Mechanism Design and Integer Programming in the Data Age

Thesis Proposal

Siddharth Prasad

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Maria-Florina Balcan (CMU; co-chair)
Tuomas Sandholm (CMU; co-chair)
Gérard Cornuéjols (CMU)
Craig Boutilier (Google)
Peter Cramton (University of Maryland)

Presented in partial fulfillment of the requirements for the degree of Doctor of Philosophy

April 26, 2024

Abstract

Modern-day human-scale marketplaces such as recommender systems, advertisement markets, matching platforms, supply chain industries, electronic commerce platforms, and others must reckon with a balancing act of (i) understanding and respecting the incentives of the system's participants, (ii) obtaining optimal outcomes subject to those incentives, and (iii) ensuring that data is used in a sound manner to improve overall efficiency. The area of *mechanism design* from economics provides a rich language and broad toolkit to understand incentives and *integer programming* is a powerful and extremely expressive optimization language that is the workhorse behind most practical solutions to real-world discrete optimization problems. This thesis studies how data-driven decisions can be integrated into fundamental algorithms from both areas to improve performance (economic, memory, run-time, *etc.*).

Within mechanism design, I focus on the design of revenue optimal mechanisms from a data-driven lens. I design algorithms, model new learning paradigms, and invent new mechanism classes for a variety of settings including two-part tariffs, combinatorial auctions, shrinking markets, and general multi-dimensional mechanism design. A highlight here is my development of the first general tunable framework for integrating side information into mechanisms to boost revenue while preserving efficiency and incentive properties. Within integer programming, I develop principled new methods for cutting plane configuration, which is one of the most important components in state-of-the-art branch-and-cut integer programming solvers. I develop a comprehensive generalization theory for cutting plane configuration that (i) unveils new geometric and combinatorial structure in the branch-and-cut algorithm and the class of Gomory cuts, (ii) improves upon and subsumes prior work via an abstract model of the underlying tree search, and (iii) is validated through experiments that demonstrate the impact of data-dependent parameter tuning. I also derive a new method of sequence-independent lifting for cutting planes that I validate through rigorous theory—by deriving broad conditions under which the new cuts define facets of the integer polytope—and extensive experiments.

I conclude by discussing promising avenues for future work, most notably my preliminary ideas on *core-selecting combinatorial auctions* with side information which calls for a blend of innovations in both auction design and integer programming/search techniques.

Contents

Chapter 1

Introduction

Modern-day human-scale marketplaces such as recommender systems, advertisement markets, matching platforms, supply chain industries, electronic commerce platforms, and others must reckon with a balancing act of (i) understanding and respecting the incentives of the system's participants, (ii) obtaining optimal outcomes subject to those incentives, and (iii) ensuring that data is used in a sound manner to improve overall efficiency. Many of these areas of great importance can be thought of as having two facets: (i) the human aspect where the input to the problem needs to be effectively aggregated from possibly strategic participants and (ii) the underlying optimization aspect where once human-provided inputs have been collected, a complex discrete optimization problem must be solved. The area of *mechanism design* from economics provides a rich language and broad toolkit to understand incentives and *integer programming* is a powerful and extremely expressive optimization language that captures most discrete optimization problems in the real world. In this thesis we study how learning from data can improve and augment fundamental algorithms and techniques for mechanism design, integer programming, and the rich interactions that ensue.

Mechanism design Mechanism design is a high-impact branch of economics and computer science that studies the implementation of socially desirable outcomes among strategic self-interested agents. Major real-world use cases include combinatorial auctions (*e.g.*, strategic sourcing, radio spectrum auctions, real estate auctions), matching markets (*e.g.*, housing allocation, ridesharing), project fundraisers, strategic routing, and many more. Revenue maximization in mechanism design is a difficult open problem that is only understood in very special cases. The seminal work of Myerson [83] characterized the revenue-optimal mechanism for the sale of a single item in the Bayesian setting, but it is not even known how to optimally sell two items.

We study the design of revenue optimal mechanisms—an area where classical economic theory has been unable to make significant progress towards closed-form characterizations (this is a barrier that can be explained by a wide swath of computational intractability results; see, for example, Conitzer and Sandholm [44])—from a data-driven lens. Machine learning theory provides tools to make concrete statements regarding the overall quality of a mechanism based on empirical performance on data. This approach provides a practical avenue towards finding high-quality solutions to otherwise intractable mechanism design problems. We show how mechanisms that are learned from (or augmented with predictions from) data can provably achieve strong revenue guarantees in a variety of different settings.

Integer programming My work on integer programming has focused on designing more principled methods for cutting plane configuration. Cutting planes are one of the most important components in state-of-the-art branch-and-cut integer programming solvers like Gurobi and CPLEX. Cutting plane selection, however, is an inexact science lacking formal mathematical guidelines. An increasingly popular approach is to use machine learning to configure cutting plane selection parameters based on a training set of integer programs from the application domain (e.g., an airline company might solve similar scheduling problems from day to day). We develop the first formal guarantees for machine-learning based cut selection; placing this line of largely empirical research on solid theoretical foundations. We also investigate fundamental topics in integer programming, specifically *sequence-independent lifting* which is a technique for strengthening cutting planes. We derive a new method of performing sequence-independent lifting and validate our techniques through theory and practice.

Learning to tune algorithms and mechanisms My work on data-driven aspects of mechanism design and integer programming follows a common theme: observations from data assist with the critical decisions that are made within existing backbone solution techniques. Within mechanism design this often involves augmenting variants of the classical Vickrey-Clarke-Groves mechanism with data to improve revenue and within integer programming this involves tuning the decisions made by the branch-and-cut algorithm. Rather than using machine learning as the complete end-to-end solution, we seek to improve the decision making of algorithms that already work to produce desired outcomes in order to, for example, improve revenue or run-time. For example, rather than using machine learning to come up with the allocation rule and payment rule of a mechanism directly, we use machine learning to tune the parameters within a class of mechanisms that already satisfy certain desirable properties. Our approach is a blend of two paradigms in the realm of machine learning for combinatorial optimization described by Bengio et al. [32]: (i) learning to initialize a tunable algorithm and (ii) a synergistic pairing where machine learning dynamically influences the intermediate decisions of a tunable algorithm. Two other recognized paradigms my work contributes to are *data-driven algorithm design* [15] and *learning-augmented algorithm* *design or algorithms with predictions [80].*

Future work: integer programming techniques in mechanism design Understanding the interplay between integer programming and the economic aspects of the problems being solved has been very fruitful with significant real-world impact; a prime example is combinatorial auctions [10, 77, 92, 93]. I want to bring my advances in integer programming and data-augmented mechanism design to the aid of computationally tough problems like the design of core-selecting combinatorial auctions. These are auction outcomes where no coalition of bidders plus the seller can jointly deviate to obtain a better outcome and have been used in various high-stakes spectrum auctions in Canada, the UK, and other countries. More generally, the operations research problems that arise in the human-scale marketplaces of the present day call for the development of synergies between fundamental optimization methods and predictions learned from data, all while balancing the incentives and needs of the participating agents. I plan to align my future research with this goal.

1.1 Outline of proposal

I summarize the main contributions of each chapter of this thesis and list citations to corresponding research papers.

Chapter 2: Learning across and within instances in mechanism design In this first chapter, we demonstrate the practical applicability of data-driven algorithm design on a problem of real-world importance; the optimization of a widely-used pricing scheme called a two-part tariff (and menus thereof). We then study the design of revenue-maximizing combinatorial auctions when there is no historical data available (a prior-free setting). We show that learning-based approaches can nonetheless yield fruitful guarantees in such settings. This chapter is based on research from the following publications:

[20] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Efficient algorithms for learning revenue-maximizing two-part tariffs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020 [link]

[22] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Learning within an instance for designing high-revenue combinatorial auctions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021 [link]

[24] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Maximizing revenue under market shrinkage and market uncertainty. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022 [link]

Chapter 3: Mechanism design with side information This chapter studies mechanism design with side information about the agents, which is a form of learning-augmented mechanism design. This chapter is based on research from the following publication:

[27] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Bicriteria multidimensional mechanism design with side information. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023 [link]

Chapter 4: Learning to cut in integer programming In this chapter we show how data-driven algorithm configuration can help tune one of the most important aspects of integer programming solvers: cutting plane selection. We develop a comprehensive generalization theory for cutting planes. This chapter is based on research from the following publications:

[23] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of tree search configuration: Cutting planes and beyond. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021 [link]

[25] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Improved sample complexity bounds for branch-and-cut. In *International Conference on Principles and Practice of Constraint Programming (CP)*, 2022 [link]

[26] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Structural analysis of branch-and-cut and the learnability of Gomory mixed integer cuts. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022 [link]

Chapter 5: New sequence-independent lifting methods While the previous chapter was about tuning cutting plane selection policies, in this chapter we derive fundamentally new families of cuts. These cuts are obtained via new sequence-independent lifting techniques we derive. This chapter is based on research from the following preprint:

[88] Siddharth Prasad, Ellen Vitercik, Maria-Florina Balcan, and Tuomas Sandholm. New sequence-independent lifting techniques for cutting planes and when they induce facets. 2023 [link]

Chapter 6: Future research I outline future directions I plan to embark on for the remainder of my PhD. These include a study of side information in (core-selecting) combinatorial auctions (both auction design aspects and scalability/computational aspects), new cutting plane generation techniques, and some new ideas towards improved large-scale integer programming.

Other work not in thesis I have also done research in learning theoretic approaches to preference elicitation and the theory of choice more broadly; specifically learning intertemporal choice [38] and incentive compatible active learning [57]. During my Google internship in 2022 I developed content prompting policies and content provider dynamics to improve user welfare in recommender ecosystems [87].

Chapter 2

Learning across and within instances in mechanism design

A recent exciting line of work proposes the use of machine learning to automatically design high-revenue mechanisms (auctions, pricing schemes, etc.) from samples. We present some new approaches that stem from this idea—including algorithms for efficient learning, and learning when samples are not even available.

In Section 2.1, we tackle algorithmic challenges of learning high-revenue two-part tariff structures from samples. A two-part tariff is a pricing scheme consisting of an up-front lump-sum fee and a per-unit fee—real-world examples include gym memberships, cell phone plans, *etc*. We present practically-efficient algorithms for learning two-part tariff structures in various scenarios. We then show that situations involving market segmentation induce computational hardness, but show how to circumvent that hardness when buyers are identically distributed.

We then present some new approaches to the elusive problem of designing high-revenue (limited supply) multi-item, multi-bidder auctions. While Section 2.1 dealt with "learning across instances", here we assume that the mechanism designer is faced with a single instance of bidders who show up. First, in Section 2.2, we present a new *learning within an instance* mechanism that generalizes and improves upon previous random-sampling mechanisms for unlimited supply, and prove strong revenue guarantees for this mechanism. Then, in Section 2.3, we show how to learn an auction that is robust to market shrinkage and market uncertainty. If there is a fixed population of buyers known to the seller, but only some random (unknown) fraction of them participate in the market, how much revenue can the seller guarantee?

2.1 Efficient algorithms for learning revenue-maximizing two-part tariffs

A *two-part tariff (TPT)* consists of an up-front lump sum fee p_1 and a fee p_2 for every additional unit purchased. Various goods and services are priced using such a scheme. For example, Keurig sells coffee machines (the up-front fee) that require proprietary coffee pods (the per unit fee). Another example is health club memberships, where participants often are required to pay an up-front fixed membership fee, as well as a monthly fee. More generally, a length L menu of TPTs is a list $((p_1^1, p_2^1), \ldots, (p_1^L, p_2^L))$ of L TPTs, and a buyer may elect to pay according to any one of the L TPTs (or not to buy anything). Menus of TPTs are also prevalent: health clubs, amusement parks, wholesale stores like Costco, cell phone companies, and credit card companies all frequently offer various tiers of membership usually consisting of lower future payments for a larger up-front payment.

In an early analysis of TPTs, Oi [86] inspires the problem via Disneyland trying to decide between charging attendees a hefty entrance fee and allowing them free access to rides, versus charging a nominal entrance fee but requiring payment for each ride. An even earlier discussion of TPTs is given by Lewis [75], where the merits and drawbacks of TPTs are discussed in contexts such as the telephone system, gas legislation, and the UK Central Electricity Board.

We study the problem of learning high-revenue menus of TPTs from buyer valuation data. This can be viewed as a form of *automated mechanism design* [44]. In our setting, the seller has access to samples from the distribution over buyers' values, but not an explicit description thereof. This differs from the usual approach taken by the economic theory literature, and instead takes the *sample-based approach to mechanism design*, introduced by Sandholm and Likhodedov [94]. Balcan et al. [19] study the sample complexity of revenue maximization, deriving a broad characterization of the number of samples needed to ensure with high probability that a mechanism that achieves high empirical revenue on the samples also generalizes well, that is, achieves high expected revenue over a freshly drawn sample. Our main goal is to provide efficient algorithms for finding menus of TPTs that achieve high empirical revenue over a given set of samples. Many of the mechanism settings studied by Balcan et al. [19] have large parameter spaces and require a number of samples that is exponential in the problem parameters to guarantee generalization. However, they show that the sample complexity of TPTs has only a mild (at most linear) dependence on the parameters, so it is reasonable to ask for sample efficient and computationally efficient algorithms for finding nearly optimal solutions. We present such algorithms, thereby providing the missing, complementary piece to the results of Balcan et al. [19]. Our algorithms also have the obvious practical uses in designing TPTs and menus thereof.

Main contributions We give efficient algorithms for finding the empirical revenue maximizing menu of TPTs when the menu length is a fixed constant. Our main result here is an $O(N^3K^3)$ algorithm when the menu length $L = 1$ in the single buyer setting, where N is the number of samples and K is the maximum quantity of the item. Our algorithm generalizes to an $O(n^3N^3K^3)$ algorithm in the multi-buyer setting with n buyers. We then give an $(NK)^{O(L)}$ algorithm for the setting where the menu length $L \geq 1$. This algorithm exploits the geometric structure of the problem—buyers' valuations partition the parameter space into several convex polytopes, and revenue maximization over each polytope reduces to solving a linear program.

We then generalize the problem to multiple markets. We prove how many samples suffice to guarantee that a two-part tariff scheme that is feasible on the samples is also feasible on a new problem instance with high probability. We then show that computing revenue-maximizing feasible prices is NP-hard even for buyers with additive valuations. Then, for bidders with identical valuation distributions, we present a condition that is sufficient for the two-part tariff scheme from the unsegmented setting to be optimal and feasible for the market-segmented setting. Finally, we prove a generalization result that states how many samples suffice so that we can compute the unsegmented solution on the samples and still be guaranteed that we get a near-optimal solution for the market-segmented setting with high probability.

2.2 Learning within an instance for designing high-revenue combinatorial auctions

In a (limited-supply) *combinatorial auction*, a seller has m indivisible items to allocate among a set S of n bidders. Combinatorial auctions have various real-world applications. Two examples include auctions for allocating licenses for bands of the electromagnetic spectrum and sourcing auctions for supply chain management. The design of truthful, high-revenue combinatorial auctions is a central problem in mechanism design. A comprehensive account of combinatorial auctions may be found in Cramton et al. [47].

A common strategy for designing truthful, high-revenue auctions when there is an *unlimited supply* of each good has been to use a random-sampling mechanism. A random-sampling mechanism splits the bidders into two groups, and applies the optimal auction for each group to the other group (thereby achieving truthfulness, since the auction run on any bidder's group is independent of her reported valuation). In unlimited-supply settings, random-sampling mechanisms satisfy strong guarantees [4, 16, 63].

However, until now, there has been no unified, general-purpose method of adapting the random-sampling approach to analyze the limited-supply setting. Limited supply poses additional significant technical challenges, since allocations of items to bidders must be feasible. For example, random-sampling with any mechanism class that allows bidders to purchase according to their demand functions would violate supply constraints. Most adaptations of random-sampling to limited supply deal with feasibility issues in complicated ways, for example, by constructing intricate revenue benchmarks to limit the number of buyers who can make a purchase [17], or by placing combinatorial constraints on the environment [53, 54].

In this work we circumvent these issues by applying auction formats that generalize the classical Vickrey-Clarke-Groves (VCG) auction [42, 66, 97] to sell all m items to a random group of participatory bidders. These auctions prescribe feasible allocations and payments (and are incentive compatible). Several parameterized generalizations of the VCG auction have been studied with the aim of increasing revenue by introducing weights to favor certain bidders or allocations. Examples include affine-maximizer auctions (AMAs) [90], virtual-valuations combinatorial auctions (VVCAs) [94], λ-auctions [70], mixed-bundling auctions with reserve prices [96], and mixed-bundling auctions [70]. However, little is known when it comes to formal approximation guarantees for these auction classes.

A direct adaptation of vanilla random sampling can do poorly when the auction class is rich due to overfitting to the training group of bidders. Our main *learning-within-an-instance (LWI) mechanism* alleviates these issues by randomly drawing a set of participatory bidders S_{par}, and then sampling several proportionally-sized learning groups from $S_{lrn} := S \setminus S_{par}$ to learn an auction that is close-to-optimal *in expectation* for a random learning group. Our approach is a form of *automated mechanism design* [44, 94].

Main contributions First, we provide the main guarantees satisfied by our LWI framework. The guarantees are derived using learning-theoretic techniques.

We then introduce a new class of auctions called *bundling-boosted auctions*. These auctions are parameterized in a way that does not depend on the number of bidders who participate in the auction (unlike most previous generalizations of the VCG auction). We prove bounds on the intrinsic complexity of bundling-boosted auctions (and a few other natural subclasses of auctions) that have no dependence on the number of bidders, and instantiate our LWI framework with this auction class.

Finally, we show how our learning-within-an-instance mechanism can be implemented in a sample and computationally efficient manner for bundling-VCG auctions and sparse bundling-boosted auctions by leveraging practically efficient routines for solving winner determination. We also show how to use structural revenue maximization to decide what auction class to use with LWI when there is a constraint on the number of learning groups.

2.3 Maximizing revenue under market shrinkage and market uncertainty

A shrinking market with uncertain buyer participation is a natural phase of products' and services' lifecycles. Current examples of great importance include media consumers—known as cord cutters—who cancel cable-TV subscriptions in favor of streaming services [5, 78], a thinning customer base for department stores due to online retailers like Amazon [48, 64], and reduced capacities for restaurants during the COVID-19 pandemic [95]. In this work we study how mechanism design can help preserve revenue in this ubiquitous challenge of a shrinking market, specifically for combinatorial auctions for limited supply. The seller has m indivisible items to allocate to a set S of n bidders. The bidders can express how much they value each possible bundle $b \subseteq \{1, \ldots, m\}$ of items. Combinatorial auctions have had wide reach in practice, from strategic sourcing to spectrum auctions to estate auctions. Cramton et al. [47] provide a survey of various aspects of combinatorial auctions. The design of revenue-maximizing combinatorial auctions in multi-item, multi-bidder settings is an elusive and difficult problem that has spurred a long and active line of research combining techniques from economics, artificial intelligence, and theoretical computer science. This is still largely an open question and a very active research area.

We start a new strand within that topic area, namely the study of shrinking markets with uncertain buyer participation. We introduce the first formal model of market shrinkage in multi-item settings, and prove the first revenue guarantees. Specifically, we show how much revenue can be preserved when only a random unknown fraction of the set S of bidders participates in the market.

Main contributions We present the first formal analysis of how much revenue can be preserved in a shrinking market, for multi-item settings. More precisely, there is a set S of n bidders that is known to the mechanism designer. Each bidder participates in the market independently with probability p, but the valuations of the bidders who participate in the market, denoted by $S_0 \subseteq S$, are unknown (what is known is that they belong to S). We present a learning-based method for designing a mechanism that satisfies the first known revenue-preservation guarantees in this setting.

After formally defining the problem setting, we precisely show how to reckon with subtleties that arise when auctions are run among a shrunken market of unknown size. We then provide and discuss a simple example of a market where reduced competition in a shrinking market drives revenue to a lower threshold than one might expect. We furthermore show that if bidders' valuation functions can depend on what items other bidders receive, there exist scenarios in which only an exponentially small (in the number of items) fraction of the revenue obtainable by even the vanilla Vickrey-Clarke-Groves (VCG) auction on S can be guaranteed on a random subset of bidders, even if a large fraction of the market shows up. For example, if 50 items are for sale and each bidder shows up independently with 90% probability, our construction yields a maximum expected revenue of roughly 7% of the VCG revenue on S. If 100 items are for sale, at most 0.52% of the VCG revenue on S can be guaranteed.

Our main theorem is the following revenue guarantee obtained via a sample-based learning algorithm. Delineability is a structural assumption introduced by Balcan et al. [19] satisfied by nearly all commonly studied auction classes. $W_{\mathcal{M}}(S)$ denotes the maximum welfare achievable by mechanisms in $\mathcal{M}$, k is a term that depends on the number of winners in any mechanism in $\mathcal{M}$, and γ is a constant that depends on S. Rev_M denotes the revenue function induced by M.

Theorem 2.3.1. *Let $\mathcal{M}$ be (d,h)-delineable class of mechanisms. A mechanism $\widetilde{M} \in \mathcal{M}$ such that*

$$\mathbb{E}[\mathsf{Rev}_{\widetilde{M}}(S_0)] \geq \Omega\left(\frac{p^2}{k^{1+\log_{1/\gamma}(4/p)}}\right) W_{\mathcal{M}}(S) - \varepsilon$$

with probability at least $1-\delta$ can be computed in $NhT + (Nh)^{O(d)}$ time, where T is the time required to generate any given hyperplane witnessing delineability of any mechanism in $\mathcal{M}$ and $N = O\big(\frac{d\log(dh)}{\varepsilon^2}\log(\frac{1}{\delta})\big)$.

To prove our theorem, we first prove that

$$\sup_{M\in\mathcal{M}} \mathbb{E}[\mathsf{Rev}_M(S_0)] \geq \Omega\left(\frac{p^2}{k^{1+\log_{1/\gamma}(4/p)}}\right) W_{\mathcal{M}}(S),$$

which is the major technical contribution of this work. Our main technique is the analysis of a novel combinatorial structure we construct called a *winner diagram*, which is a graph that concisely captures all possible executions of an auction on an uncertain set of bidders. Via a probabilistic method argument that randomizes over a subgraph of the winner diagram, we arrive at a general possibility result: *if $\mathcal{M}$ is a sufficiently rich class of mechanisms, there always exists an $M \in \mathcal{M}$ that is robust to uncertainty/shrinkage in the market*. This implies our bound on $\sup \mathbb{E}[\mathsf{Rev}_M(S_0)]$. We primarily focus on the case where bidders participate in the market independently with probability p, but show how to generalize our results to any distribution over submarkets. Our bound is a parameterized guarantee that has interesting applications to practically motivated auction constraints: (1) limiting the number of winners and (2) bundling constraints on items.

We then present a learning algorithm to compute a mechanism $\widetilde{M}$ such that

$$\mathbb{E}[\mathsf{Rev}_{\widetilde{M}}(S_0)] \geq \sup_M \mathbb{E}[\mathsf{Rev}_M(S_0)] - \varepsilon$$

with high probability, which proves Theorem 2.3.1. Our algorithm exploits geometric structure and a linear-programming approach over hyperplane arrangements. We show the run-time of our procedure is computationally tractable for a specific auction class by leveraging practically-efficient routines for solving the winner determination problem.

The research covered in this chapter is joint work with Nina Balcan and Tuomas Sandholm.

Chapter 3

Multidimensional mechanism design with side information

Mechanism design is a high-impact branch of economics and computer science that studies the implementation of socially desirable outcomes among strategic self-interested agents. Major real-world use cases include combinatorial auctions (*e.g.*, strategic sourcing, radio spectrum auctions), matching markets (*e.g.*, housing allocation, ridesharing), project fundraisers, and many more. The two most commonly studied objectives in mechanism design are *welfare maximization* and *revenue maximization*. In many settings, welfare maximization, or *efficiency*, is achieved by the classic Vickrey-Clarke-Groves (VCG) mechanism [42, 66, 97]. Revenue maximization is a much more elusive problem that is only understood in very special cases. The seminal work of Myerson [83] characterized the revenue-optimal mechanism for the sale of a single item in the Bayesian setting, but it is not even known how to optimally sell two items to multiple buyers. It is known that welfare and revenue are generally competing objectives and optimizing one can come at the great expense of the other [1, 7, 9, 55, 72].

In this chapter we study how *side information* (or *predictions*) about the agents can help with *bicriteria* optimization of both welfare and revenue. Side information can come from a variety of sources that are abundantly available in practice such as predictions from a machine-learning model trained on historical agent data, advice from domain experts, or even the mechanism designer's own gut instinct. Mechanism design approaches that exploit the proliferation of agent data have in particular witnessed a great deal of success both in theory [16, 19, 79, 82] and in practice [56, 58, 92, 98]. In contrast to the typical Bayesian approach to mechanism design that views side information through the lens of a prior distribution over agents, we adopt a prior-free perspective that makes no assumptions on the correctness, accuracy, or source of the side information. A nascent line of work (that is part of a larger agenda on augmenting algorithms with machine-learned predictions [80]) has begun to examine the challenge of exploiting predictions (of *a priori* unknown quality) when agents are self-interested, but only for fairly specific problem settings [3, 28–30, 61, 102]. We contribute to this line of work with a general side-information-dependent meta-mechanism for a wide swath of multidimensional mechanism design problems that aim for high social welfare and high revenue.

Here we provide a few examples of the forms of side information we consider in various multidimensional mechanism design scenarios. *(1) The owner of a new coffee shop sets prices based on the observation that most customers are willing to pay $9 for a coffee and a croissant, and are willing to pay $5 for either item individually. (2) A real-estate agent believes that a particular buyer values a high-rise condominium with a city view three times more than one on the first floor. Alternately, the seller might know for a fact that the buyer values the first property three times more than the second based on set factors such as value per square foot. (3) A homeowner association is raising funds for the construction of a new swimming pool within a townhome complex. Based on the fact that a particular resident has a family with children, the association estimates that this resident is likely willing to contribute at least $300 if the pool is opened within a block of the resident's house but only $100 if outside a two-block radius.* These are all examples of side information available to the mechanism designer that may or may not be useful or accurate. Our methodology allows us to derive welfare and revenue guarantees under different assumptions on the veracity of the side information. We study two slightly different settings: one in which the side information can be completely bogus, and another in which the side information is constrained to be valid.

Main contributions Our main contribution is a versatile meta-mechanism that integrates side information about agent types with the bicriteria goal of simultaneously optimizing welfare and revenue.

First, we present the weakest-competitor VCG mechanism introduced by Krishna and Perry [73] and prove that it is revenue-optimal among all efficient mechanisms in the prior-free setting (extending their work which was in the Bayesian setting for a fixed known prior).

Next, we present our meta-mechanism for mechanism design with side information. It generalizes the mechanism of Krishna and Perry [73]. We introduce the notion of a weakest-competitor set and a weakest-competitor hull, which are constructions that are crucial to understanding the payments and incentive properties of our meta-mechanism.

We then prove that our meta-mechanism—when carefully instantiated—achieves strong welfare and revenue guarantees that are parameterized by errors in the side information. Our mechanism works by independently expanding the input predictions, where the expansion radius for each prediction is drawn randomly from a logarithmic discretization of the diameter of the ambient type space. Our mechanism achieves the efficient welfare OPT and revenue *at least* $\Omega(\mathrm{OPT}\,/\log H)$ when the side information is highly informative and accurate, where H is an upper bound on any agent's value for any outcome. Its revenue approaches OPT if its initialization parameters are chosen wisely. Its performance decays gradually as the quality of the side information decreases (whereas naïve approaches suffer from huge discontinuous drops in performance). *Prior-free efficient welfare* OPT*, or total social surplus, is the strongest possible benchmark for both welfare and revenue.* Finally, we extend our methods to more general, more expressive side information languages including (1) uncertainty and (2) information involving multiple agents.

Finally, we use our meta-mechanism to derive new results in a setting where each agent's type is determined by a constant number of parameters. Specifically, agent types lie on constant-dimensional subspaces (of the potentially high-dimensional ambient type space) that are known to the mechanism designer. *For example, in the condominium example from the introduction, an agent's value per square foot might completely determine her value for each property.* When each agent's true type is known to lie in a particular k-dimensional subspace of the ambient type space, we show how to use our meta-mechanism to guarantee revenue at least $\Omega(\mathrm{OPT}\,/k(\log H)^k)$ while simultaneously guaranteeing welfare at least $\mathrm{OPT}\,/\log H$.

Traditionally it is known that welfare and revenue are at odds and maximizing one objective comes at the expense of the other. Our results show that side information can help mitigate this difficulty.

The research covered in this chapter is joint work with Nina Balcan and Tuomas Sandholm.

Chapter 4

Learning to cut in integer programming

The incorporation of cutting planes within the branch-and-bound algorithm, known as branch-and-cut, forms the backbone of modern integer programming solvers. These solvers are the foremost method for solving various discrete optimization problems. Choosing cutting planes effectively is a major research topic in the theory and practice of integer programming.

This chapter covers our recent generalization theory that provides provable guarantees for machine learning approaches to cutting plane selection. These guarantees are obtained via a structural analysis of branch-and-cut, in which we pin down conditions for different cutting planes to lead to identical executions of branch-and-cut. First (Section 4.1), we present such guarantees for the canonical family of Chvátal-Gomory cuts. Then (Section 4.2), we show how to extend the underlying ideas behind this theory to derive sample complexity guarantees for tuning all critical components of branch-and-cut simultaneously. This includes node selection, branching/variable selection, and cut selection. Finally (Section 4.3), we extend our theory to the class of Gomory mixed integer cuts, one of the most practically important cutting plane families in state-of-the-art solvers. This requires a deeper structural analysis of the branch-and-cut algorithm that pins down its behavior as a function of general cut parameters. Our structural analysis uncovers fundamental geometric and combinatorial properties of branch-and-cut.

4.1 Learning Chvátal-Gomory cuts

As our first main contribution, we bound the *sample complexity* of learning high-performing cutting planes. Fixing a family of cutting planes, these guarantees bound the number of samples sufficient to ensure that for any sequence of cutting planes from the family, its average performance over the samples is close to its expected performance. We measure performance in terms of the size of the search tree branch-and-cut builds. Our guarantees apply to the parameterized family of

Chvátal-Gomory (CG) cuts [41, 65], one of the most widely-used families of cutting planes.

The overriding challenge is that to provide guarantees, we must analyze how the tree size changes as a function of the cut parameters. This is a sensitive function—slightly shifting the parameters can cause the tree size to shift from constant to exponential in the number of variables. Our key technical insight is that as the parameters vary, the entries of the cut (i.e., the vector $\boldsymbol{\alpha}$ and offset β of the cut $\boldsymbol{\alpha}^T\boldsymbol{x} \leq \beta$) are multivariate polynomials of bounded degree. The number of terms defining the polynomials is exponential in the number of parameters, but we show that the polynomials can be embedded in a space with dimension sublinear in the number of parameters. This insight allows us to better understand tree size as a function of the parameters. We then leverage results by Balcan et al. [21] that show how to use structure exhibited by dual functions (measuring an algorithm's performance, such as its tree size, as a function of its parameters) to derive sample complexity bounds.

Our second main contribution is a sample complexity bound for learning cut-selection policies, which allows branch-and-cut to adaptively select cuts as it solves the input IP. These cut-selection policies assign a number of real-valued scores to a set of cutting planes and then apply the cut that has the maximum weighted sum of scores. Tree size is a volatile function of these weights, though we prove that it is piecewise constant, which allows us to prove our sample complexity bound.

4.2 Learning to tune branch-and-cut and general tree search

Our main contribution in this section is a formalization of a general model of tree search (Algorithm 1) that allows us to improve and generalize prior results on the sample complexity of tuning branch-and-cut. In this model, the algorithm repeatedly chooses a leaf node of the search tree, performs a series of actions (for example, a cutting plane to apply and a constraint to branch on), and adds children to that leaf in the search tree. The algorithm will also fathom nodes when applicable. The node and action selection are governed by *scoring rules*, which assign a real-valued score to each node and possible action. For example, a node-selection scoring rule might equal the objective value of the node's LP relaxation. We focus on general tree search with *path-wise* scoring rules. At a high level, a score of a node or action is path-wise if its value only depends on information contained along the path between the root and that node, as is often the case in branch-and-cut. Many commonly used scoring rules are path-wise including the efficacy [13], objective parallelism [2], directed cutoff distance [59], and integral support [100] scoring rules, all used for cut selection by the leading open-souce solver SCIP [59]; the best-bound scoring rule for node selection; and the linear, product, and most-fractional scoring rules for variable selection using strong branching [2]. We show how this general model of tree search captures a wide array of branch-and-cut components, including node selection, general branching constraint selection, and

Algorithm 1 Tree search

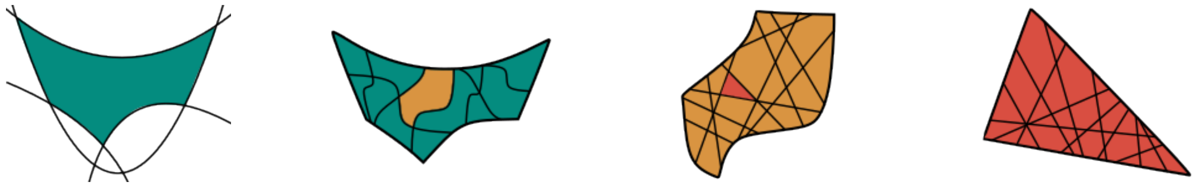
Input: Root node Q, depth limit Δ

Initialize $\mathcal{T} = Q$.
while $\mathcal{T}$ contains an unfathomed leaf **do**
  Select a leaf Q of $\mathcal{T}$ that maximizes $\texttt{nscore}(\mathcal{T}, Q)$.
  if $\texttt{depth}(Q) = \Delta$ or $\texttt{fathom}(\mathcal{T}, Q, \texttt{None})$ **then**
    Fathom Q.
  else
    Select an action $A \in \texttt{actions}(\mathcal{T}, Q)$ that maximizes $\texttt{ascore}(\mathcal{T}, Q, A)$.
    if $\texttt{fathom}(\mathcal{T}, Q, A)$ **then**
      Fathom Q.
    else if $\texttt{children}(\mathcal{T}, Q, A) = \emptyset$ **then**
      Fathom Q.
    else
      Add all nodes in $\texttt{children}(\mathcal{T}, Q, A)$ to $\mathcal{T}$ as children of Q.

cutting plane selection, simultaneously. We also provide experimental evidence that, in the case of cutting plane selection, the data-dependent tuning suggested by our model can lead to dramatic reductions in the number of nodes expanded by branch-and-cut.

Our main structural result shows that for any IP, the tree search parameter space can be partitioned into a finite number of regions such that in any one region, the resulting search tree is fixed. This is in spite of the fact that the branch-and-cut search tree can be an extremely unstable function of its parameters, with minuscule changes leading to exponentially better or worse performance [18, 23]. By analyzing the complexity of this partition, we prove our sample complexity bound. In particular, we relate the complexity of the partition to the *pseudo-dimension* of the set of functions that measure the performance of branch-and-cut as a function of the input IP.

We show that the pseudo-dimension is only polynomial in the depth of the tree (which is, for example, at most the number of variables in the case of binary integer programming). By contrast, we might naïvely expect the pseudo-dimension to grow linearly with the number of arithmetic operations required to compute the branch-and-cut tree (as in Theorem 8.4 in Anthony and Bartlett [8]), which is exponential in the depth of the tree. In fact, our bound is exponentially smaller than the pseudo-dimension bound of prior research by Balcan et al. [23], which grows linearly with the total number of nodes in the tree. Their results apply to any type of scoring rule, path-wise or otherwise. By taking advantage of the path-wise structure, we are able to reason inductively over the depth of the tree, leading to our exponentially improved bound. Our results recover those of Balcan et al. [18], who only studied path-wise scoring rules for single-variable selection for branching. In contrast, we are able to handle many more of the critical components of tree search: node selection, general branching constraint selection, and cutting plane selection.



(a) LP optimum closed form (b) Invariant branching (c) Invariant LP integrality (d) Invariant B&C execution

Figure 4.1: Our branch-and-cut analysis involves successive refinements to our partition of the parameter space.

4.3 Structural analysis of branch-and-cut and the learnability of Gomory mixed integer cuts

In this section we study the learnability of Gomory mixed integer (GMI) cuts. In order to prove our sample complexity bound for GMI cuts, we analyze how the branch-and-cut tree varies as a function of the cut parameters on any IP. We prove that the set of all possible cuts can be partitioned into a finite number of regions such that within any one region, branch-and-cut builds the exact same search tree. Moreover, the boundaries between regions are defined by low-degree polynomials. The simplicity of this function allows us to prove our sample complexity bound. The buildup to this result consists of three main contributions, each of which we believe may be of independent interest:

1. Our first main contribution addresses a fundamental question in linear programming: how does an LP's solution change when new constraints are added? As the constraints vary, the solution will jump from vertex to vertex of the LP polytope. We prove that one can partition the set of all possible constraint vectors into a finite number of regions such that within any one region, the LP's solution has a clean closed form. Moreover, we prove that the boundaries defining this partition have a specific form, defined by degree-2 polynomials.

2. We build on this result in our second main contribution: a novel analysis of how the entire branch-and-cut search tree changes as a function of the cuts added at the root. Our analysis of how the branch-and-cut search tree changes as a function of the cuts added has four steps, illustrated by Figure 4.1:

   (a) First, we use our result about LPs to show that the cut parameter space can be partitioned into regions such that in any one region, the LP optimal solution at any node of the branch-and-cut search tree has a clean closed form, as illustrated in Figure 4.1a.

   (b) We use this result to show that each region can be further partitioned (as illustrated in Figure 4.1b) such that no matter what cut we employ in any one region, all of the branching decisions that branch-and-cut makes are fixed. Intuitively, this is because the branching decisions depend on the LP relaxation, which has a closed-form solution in any one region.

(c) Next, we show that each region from Figure 4.1b can be further partitioned into regions (illustrated in Figure 4.1c) where in any one region, for every node in the branch-and-cut tree, the integrality of that node's LP relaxation is invariant no matter what cut in that region we use.

(d) Finally, we show that each of these regions can be further subdivided into regions (as in Figure 4.1d) where the nodes that branch-and-cut fathoms are fixed, so the tree it builds is fixed.

3. This result allows us to prove sample complexity bounds for learning high-performing cutting planes from the class of GMI cuts, our third main contribution. Our key technical insight is that the GMI cutting plane coefficients can be viewed as a mapping that embeds our polynomial partition from the previous step (Figure 4.1) into the space of GMI cut parameters. We prove that the resulting embedding does not distort the polynomial hypersurfaces too much: the embedded hypersurfaces are still polynomial, with only slightly larger degree.

The research covered in this chapter is joint work with Nina Balcan, Tuomas Sandholm, and Ellen Vitercik.

Chapter 5

New sequence-independent lifting techniques for cutting planes

The previous chapter dealt with *learning to cut*, wherein we were interested in tuning policies to select from a pre-defined pool of cuts. In this chapter, we devise a new technique for generating cuts—via a new approach to *sequence-independent lifting*.

Lifting is a technique for strengthening cutting planes for integer programs by increasing the coefficients of variables that are not in the cut. We study lifting methods for valid cuts of *knapsack polytopes*, which have the form $\mathrm{conv}(P)$ where

$$P = \left\{ \boldsymbol{x} \in \{0,1\}^n : \sum_{j=1}^{n} a_j x_j \le b \right\}$$

for $a_1, \ldots, a_n, b \in \mathbb{N}$ with $0 < a_1, \ldots, a_n \le b$. We interpret P as the set of feasible packings of n items with weights $a_1, \ldots, a_n$ into a knapsack of capacity b. Such *knapsack constraints* arise in binary integer programs from various industrial applications such as resource allocation, auctions, and container packing. They are a very general and expressive modeling tool, as any linear constraint involving binary variables admits an equivalent knapsack constraint by replacing negative-coefficient variables with their complements. A *minimal cover* is a set $C \subseteq \{1, \ldots, n\}$ such that $\sum_{j \in C} a_j > b$ and $\sum_{j \in C \setminus \{i\}} a_j \le b$ for all $i \in C$. That is, the items in C cannot all fit in the knapsack, but any proper subset of C can. The *minimal cover cut* corresponding to C is the inequality

$$\sum_{j \in C} x_j \le |C| - 1,$$

which enforces that the items in C cannot all be selected. A *lifting* of the minimal cover cut is any valid inequality of the form

$$\sum_{j\in C} x_j + \sum_{j\notin C} \alpha_j x_j \le |C| - 1. \tag{5.1}$$

The lifting coefficients α_j are often computed one-by-one—a process called *sequential lifting* that depends on the lifting order. Sequential lifting can be expensive since one must solve an optimization problem for each coefficient. Furthermore, one must reckon with the question of what lifting order to use. To lessen this computational burden, the lifting coefficients can be computed simultaneously. This method is called *sequence-independent lifting* and is the focus of this work. Our contributions include: (i) a generalization of the seminal sequence-independent lifting method developed by Gu et al. [67] and a correction of their proposed generalization; (ii) the first broad conditions under which sequence-independent liftings that are efficiently computable from the underlying cover—via our new techniques—define facets of $\mathrm{conv}(P)$; and (iii) new cover cut generation methods that, together with our new lifting techniques, display promising practical performance in experiments.

Preliminaries on sequence-independent lifting We begin with an overview of the *lifting function* $f : [0, b] \to \mathbb{R}$ associated with a minimal cover C, defined by

$$f(z) = |C| - 1 - \max\left\{\sum_{j\in C} x_j : \sum_{j\in C} a_j x_j \le b - z, x_j \in \{0,1\}\right\}.$$

For $i \notin C$, the value $f(a_i)$ is the maximum possible coefficient α_i such that $\sum_{j\in C} x_j + \alpha_i x_i \le |C| - 1$ is valid for $\mathrm{conv}(P)$. The lifting function has a more convenient closed form due to Balas [12]. First, relabel the items so $C = \{1, \dots, t\}$ and $a_1 \ge \cdots \ge a_t$. Let $\mu_0 = 0$ and for $h = 1, \dots, t$ let $\mu_h = a_1 + \cdots + a_h$. Let $\lambda = \mu_t - b > 0$ be the cover's excess weight. Then,

$$f(z) = \begin{cases} 0 & 0 \le z \le \mu_1 - \lambda \\ h & \mu_h - \lambda < z \le \mu_{h+1} - \lambda. \end{cases}$$

The lifting function has an intuitive interpretation: $f(z)$ is the maximum h such that an item of weight z cannot be brought into in C and fit in the knapsack, even if we are allowed to discard any h items from C. The lifting function f may be used to maximally lift a *single* variable not in the cover. To lift a second variable, a new lifting function must be computed. This (order-dependent) process can be continued to lift all remaining variables, and is known as *sequential lifting*. Conforti et al. [43] and Hojny et al. [68] contain further details.

Superadditivity and sequence-independent lifting. A function $g : D \to \mathbb{R}$ is superadditive if

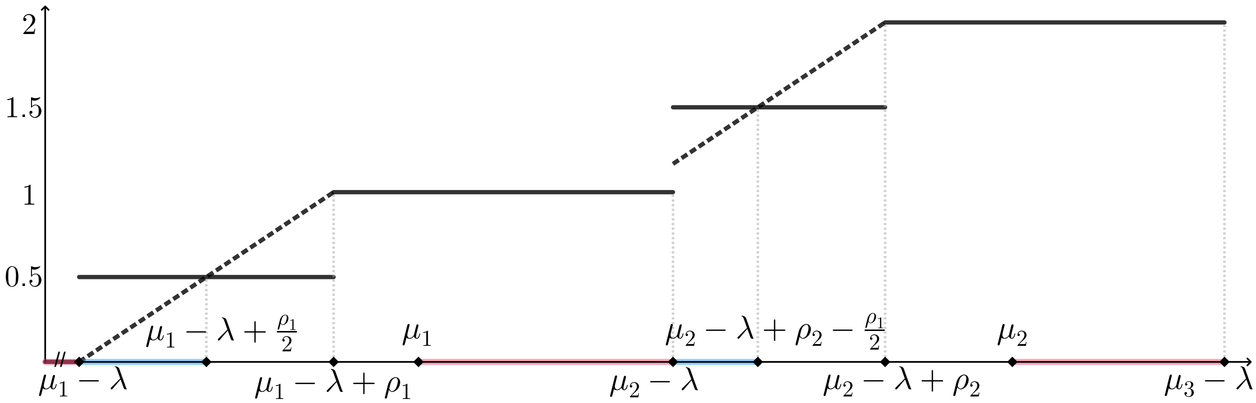


Figure 5.1: The PC lifting function g_0 is the piecewise constant step function depicted by the solid black lines. The GNS lifting function g_{1/ρ_1} is obtained by replacing the solid lines in the intervals S_h with the depicted dashed lines. If all coefficients of variables being lifted lie in the blue and red regions with at least three coefficients in the leftmost blue region, PC lifting is facet-defining and dominates GNS lifting (Theorem 5.0.1).

$g(u+v) \geq g(u) + g(v)$ for all $u, v, u+v \in D$. If $g \leq f$ is superadditive,

$$\sum_{j \in C} x_j + \sum_{j \notin C} g(a_j) x_j \leq |C| - 1$$

is a valid *sequence-independent* lifting for $\operatorname{conv}(P)$. This result is due to Wolsey [101]; Gu et al. [67] generalize to mixed 0-1 integer programs. The lifting function f is generally not superadditive. Gu et al. [67] construct a superadditive function $g \leq f$ as follows. Let $\rho_h = \max\{0, a_{h+1} - (a_1 - \lambda)\}$ be the excess weight of the cover if the heaviest item is replaced with a copy of the $(h+1)$-st heaviest item. For $h \in \{0, \ldots, t-1\}$, let $F_h = (\mu_h - \lambda + \rho_h, \mu_{h+1} - \lambda]$ and for $h \in \{1, \ldots, t-1\}$, let $S_h = (\mu_h - \lambda, \mu_h - \lambda + \rho_h]$. S_h is nonempty if and only if $\rho_h > 0$. For $w : [0, \rho_1] \to [0, 1]$, Gu et al. define

$$g_w(z) = \begin{cases} 0 & z = 0 \\ h & z \in F_h \qquad h = 0, \ldots, t-1 \\ h - w(\mu_h - \lambda + \rho_h - z) & z \in S_h \qquad h = 1, \ldots, t-1. \end{cases}$$

Gu et al. prove that for $w(x) = x/\rho_1$, g_w is superadditive. We call this particular lifting function the *Gu-Nemhauser-Savelsbergh (GNS) lifting function*. Furthermore, g_w is undominated, that is, there is no superadditive g' with $f \geq g' \geq g_w$ and $g'(z') > g_w(z')$ for some $z' \in [0, b]$.

Main contributions First, we prove that under a certain condition, g_w is superadditive for any linear symmetric function w. This generalizes Gu et al.'s result [67] for $w(x) = x/\rho_1$ and furthermore corrects an error in their proposed generalization, which incorrectly claims w can be any symmetric function. Of particular interest is the constant function $w = 1/2$; we call the resulting lifting *piecewise-constant (PC) lifting*. We then give a thorough comparison of PC and GNS lifting. We show that GNS lifting can be arbitrarily worse than PC lifting, and characterize the full domination criteria between the two methods. Next, we provide a broad set of conditions under which PC lifting defines facets of $\mathrm{conv}(P)$:

Theorem 5.0.1. *Let* $C = \{1, \dots, t\}$, $a_1 \geq \cdots \geq a_t$, *be a minimal cover such that* $\mu_1 - \lambda \geq \rho_1 > 0$. *Suppose* $|\{j \notin C : a_j \in S_1\}| \geq 3$ *and for all* $j \notin C$*:*

$$\begin{aligned} a_j \in S_h &\implies \rho_h > \frac{\rho_1}{2} \text{ and } a_j \leq \mu_h - \lambda + \rho_h - \frac{\rho_1}{2}, \\ a_j \in F_h &\implies a_j \geq \mu_h. \end{aligned}$$

Then, the cut

$$\sum_{j \in C} x_j + \sum_{j \notin C} g_0(a_j) x_j \leq |C| - 1,$$

obtained via PC lifting, defines a facet of $\mathrm{conv}(P)$.

To our knowledge, these are the first conditions for facet-defining sequence-independent liftings that are efficiently computable from the underlying cover. Figure 5.1 illustrates the sufficient conditions of the theorem.

We experimentally evaluate our new lifting techniques in conjunction with a number of novel cover cut generation techniques. Our cut generation techniques do not solve expensive NP-hard separation problems (which has been the norm in prior research [71]). Instead, we cheaply generate many candidate cover cuts based on qualitative criteria, lift them, and check for separation only before adding the cut. This approach is effective in experiments with CPLEX.

The research covered in this chapter is joint work with Nina Balcan, Tuomas Sandholm, and Ellen Vitercik.

Chapter 6

Future research directions

I outline some promising directions for future research. I will not be able to finish most of the projects discussed here, and might even end up working on ideas different than those listed here, in the remaining year of my PhD.

6.1 Core-selecting auctions with side information

In this section I describe a computational problem in the realm of combinatorial auctions that in practice is solved via iterative integer programming methods (constraint generation). My proposed research direction here augments the auction design aspect with side information as in Chapter 3; this begets a variety of interesting and challenging questions that require innovations at the confluence of auction design and mathematical programming techniques to ensure scalable solutions.

A prominent criticism of the VCG mechanism is that it can unfairly allocate items to bidders and charge them only modestly when there are other losing bidders who would have been willing to pay significantly more for those items. The *core* is a solution concept from cooperative game theory that avoids this issue by stipulating that no coalition of bidders plus the seller should be able to jointly deviate to a better outcome for every agent in the coalition (including the seller). A core selecting combinatorial auction is one that produces an outcome within the core. Core-selecting auctions are of great practical importance and have been implemented in various high-stakes auctions. For example, variations of the *quadratic* core-selecting rule of Day and Cramton [49] have been used in Canada [69] and the UK [46, 49] to run spectrum auctions, and by the Federal Aviation Administration to auction landing-slot rights in New York City airports [49, 89].

My work on side-information in mechanism design is of particular relevance in the context of core-selecting auctions; there is a nice agenda of information and auction design, and of computation and mathematical programming techniques that will comprise my research focus for the next several months at least. It is well known (and easy to show) that in general there is no incentive compatible core-selecting auction (in the special case that VCG lies in the core, then VCG is the only incentive compatible core-selecting auction). However, with side information about the bidders this is no longer necessarily the case; the *weakest-competitor VCG mechanism* originally introduced by Krishna and Perry [73] and generalized by Balcan et al. [27] offers a new perspective of incentives and the core. I now outline four main components of a research agenda that studies core-selecting combinatorial auctions with side information, and describe some preliminary progress.

New solution concepts. We define a refinement/strengthening of the core that incorporates side-information called the *weakest-competitor core* (borrowing terminology from Balcan et al. [27]). The definition is a natural extension of weakest-competitor VCG prices: the aggregate payment of any group of winners must exceed the externality they impose on the other bidders if that group was replaced by their weakest competitors. There are various practical situations in which weakest-competitor core prices might be a desirable alternative to vanilla core prices. The first component of the research agenda involves fleshing out compelling and practically-relevant uses for side information—both in terms of weakest-competitor VCG prices and in terms of weakest-competitor core prices. So far, one such use is aimed towards mitigating fairness issues: *e.g.,* if a particular bidder is known to have a massive amount of spending power, that should be reflected in prices. Vanilla core pricing does not address issues of fariness/envy that can arise from this kind of knowledge.

Computational aspects. There are two important and novel computational problems we must reckon with. The first is the computation of weakest-competitor VCG prices and the second is the computation of weakest-competitor core prices.

Computing weakest-competitor VCG prices: The LP formulation of Balcan et al. [27] to compute weakest-competitor VCG prices has an exponential number of constraints (one for every feasible allocation), and so we have developed two constraint generation approaches to solve it. The problem of finding the most violated constraint is a winner determination problem in both formulations. However, the difficulty of the constraint generation procedure depends on the form/language of side information as well. I plan to develop some working implementations for practically relevant side information languages soon. (A comparison of the two constraint generation formulations in terms of number of winner determination calls, rate of price convergence, *etc.*, is an interesting starting point for this agenda.)

Finding weakest-competitor core prices: Optimizing over the weakest-competitor core appears to be a computationally tough problem. For example, a practical implementation of the quadratic core-selecting rule already involves the solution of many integer programs via constraint generation. The analogous implementation of a quadratic weakest-competitor-core-selecting rule is certainly no easier, and possibly much harder due to the additional consideration of computing weakest competitors. I believe it will be necessary to design custom search algorithms here.

Empirical evaluation and realistic side-information generators. To empirically test our proposed approaches, realistic distributions/generators of side information will be needed. Here, I plan to build directly atop the Combinatorial Auction Test Suite (CATS) [76] that is a mainstay of auction design research (some other lesser-studied generators worth looking into include the Spectrum Auction Test Suite (SATS) [99] and the distributions for TV advertisement markets in Goetzendorff et al. [62]). Such side-information distributions should have direct ties to information an auctioneer would have access to in real settings: preliminary examples include known appraisal values for items, known relative spending powers of different bidders, known complementarity/substitutability structures on the bids, *etc.*

Side-information languages. How should side information be best expressed? This analogous question in the context of bid elicitation spurred a productive line of research on *bidding languages* (starting with Sandholm [91] and Nisan [85], see also, *e.g.,* Boutilier and Hoos [35]). Can the paradigm of *expressive bidding* [92] be used as an analogy for a new paradigm, *expressive side information*, that might enable significant economic improvements? As a first step, a systematic syntactic characterization of side information is needed. How the expression of information affects the resulting computational tasks is a deep and wide-open research question (owing in part to the fact that the study of weakest-competitor computation has only been initiated in the past year by Balcan et al. [27]). Bichler et al. [33] introduce a new bidding language for spectrum auctins that is also worth looking into.

Some other peripheral ideas worth exploring are listed below:

- If weakest-competitor VCG prices are not in the core, then those prices serve as a new *reference point* that has not been studied before. Here, reference point means some set of prices with respect to which a core point is computed to optimize some objective (*e.g.*, the core prices that minimize Euclidean distance to the reference point). Bünz et al. [37] conduct a computational study of various reference points and various core-selection rules (totaling 366 distinct configurations of core-selecting rules).

- The core constraint generation technique (and variations) [36, 49, 50] starts by initializing VCG prices and iteratively updates prices by augmenting the formulation with the most violated core constraint at every round. One possible way to speed this method up would be to start with a carefully chosen initial (small) set of core constraints. The problem of choosing starting coalitions, or *coalition seeding*, to initialize core constraint generation has not been studied, to the best of my knowledge.

- Multi-unit generalizations: the winner determination problem for the multi-unit generalization of the combinatorial auction setup contains many knapsack constraints. There might be a way to cleverly generate cover cuts (and lift them) in a way that shares/reuses information across iterations in the methods for finding core points.

Some relevant computational work on core-selecting combinatorial auctions: Day and Raghavan [50] propose a core constraint generation approach for core pricing and Day and Cramton [49] solve a quadratic program (also using constraint generation) to compute core prices that are closest to VCG prices in Euclidean distance. Bünz et al. [36] propose different techniques for improving and solving the constraint generation formulation and Niazadeh et al. [84] pose an entirely different algorithmic approach (based on a water filling idea) that is fast but only converges to *some* Pareto-optimal core point (as opposed to the particular Pareto-optimal core point that minimizes, e.g., Euclidean distance to VCG prices). Also, the experiments of Niazadeh et al. [84] are in an advertising setting where the winner determination problem is computationally easy; it admits a polynomial-time dynamic programming algorithm.

6.1.1 Other related directions: iterative combinatorial auctions and sample complexity of learning weakest competitors

I (very) briefly sketch two other related directions for future work. While the agenda detailed above will be my primary mechanism design focus for now, these are two other related project ideas.

Iterative combinatorial auctions and nonlinear pricing

Market clearing in combinatorial auctions is a fundamental problem [34]. It is well known that outside of very restricted classes of bidder valuations, (anonymous) linear item pricing is insufficient to clear the market. Recently, an adaptive iterative combinatorial auction has been proposed [74] that uses *polynomial prices*, increasing the expressivity of the pricing structure as needed (such a procedure could be a candidate to replace the *clock phase* of the combinatorial clock auction [11]). This procedure involves cut generation and column generation, and a deeper dive into the integer programming techniques called for here could be a fruitful research direction. In addition, how can side information guide the design of such procedures?

Sample complexity of (batch and online) weakest-competitor learning

The weakest-competitor VCG mechanism, and more generally the meta-mechanism of Balcan et al. [27], can be viewed as a parametric mechanism where the parameters describe the set of weakest competitors. If agent valuation profiles are drawn i.i.d. from some fixed distribution, can one learn a revenue-maximizing weakest-competitor set from data to use on future agent valuation profiles (this is the setup of sample-based automated mechanism design, as in Balcan et al. [19])? What is the sample complexity of empirical revenue maximization? Can (potentially approximate) empirical revenue maximization be performed in a computationally efficient way? In the online setting, can we develop computationally efficient learning algorithms with low regret?

6.2 New cutting plane generation and selection techniques

In this section I outline some important research questions that arise from my work on cutting planes in integer programming.

6.2.1 Rank-two Gomory cuts and cut generating functions

The standard method of generating Gomory cuts in integer programming solvers for the past several decades works as follows. At a given node of branch-and-cut, the LP relaxation of the subproblem is solved from which an optimal LP tableau is obtained. Each row of the tableau gives rise to a Gomory cut that is guaranteed to separate the LP optimum. While the Gomory rounding procedure is more general than this, using the guidance provided by the tableau guarantees separation. Our idea here is to harness the power of the general rounding procedure directly atop the cuts provided by the tableau. Any choice of multipliers here is guaranteed to separate the LP optimum; this yields an infinite family of cuts that can be optimized over (either exactly or with machine learning). Cornuéjols et al. [45] and Andersen et al. [6] explore a similar idea, but they operate directly on the rows of the simplex tableau (which can be thought of as rank-0 inequalities). Our idea is to operate on the Gomory cuts (rank-1 inequalities) derived from the simplex tableau. Chételat and Lodi [39] also study a similar idea of running optimization algorithms like gradient descent to tune GMI multipliers.

Cut generating functions are a very general framework for deriving cutting planes (the Gomory mixed integer cut can be derived as a special case of this framework). So far, "the jury is still out on the practical usefulness of [cut generating functions]" [31, 43]. Finding fast and cheap ways to exploit this framework to produce new and useful cuts is a compelling research direction. The rank-2 GMI cut idea can likely be cast in this framework as well, and might be indicative of a more general idea here.

6.2.2 Dominance relations for lifted cover inequalities

In our work on sequence-independent lifting [88] we showed that rethinking cover cut generation routines can be very effective in practical settings, leading to dramatically smaller search trees than those built by CPLEX. The norm in all prior research has been to solve (or approximate) NP-hard separation routines to furnish a single most-violated cover cut. In contrast, in our work, we cheaply generate many candidate cover cuts based on qualitative criteria, lift them, and check for separation only before adding the cut. This approach turns out to work well in practice, and is more aligned with the analogous practice for Gomory cuts [14].

Our approach raises a fundamental question: how can one determine the best set of minimal covers to lift? When does one lifted cover cut dominate another, and can this domination be determined from the covers themselves (before lifting)? A methodical answer to this question grounded in theory would likely directly translate to practical gains in the implementations in our work [88].

6.3 Large-scale integer programming

So far, my work on integer programming has focused on problems that are small enough to be fed as input to a solver. A large class of integer programming formulations that arise in industrial applications, however, involve too many variables to be enumerated explicitly (often coming from a combinatorial representation of the variables). Prominent examples include the cutting stock problem [60], various vehicle routing problems [52], aircraft routing and scheduling [51], and so on. In this case, iterative techniques such as *column generation* and *branch-and-price* are used.

6.3.1 Learning column selection policies for column generation and branch-and-price

The column selection step in column generation and branch-and-price is an interesting area for data-driven decision making. In particular, the use of common scoring/selection criteria for cutting plane selection could help for column selection. Usual column generation proceeds by adding, at each iteration, the column with maximum reduced cost, found by solving a pricing problem. Motivated by the dual viewpoint of reduced cost measuring constraint violation in the dual, we can port over commonly used cutting plane selection criteria to this setting. For example, we might add the column corresponding to the dual constraint that maximizes efficacy, or some other criterion. Alternately, we might wish to learn a good weighting of scores as we have done in the previous chapter on learning to cut.

Of course, the resulting pricing problems with modified scores are likely even harder to solve than the original pricing problem. Thus, a more practical method would be to use the pricing problem to obtain a set of columns all with negative reduced cost, and exhaustively search through that set to find the column that maximizes a given score. That column would then be added to the restricted master problem.

Another idea here: Morabit et al. [81] propose a one-step lookahead column selection rule (used in a training procedure for a ML-based column selection rule) that is reminiscent of strong branching. Pseudocost branching serves to approximate strong branching without the expensive computations of strong branching; perhaps there is a similar idea that can be used here. The main challenge in validating these ideas is that we are somewhat restricted to specific problems where the pricing problem can be solved efficiently (and more generally, columns with negative reduced cost can be partially enumerated/ranked as is done by Morabit et al. [81] and Chi et al. [40]).

6.4 Timeline until graduation

I plan to graduate at the end of Spring term, 2025—at the end of my sixth year. Figure 6.1 is a rough timeline listing the things I would like to get done before then. The timeline and list of todo items is non-binding and failure to adhere to any part of it will not delay my graduation.

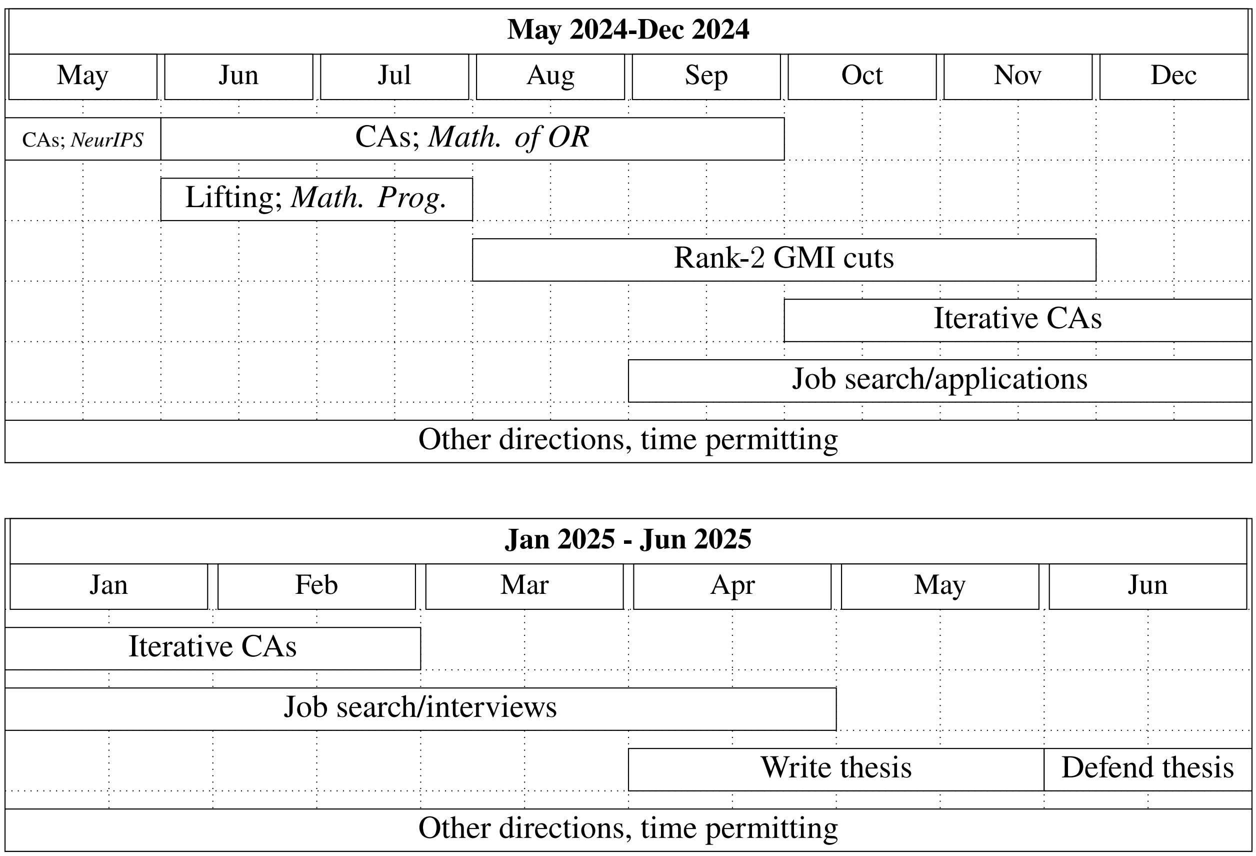


Figure 6.1: Proposed timeline to thesis defense

Bibliography

[1] Vineet Abhishek and Bruce Hajek. Efficiency loss in revenue optimal auctions. In *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010.

[2] Tobias Achterberg. Constraint integer programming. 2007.

[3] Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. Learning-augmented mechanism design: Leveraging predictions for facility location. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC)*, 2022.

[4] Saeed Alaei, Azarakhsh Malekian, and Aravind Srinivasan. On random sampling auctions for digital goods. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 187–196, 2009.

[5] Nicole P Aliloupour. The impact of technology on the entertainment distribution market: The effects of Netflix and Hulu on cable revenue. 2016.

[6] Kent Andersen, Gérard Cornuéjols, and Yanjun Li. Reduce-and-split cuts: Improving the performance of mixed-integer Gomory cuts. *Management Science*, 51(11):1720–1732, 2005.

[7] Elliot Anshelevich, Koushik Kar, and Shreyas Sekar. Pricing to maximize revenue and welfare simultaneously in large markets. In *International Conference on Web and Internet Economics (WINE)*. Springer, 2016.

[8] Martin Anthony and Peter Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 1999.

[9] Lawrence Ausubel and Paul Milgrom. The lovely but lonely Vickrey auction. *Combinatorial auctions*, 2006.

[10] Lawrence M Ausubel and Oleg Baranov. A practical guide to the combinatorial clock auction. *Economic Journal*, 127(605), 2017.

[11] Lawrence M. Ausubel, Peter Cramton, and Paul Milgrom. The clock-proxy auction: A practical combinatorial auction design. In *Combinatorial Auctions*, chapter 5. MIT Press, 2006.

[12] Egon Balas. Facets of the knapsack polytope. *Mathematical programming*, 8(1):146–164, 1975.

[13] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, 1996.

[14] Egon Balas, Sebastian Ceria, Gérard Cornuéjols, and N Natraj. Gomory cuts revisited. *Operations Research Letters*, 19(1):1–9, 1996.

[15] Maria-Florina Balcan. Data-driven algorithm design. In Tim Roughgarden, editor, *Beyond Worst Case Analysis of Algorithms*. Cambridge University Press, 2020.

[16] Maria-Florina Balcan, Avrim Blum, Jason D Hartline, and Yishay Mansour. Mechanism design via machine learning. In *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2005.

[17] Maria-Florina Balcan, Nikhil Devanur, Jason D Hartline, and Kunal Talwar. Random sampling auctions for limited supply. 2007. Technical report, Carnegie Mellon University.

[18] Maria-Florina Balcan, Travis Dick, Tuomas Sandholm, and Ellen Vitercik. Learning to branch. In *International Conference on Machine Learning (ICML)*, 2018.

[19] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *ACM Conference on Economics and Computation (EC)*, 2018.

[20] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Efficient algorithms for learning revenue-maximizing two-part tariffs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[21] Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms? generalization guarantees for data-driven algorithm design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2021.

[22] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Learning within an instance for designing high-revenue combinatorial auctions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[23] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of tree search configuration: Cutting planes and beyond. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[24] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Maximizing revenue under market shrinkage and market uncertainty. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[25] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Improved sample complexity bounds for branch-and-cut. In *International Conference on Principles and Practice of Constraint Programming (CP)*, 2022.

[26] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Structural analysis of branch-and-cut and the learnability of Gomory mixed integer cuts. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[27] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Bicriteria multidimensional mechanism design with side information. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[28] Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. Strategyproof scheduling with predictions. In *Innovations in Theoretical Computer Science (ITCS)*, 2023.

[29] Santiago Balseiro, Christian Kroer, and Rachitesh Kumar. Single-leg revenue management with advice. *arXiv preprint arXiv:2202.10939*, 2022.

[30] Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. Online nash social welfare maximization with predictions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022.

[31] Amitabh Basu and Sriram Sankaranarayanan. Can cut-generating functions be good and efficient? *SIAM Journal on Optimization*, 29(2):1190–1210, 2019.

[32] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[33] Martin Bichler, Paul Milgrom, and Gregor Schwarz. Taming the communication and computation complexity of combinatorial auctions: the fuel bid language. *Management Science*, 69(4):2217–2238, 2023.

[34] Sushil Bikhchandani and Joseph M. Ostroy. The package assignment model. *Journal of Economic Theory*, 107:377–406, 2002.

[35] Craig Boutilier and Holger Hoos. Bidding languages for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1211–1217, Seattle, WA, 2001.

[36] Benedikt Bünz, Sven Seuken, and Benjamin Lubin. A faster core constraint generation algorithm for combinatorial auctions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[37] Benedikt Bünz, Benjamin Lubin, and Sven Seuken. Designing core-selecting payment rules: A computational search approach. *Information Systems Research*, 33(4):1157–1173, 2022.

[38] Zachary Chase and Siddharth Prasad. Learning time dependent choice. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2019.

[39] Didier Chételat and Andrea Lodi. Continuous cutting plane algorithms in integer programming. *Operations Research Letters*, 51(4):439–445, 2023.

[40] Cheng Chi, Amine Aboussalah, Elias Khalil, Juyoung Wang, and Zoha Sherkat-Masoumi. A deep reinforcement learning framework for column generation. *Advances in Neural Information Processing Systems*, 35:9633–9644, 2022.

[41] Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete mathematics*, 4(4):305–337, 1973.

[42] Ed H. Clarke. Multipart pricing of public goods. *Public Choice*, 1971.

[43] Michele Conforti, Gérard Cornuéjols, Giacomo Zambelli, et al. *Integer programming*, volume 271. Springer, 2014.

[44] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 103–110, Edmonton, Canada, 2002.

[45] Gérard Cornuéjols, Yanjun Li, and Dieter Vandenbussche. K-cuts: A variation of Gomory mixed integer cuts from the lp tableau. *INFORMS Journal on Computing*, 15(4):385–396, 2003.

[46] Peter Cramton. Spectrum auction design. *Review of industrial organization*, 42:161–190, 2013.

[47] Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[48] Michael A Cusumano. Amazon and Whole Foods: follow the strategy (and the money). *Communications of the ACM*, 60(10):24–26, 2017.

[49] Robert Day and Peter Cramton. Quadratic core-selecting payment rules for combinatorial auctions. *Operations Research*, 60(3):588–603, 2012.

[50] Robert Day and S. Raghavan. Fair payments for efficient allocations in public sector combinatorial auctions. *Management Science*, 53(9):1389–1406, 2007.

[51] Guy Desaulniers, Jacques Desrosiers, Yvan Dumas, Marius M Solomon, and François Soumis. Daily aircraft routing and scheduling. *Management Science*, 43(6):841–855, 1997.

[52] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.

[53] Nikhil R Devanur and Jason D Hartline. Limited and online supply and the Bayesian foundations of prior-free mechanism design. In *EC*, 2009.

[54] Nikhil R Devanur, Jason D Hartline, and Qiqi Yan. Envy freedom and prior-free mechanism design. *Journal of Economic Theory*, 2015.

[55] Ilias Diakonikolas, Christos Papadimitriou, George Pierrakos, and Yaron Singer. Efficiency-revenue trade-offs in auctions. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2012.

[56] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *International Conference on Machine Learning (ICML)*, pages 1706–1715. PMLR, 2019.

[57] Federico Echenique and Siddharth Prasad. Incentive compatible active learning. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2020.

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

[59] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, et al. The scip optimization suite 7.0. 2020.

[60] Paul C Gilmore and Ralph E Gomory. A linear programming approach to the cutting stock problem—part ii. *Operations research*, 11(6):863–888, 1963.

[61] Vasilis Gkatzelis, Kostas Kollias, Alkmini Sgouritsa, and Xizhi Tan. Improved price of anarchy via predictions. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC)*, 2022.

[62] Andor Goetzendorff, Martin Bichler, Pasha Shabalin, and Robert W Day. Compact bid languages and core pricing in large multi-item auctions. *Management Science*, 61(7):1684–1703, 2015.

[63] Andrew V. Goldberg, Jason D. Hartline, and Andrew Wright. Competitive auctions and digital goods. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.

[64] Maris Goldmanis, Ali Hortaçsu, Chad Syverson, and Önsel Emre. E-commerce and the market structure of retail industries. *The Economic Journal*, 120(545):651–682, 2010.

[65] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275 – 278, 1958.

[66] Theodore Groves. Incentives in teams. *Econometrica*, 1973.

[67] Zonghao Gu, George L Nemhauser, and Martin WP Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4(1):109–129, 2000.

[68] Christopher Hojny, Tristan Gally, Oliver Habeck, Hendrik Lüthen, Frederic Matter, Marc E Pfetsch, and Andreas Schmitt. Knapsack polytopes: a survey. *Annals of Operations Research*, 292:469–517, 2020.

[69] Innovation, Science and Economic Development Canada. Mathematical formulations for winner and price determination for the combinatorial clock auction in the 600 mhz band. Technical report, January 2019.

[70] Philippe Jehiel, Moritz Meyer-Ter-Vehn, and Benny Moldovanu. Mixed bundling auctions. *Journal of Economic Theory*, 2007.

[71] Konstantinos Kaparis and Adam N Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical programming*, 124:69–91, 2010.

[72] Robert Kleinberg and Yang Yuan. On the ratio of revenue to welfare in single-parameter mechanism design. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC)*, 2013.

[73] Vijay Krishna and Motty Perry. Efficient mechanism design. *Available at SSRN 64934*, 1998.

[74] Sébastien Lahaie and Benjamin Lubin. Adaptive-price combinatorial auctions. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 749–750, 2019.

[75] W Arthur Lewis. The two-part tariff. *Economica*, 8(31):249–270, 1941.

[76] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 66–76, Minneapolis, MN, 2000.

[77] Kevin Leyton-Brown, Paul Milgrom, and Ilya Segal. Economics and computer science of a radio spectrum reallocation. *Proceedings of the National Academy of Sciences*, 114(28): 7202–7209, 2017.

[78] Victor J Massad. Understanding the cord-cutters: An adoption/self-efficacy approach. *International Journal on Media Management*, 20(3):216–237, 2018.

[79] Andres Muñoz Medina and Sergei Vassilvitskii. Revenue optimization with approximate bid predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[80] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 2022.

[81] Mouad Morabit, Guy Desaulniers, and Andrea Lodi. Machine-learning–based column selection for column generation. *Transportation Science*, 55(4):815–831, 2021.

[82] Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In *Conference on Learning Theory (COLT)*, 2016.

[83] Roger Myerson. Optimal auction design. *Mathematics of operations research*, 1981.

[84] Rad Niazadeh, Jason Hartline, Nicole Immorlica, Mohammad Reza Khani, and Brendan Lucier. Fast core pricing for rich advertising auctions. *Operations Research*, 70(1):223–240, 2022.

[85] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 1–12, Minneapolis, MN, 2000.

[86] Walter Y Oi. A Disneyland dilemma: Two-part tariffs for a Mickey Mouse monopoly. *The Quarterly Journal of Economics*, 85(1):77–96, 1971.

[87] Siddharth Prasad, Martin Mladenov, and Craig Boutilier. Content prompting: Modeling content provider dynamics to improve user welfare in recommender ecosystems. 2023.

[88] Siddharth Prasad, Ellen Vitercik, Maria-Florina Balcan, and Tuomas Sandholm. New sequence-independent lifting techniques for cutting planes and when they induce facets. 2023.

[89] Stephen Rassenti, Vernon Smith, and Robert Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell J. of Economics*, 13(2):402–417, 1982.

[90] Kevin Roberts. The characterization of implementable social choice rules. In J-J Laffont, editor, *Aggregation and Revelation of Preferences*. 1979.

[91] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, January 2002.

[92] Tuomas Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3):45–45, 2007.

[93] Tuomas Sandholm. Very-large-scale generalized combinatorial multi-attribute auctions: Lessons from conducting $60 billion of sourcing. In Zvika Neeman, Alvin Roth, and Nir Vulkan, editors, *Handbook of Market Design*. Oxford University Press, 2013.

[94] Tuomas Sandholm and Anton Likhodedov. Automated design of revenue-maximizing combinatorial auctions. *Operations Research*, 2015.

[95] Hyoung Ju Song, Jihwan Yeon, and Seoki Lee. Impact of the COVID-19 pandemic: Evidence from the US restaurant industry. *International Journal of Hospitality Management*, 92: 102702, 2021.

[96] Pingzhong Tang and Tuomas Sandholm. Mixed-bundling auctions with reserve prices. In *AAMAS*, 2012.

[97] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 1961.

[98] William E Walsh, David C Parkes, Tuomas Sandholm, and Craig Boutilier. Computing reserve prices and identifying the value distribution in real-world auctions with market disruptions. In *AAAI*, pages 1499–1502, 2008.

[99] Michael Weiss, Benjamin Lubin, and Sven Seuken. Sats: A universal spectrum auction test suite. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 51–59, 2017.

[100] Franz Wesselmann and Uwe Stuhl. Implementing cutting plane management and selection techniques. In *Technical Report*. University of Paderborn, 2012.

[101] Laurence A Wolsey. Valid inequalities and superadditivity for 0–1 integer programs. *Mathematics of Operations Research*, 2(1):66–77, 1977.

[102] Chenyang Xu and Pinyan Lu. Mechanism design with predictions. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.