepel. Bounds and dynamics for empirical game theoretic analysis. *Autonomous Agents and Multi-Agent Systems*, 34(1):7, 2020.

[123] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[124] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[125] E. A. Viqueira, C. Cousins, Y. Mohammad, and A. Greenwald. Parametric mechanism design under uncertainty. In *To Appear on the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, 2019.

[126] E. A. Viqueira, C. Cousins, and A. Greenwald. Learning competitive equilibria in noisy combinatorial markets. *arXiv preprint arXiv:2101.09551*, 2021.

[127] J. Von Neumann, O. Morgenstern, and H. W. Kuhn. *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

[128] Y. Vorobeychik. Probabilistic analysis of simulation-based games. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 20(3):16, 2010.

[129] Y. Vorobeychik and M. P. Wellman. Stochastic search methods for nash equilibrium approximation in simulation-based games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 1055–1062. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[130] Y. Vorobeychik, C. Kiekintveld, and M. P. Wellman. Empirical mechanism design: methods, with application to a supply-chain scenario. In *Proceedings of the 7th ACM conference on Electronic commerce*. ACM, 2006.

[131] Y. Vorobeychik, M. P. Wellman, and S. Singh. Learning payoff functions in infinite games. *Machine Learning*, 67(1-2):145–168, 2007.

[132] L. Walras. *Elements of Pure Economics: Or the Theory of Social Wealth*. Elements of Pure Economics, Or the Theory of Social Wealth. Routledge, 2003. ISBN 9780415313407. URL https://books.google.com/books?id=hwjRD3z0Qy4C.

[133] M. Weiss, B. Lubin, and S. Seuken. Sats: A universal spectrum auction test suite. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 51–59. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[134] M. P. Wellman. Methods for empirical game-theoretic analysis. In *AAAI*, pages 1552–1556, 2006.

[135] M. P. Wellman. Trading agents. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–107, 2011.

[136] M. P. Wellman, P. R. Jordan, C. Kiekintveld, J. Miller, and D. M. Reeves. Empirical game-theoretic analysis of the tac market games. In *AAMAS-06 Workshop on Game-Theoretic and Decision-Theoretic Agents*, volume 10, 2006.

[137] M. P. Wellman, A. Greenwald, and P. Stone. *Autonomous bidding agents: Strategies and lessons from the trading agent competition*. Mit Press, 2007.

[138] M. P. Wellman, T. H. Kim, and Q. Duong. Analyzing incentives for protocol compliance in complex domains: A case study of introduction-based routing. *arXiv preprint arXiv:1306.0388*, 2013.

[139] B. Wiedenbeck. Approximate game theoretic analysis for large simulation-based games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1745–1746, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-2738-1. URL http://dl.acm.org/citation.cfm?id=2615731.2616156.

[140] B. Wiedenbeck, F. Yang, and M. P. Wellman. A regression approach for modeling games with many symmetric players. In *32nd AAAI Conference on Artificial Intelligence*, 2018.

[141] S. Zamir et al. Bayesian games: Games with incomplete information. Technical report, N/A, 2008.

[142] B. H. Zhang and T. Sandholm. Finding and certifying (near-) optimal strategies in black-box extensive-form games. *arXiv preprint arXiv:2009.07384*, 2020.