Mechanism Design and Integer Programming in the Data Age

Siddharth Prasad

CMU-CS-25-125

August 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Maria-Florina Balcan, Co-Chair
Tuomas Sandholm, Co-Chair
Gérard Cornuéjols
Craig Boutilier (Google)
Peter Cramton (University of Maryland)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under award numbers IIS-1901403, IIS-2312342, IIS-1718457, IIS-1617590, CCF-1910321, CCF-1733556, CCF-1535967, and SES-1919453, the Toyota Technological Institute at Chicago under award number T00311601 (Defense Advanced Research Projects Agency under award number HR00112020003), the U.S. Army Contracting Command under award numbers W911NF-22-1-0266, W911NF-17-1-0082, W911NF-20-1-0081, a Vannevar Bush Faculty Fellowship ONR N00014-23-1-2876, and NIH award A240108S001. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Mechanism Design, Market Design, Auctions, Integer Programming, Cutting Planes, Branch-and-Cut, Machine Learning, Algorithm Configuration

Dedicated to Asu Thatha, Indira Pati, Sujatha Athai,
Naani Thatha, and Kamala Pati.

Abstract

This thesis focuses on improving computational and economic aspects of mechanism design, and on improving critical components of integer programming algorithms. Various marketplaces in the world today, from spectrum allocation to strategic sourcing to display advertisements to financial exchanges and more, benefit from carefully engineered rules to govern the efficient exchange of items. Mechanism design offers a principled way to design the rules to such market-based systems in order to implement desired market outcomes subject to strategic self-interested participants. It is the prominent approach to many market design problems and has been deployed in the real world with high impact. On the computational front, integer programming is the go-to method for solving discrete optimization problems that arise in market design applications and beyond.

Within mechanism design, our focus is on the design of better mechanisms that take advantage of any and all information available to the mechanism designer. Our new mechanisms provably generalize and improve the state of the art, and significantly expand the scope of what forms of information can be expressed and used to boost performance. We apply our advances in mechanism design to combinatorial markets where bidders have complex, combinatorial preferences over a rich space of outcomes. Here, our new combinatorial auctions directly improve over existing designs that have been used to conduct high-stakes auctions around the world.

Within integer programming, our focus is on the theory and practice of cutting planes, which are one of the most critical components of integer programming solvers. We invent new cutting planes that deliver strong theoretical and practical performance, and develop a comprehensive generalization theory for data-driven parameter configuration within the branch-and-cut algorithm.

In both areas, we fundamentally advance the classical state of knowledge and introduce new data-driven perspectives, all in support of the thesis that *high performance—e.g., revenue, social welfare, run-time, memory, etc.—in marketplaces can only be fully realized by a synergy of approaches in mechanism design, integer programming, and machine learning.*

Acknowledgments

First and foremost, I would like to thank my PhD advisors Nina Balcan and Tuomas Sandholm. Working with them was an absolute thrill and a deeply enlightening experience. I learned from them the art of carving out the most important research problems to work on in a manner that delivers both elegance and practical impact. The exalted opportunity to complete a PhD under their guidance has immeasurably shaped the way I think about the important things in work and in life.

Thank you to Gérard Cornuéjols, Craig Boutilier, and Peter Cramton for being a fantastic thesis committee. They are giants in their respective fields and it is a great honor to have had them engage with my research and to have learned from them. Special thanks to Craig Boutilier and Martin Mladenov for their enriching mentorship and collaboration during my internship at Google. Thanks to Ellen Vitercik for a very productive and enjoyable collaboration on integer programming.

Caltech was an amazing place to kindle a love for research as an undergraduate and I would not be on my current path without my mentors from that time. Special thanks to Federico Echenique and Adam Wierman for being excellent research advisors and for introducing me to the fascinating area of computational economics. Thanks are also due to Alexander Kechris and Adam Sheffer, who patiently guided me through my first real experiences with open-ended mathematical research.

Thank you to my friends and colleagues, including my officemates and labmates, at CMU and elsewhere for the camaraderie. Special thanks to Matt Kempster for being a close friend for the last ten years.

Finally, and most importantly, thank you to my family. The PhD is just a small fruit of the immense sacrifices made by my parents and grandparents. Thank you to Amma, Appa, and Suhas for being a nonstop source of support, guidance, encouragement, and entertainment. Thank you to Morgan for your constant love and unconditional support. PhD life would have been a slog without you. And lastly, thanks to Rudy for being the greatest cat.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Markets are everywhere. Spectrum allocation, sourcing and procurement, display advertisements, financial exchanges, organ exchanges, ride sharing, supply chain industries, and electronic commerce are just some of the domains that benefit from carefully engineered rules to govern the purchase and/or exchange of items. The need for better *market design* is at the core of some of humanity's biggest problems as well: just one example is the importance of better designs of electricity, water, and carbon emission markets to mitigate the rate and magnitude of climate change while allocating essential resources to those who most need them.

In nearly every domain, the market designer must reckon with tradeoffs between market objectives (like social welfare, gains from trade, or revenue), treatment of market participants in terms of incentives and fairness, judicious use of computational resources, and real world complexities like legacy market structures and regulations. The ability to effectively learn from data in today's world presents significant opportunities to improve the efficiency of market-based systems in all aspects. But, data must be used in a sound manner that respects the interests and incentives of those human participants without whom there would be no market to speak of. The opportunities and challenges of data are further amplified in modern massive-scale online markets such as those present in recommender ecosystems and display advertisements.

Mechanism design is a subfield of economics, computer science, and operations research that offers a principled way to design the rules to market-based systems such as those described above in order to implement desirable outcomes subject to strategic self-interested participants. It is the prominent approach to a large swathe of market design problems, especially those where the exchange of goods can be implemented through monetary transfers.

On the computational front, nearly all success stories of mechanism design in the real world rely and have relied on *integer programming*. In strategic sourcing, the market clearing problem is typically solved via integer programming; CombineNet in the 2000s developed custom branch-and-cut algorithms to take advantage of the rich economic design space [Sandholm et al., 2006, Sandholm, 2007, 2013]. The OneChronos Alternative Trading System uses integer programming to clear point-in-time auctions that prioritize trade execution quality over speed—a departure from how most financial exchanges operate. Core-selecting combinatorial auctions used for spectrum allocation around the world solve large integer programs via iterative methods to compute equitable prices [Day and Cramton, 2012]. Beyond market design, nearly every industry uses integer programming to model and solve the discrete optimization problems that arise in diverse business applications from NFL game scheduling to post-pandemic return-to-office planning to delivery truck routing.

This thesis focuses on improving computational and economic aspects of mechanism design, and on further improving the components of integer programming algorithms that make solvers like Gurobi, CPLEX, Xpress, HiGHS, SCIP, and others a go-to technology for market design applications and beyond.

Within mechanism design, our focus is on the design of better pricing structures that take into account any and all information available to the mechanism designer. The main application is to the design of combinatorial markets where bidders have complex, combinatorial preferences over outcomes. We also investigate a few other novel learning models for mechanism design.

Within integer programming, our focus is on the theory and practice of *cutting planes*, which are one of the most critical components of integer programming solvers. We invent new cutting planes with strong theoretical and practical properties, and develop a comprehensive generalization theory for data-driven cutting plane configuration (which has since turned into an active area of research).

In both areas, we fundamentally advance the classical state of knowledge and introduce new data-driven perspectives, all in support of the thesis that *high performance—e.g., revenue, social welfare, run-time, memory, etc.—in marketplaces can only be fully realized by a synergy of approaches in mechanism design, integer programming, and machine learning.*

Part I. Cutting Plane Theory and Configuration for Integer Programming

The first part of this thesis studies cutting planes, which are some of the most important components of modern integer programming solvers. Cutting planes are constraints added to an integer program—throughout branch-and-cut tree search—that chop off infeasible fractional solutions while preserving the set of integer feasible solutions. They lead to tighter dual bounds and thus allow tree search to terminate earlier. The criticality of cutting planes to practical integer programming for real-world problems cannot be understated. Without cutting planes, integer programming solvers would not be a viable commercial technology [Bixby et al., 1999, Cornuéjols, 2007]. The work covered in this part is joint with Nina Balcan, Tuomas Sandholm, and Ellen Vitercik.

Chapter 2: Primer on Integer Programming, Cutting Planes, and Branch-and-Cut Tree Search This chapter is a primer on the important and relevant aspects of integer programming: it covers integer programming formulations, an overview of cutting planes and polyhedral theory, and a description of the branch-and-cut tree search algorithm that forms the backbone of all successful integer programming solvers.

Chapter 3: New sequence-Independent Lifting Techniques for Cover Inequalities and When They Induce Facets We study a class of cutting planes called *lifted cover inequalities*, which are implemented in every integer programming solver. We invent a new technique for generating lifted cover inequalities, correct an error in a proposed technique from the seminal work on this topic [Gu et al., 2000], characterize when our technique yields *facet-defining cuts* (which are the gold standard for cutting planes since they are in a formal sense the strongest kind of cut), and conduct experiments that validate the practical use of our new class of cuts.

Chapter 4: Learning to Tune Branch-and-Cut Cutting plane selection—the question of what cuts to add during branch-and-cut tree search—is an inexact science that had generally relied upon heuristics and rule of thumb. In this chapter, we develop the first formal guarantees for machine-learning based cut selection by bounding how large the training set should be to ensure that for any cutting plane configuration, its average performance over the training set is close to its expected future performance. En route, we conduct a novel structural analysis of the branch-and-cut algorithm that sheds new geometric and combinatorial insights on (i) general tree search algorithms and (ii) important families of cutting planes such as Gomory cuts (which were invented in the 1950s [Gomory, 1958] and integrated into integer programming solvers in the late 1990s [Balas et al., 1996b, Cornuéjols, 2007]).

Part II. Mechanism Design with Side Information With Applications to Combinatorial Markets

The second part of this thesis is focused on improving mechanism design based on any knowledge the mechanism designer might have about the participating agents. We develop a novel and highly-flexible framework that can integrate nearly anything the mechanism designer knows about agents, and we show how this ability helps obtain mechanisms with better economic properties than existing ones. The work covered in this part is joint with Nina Balcan and Tuomas Sandholm.

Chapter 5: Bicriteria Multidimensional Mechanism Design with Side Information We develop a versatile methodology for multidimensional mechanism design that incorporates side information about agents to generate high welfare and high revenue simultaneously. Side information sources include advice from domain experts, predictions from machine learning models, and even the mechanism designer's gut instinct. We design a tunable mechanism that integrates side information with an improved Vickrey-Clarke-Groves-like mechanism based on weakest types, which are agent types that generate the least welfare. We show that our mechanism, when carefully tuned, generates welfare and revenue competitive with the prior-free total social surplus, and its performance decays gracefully as the side information quality decreases. We consider a number of side information formats including distribution-free predictions, predictions that express uncertainty, agent types constrained to low-dimensional subspaces of the ambient type space, and the traditional setting with known priors over agent types. In each setting we design mechanisms based on weakest types and prove performance guarantees.

Chapter 6: Weakest Bidder Types and New Core-Selecting Combinatorial Auctions Core-selecting combinatorial auctions are popular auction designs that constrain prices to eliminate the incentive for any group of bidders—with the seller—to renegotiate for a better deal. They help overcome the low-revenue issues of classical combinatorial auctions. We introduce a new class of core-selecting combinatorial auctions that leverage bidder information available to the auction designer through type spaces. We show that our designs can overcome the well-known impossibility of incentive-compatible core-selecting combinatorial auctions, and prove that they minimize the sum of bidders' incentives to deviate from truthful bidding. We develop new constraint generation techniques—and build upon existing quadratic programming techniques—to compute core prices, and conduct experiments to evaluate the incentive, revenue, fairness, and computational merits of our new auctions.

Chapter 7: Revenue-Optimal Efficient Mechanism Design with General Type Spaces We derive the revenue-optimal efficient (welfare-maximizing) mechanism in a general multidimensional mechanism design setting when type spaces—that is, the underlying domains from which agents' values come from—can capture arbitrarily complex informational constraints about the agents. Prior work (dating from Green and Laffont [1979], Holmström [1979], Myerson [1981]) has only dealt with connected type spaces, which are not expressive enough to capture many natural kinds of constraints such as disjunctive constraints. We provide two characterizations of the optimal mechanism based on allocations and connected components; both make use of an underlying network flow structure to the mechanism design. Our results significantly generalize and improve the prior state of the art in revenue-optimal efficient mechanism design. They also considerably expand the scope of what forms of agent information can be expressed and used to improve revenue.

Chapter 8: Learning to Generate Artificial Competition We show how an auction designer can inject competition into auctions to boost revenue while striving to maintain efficiency. First, we invent a new auction family that enables the auction designer to specify competition in a precise, expressive, and interpretable way. We then introduce a new model of bidder behavior and individual rationality to understand how bidders act when prices are too competitive. Under our bidder behavior model, we use our new competitive auction class to study revenue-optimal efficient mechanism design under three different knowledge models for the auction designer: knowledge of full bidder value distributions, knowledge of bidder value quantiles, and knowledge of historical bidder valuation data.

Part III. Other Learning Models for Mechanism Design

There has been significant work on a subfield of automated mechanism design where the designer only has samples from the valuation distribution, initiated by Likhodedov and Sandholm [2004], Balcan et al. [2005]. This has been used to design high-revenue auctions, pricing structures, lotteries, and many other mechanisms [Morgenstern and Roughgarden, 2015, 2016, Balcan et al., 2016, 2018d]. The final part of this thesis investigates a few applications of this framework. The work covered in this part is joint with Nina Balcan and Tuomas Sandholm.

Chapter 9: Learning Revenue-Maximizing Two-Part Tariffs A two-part tariff is a pricing scheme that consists of an up-front lump sum fee and a per unit fee. Various products in the real world are sold via a menu, or list, of two-part tariffs—for example gym memberships, cell phone data plans, etc. We develop algorithms for learning high-revenue menus of two-part tariffs from buyer valuation data, in the setting where the mechanism designer has access to samples from the distribution over buyers' values rather than an explicit description thereof. Our algorithms have clear direct uses, and provide the missing piece for the recent generalization theory of two-part tariffs.

Chapter 10: Within-Instance Mechanism Design We present applications of sample-based automated mechanism design to multi-item, multi-bidder revenue maximization (for limited supply) when samples are not available. First, we present a learning-within-an-instance mechanism that generalizes and improves upon prior random-sampling mechanisms for unlimited supply, and prove revenue guarantees for it. Second, we show how to design an auction that is robust to market shrinkage and uncertainty: if there is a fixed population of buyers known to the seller, but only some random (unknown) fraction of them participate, how much revenue can the seller guarantee?

Part I

Cutting Plane Theory and Configuration for Integer Programming

Chapter 2

Primer on Integer Programming, Cutting Planes, and Branch-and-Cut Tree Search

Integer programming is one of the most broadly-applicable methods for optimization, used to formulate problems from operations research (such as routing, scheduling, and pricing), machine learning (such as adversarially-robust learning and clustering), economics and market design (such as auctions, revenue management, and efficient trade/exchange), and many other areas cutting across the sciences. Branch-and-cut (B&C) is the most widely-used algorithm for solving integer programs (IPs), and cutting planes are some of the most important components of B&C. We next review these key concepts.

Integer and linear programs. An *integer program* (IP) is an optimization problem of the form

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{x} \in \mathbb{Z}^n\}, \tag{2.1}$$

where $\boldsymbol{c} \in \mathbb{R}^n$ is the objective vector, $A \in \mathbb{Z}^{m\times n}$ is the constraint matrix, and $\boldsymbol{b} \in \mathbb{Z}^m$ is the constraint vector. The *linear programming (LP) relaxation* is formed by removing the integrality constraints: $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$. We denote the optimal solution to the above IP by $\boldsymbol{x}^*_{\text{IP}}$ and its LP-optimal solution by $\boldsymbol{x}^*_{\text{LP}}$. Let $z^*_{\text{LP}} = \boldsymbol{c}^\top \boldsymbol{x}^*_{\text{LP}}$. Integer programming is NP-complete.

Cutting planes. A *cutting plane* is a constraint $\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta$. Let $\mathcal{P}$ be the feasible region of the LP relaxation of the above IP and $\mathcal{P}_{\mathcal{I}} = \mathcal{P} \cap \mathbb{Z}^n$ be the IP's feasible set. A cut is *valid* if it is satisfied by every integer point in $\mathcal{P}_{\mathcal{I}}$, that is, $\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta$ for all $\boldsymbol{x} \in \mathcal{P}_{\mathcal{I}}$. A valid cut *separates* a point $\boldsymbol{x} \in \mathcal{P} \setminus \mathcal{P}_{\mathcal{I}}$ if $\boldsymbol{\alpha}^\top \boldsymbol{x} > \beta$. We refer to a cut both by its parameters $(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^{n+1}$ and the halfspace $\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta$ in $\mathbb{R}^n$. A valid cut that separates $\boldsymbol{x}^*_{\text{LP}}$ improves the LP estimate used in branch-and-bound, detailed next.

Branch-and-cut. We provide a high-level overview of Branch-and-Cut (B&C) based on the textbook presentation in Conforti et al. [2014]. B&C is the de facto algorithm for solving integer programs. Given an IP, B&C searches the IP's feasible region by building a binary search tree. B&C solves the LP relaxation of the input IP and then adds any number of cutting planes. It stores this information at the tree's root. Let $\boldsymbol{x}^*_{\text{LP}} = (\boldsymbol{x}^*_{\text{LP}}[1], \ldots, \boldsymbol{x}^*_{\text{LP}}[n])$ be the solution to the LP relaxation with the addition of the cutting planes. B&C next uses a *variable selection policy* to choose a variable x_i to branch on. This means that it splits the IP's feasible region in two: one set where $x_i \leq \lfloor \boldsymbol{x}^*_{\text{LP}}[i] \rfloor$ and the other where $x_i \geq \lceil \boldsymbol{x}^*_{\text{LP}}[i] \rceil$. It is critical that the branching on x_i is done in this exact manner. Any other partition of the solution space of the form $x_i \leq k, x_i \geq k+1$, $k \neq \lfloor \boldsymbol{x}^*_{\text{LP}}[i] \rfloor$ would yield a redundant subproblem that will have the same LP solution as its parent. The left child of the root now corresponds to the IP with a feasible region defined by the first subset and the right child likewise corresponds to the second subset. B&C then chooses a leaf using a *node selection policy* and recurses, adding any number of cutting planes, branching on a variable, and so on. B&C *prunes* a node—which means that it will never branch on that node—if 1) the LP relaxation at the node is infeasible, 2) the optimal solution to the LP relaxation is integral, or 3) the optimal solution to the LP relaxation is no better than the best integral solution found thus far. Eventually, B&C will prune every leaf, at which point it has found the globally optimal integral solution. The final tree built by B&C serves as a proof of optimality.

In Chapter 4, we study data-dependent tuning of branch-and-cut hyperparameters. There, we assume there is a bound κ on the size of the tree B&C is allowed to build before termination, as is common in prior research [Hutter et al., 2009, Kleinberg et al., 2017, 2019, Balcan et al., 2018a].

Chapter 3

New Sequence-Independent Lifting Techniques for Cover Inequalities and When They Induce Facets

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 $\operatorname{conv}(P)$ where

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

for $a_1, \ldots, a_n, b \in \mathbb{N}$ with $0 < a_1, \ldots, a_n \leq b$. 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 $\sum_{j \in C \setminus \{i\}} a_j \leq b$ 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 inequality/cut* corresponding to C is the inequality

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

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

$$\sum_{j \in C} x_j + \sum_{j \notin C} \alpha_j x_j \leq |C| - 1. \tag{3.1}$$

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. [2000] 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 $\mathrm{conv}(P)$ (facet-defining cuts are the gold standard for cutting planes since they are in a formal sense the strongest kind of cutting plane); and (iii) new cover inequality generation methods that, together with our new lifting techniques, display promising practical performance in experiments.

Preliminaries on cutting planes and sequence-independent lifting

Facet-defining cuts A cut $\boldsymbol{a}^\top \boldsymbol{x} \leq b$ is a *facet* of $\mathrm{conv}(P)$ if it defines a dimension-$(\dim(P)-1)$ face of $\mathrm{conv}(P)$. Facet-defining cuts are thus the strongest kind of cutting plane and provide the best dual bounds for use within branch-and-cut.

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

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

For $i \notin C$, the value $f(a_i)$ is the maximum possible coefficient α_i such that $\sum_{j\in C} x_j + \alpha_i x_i \leq |C| - 1$ is valid for $\mathrm{conv}(P)$. The lifting function has a more convenient closed form due to Balas [1975]. First, relabel the items so $C = \{1, \dots, t\}$ and $a_1 \geq \cdots \geq a_t$. 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,

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

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. [2014] and Hojny et al. [2020] contain further details.

Superadditivity and sequence-independent lifting A function $g : D \to \mathbb{R}$ is superadditive if $g(u + v) \geq g(u) + g(v)$ 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 $\mathrm{conv}(P)$. This result is due to Wolsey [1977]; Gu et al. [2000] generalize to mixed 0-1 integer programs. The lifting function f is generally not superadditive. Gu et al. [2000] 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 $h \in \{0, \dots, t-1\}$, let $F_h = (\mu_h - \lambda + \rho_h, \mu_{h+1} - \lambda]$ and for $h \in \{1, \dots, t-1\}$, 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 $g_w(z) =$

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

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 $z' \in [0, b]$.

Our contributions

In Section 3.1, we prove that under a certain condition, g_w is superadditive for any linear symmetric function w. This generalizes the Gu et al. [2000] result 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*. In Section 3.1.1 we 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. In Section 3.1.2, we provide a broad set of conditions under which PC lifting defines facets 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.*[1] Furthermore, PC-lifted cuts only have integral and half-integral coefficients making them practically relevant for solvers. In Section 3.1.2 we give an example that shows PC lifting can be significantly stronger than another half-integral sequence-independent lifting procedure due to Letchford and Souli [2019] that is currently implemented in the FICO Xpress solver [Perregard, 2024].

In Section 3.2, we experimentally evaluate our 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 Kaparis and Letchford [2010]). 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.

Related work

Cover cuts and their associated separation routines were first shown to be useful in a branch-and-cut framework by Crowder et al. [1983]. Since then, there has been a large body of work studying various computational aspects, both theoretical and practical, of cover cuts, separation

[1]Balas [1975] proved that lifting coefficients are sometimes fully determined independent of the lifting order, in which case sequential and sequence-independent lifting are the same and yield a facet. When sequence-independent lifting can be non-trivially performed, ours is the first such result.

routines, and lifting. The seminal work of Gu et al. [2000] showed how sequence-independent lifting can be performed efficiently using g_w for $w(x) = x/\rho_1$. Gu et al. [1998] perform a computational study of sequential lifting, and Wolter [2006] presents some computational results on the interaction between the sequence-independent lifting technique of Gu et al. [2000] and different separation techniques. To our knowledge, this is the only computational study of sequence-independent lifting published to date. Our computational study takes a different approach than prior work. Rather than solving separation problems exactly, which involves expensive optimization, we generate large pools of candidate cover cuts, lift them, and check for separation before adding cuts to the formulation. This approach proves to be effective in our experiments. (The separation problem is NP-hard [Klabjan et al., 1998, Gu et al., 1999], but checking violation is a trivial linear time operation. More on separation can be found in Kaparis and Letchford [2010].) Marchand et al. [2002] and Letchford and Souli [2019, 2020] present other sequence-independent lifting functions based on superadditivity.

3.1 New Sequence-Independent Lifting Functions: Structure and Properties

We generalize the result of Gu et al. [2000] and also point out an error in their suggested generalization. Gu et al. claim that if $\mu_1 - \lambda \geq \rho_1$, then g_w is superadditive for any nondecreasing $w : [0, \rho_1] \to [0, 1]$ such that $w(x) + w(\rho_1 - x) = 1$. This claim is incorrect (we provide counterexamples in App. A). We show that this claim is correct when restricted to linear w.

Theorem 3.1.1. *For $k \in [0, 1/\rho_1]$, let $w_k(x) = kx + \frac{1-k\rho_1}{2}$, and let $g_k = g_{w_k}$. If $\mu_1 - \lambda \geq \rho_1$, g_k is superadditive and undominated.*

The GNS lifting function is given by g_{1/ρ_1}. The proof of Theorem 3.1.1 follows the proof that g_{1/ρ_1} is superadditive Gu et al. [2000] with a few key modifications; we defer it to App. A. Of particular interest is the superadditive lifting function g_0, which we refer to as the *piecewise-constant (PC) lifting function*. The condition $\mu_1 - \lambda \geq \rho_1$ is necessary for superadditivity of g_0 (proven in App. A). The following result shows that the lifting obtained via g_k is dominated by the union of the liftings obtained via g_0 (PC lifting) and g_{1/ρ_1} (GNS lifting). Thus, in order to get as close to $\text{conv}(P)$ as possible, it suffices to study these two lifting functions.

Proposition 3.1.2. *Let $k \in (0, 1/\rho_1)$. If $\sum_{j\in C} x_j + \sum_{j\notin C} g_0(a_j)x_j \leq |C| - 1$ and $\sum_{j\in C} x_j + \sum_{j\notin C} g_{1/\rho_1}(a_j)x_j \leq |C| - 1$, then $\sum_{j\in C} x_j + \sum_{j\notin C} g_k(a_j)x_j \leq |C| - 1$.*

Proof. We have $g_k(z) = k\rho_1 g_{1/\rho_1}(z) + (1 - k\rho_1)g_0(z)$ by direct computation, so g_k lifting is a convex combination of GNS and PC lifting. □

Example 3.1.3. Let $C = \{1, 2, 3, 4\}$ and consider a knapsack constraint of the form $16x_1 + 14x_2 + 13x_3 + 9x_4 + \sum_{j\notin C} a_j x_j \leq 44$. C is a minimal cover with $\mu_1 = 16$, $\mu_2 = 30$, $\mu_3 = 43$, $\mu_4 = 52$, $\lambda = 8$, $\rho_1 = 6$, $\rho_2 = 5$, $\rho_3 = 1$, and $\mu_1 - \lambda \geq \rho_1$. Fig. 3.1 depicts g_0 and g_{1/ρ_1} truncated to the domain $[\mu_1 - \lambda, \mu_3 - \lambda] = [8, 35]$.

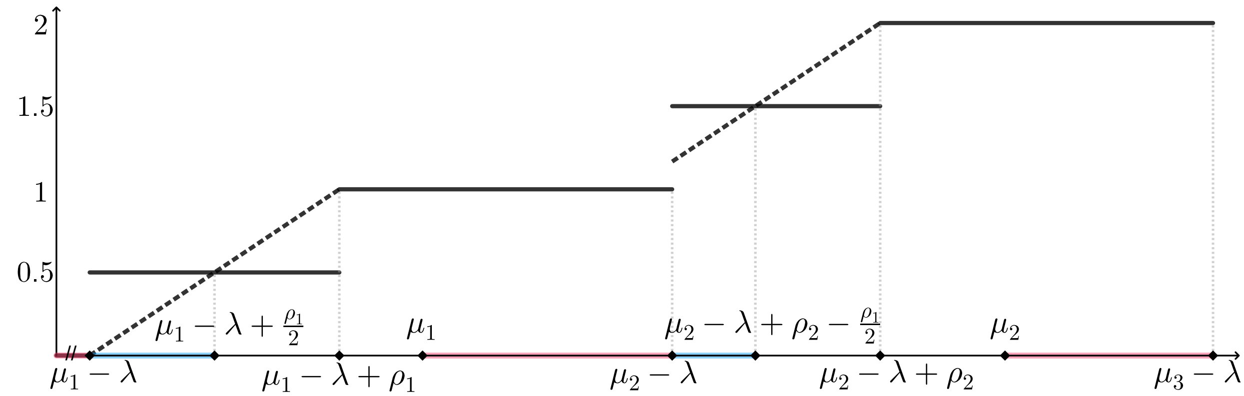


Figure 3.1: The PC lifting function g_0 is the piecewise constant step function depicted by the solid black lines. The GNS lifting function g_{1/ρ_1} is obtained by replacing the solid lines in the intervals S_h with the depicted dashed lines. If all coefficients of variables being lifted lie in the blue and red regions with at least three coefficients in the leftmost blue region, PC lifting is facet-defining and dominates GNS lifting (Theorem 3.1.8).

3.1.1 Comparisons between PC and GNS lifting

The following result shows GNS lifting can be arbitrarily worse than PC lifting.

Proposition 3.1.4. *For any $\varepsilon > 0, t \in \mathbb{N}$ there exists a knapsack constraint with a minimal cover C of size t such that PC lifting yields*

$$\sum_{j \in C} x_j + \sum_{j \notin C} \frac{1}{2} x_j \leq |C| - 1$$

and GNS lifting is dominated by

$$\sum_{j \in C} x_j + \sum_{j \notin C} \varepsilon x_j \leq |C| - 1.$$

The proof is in Appendix A. At a high level, we construct an instance where the length of S_1, which is ρ_1, is large, and consider coefficients that are at the leftmost part S_1. GNS barely lifts such coefficients, while PC yields lifting coefficients of $1/2$. The next proposition fully characterizes the domination criteria between PC and GNS lifting. Its proof is immediate from the plots in Fig. 3.1.

Proposition 3.1.5. *Assume $\mu_1 - \lambda \geq \rho_1$. Furthermore, suppose $\{j \notin C : \exists h \text{ s.t. } a_j \in S_h\} \neq \emptyset$ (else, GNS and PC trivially produce the same cut). If, for all $j \notin C$,*

1. *$a_j \in S_h \implies \rho_h > \frac{\rho_1}{2}$ and $a_j \leq \mu_h - \lambda + \rho_h - \frac{\rho_1}{2}$ with at least one such $a_j \in S_h$ satisfying $a_j < \mu_h - \lambda + \rho_h - \frac{\rho_1}{2}$, PC strictly dominates GNS.*
2. *$a_j \in S_h \implies \rho_h > \frac{\rho_1}{2}$ and $a_j = \mu_h - \lambda + \rho_h - \frac{\rho_1}{2}$, PC and GNS yield the same cut.*
3. *$a_j \in S_h \implies$ ($\rho_h \leq \frac{\rho_1}{2}$) or ($\rho_h > \frac{\rho_1}{2}$ and $a_j > \mu_h - \lambda + \rho_h - \frac{\rho_1}{2}$), GNS strictly dominates PC.*
4. *Otherwise, neither PC nor GNS dominates the other.*

Example 3.1.6. Consider the constraint $16x_1 + 14x_2 + 13x_3 + 9x_4 + a_5x_5 + a_6x_6 + a_7x_7 \leq 44$ with minimal cover $C = \{1, 2, 3, 4\}$.

1. Let $a_5 = 9$, $a_6 = 10$, $a_7 = 23$. GNS yields

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{6}x_5 + \frac{1}{3}x_6 + \frac{4}{3}x_7 \leq 3.$$

   PC yields the dominant cut

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}x_5 + \frac{1}{2}x_6 + \frac{3}{2}x_7 \leq 3.$$

2. Let $a_5 = 11$, $a_6 = 17$, $a_7 = 24$. GNS and PC both yield the cut

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}x_5 + x_6 + \frac{3}{2}x_7 \leq 3.$$

3. Let $a_5 = 12$, $a_6 = 13$, $a_7 = 26$. GNS yields

$$x_1 + x_2 + x_3 + x_4 + \frac{2}{3}x_5 + \frac{5}{6}x_6 + \frac{11}{6}x_7 \leq 3.$$

   PC yields the weaker cut

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}x_5 + \frac{1}{2}x_6 + \frac{3}{2}x_7 \leq 3.$$

4. Let $a_5 = 9$, $a_6 = 13$, $a_7 = 24$. GNS yields

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{6}x_5 + \frac{5}{6}x_6 + \frac{3}{2}x_7 \leq 3.$$

   PC yields

$$x_1 + x_2 + x_3 + x_4 + \frac{1}{2}x_5 + \frac{1}{2}x_6 + \frac{3}{2}x_7 \leq 3.$$

   Neither cut is dominating.

Open question: Gu et al. [1999] and Hunsaker and Tovey [2005] construct knapsack problems where branch-and-cut requires a tree of exponential size, even when *all lifted cover inequalities* (all inequalities of the form (3.1)) are added to the formulation. Do there exist knapsack problems where branch-and-cut requires exponential-size trees when all GNS-lifted cover inequalities are added, but does not when PC-lifted cover inequalities are added (or vice versa)?

3.1.2 Facet defining inequalities from sequence-independent lifting

We now provide a broad set of sufficient conditions under which PC lifting yields facet-defining inequalities. Our result relies on the following complete characterization of the facets of the knapsack polytope obtained from lifting minimal cover cuts, due to Balas and Zemel [Balas and Zemel, 1978, Hojny et al., 2020], which we restate using the notation and terminology of Gu et al. [2000] and Conforti et al. [2014, Section 7.2]. Given a minimal cover C and $j \notin C$, let $h(j)$ be the index such that $\mu_{h(j)} \leq a_j < \mu_{h(j)+1}$.

Theorem 3.1.7 (Balas and Zemel [1978]). *Let C be a minimal cover. Let $J = \{j \notin C : a_j > \mu_{h(j)+1} - \lambda\}$ and let $I = (\{1, \ldots, n\} \setminus C) \setminus J$. Let $\mathcal{Q}(J) = \{Q \subseteq J : \sum_{j \in Q} a_j \leq b, Q \neq \emptyset\}$. The inequality*

$$\sum_{j \in C} x_j + \sum_{j \notin C} \alpha_j x_j \leq |C| - 1$$

is a facet of $\operatorname{conv}(P)$ *if and only if $\alpha_j = h(j) + \delta_j \cdot \mathbf{1}(j \in J)$ where each $\delta_j \in [0, 1]$ and $\boldsymbol{\delta} = (\delta_j)_{j \in J}$ is a vertex of the polyhedron $T =$*

$$\left\{ \boldsymbol{\delta} \in \mathbb{R}^{|J|} : \sum_{j \in Q} \delta_j \leq f\Big(\sum_{j \in Q} a_j\Big) - \sum_{j \in Q} h(j) \; \forall \, Q \in \mathcal{Q}(J) \right\}.$$

The characterization of facets based on T in Theorem 3.1.7 does not translate to a tractable method of deriving facet-defining inequalities, since one would need to enumerate the vertices of T. In fact, Hartvigsen and Zemel [1992] show the question of determining whether or not a cutting plane is a valid lifted cover cut is co-NP-complete. Deciding whether or not a cutting plane is a facet-defining lifted cover cut is in $\mathsf{D^P}$ ($\mathsf{D^P}$ is a complexity class introduced by Papadimitriou and Yannakakis [1982] to characterize the complexity of facet recognition). Critically, these complexity results hold when the underlying cover is given as input.

Our result, to the best of our knowledge, provides *the first broad set of sufficient conditions under which sequence-independent liftings that can be efficiently computed from the underlying minimal cover—via PC lifting—are facet defining.* We now state and prove our result.

Theorem 3.1.8. *Let $C = \{1, \ldots, t\}$, $a_1 \geq \cdots \geq a_t$, 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 $j \notin C$:*

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

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 $\operatorname{conv}(P)$.

Proof. First we show that $J = \cup_{h \geq 1} \{j \notin C : a_j \in S_h\}$ and $I = \cup_{h \geq 0} \{j \notin C : a_j \in F_h\}$. Let $j \notin C$ be such that $a_j \in S_h$. We have $a_j > \mu_h - \lambda > \mu_{h-1}$ (as $\lambda \leq a_i$ for any $i \in C$) and $a_j \leq \mu_h - \lambda + \rho_h - \frac{\rho_1}{2} < \mu_h - \lambda + \rho_h < \mu_h$ (the final inequality follows directly from expanding out μ_h and ρ_h). So $h(j) = h - 1$, and as $a_j > \mu_h - \lambda = \mu_{h(j)+1} - \lambda$, $j \in J$. Next, let $j \notin C$ be such that $a_j \in F_h$. Then, $a_j \leq \mu_{h+1} - \lambda < \mu_{h+1}$, and by assumption $a_j \geq \mu_h$, so $h(j) = h$. Therefore $a_j \leq \mu_{h(j)+1} - \lambda$, and so $j \in I$. The facts that $a_j \in S_h \implies h(j) = h - 1$ and $a_j \in F_h \implies h(j) = h$ will be used repeatedly in the remainder of the proof.

We now determine the constraints defining the polyhedron $T \subset \mathbb{R}^{|J|}$ in Theorem 3.1.7. (For $j \in I$, PC lifting produces a coefficient of $h(j)$, as required by Theorem 3.1.7.) Partition J into two sets $J = J_1 \cup J_{>1}$ where $J_1 = \{j \notin C : a_j \in S_1\}$ and $J_{>1} = \cup_{h>1} \{j \notin C : a_j \in S_h\}$. Each singleton $Q = \{j\} \in \mathcal{Q}(J)$ yields the constraint $\delta_j \leq 1$. Consider now $Q = \{j_1, j_2\} \in \mathcal{Q}(J)$.

We consider two cases, one where $j_1 \in J_1$ and the other where $j_1 \in J_{>1}$. First, let $j_1 \in J_1$. Let h be such that $a_{j_2} \in S_h$. We have $a_{j_1} + a_{j_2} > a_{j_2} > \mu_h - \lambda$, so $f(a_{j_1} + a_{j_2}) \geq h$, and $a_{j_1} + a_{j_2} \leq \mu_1 - \lambda + \frac{\rho_1}{2} + \mu_h - \lambda + \rho_h - \frac{\rho_1}{2} = a_1 - \lambda + \mu_h - \lambda + (a_{h+1} - a_1 + \lambda) = \mu_h + a_{h+1} - \lambda = \mu_{h+1} - \lambda$, so $f(a_{j_1} + a_{j_2}) \leq h$. Therefore $f(a_{j_1} + a_{j_2}) = h$, and we get the constraint $\delta_{j_1} + \delta_{j_2} \leq f(a_{j_1} + a_{j_2}) - h(j_1) - h(j_2) = h - 0 - (h-1) = 1$. Suppose now that $j_1, j_2 \in J_{>1}$ (if $j_2 \in J_1$ that is handled by the first case). Let h_1, h_2 be the integers such that $a_{j_1} \in S_{h_1}$ and $a_{j_2} \in S_{h_2}$. We have $a_{j_1} + a_{j_2} > \mu_{h_1} - \lambda + \mu_{h_2} - \lambda = (a_1 + \cdots + a_{h_1}) + (a_1 + \cdots + a_{h_2}) - 2\lambda > (a_1 + \cdots + a_{h_1 - 1}) + (a_1 + \cdots + a_{h_2}) - \lambda > \mu_{h_1 + h_2 - 1} - \lambda$ so $f(a_{j_1} + a_{j_2}) \geq h_1 + h_2 - 1$, and $f(a_{j_1} + a_{j_2}) - h(j_1) - h(j_2) \geq 1$. So for such pairs, we obtain a constraint $\delta_{j_1} + \delta_{j_2} \leq H(j_1, j_2)$, where $H(j_1, j_2) \geq 1$. Finally, we consider $Q \in \mathcal{Q}(J)$ with $|Q| \geq 3$. For $j \in Q$ let h_j be the integer such that $a_j \in S_{h_j}$. We claim that

$$\sum_{j \in Q} a_j > \mu_{\sum_{j \in Q} h_j - \lfloor |Q|/2 \rfloor} - \lambda.$$

This claim is succinctly proven using the quantities used to prove Theorem 3.1.1 (defined by Gu et al. [2000] to prove superadditivity of g_{1/ρ_1}). To avoid notational clutter, we defer its proof to App. A. The claim implies $f(\sum_{j \in Q} a_j) \geq \sum_{j \in Q} h_j - \lfloor |Q|/2 \rfloor$, so the constraint induced by Q is of the form $\sum_{j \in Q} \delta_j \leq H(Q)$, where

$$H(Q) := f\Big(\sum_{j \in Q} a_j\Big) - \sum_{j \in Q} h(j) \geq \sum_{j \in Q} h_j - \lfloor |Q|/2 \rfloor - \sum_{j \in Q} (h_j - 1) = \lceil |Q|/2 \rceil.$$

We can now write down a complete description of T as

$$\left\{ \boldsymbol{\delta} \in \mathbb{R}^{|J|} : \begin{array}{l} (1)\, \delta_j \leq 1 \,\forall\, j \in J \\ (2)\, \delta_i + \delta_j \leq 1 \,\forall\, (i,j) \in J_1 \times J \\ (3)\, \delta_i + \delta_j \leq H(i,j) \,\forall\, (i,j) \in J_{>1} \times J_{>1} \\ (4)\, \sum_{j \in Q} \delta_j \leq H(Q) \,\forall\, Q \in \mathcal{Q}(J), |Q| \geq 3 \end{array} \right\}$$

where $H(i,j) \geq 1$ for all $(i,j) \in J_{>1} \times J_{>1}$ and $H(Q) \geq \lceil |Q|/2 \rceil$ for all $Q \in \mathcal{Q}(J)$, $|Q| \geq 3$. We argue that $\boldsymbol{\delta} = (1/2, \ldots, 1/2)$ is a vertex of T. Constraints of type (1), (3), and (4) are satisfied, and type (2) constraints are tight. Fix distinct $j_1, j_2, j_3 \in J_1$. The set of $|J|$ type (2) constraints $\{\delta_j + \delta_{j_1} \leq 1 \;\forall\; j \in J \setminus \{j_1\}\} \cup \{\delta_{j_2} + \delta_{j_3} \leq 1\}$ is linearly independent, and hence $\boldsymbol{\delta} = (1/2, \ldots, 1/2)$ is a vertex of T. PC lifting produces precisely the coefficients prescribed by Theorem 3.1.7: $g_0(a_j) = h(j)$ for $j \in I$ and $g_0(a_j) = h(j) + \frac{1}{2}$ for $j \in J$, so we are done. □

Fig. 3.1 illustrates the sufficient conditions of Theorem 3.1.8. While a facet-defining PC lifting can be efficiently obtained given a minimal cover satisfying the sufficient conditions of Theorem 3.1.8, we do not show how to find a minimal cover satisfying these conditions. We conjecture that finding such a cover is NP-hard.

Example 3.1.9. Consider the constraint $16x_1 + 14x_2 + 13x_3 + 9x_4 + 9x_5 + 10x_6 + 11x_7 + 23x_8 \leq 44$ with minimal cover $C = \{1, 2, 3, 4\}$. GNS lifting yields the cut $x_1 + x_2 + x_3 + x_4 + \frac{1}{6}x_5 + \frac{1}{3}x_6 + \frac{1}{2}x_7 + \frac{4}{3}x_8 \leq 3$ and PC lifting yields the strictly dominant facet-defining cut $x_1 + x_2 + x_3 + x_4 + \frac{1}{2}(x_5 + x_6 + x_7) + \frac{3}{2}x_8 \leq 3$.

Other half-integral liftings There exist facet-defining lifted cover inequalities with half-integral coefficients that cannot be obtained via PC lifting. Balas and Zemel [1978] provide an example of a knapsack constraint and minimal cover for which $\boldsymbol{\delta} = (1/2, \ldots, 1/2)$ is a vertex of T, but $\mu_1 - \lambda \geq \rho_1$ *does not hold*. It is an interesting open question to investigate when such facets arise and how to (efficiently) find them. The lifting procedure of Letchford and Souli [2019] also produces half-integral coefficients, but it is unclear when it can yield facets (it produces cuts that in general are incomparable with ours as their lifting function is also undominated). In Example 3.1.9, Letchford and Souli's lifting technique (which is implemented in version 9.5 of the FICO Xpress solver [Perregard, 2024]) yields the cut $x_1 + x_2 + x_3 + x_4 + x_8 \leq 3$, a significantly weaker cut than the facet-defining PC cut. The numerical properties of half-integral cuts make them desirable for implementation within a solver, and the computational overhead of PC lifting (sorting the cover elements) is the same as Letchford and Souli's lifting.

It would be interesting to derive similar conditions under which GNS lifting defines facets. The following sufficient conditions are immediate, but it is likely that a stronger result could be derived. We leave this as an open question.

Proposition 3.1.10. *If for all $j \notin C$, $a_j \in S_h \implies a_j = \mu_h - \lambda + \rho_h$, GNS lifting strictly dominates PC lifting and defines a facet of* $\operatorname{conv}(P)$.

Proof. The condition implies g_{1/ρ_1} coincides with the lifting function f on all a_j, $j \notin C$, which (*e.g.*, Prop. 7.2 in Conforti et al. [2014]) means GNS lifting is facet defining. □

3.2 Experimental Evaluation

We evaluate our new sequence-independent lifting techniques in conjunction with a number of methods for generating the minimal cover cuts that are to be lifted. We describe each component of the experimental setup below.

Cover cut generation Let $\sum_{j=1}^{n} a_j x_j \leq b$ be a knapsack constraint, let $\boldsymbol{x}^{\mathrm{LP}}$ be the LP optimal solution at a given node of the branch-and-cut tree, and let $\mathcal{I} = \{i : x_i^{\mathrm{LP}} > 0\}$. Enumerate $\mathcal{I}$ as $\mathcal{I} = \{1, \ldots, s\}$ (relabeling variables if necessary). We do not include variables not in $\mathcal{I}$ in any of the minimal covers, since these do not contribute to the violation of $\boldsymbol{x}^{\mathrm{LP}}$ (though such variables may be assigned a nonzero coefficient in the lifted cover cut). Next, we present the five cover cut generation techniques that we use in experiments.

Contiguous covers. First, sort the variables in $\mathcal{I}$ in descending order of weight; without loss of generality relabel them so that $a_1 \geq a_2 \geq \cdots \geq a_s$. For each $i \in \{1, \ldots, s\}$, let $j \in \{i+1, \ldots, s\}$ be such that $C = \{i, i+1, \ldots, j\}$ is a minimal cover (if such a j exists). This is the *contiguous* cover starting at i. We generate all such contiguous cover cuts for each i. (Balcan et al. [2022b] introduced these cover cuts, though they did not lift them nor did they restrict to $x_i^{\mathrm{LP}} > 0$.)

Spread covers. As before, sort the variables in $\mathcal{I}$ in descending order of weight; $a_1 \geq \cdots \geq a_s$. For each i, let $j \in \{i+1, \ldots, s\}$ be *maximal* (if such a j exists) such that there exists $k \in \{j+1, \ldots, s\}$ such that $C = \{i\} \cup \{j, \ldots, k\}$ is a minimal cover. Intuitively, i can be thought of as a heavy "head", and the "tail" from j to k is as light as possible. We coin this the *spread* cover with head i. We generate all such spread cover cuts for each i.

Algorithm 1 Lifted cover cut generation at a node of branch-and-cut

Input: IP data $\boldsymbol{c}$, $\boldsymbol{A}$, $\boldsymbol{b}$, LP optimum at current node $\boldsymbol{x}^{\mathrm{LP}}$, per-node cut limit ℓ

Initialize cuts $= \emptyset$.
for each knapsack constraint $\boldsymbol{a}^\top \boldsymbol{x} \le b$ **do**
  for each cut $\in$ `CoverCuts`$(\boldsymbol{a}, b, \boldsymbol{c}, \boldsymbol{x}^{\mathrm{LP}})$ **do**
    for each liftedcut $\in$ `Lift`(cut) **do**
      if liftedcut separates $\boldsymbol{x}^{\mathrm{LP}}$ **then**
        cuts $\leftarrow$ cuts $\cup$ {liftedcut}.
Add the top $\min\{\ell, |\text{cuts}|\}$ cuts in cuts with respect to efficacy.

Heaviest contiguous cover. We define this as the contiguous cover starting at 1.
Default cover. Sort (and relabel) the variables in $\mathcal{I}$ in descending order of LP value so that $x_1^{\mathrm{LP}} \ge \cdots \ge x_s^{\mathrm{LP}}$. Let j be minimal so that $\{1, \dots, j\}$ is a cover. Then, evict items, lightest first, until the cover is minimal. We coin this the *default* cover. These cover cuts are loosely based on the "default" separation routine implemented by Gu et al. [1998]. (Their routine was for sequential lifting and does not directly carry over to our setting.) Wolter [2006] tested similar routines.
Bang-for-buck cover. Sort (and relabel) the variables in $\mathcal{I}$ in descending order of "bang-for-buck" so that $\frac{c_1}{a_1} \ge \cdots \frac{c_s}{a_s}$, where c_i is variable i's objective coefficient. Let j be minimal so that $\{1, \dots, j\}$ is a cover. Then, evict items, lightest first, until the cover is minimal. We coin this the *bang-for-buck* cover.
Example 3.2.1. Consider the knapsack constraint

$$10x_1 + 9x_2 + 8x_3 + 7x_4 + 6x_5 + 6x_6 + 5x_7 + 4x_8 \le 26,$$

suppose the LP optimal solution at the current branch-and-cut node is

$$\boldsymbol{x}^{\mathrm{LP}} = (0.1, 0.8, 0.7, 0.4, 0, 1, 0.2, 0.8),$$

and suppose $\boldsymbol{c} = (5, 7, 9, 1, 2, 6, 6, 5)$. We have $\mathcal{I} = \{1, 2, 3, 4, 6, 7, 8\}$. The contiguous minimal covers are $\{1, 2, 3\}$, $\{2, 3, 4, 6\}$, $\{3, 4, 6, 7, 8\}$. The spread minimal covers are $\{1, 4, 6, 7\}$, $\{2, 4, 6, 7\}$, $\{3, 4, 6, 7, 8\}$. The heaviest contiguous cover is $\{1, 2, 3\}$. The default cover is $\{2, 3, 6, 8\}$. The bang-for-buck cover is $\{2, 3, 6, 7\}$.

Lifting We evaluate three lifting methods. (1) *GNS:* The cover cut is lifted using g_{1/ρ_1}. (2) *PC:* If $\mu_1 - \lambda \ge \rho_1$ the cover cut is lifted using g_0, and otherwise it is lifted using g_{1/ρ_1}. (3) *Smart:* If $\mu_1 - \lambda \ge \rho_1$, two liftings are generated: one with g_0 and one with g_{1/ρ_1}. If either lifting dominates the other, the weaker lifting is discarded. If $\mu_1 - \lambda < \rho_1$, the cover cut is only lifted using g_{1/ρ_1}.

Integration with branch-and-cut Alg. 1 is the pseudocode for adding lifted cover cuts at a node of the branch-and-cut tree and is called once at every node of the tree. It uses the prescribed lifting method `Lift` atop the prescribed cover cut generation method `CoverCuts` for each constraint, and adds the resulting lifted cut to a set of candidate cuts if it separates the current LP optimum $\boldsymbol{x}^{\mathrm{LP}}$. It adds the ℓ deepest cuts among the candidate cuts to the formulation. (The depth or *efficacy* of a cut $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ is the quantity $\frac{\boldsymbol{\alpha}^\top \boldsymbol{x}^{\mathrm{LP}} - \beta}{\|\boldsymbol{\alpha}_2\|}$ and measures the distance between the cut and $\boldsymbol{x}^{\mathrm{LP}}$.) The ℓ cuts are added in a single round at the current node, and no further cuts are generated at that node. Alg. 1 does not solve NP-hard separation problems and instead relies on fast ways of generating candidate cuts through `CoverCuts`, and only adds those that separate $\boldsymbol{x}^{\mathrm{LP}}$.

Experimental results We evaluated our techniques on five problem distributions: two distributions over winner-determination problems in multi-unit combinatorial auctions (*decay-decay* [Sandholm et al., 2002] and *multipaths* from CATS version 1.0 [Leyton-Brown et al., 2000]) and three distributions over multiple knapsack problems (*uncorrelated* and *weakly correlated* benchmark distributions from Fukunaga [2011] and *Chvátal* from Balcan et al. [2022b]). We ran experiments in C++ using the callable library of CPLEX version 20.1 on a 64-core machine, and implemented Alg. 1 with $\ell = 10$ within a cut callback. We generated 100 integer programs from each distribution. We set a 1 hour time limit, allocated 16GB of RAM, and used one thread for each integer program.

We present illustrative results in Fig. 3.2, focusing on settings where PC lifting had significant impact. The full set of experiments are in the full version of the paper.[2] We focus on tree size as the performance measure, but we provide full run-time results in the full paper. Each curve is labeled `setting/lifting/covers` or `setting/CPLEX`. The `setting` label is either empty, `np`, or `d`. Empty corresponds to all cuts, all heuristics, and presolve off. A setting of `np` denotes presolve off and all other settings untouched. A setting of `d` denotes the default settings (this setting is incompatible with our lifting implementations since presolve is incompatible with cut callbacks). If an underscore follows the label, CPLEX's internal cover cut generation is turned off; otherwise it is on. To ensure fair comparisons, in all CPLEX runs not involving our new techniques (those labeled `setting/CPLEX`), we register a dummy cut callback that does nothing. This disables proprietary techniques such as "dynamic search" which do not support callbacks and thus do not support experiments of the type required in this work.

We now describe the performance plots (Fig. 3.2), which display how many instances each method was able to solve using trees smaller than the prescribed size on the x-axis within the 1 hour time limit. *Weakly correlated, contiguous covers (top left):* GNS solves 81 instances, Smart solves 83 instances, and PC solves 88 instances. These all outperform default CPLEX which solves 80 instances and builds trees an order of magnitude larger. *Uncorrelated, contiguous covers (top right):* GNS solves 43 instances, Smart solves 48 instances, and PC solves 55 instances, while default CPLEX solves 78 instances. *Chvátal, heaviest covers (middle left):* PC lifting dramatically outperforms the other methods. Here, all CPLEX parameters are turned off, so we are directly comparing our lifted cover cut implementations with CPLEX's own cover cut generation routines. PC and Smart strictly outperform GNS and CPLEX (GNS is the only one unable to solve all 100 problems), and the largest tree size required by PC is an order of magnitude smaller than any of the other methods. This translates into a run-time improvement as well due to PC lifting. On the auction instances, there is little discernible performance difference

[2] `https://arxiv.org/pdf/2401.13773`

between the lifting methods. *Decay-decay, spread covers (middle right):* our lifting implementations with all other CPLEX parameters off dramatically outperform default CPLEX on problems requiring trees of size $> 10^4$ (though default CPLEX solves all 100 problems while our methods do not.) *Multipaths, spread covers (bottom left):* we once again directly compare our lifted cover cut implementations with CPLEX's internal cover cut generation. Here, spread covers yield over an order of magnitude reduction in tree size relative to CPLEX, while default covers perform extremely poorly. We observed that contiguous and spread covers generally resulted in the best performance (with heaviest contiguous covers on par), and default and bang-for-buck covers performed much worse.

While our techniques often led to significantly smaller trees than CPLEX, this did not translate to significant run-time improvements. However, in most settings our techniques were not too much slower than CPLEX and sometimes they were faster. Full run-time performance plots are in the full paper. A possibility for this is that we did not limit the total number of cuts added throughout the tree, causing the formulation to grow very large. We ran an experiment to investigate the run-time impact of varying the number of cuts allowed (i) at each node (ℓ in Alg. 1) and (ii) throughout the entire search tree. We plot the average run-time (time limit of 1 hour) over the first 10 weakly-correlated instances using `pc/contiguous`, visualized as a heatmap (Fig. 3.2; bottom right). The best settings (limiting the overall number of cuts to 800) yield an average run-time of around 300 seconds and solved all 10 instances. Default CPLEX (`d/CPLEX`) averaged roughly 900 seconds and only solved 9 instances to optimality.

3.3 Conclusions and Future Research

In this chapter we showed that PC lifting can be a useful alternative to GNS lifting. We proved that under some sufficient conditions, PC lifting is facet-defining. To our knowledge, these are the first broad conditions for facet-defining sequence-independent liftings that can be efficiently computed from the underlying cover. We invented new cover cut generation routines, which in conjunction with our new lifting techniques, displayed promising practical performance.

There are a number of important future research directions that stem from our findings. First, a much more extensive experimental evaluation of PC lifting is needed. We have made several simplifying design choices, including (i) adding only one wave of lifted cover cuts at each node; (ii) ranking cuts solely based on efficacy (efficacy is not always the best quality to prioritize Balcan et al. [2022b,c], Turner et al. [2023]); (iii) keeping ℓ constant across nodes while it could be a tuned hyperparameter, or even adjusted dynamically during search. Our experiments show promise even with these fixed choices, but a more thorough suite of tests could find how to best exploit the potential of our new techniques. Another direction here is to use machine learning (which has already been used to tune cutting plane selection policies [Balcan et al., 2021d, Li et al., 2023]) to decide when to use PC or GNS lifting. There might also be additional ways of determining what lifting method to use based on problem structure, and detecting that automatically. Finally, PC lifting possesses strong numerical properties since it always involves half-integral coefficients. That, along with its ability to yield facets, makes PC lifting practically relevant for solvers.

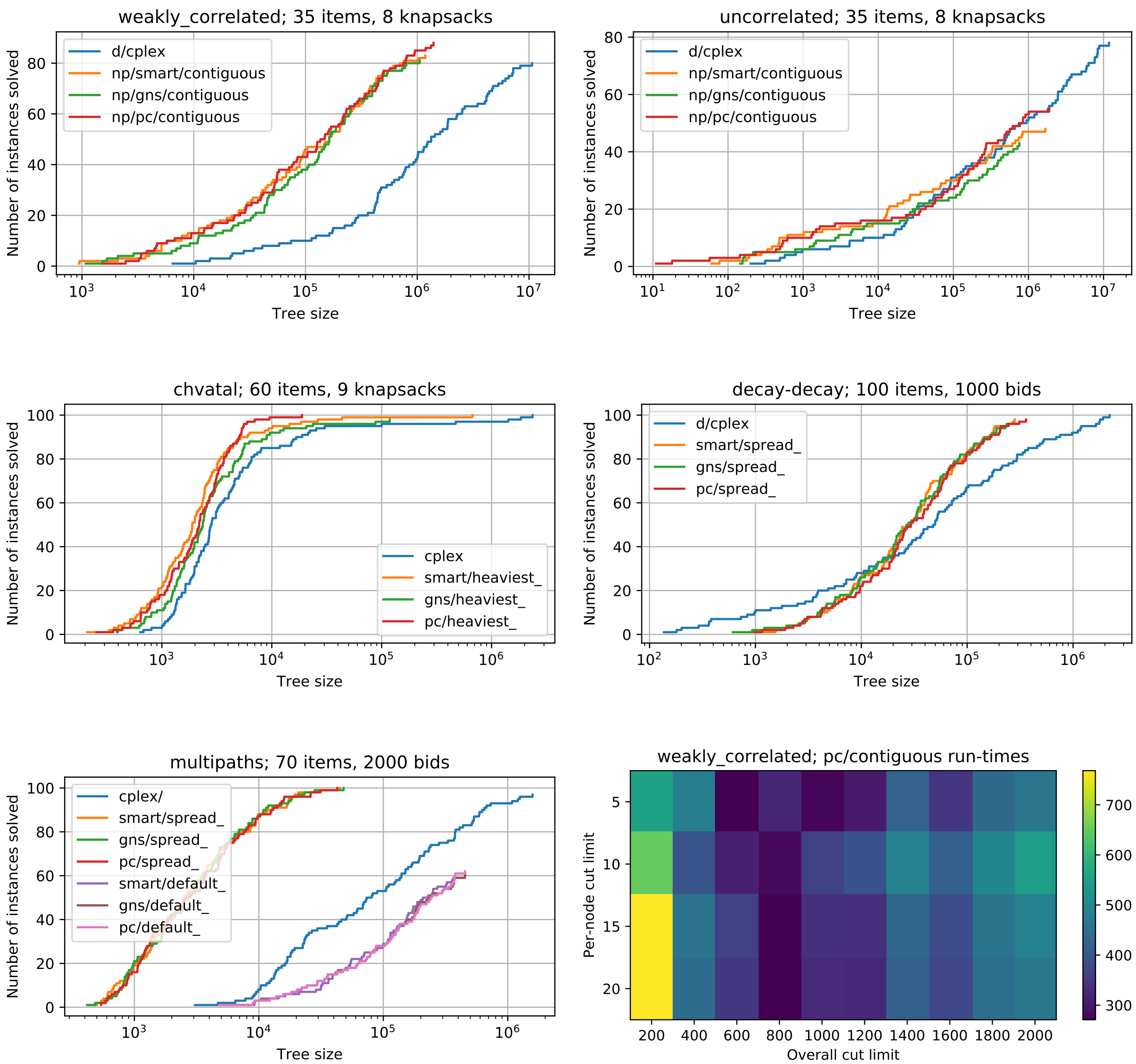


Figure 3.2: Illustrative experiments comparing different lifting methods. The first five plots are performance plots for the five different problem distributions, with various parameter settings. The heatmap illustrates the effect of varying the per-node cut limit and overall cut limit on run-time (avg. over first 10 weakly-correlated instances).

Chapter 4

Learning to Tune Branch-and-Cut

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.

The main components of B&C can be tuned and tweaked in a myriad of ways. The best IP solvers like Gurobi, CPLEX, XPress, HiGHS, SCIP, and others employ an array of heuristics to make decisions at every stage of B&C to reduce the solving time as much as possible, and give the user freedom to tune the multitude of parameters influencing the search through the space of feasible solutions. However, tuning the parameters that control B&C in a principled way is an inexact science with little to no formal mathematical guidelines. A rapidly growing line of work studies machine-learning approaches to speeding up the various aspects of B&C—in particular investigating whether high-performing B&C parameter configurations can be learned from a *training set* of typical IPs from the particular application at hand [Alvarez et al., 2017, Horvitz et al., 2001, Sandholm, 2013, Xu et al., 2008, Hutter et al., 2009, Leyton-Brown et al., 2009, Kadioglu et al., 2010, Xu et al., 2011, Khalil et al., 2016]. Complementing the substantial number of experimental approaches using machine learning for B&C, we develop a generalization theory that aims to provide a rigorous foundation for how well any B&C configuration learned from training IP data will perform on new unseen IPs. In particular, we provide *sample complexity guarantees* that bound how large the training set should be to ensure that *no matter how the parameters are configured* (i.e., using any approach from prior research), the average performance of branch-and-cut over the training set is close to its expected future performance. Sample complexity bounds are important because with too small a training set, learning is impossible: a configuration may have strong average performance over the training set but terrible expected performance on future IPs. If the training set is too small, then no matter how the parameters are tuned, the resulting configuration will not have reliably better performance than any default configuration. State-of-the-art parameter tuning methods have historically come without any provable guarantees, and our results fill in that gap for a wide array of tunable B&C parameters.

This chapter covers our 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 4.3), we present such guarantees for the canonical family of Chvátal-Gomory cuts. Then (Section 4.4), 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 4.5), 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 Theory Background

We first define the formal notion of a *sample complexity bound*. The following definitions are completely general, but we situate them in the context of algorithm configuration. Let $\mathcal{X}$ denote the domain of possible inputs to an algorithm (for example, the set of all IPs with n variables and m constraints). Let $\mathcal{D}$ be any unknown distribution over $\mathcal{X}$ (for example, representing IP instances that model procurement auction clearing problems that a company solves daily). The *sample complexity* of a class of real valued functions $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$ is the minimum number of independent samples required from $\mathcal{D}$ so that with high probability over the samples, the empirical value of f on the samples is a good approximation of the expected value of f over $\mathcal{D}$, uniformly over all $f \in \mathcal{F}$. In the settings we study, $\mathcal{F}$ is a collection of algorithms/algorithm configurations; each $f \in \mathcal{F}$ measures, for example, the tree size built by branch-and-cut when using the parameter tuning prescribed by f.

Formally, given an error parameter ε and confidence parameter δ, the sample complexity $N_{\mathcal{F}}(\varepsilon, \delta)$ is the minimum $N_0 \in \mathbb{N}$ such that for any $N \geq N_0$,

$$\Pr_{x_1,\ldots,x_N \sim \mathcal{D}}\left(\sup_{f \in \mathcal{F}}\left|\frac{1}{N}\sum_{i=1}^{N} f(x_i) - \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f(x)]\right| \leq \varepsilon\right) \geq 1 - \delta$$

for all distributions $\mathcal{D}$ supported on $\mathcal{X}$. Equivalently, one can analyze the error rate $\varepsilon_{\mathcal{F}}(N, \delta)$ between the empirical value of any $f \in \mathcal{F}$ and its true expected value in terms of the number of training samples N and the confidence parameter δ. $N_{\mathcal{F}}(\varepsilon, \delta)$ is the number of samples required to achieve a prescribed error bound ε, while $\varepsilon_{\mathcal{F}}(N, \delta)$ provides an error bound for any number N of samples at hand. To prove bounds on $N_{\mathcal{F}}(\varepsilon, \delta)$ and $\varepsilon_{\mathcal{F}}(N, \delta)$, we rely on the notion of *pseudo-dimension* [Pollard, 1984], a well-known measure of a function class's *intrinsic complexity*. The pseudo-dimension of a collection of real-valued functions $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$, denoted by $\mathsf{Pdim}(\mathcal{F})$, is the largest positive integer N such that there exist N inputs $x_1, \ldots, x_N \in \mathcal{X}$ and N thresholds $r_1, \ldots, r_N \in \mathbb{R}$ such that

$$|\{(\text{sign}(f(x_1) - r_1), \ldots, \text{sign}(f(x_N) - r_N)) : f \in \mathcal{F}\}| = 2^N.$$

Pseudo-dimension is related to sample complexity via the following *uniform convergence* theorem [Pollard, 1984, Anthony and Bartlett, 1999]. If functions in $\mathcal{F}$ have bounded range $[-H, H]$, then

$$N_{\mathcal{F}}(\varepsilon, \delta) = O\left(\frac{H^2}{\varepsilon^2}\left(\mathsf{Pdim}(\mathcal{F}) + \ln(1/\delta)\right)\right) \text{ and } \varepsilon_{\mathcal{F}}(N, \delta) = O\left(H\sqrt{\frac{\mathsf{Pdim}(\mathcal{F}) + \ln(1/\delta)}{N}}\right).$$

When the range of $\mathcal{F}$ is $\{0, 1\}$, the pseudo-dimension is equivalent to the *VC dimension* [Vapnik and Chervonenkis, 1971].

4.2 Related Work

A growing body of research studies how machine learning can be used to speed up the time it takes to solve integer programs, primarily from an empirical perspective, whereas we study this problem from a theoretical perspective. This line of research has included general parameter tuning procedures [Hutter et al., 2009, Kadioglu et al., 2010, Hutter et al., 2011, Sandholm, 2013], which are not restricted to any one aspect of B&C. Researchers have also honed in on specific aspects of tree search and worked towards improving those using machine learning. These include variable selection [Khalil et al., 2016, Alvarez et al., 2017, Di Liberto et al., 2016, Balcan et al., 2018a, Gasse et al., 2019, Gupta et al., 2020], general branching constraint selection [Yang et al., 2020], cut selection [Sandholm, 2013, Tang et al., 2020, Huang et al., 2021], node selection [Sabharwal et al., 2012, He et al., 2014], and heuristic scheduling [Khalil et al., 2017, Chmiela et al., 2021]. Machine learning approaches to large neighborhood search have also been used to speed up solver runtimes [Song et al., 2020].

This chapter contributes to a growing research area dubbed *data-driven algorithm design* that provides sample complexity guarantees for algorithm configuration, often by using structure exhibited by the algorithm's performance as a function of its parameters [Gupta and Roughgarden, 2017, Balcan et al., 2017, 2018a, 2021a, Balcan, 2020]. This line of research has studied algorithms for clustering [Balcan et al., 2017, 2018c, 2020a], decision tree learning [Balcan and Sharma, 2024], semi-supervised learning [Balcan and Sharma, 2021], computational biology [Balcan et al., 2021a], integer programming [Balcan et al., 2018a], AI search and planning [Sakaue and Oki, 2022], topics in scientific computing such as linear system solving [Khodak et al., 2024] and numerical linear algebra [Bartlett et al., 2022], and various other computational problems. The main contribution of this chapter is to provide a sharp yet general analysis of the performance of branch-and-cut tree search as a function of its parameters, with a focus on cutting planes.

A related line of research provides algorithm configuration procedures with provable guarantees that are agnostic to the specific algorithm that is being configured [Kleinberg et al., 2017, Weisz et al., 2018] and are particularly well-suited for algorithms with a finite number of possible configurations (though they can be applied to algorithms with infinite parameter spaces by randomly sampling a finite set of configurations).

4.2.1 Work subsequent to our initial publications

The three papers [Balcan et al., 2021d, 2022b,c] covered in this chapter helped kindle an active research effort on understanding machine learning for cutting plane configuration, both theoretical and applied. On the applied front, learning to cut has become a significant research agenda. Areas of study include imitation learning [Paulus et al., 2022], reinforcement learning [Mana et al., 2024], learning to stop cut generation [Ling et al., 2024], learning to remove cuts [Puigdemont et al., 2024], learning to toggle cutting plane selectors [Li et al., 2023, Lawless et al., 2024], graph neural networks [Deza et al., 2025], and learning for better cut separation [Guaje et al., 2024, Becu et al., 2024]. The research in this chapter provides a theoretical backing for all these works. The theory developed in this chapter has also been explicitly applied and extended to understand sample complexity of other tunable aspects of integer programming. For example, Cheng and Basu [2024] show how to learn cut generating functions, which represent a general abstraction for deriving valid cuts. Cheng et al. [2024], Cheng and Basu [2025] adapt and generalize some of the results in Balcan et al. [2021d, 2022b] to cut-selection policies with piecewise polynomial structure—specifically those parameterized by the weights of a neural network—thus moving beyond the policies we study that are parameterized by mixture weights of existing cut quality metrics (though piecewise polynomial structure is prevalent throughout our structural analyses). Finally, Deza and Khalil [2023] provides a survey of machine learning approaches for cutting planes (though only covering pre-2023 efforts).

Beyond integer programming, Sakaue and Oki [2024] study data-driven acceleration of LP solvers. They note the thematic similarity of their analyses to those in Balcan et al. [2022c] (Section 4.5). Finally, the tools from data-driven algorithm design used to prove our bounds, particularly in Section 4.5 [Balcan et al., 2022c], have since been significantly extended as well. Balcan et al. [2025a,b] show how to establish sample complexity bounds for algorithm configuration settings that can have more complex structure than what we encounter here.

4.3 Sample Complexity of Learning Chvátal-Gomory Cuts

As our first main contribution, we bound the *sample complexity* of learning good cutting planes that lead to small branch-and-cut trees. 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 in this section apply to the parameterized family of *Chvátal-Gomory (CG) cuts* [Chvátal, 1973, Gomory, 1958], one of the most canonical 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 $\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta$) 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. [2021a] 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 CG 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.

Chvátal-Gomory cuts The family of *Chvátal-Gomory (CG) cuts* [Chvátal, 1973, Gomory, 1958] are parameterized by vectors $\boldsymbol{u} \in \mathbb{R}^m$. The CG cut defined by $\boldsymbol{u} \in \mathbb{R}^m$ is the hyperplane $\lfloor \boldsymbol{u}^\top A \rfloor \boldsymbol{x} \leq \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor$, which is guaranteed to be valid. We restrict to $\boldsymbol{u} \in [0, 1)^m$. This is without loss of generality, since the facets of $\mathcal{P} \cap \{\boldsymbol{x} \in \mathbb{R}^n : \lfloor \boldsymbol{u}^\top A \rfloor \boldsymbol{x} \leq \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor \, \forall \boldsymbol{u} \in \mathbb{R}^m\}$ can be described by the finitely many $\boldsymbol{u} \in [0, 1)^m$ such that $\boldsymbol{u}^\top A \in \mathbb{Z}^n$ (Lemma 5.13 of Conforti et al. [2014]).

4.3.1 Learning a single CG cut

We bound the pseudo-dimension of the set of functions $f_{\boldsymbol{u}}$ for $\boldsymbol{u} \in [0, 1]^m$, where $f_{\boldsymbol{u}}(\boldsymbol{c}, A, \boldsymbol{b})$ is the size of the tree B&C builds when it applies the CG cut defined by $\boldsymbol{u}$ at the root. To do so, we take advantage of structure exhibited by the class of *dual* functions, each of which is defined by a fixed IP $(\boldsymbol{c}, A, \boldsymbol{b})$ and measures tree size as a function of the parameters $\boldsymbol{u}$. In other words, each dual function $f^*_{\boldsymbol{c},A,\boldsymbol{b}} : [0, 1]^m \to \mathbb{R}$ is defined as $f^*_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{u}) = f_{\boldsymbol{u}}(\boldsymbol{c}, A, \boldsymbol{b})$. Our main result in this section is a proof that the dual functions are well-structured (Lemma 4.3.2), which then allows us to apply a result by Balcan et al. [2021a] to bound $\mathsf{Pdim}(\{f_{\boldsymbol{u}} : \boldsymbol{u} \in [0, 1]^m\})$ (Theorem 4.3.3). Proving that the dual functions are well-structured is challenging because they are volatile: slightly perturbing $\boldsymbol{u}$ can cause the tree size to shift from constant to exponential in n, as we prove in the following theorem.

Theorem 4.3.1. *For any integer n, there exists an integer program $(\boldsymbol{c}, A, \boldsymbol{b})$ with two constraints and n variables such that if $\frac{1}{2} \leq u[1] - u[2] < \frac{n+1}{2n}$, then applying the CG cut defined by $\boldsymbol{u}$ at the root causes B&C to terminate immediately. Meanwhile, if $\frac{n+1}{2n} \leq u[1] - u[2] < 1$, then applying the CG cut defined by $\boldsymbol{u}$ at the root causes B&C to build a tree of size at least $2^{(n-1)/2}$.*

Proof. Without loss of generality, we assume that n is odd. We define the integer program

$$\begin{array}{ll} \text{maximize} & 0 \\ \text{subject to} & 2x[1] + \cdots + 2x[n] = n \\ & \boldsymbol{x} \in \{0,1\}^n, \end{array} \tag{4.1}$$

which is infeasible because n is odd. Jeroslow [1974] proved that without the use of cutting planes or heuristics, B&C will build a tree of size $2^{(n-1)/2}$ before it terminates. Rewriting the equality constraint as $2x[1] + \cdots + 2x[n] \leq n$ and $-2\,(x[1] + \cdots + x[n]) \leq -n$, a CG cut defined by the vector $\boldsymbol{u} \in \mathbb{R}^2_{\geq 0}$ will have the form $\lfloor 2(u[1] - u[2]) \rfloor (x[1] + \cdots + x[n]) \leq \lfloor n\,(u[1] - u[2]) \rfloor$.

Suppose that $\frac{1}{2} \leq u[1] - u[2] < \frac{n+1}{2n}$. On the left-hand-side of the constraint, $\lfloor 2(u[1] - u[2]) \rfloor = 1$. On the right-hand-side of the constraint, $n\,(u[1] - u[2]) < \frac{n+1}{2}$. Since n is odd, $\frac{n+1}{2}$ is an integer, which means that $\lfloor n\,(u[1] - u[2]) \rfloor \leq \frac{n-1}{2}$. Therefore, the CG cut defined by $\boldsymbol{u}$ satisfies the inequality $x[1] + \cdots + x[n] \leq \frac{n-1}{2}$. The intersection of this halfspace with the feasible region of the original integer program (Equation (4.9)) is empty, so applying this CG cut at the root will cause B&C to terminate immediately.

Meanwhile, suppose that $\frac{n+1}{2n} \leq u[1] - u[2] < 1$. Then it is still the case that $\lfloor 2(u[1] - u[2]) \rfloor = 1$. Also, $n\,(u[1] - u[2]) \geq \frac{n+1}{2}$, which means that $\lfloor n\,(u[1] - u[2]) \rfloor \geq \frac{n+1}{2}$. Therefore, the CG cut defined by $\boldsymbol{u}$ dominates the inequality $x[1] + \cdots + x[n] \leq \frac{n+1}{2}$. The intersection of this halfspace with the feasible region of the original integer program is equal to the integer program's feasible region, so by Jeroslow's result, applying this CG cut at the root will cause B&C to build a tree of size at least $2^{(n-1)/2}$ before it terminates. □

This theorem shows that the dual tree-size functions can be extremely sensitive to perturbations in the CG cut parameters. However, we are able to prove that the dual functions are piecewise-constant.

Lemma 4.3.2. *For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are $O(\|A\|_{1,1} + \|\boldsymbol{b}\|_1 + n)$ hyperplanes that partition $[0,1]^m$ into regions where in any one region R, the dual function $f^*_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{u})$ is constant for all $\boldsymbol{u} \in R$.*

Proof. Let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^m$ be the columns of A. Let $A_i = \|\boldsymbol{a}_i\|_1$ and $B = \|\boldsymbol{b}\|_1$, so for any $\boldsymbol{u} \in [0,1]^m$, $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor \in [-A_i, A_i]$ and $\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor \in [-B, B]$. For each integer $k_i \in [-A_i, A_i]$, we have $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor = k_i \iff k_i \leq \boldsymbol{u}^\top \boldsymbol{a}_i < k_i + 1$. There are $\sum_{i=1}^n 2A_i + 1 = O(\|A\|_{1,1} + n)$ such halfspaces, plus an additional $2B + 1$ halfspaces of the form $k_{n+1} \leq \boldsymbol{u}^\top \boldsymbol{b} < k_{n+1} + 1$ for each $k_{n+1} \in \{-B, \ldots, B\}$. In any region R defined by the intersection of these halfspaces, the vector $(\lfloor \boldsymbol{u}^\top \boldsymbol{a}_1 \rfloor, \ldots, \lfloor \boldsymbol{u}^\top \boldsymbol{a}_n \rfloor, \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor)$ is constant for all $\boldsymbol{u} \in R$, and thus so is the resulting cut. □

Combined with the main result of Balcan et al. [2021a], this lemma implies the following bound.

Theorem 4.3.3. *Let $\mathcal{F}_{\alpha,\beta}$ denote the set of all functions $f_{\boldsymbol{u}}$ for $\boldsymbol{u} \in [0,1]^m$ defined on the domain of IPs $(\boldsymbol{c}, A, \boldsymbol{b})$ with $\|A\|_{1,1} \leq \alpha$ and $\|\boldsymbol{b}\|_1 \leq \beta$. Then,* $\mathsf{Pdim}(\mathcal{F}_{\alpha,\beta}) = O(m \log(m(\alpha + \beta + n)))$.

This theorem implies that $\widetilde{O}(\kappa^2 m/\varepsilon^2)$ samples are sufficient to ensure that with high probability, for every CG cut, the average size of the tree B&C builds upon applying the cutting plane is within ϵ of the expected size of the tree it builds (the $\widetilde{O}$ notation suppresses logarithmic terms).

4.3.2 Learning a sequence of CG cuts

We now determine the sample complexity of making w sequential CG cuts at the root. The first cut is defined by m parameters $\boldsymbol{u}_1 \in [0,1]^m$ for each of the m constraints. Its application leads to the addition of the row $\lfloor \boldsymbol{u}_1^\top A \rfloor \boldsymbol{x} \leq \lfloor \boldsymbol{u}_1^\top \boldsymbol{b} \rfloor$ to the constraint matrix. The next cut is then be defined by $m+1$ parameters $\boldsymbol{u}_2 \in [0,1]^{m+1}$ since there are now $m+1$ constraints. Continuing in this fashion, the wth cut is be defined by $m + w - 1$ parameters $\boldsymbol{u}_w \in [0,1]^{m+w-1}$. Let $f_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_w}(\boldsymbol{c}, A, \boldsymbol{b})$ be the size of the tree B&C builds when it applies the CG cut defined by $\boldsymbol{u}_1$, then applies the CG cut defined by $\boldsymbol{u}_2$ to the new IP, and so on, all at the root of the search tree.

As in Section 4.3.1, we bound the pseudo-dimension of the functions $f_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_w}$ by analyzing the structure of the dual functions $f^*_{\boldsymbol{c},A,\boldsymbol{b}}$, which measure tree size as a function of the parameters $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w$. Specifically, $f^*_{\boldsymbol{c},A,\boldsymbol{b}} : [0,1]^m \times \cdots \times [0,1]^{m+w-1} \to \mathbb{R}$, where $f^*_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w) = f_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_w}(\boldsymbol{c}, A, \boldsymbol{b})$. The analysis in this section is more complex because the s^{th} cut (with $s \in \{2, \ldots, W\}$) depends not only on the parameters $\boldsymbol{u}_s$ but also on $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{s-1}$. We prove that the dual functions are again piecewise-constant, but in this case, the boundaries between pieces are defined by multivariate polynomials rather than hyperplanes.

Lemma 4.3.4. *For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are $O(w2^w \|A\|_{1,1} + 2^w \|\boldsymbol{b}\|_1 + nw)$ multivariate polynomials in $\leq w^2 + mw$ variables of degree $\leq w$ that partition $[0,1]^m \times \cdots \times [0,1]^{m+w-1}$ into regions where in any one region R, $f^*_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w)$ is constant for all $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w) \in R$.*

Proof. Let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^m$ be the columns of A. For $\boldsymbol{u}_1 \in [0,1]^m, \ldots, \boldsymbol{u}_w \in [0,1]^{m+w-1}$, define $\widetilde{\boldsymbol{a}}_i^1 \in [0,1]^m, \ldots, \widetilde{\boldsymbol{a}}_i^w \in [0,1]^{m+w-1}$ for each $i = 1, \ldots, n$ such that $\widetilde{\boldsymbol{a}}_i^s$ is the ith column of the constraint matrix after applying cuts $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{s-1}$. More precisely, $\widetilde{\boldsymbol{a}}_i^1 \in [0,1]^m, \ldots, \widetilde{\boldsymbol{a}}_i^w \in [0,1]^{m+w-1}$ are defined recursively as

$$\widetilde{\boldsymbol{a}}_i^1 = \boldsymbol{a}_i$$
$$\widetilde{\boldsymbol{a}}_i^s = \begin{bmatrix} \widetilde{\boldsymbol{a}}_i^{s-1} \\ \boldsymbol{u}_{s-1}^\top \widetilde{\boldsymbol{a}}_i^{s-1} \end{bmatrix}$$

for $s = 2, \ldots, w$. Similarly, define $\widetilde{\boldsymbol{b}}^s$ to be the constraint vector after applying the first $s-1$ cuts:

$$\widetilde{\boldsymbol{b}}^1 = \boldsymbol{b}$$
$$\widetilde{\boldsymbol{b}}^s = \begin{bmatrix} \widetilde{\boldsymbol{b}}^{s-1} \\ \boldsymbol{u}_{s-1}^\top \widetilde{\boldsymbol{b}}^{s-1} \end{bmatrix}$$

for $s = 2, \ldots, w$. (These vectors depend on the cut vectors, but we will suppress this dependence for the sake of readability).

We prove this lemma by showing that there are $O(w2^w \|A\|_{1,1} + 2^w \|\boldsymbol{b}\|_1 + nw)$ hypersurfaces determined by polynomials that partition $[0,1]^m \times \cdots \times [0,1]^{m+w-1}$ into regions where in any one region R, the w cuts

$$\sum_{i=1}^{n} \lfloor \boldsymbol{u}_1^\top \widetilde{\boldsymbol{a}}_i^1 \rfloor x[i] \leq \lfloor \boldsymbol{u}_1^\top \widetilde{\boldsymbol{b}}^1 \rfloor$$
$$\vdots$$
$$\sum_{i=1}^{n} \lfloor \boldsymbol{u}_w^\top \widetilde{\boldsymbol{a}}_i^w \rfloor x[i] \leq \lfloor \boldsymbol{u}_w^\top \widetilde{\boldsymbol{b}}^w \rfloor$$

are invariant across all $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_w) \in R$. To this end, let $A_i = \|\boldsymbol{a}_i\|_1$ and $B = \|\boldsymbol{b}\|_1$. For each $s \in \{1, \ldots, w\}$, we claim that

$$\lfloor \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s \rfloor \in \left[-2^{s-1} A_i, 2^{s-1} A_i\right].$$

We prove this by induction. The base case of $s = 1$ is immediate since $\widetilde{\boldsymbol{a}}_i^1 = \boldsymbol{a}_i$ and $\boldsymbol{u} \in [0,1]^m$. Suppose now that the claim holds for s. By the induction hypothesis,

$$\left\|\widetilde{\boldsymbol{a}}_i^{s+1}\right\|_1 = \left\|\begin{bmatrix} \widetilde{\boldsymbol{a}}_i^s \\ \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s \end{bmatrix}\right\|_1 = \|\widetilde{\boldsymbol{a}}_i^s\|_1 + \left|\boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s\right| \leq 2 \|\widetilde{\boldsymbol{a}}_i^s\|_1 \leq 2^s A_i,$$

so

$$\lfloor \boldsymbol{u}_{s+1}^\top \widetilde{\boldsymbol{a}}_i^{s+1} \rfloor \in \left[-\left\|\widetilde{\boldsymbol{a}}_i^{s+1}\right\|_1, \left\|\widetilde{\boldsymbol{a}}_i^{s+1}\right\|_1\right] \subseteq [-2^s A_i, 2^s A_i],$$

as desired. Now, for each integer $k_i \in [-2^{s-1} A_i, 2^{s-1} A_i]$, we have

$$\lfloor \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s \rfloor = k_i \iff k_i \leq \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s < k_i + 1.$$

$\boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s$ is a polynomial in variables $\boldsymbol{u}_1[1], \ldots, \boldsymbol{u}_1[m], \boldsymbol{u}_2[1], \ldots, \boldsymbol{u}_2[m+1], \ldots, \boldsymbol{u}_s[1], \ldots, \boldsymbol{u}_s[m+s-1]$, for a total of $\leq ms + s^2$ variables. Its degree is at most s. There are thus a total of

$$\sum_{s=1}^{w} \sum_{i=1}^{n} (2 \cdot 2^{s-1} A_i + 1) = O\left(w 2^w \|A\|_{1,1} + nw\right)$$

polynomials each of degree at most w plus an additional $\sum_{s=1}^{w}(2 \cdot 2^{s-1} B + 1) = O(2^w B + w)$ polynomials of degree at most w corresponding to the hypersurfaces of the form

$$k_{n+1} \leq \boldsymbol{u}_s^\top \widetilde{\boldsymbol{b}}^s < k_{n+1} + 1$$

for each s and each $k_{n+1} \in \{-2^{s-1}B, \ldots, 2^{s-1}B\}$. This yields a total of $O(w2^w \|A\|_{1,1} + 2^w \|\boldsymbol{b}\|_1 + nw)$ polynomials in $\leq mw + w^2$ variables of degree $\leq w$. □

We now use this structure to provide a pseudo-dimension bound.

Theorem 4.3.5. *Let $\mathcal{F}_{\alpha,\beta}$ denote the set of all functions $f_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_w}$ for $\boldsymbol{u}_1 \in [0,1]^m, \ldots, \boldsymbol{u}_w \in [0,1]^{m+w-1}$ defined on the domain of integer programs $(\boldsymbol{c}, A, \boldsymbol{b})$ with $\|A\|_{1,1} \leq \alpha$ and $\|\boldsymbol{b}\|_1 \leq \beta$. Then,* $\mathsf{Pdim}(\mathcal{F}_{\alpha,\beta}) = O(mw^2 \log(mw(\alpha + \beta + n)))$.

Proof. The space of polynomials induced by the sth cut, that is, $\{k + \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s : \boldsymbol{a}_i \in \mathbb{R}^m, k \in \mathbb{R}\}$, is a vector space of dimension $\leq 1+m$. This is because for every $j = 1, \ldots, m$, all monomials that contain a variable $\boldsymbol{u}_s[j]$ for some s have the same coefficient (equal to $\boldsymbol{a}_i[j]$ for some $1 \leq i \leq n$). Explicit spanning sets are given by the following recursion. For each $j = 1, \ldots, m$ define $\widetilde{\boldsymbol{u}}_1[j], \ldots, \widetilde{\boldsymbol{u}}_w[j]$ recursively as

$$\widetilde{\boldsymbol{u}}_1[j] = \boldsymbol{u}_1[j]$$

$$\widetilde{\boldsymbol{u}}_s[j] = \boldsymbol{u}_s[j] + \sum_{\ell=1}^{s-1} \boldsymbol{u}_s[m+\ell]\widetilde{\boldsymbol{u}}_\ell[j]$$

for $s = 2, \ldots, w$. Then, $\{k + \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s : \boldsymbol{a}_i \in \mathbb{R}^m, k \in \mathbb{R}\}$ is contained in $\operatorname{span}\{1, \widetilde{\boldsymbol{u}}_s[1], \ldots, \widetilde{\boldsymbol{u}}_s[m]\}$. It follows that

$$\dim\left(\bigcup_{s=1}^{w} \{k + \boldsymbol{u}_s^\top \widetilde{\boldsymbol{a}}_i^s : \boldsymbol{a}_i \in \mathbb{R}^m, k \in \mathbb{R}\}\right) \leq 1 + mw.$$

The dual space thus also has dimension $\leq 1 + mw$. The VC dimension of the family of $0/1$ classifiers induced by a finite-dimensional vector space of functions is at most the dimension of the vector space. Thus, the VC dimension of the set of classifiers induced by the dual space is $\leq 1 + mw$. Finally, applying the main result of Balcan et al. [2021a] in conjunction with Lemma 4.3.4 gives the desired pseudo-dimension bound. □

The sample complexity of learning W sequential cuts is thus $\widetilde{O}(\kappa^2 mw^2/\epsilon^2)$.

4.3.3 Learning waves of simultaneous CG cuts

We now determine the sample complexity of making w sequential waves of cuts at the root, each wave consisting of k simultaneous CG cuts. Given vectors $\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^k \in [0,1]^m, \boldsymbol{u}_2^1, \ldots, \boldsymbol{u}_2^k \in [0,1]^{m+k}, \ldots, \boldsymbol{u}_w^1, \ldots, \boldsymbol{u}_w^k \in [0,1]^{m+k(w-1)}$, let $f_{\boldsymbol{u}_1^1,\ldots,\boldsymbol{u}_1^k,\ldots,\boldsymbol{u}_w^1,\ldots,\boldsymbol{u}_w^k}(\boldsymbol{c}, A, \boldsymbol{b})$ be the size of the tree B&C builds when it applies the CG cuts defined by $\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^k$, then applies the CG cuts defined by $\boldsymbol{u}_2^1, \ldots, \boldsymbol{u}_2^k$ to the new IP, and so on, all at the root of the search tree. The proof follows from the observation that w waves of k simultaneous cuts can be viewed as making kw sequential cuts with the restriction that cuts within each wave assign nonzero weight only to constraints from previous waves.

Theorem 4.3.6. *Let $\mathcal{F}_{\alpha,\beta}$ be the set of all functions $f_{\boldsymbol{u}_1^1,\ldots,\boldsymbol{u}_1^k,\ldots,\boldsymbol{u}_w^1,\ldots,\boldsymbol{u}_w^k}$ for*

$$\boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_1^k \in [0,1]^m, \ldots, \boldsymbol{u}_w^1, \ldots, \boldsymbol{u}_w^k \in [0,1]^{m+k(w-1)}$$

defined on the domain of integer programs $(\boldsymbol{c}, A, \boldsymbol{b})$ with $\|A\|_{1,1} \leq \alpha$ and $\|\boldsymbol{b}\|_1 \leq \beta$. Then, $\mathsf{Pdim}(\mathcal{F}_{\alpha,\beta}) = O(mk^2w^2 \log(mkw(\alpha + \beta + n)))$.

Proof. Applying cuts $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^k \in [0,1]^m$ simultaneously is equivalent to sequentially applying the cuts

$$\boldsymbol{u}^1 \in [0,1]^m, \begin{bmatrix}\boldsymbol{u}^2\\0\end{bmatrix} \in [0,1]^{m+1}, \begin{bmatrix}\boldsymbol{u}^3\\0\\0\end{bmatrix} \in [0,1]^{m+2}, \ldots, \begin{bmatrix}\boldsymbol{u}^k\\0\\\vdots\\0\end{bmatrix} \in [0,1]^{m+k-1}.$$

Thus, the set in question is a subset of $\left\{f_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{kw}} : \boldsymbol{u}_1 \in [0,1]^m, \ldots, \boldsymbol{u}_{kw} \in [0,1]^{m+kw-1}\right\}$ and has pseudo-dimension $O(mk^2w^2\log(mkw(\alpha+\beta+n)))$ by Theorem 4.3.5. □

So, the sample complexity of learning W waves of k cuts is $\widetilde{O}(\kappa^2 mk^2w^2/\epsilon^2)$.

4.3.4 Learning cut selection policies

In this section, we bound the sample complexity of learning *cut-selection policies* at the root, that is, functions that take as input an IP and output a candidate cut. Using scoring rules is a more nuanced way of choosing cuts since it allows for the cut parameters to depend on the input IP.

Formally, let $\mathcal{I}_m$ be the set of IPs with m constraints (the number of variables is always fixed at n) and let $\mathcal{H}_m$ be the set of all hyperplanes in $\mathbb{R}^m$. A *scoring rule* is a function $\texttt{score} : \cup_m(\mathcal{H}_m \times \mathcal{I}_m) \to \mathbb{R}_{\geq 0}$. The real value $\texttt{score}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, (\boldsymbol{c}, A, \boldsymbol{b}))$ is a measure of the quality of the cutting plane $\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta$ for the IP $(\boldsymbol{c}, A, \boldsymbol{b})$. We will see explicit examples of scoring rules in the next chapter.

Suppose $\texttt{score}_1, \ldots, \texttt{score}_d$ are d different scoring rules. We now bound the sample complexity of learning a combination of these scoring rules that guarantee a low expected tree size. Our high-level proof technique is the same as in the previous section: we establish that the dual tree-size functions are piecewise structured, and then apply the general framework of Balcan et al. [2021a] to obtain pseudo-dimension bounds.

Theorem 4.3.7. *Let $\mathcal{C}$ be a set of cutting-plane parameters such that for every IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there is a decomposition of $\mathcal{C}$ into $\leq r$ regions such that the cuts generated by any two vectors in the same region are the same. Let $\texttt{score}_1, \ldots, \texttt{score}_d$ be d scoring rules. For $\boldsymbol{\mu} \in \mathbb{R}^d$, let $f_{\boldsymbol{\mu}}(\boldsymbol{c}, A, \boldsymbol{b})$ be the size of the tree B&C builds when it chooses a cut from $\mathcal{C}$ to maximize $\mu[1]\texttt{score}_1(\cdot, (\boldsymbol{c}, A, \boldsymbol{b})) + \cdots + \mu[d]\texttt{score}_d(\cdot, (\boldsymbol{c}, A, \boldsymbol{b}))$. Then,* $\mathsf{Pdim}(\{f_{\boldsymbol{\mu}} : \boldsymbol{\mu} \in \mathbb{R}^d\}) = O(d\log(rd))$.

Proof. Fix an integer program $(\boldsymbol{c}, A, \boldsymbol{b})$. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r \in \mathcal{C}$ be representative cut parameters for each of the r regions. Consider the hyperplanes $\sum_{i=1}^d \mu[i]\texttt{score}_i(\boldsymbol{u}_s) = \sum_{i=1}^d \mu[i]\texttt{score}_i(\boldsymbol{u}_t)$ for each $s \neq t \in \{1, \ldots, r\}$ (suppressing the dependence on $\boldsymbol{c}, A, \boldsymbol{b}$). These $O(r^2)$ hyperplanes partition $\mathbb{R}^d$ into regions such that as $\boldsymbol{\mu}$ varies in a given region, the cut chosen from $\mathcal{C}$ is invariant. The desired pseudo-dimension bound follows from the main result of Balcan et al. [2021a]. □

Theorem 4.3.7 can be directly instantiated with the class of CG cuts. Combining Lemma 4.3.2 with the basic combinatorial fact that k hyperplanes partition $\mathbb{R}^m$ into at most k^m regions, we get that the pseudo-dimension of $\{f_{\boldsymbol{\mu}} : \boldsymbol{\mu} \in \mathbb{R}^d\}$ defined on IPs with $\|A\|_{1,1} \leq \alpha$ and $\|\boldsymbol{b}\|_1 \leq \beta$ is $O(dm\log(d(\alpha+\beta+n)))$. Instantiating Theorem 4.3.7 with the set of all sequences of w

CG cuts requires the following extension of scoring rules to sequences of cutting planes. A *sequential scoring rule* is a function that takes as input an IP $(\boldsymbol{c}, A, \boldsymbol{b})$ and a sequence of cutting planes $h_1, \ldots, h_w$, where each cut lives in one higher dimension than the previous. It measures the quality of this sequence of cutting planes when applied one after the other to the original IP. Every scoring rule `score` can be naturally extended to a sequential scoring rule $\overline{\texttt{score}}$ defined by $\overline{\texttt{score}}(h_1, \ldots, h_w, (\boldsymbol{c}^0, A^0, \boldsymbol{b}^0)) = \sum_{i=0}^{w-1} \texttt{score}(h_{i+1}, (\boldsymbol{c}^i, A^i, \boldsymbol{b}^i))$, where $(\boldsymbol{c}^i, A^i, \boldsymbol{b}^i)$ is the IP after applying cuts $h_1, \ldots, h_{i-1}$.

Corollary 4.3.8. *Let $\mathcal{C} = [0,1]^m \times \cdots \times [0,1]^{m+w-1}$ denote the set of possible sequences of w Chvátal-Gomory cut parameters. Let $\mathit{score}_1, \ldots, \mathit{score}_d : \mathcal{C} \times \mathcal{I}_m \times \cdots \times \mathcal{I}_{m+w-1} \to \mathbb{R}$ be d sequential scoring rules and let $f_{\boldsymbol{\mu}}(\boldsymbol{c}, A, \boldsymbol{b})$ be as in Theorem 4.3.7 for the class $\mathcal{C}$. Then,* $\mathsf{Pdim}(\{f^w_{\boldsymbol{\mu}} : \boldsymbol{\mu} \in \mathbb{R}^d\}) = O(dmw^2 \log(dw(\alpha + \beta + n)))$.

Proof. In Lemma 4.3.4 and Theorem 4.3.5 we showed that there are $O(w2^w\alpha + 2^w\beta + nw)$ multivariate polynomials that belong to a family of polynomials $\mathcal{G}$ with $\mathsf{VC}(\mathcal{G}^*) \leq 1 + mw$ ($\mathcal{G}^*$ denotes the dual of $\mathcal{G}$) that partition $\mathcal{C}$ into regions such that resulting sequence of cuts is invariant in each region. By Claim 3.5 by Balcan et al. [2021a], the number of regions is

$$O(w2^w\alpha + 2^w\beta + nw)^{\mathsf{VC}(\mathcal{G}^*)} \leq O(w2^w\alpha + 2^w\beta + nw)^{1+mw}.$$

The corollary then follows from Theorem 4.3.7. □

These results bound the sample complexity of learning cut-selection policies based on scoring rules, which allow the cuts B&C that selects to depend on the input IP.

4.4 Sample Complexity Bounds for Branch-and-Cut and General Tree Search

Our main contribution in this section is a formalization of a general model of tree search (Algorithm 2) 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 [Balas et al., 1996a], objective parallelism [Achterberg, 2007], directed cutoff distance [Gamrath et al., 2020], and integral support [Wesselmann and Stuhl, 2012] scoring rules, all used for cut selection by SCIP [Gamrath et al., 2020], a leading open-source solver; the best-bound scoring rule for node selection; and the linear, product, and most-fractional scoring rules for variable selection using strong branching [Achterberg, 2007]. 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 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 [Balcan et al., 2018a, 2021d]. 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 [1999]), 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. [2021d], 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. [2018a], 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.

4.4.1 Main tree search model

In this section we present our general tree search model and situate it within the framework of sample complexity. Balcan et al. [2021d], Vitercik [2021] studied the sample complexity of a much more general formulation of a tunable search algorithm without any inherent tree structure. Our formulation explicitly builds a tree.

Tree search starts with a root node. In each round of tree search, a leaf node Q is selected. At this node, one of three things may occur: (1) Q is fathomed, meaning it is never visited again, (2) some action is taken at Q, and then it is fathomed, or (3) some action is taken at Q, and then some number of children nodes of Q are added to the tree. (For example, an action might represent a decision about which variable to branch on.) This process repeats until the tree has no unfathomed leaves. More formally, there are functions `actions`, `children`, and `fathom` prescribing how the search proceeds. Given a partial tree $\mathcal{T}$ and a leaf Q of $\mathcal{T}$, $\texttt{actions}(\mathcal{T}, Q)$ outputs a set of actions available at Q. Given a partial tree $\mathcal{T}$, a leaf Q of $\mathcal{T}$, and an action $A \in \texttt{actions}(\mathcal{T}, Q)$, $\texttt{fathom}(\mathcal{T}, Q, A) \in \{\texttt{true}, \texttt{false}\}$ is a Boolean function used to determine when to fathom a leaf Q of $\mathcal{T}$ given that action $A \in \texttt{actions}(\mathcal{T}, Q) \cup \{\texttt{None}\}$ was taken at Q, and $\texttt{children}(\mathcal{T}, Q, A)$ outputs a (potentially empty) list of nodes representing the children of Q to be added to the search tree given that action A was taken at Q. Finally, $\texttt{nscore}(\mathcal{T}, Q)$ is a node-selection score that outputs a real-valued score for each leaf of $\mathcal{T}$, and $\texttt{ascore}(\mathcal{T}, Q, A)$ is an action-selection score that outputs a real-valued score for each action $A \in \texttt{actions}(\mathcal{T}, Q)$. These scores are heuristics that are meant to indicate the quality of exploring a node or performing an action.

Many aspects of B&C are governed by scoring rules [Achterberg, 2007]. For example, commonly used scoring rules for cutting plance selection include *efficacy* [Balas et al., 1996a], which is the perpendicular distance from the current LP solution to the cutting plane; *parallelism* [Achterberg, 2007], which measures the angle between the objective and the normal vector to the cutting plane; and *directed cutoff* [Gamrath et al., 2020], which is the distance from the current LP solution to the cutting plane along the direction of the line segment connecting the LP solution to the current best incumbent integer solution For node selection, under the commonly used best-first node selection policy, $\texttt{nscore}(\mathcal{T}, Q)$ equals the objective value of the LP relaxation of the IP represented by the node Q. Finally, for variable selection, popular scoring rules include a maximum change in LP objective value after branching on the variable (where the maximum is taken over the two resulting children), the minimum change in the LP objective value, linear combinations of these two values, and the product of these two values [Achterberg, 2007]. Algorithm 2 is a formal description of tree search using these functions.

The key condition that enables us to derive stronger sample complexity bounds compared to prior research is the notion of a *path-wise* function, which was also used in prior research but only in the context of variable selection Balcan et al. [2018a].

Definition 4.4.1 (Path-wise functions)**.** A function f on tree-leaf pairs is path-wise if for all $\mathcal{T}$ and $Q \in \mathcal{T}$, $f(\mathcal{T}, Q) = f(\mathcal{T}_Q, Q)$, where $\mathcal{T}_Q$ is the path from the root of $\mathcal{T}$ to Q. A function g on tree-leaf-action triples is path-wise if for all A, $f_A(\mathcal{T}, Q) := g(\mathcal{T}, Q, A)$ is path-wise.

Algorithm 2 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.

We assume that `actions`, `ascore`, `nscore` and `children` are path-wise, though `fathom` is not necessarily path-wise.

Many commonly-used scoring rules are path-wise. For example, scoring rules are often functions of the LP relaxation of the IP represented by a given node, and these scoring rules are path-wise. Specific examples include the efficacy, objective parallelism, directed cutoff distance, and integral support scoring rules used for cut selection; the best-bound scoring rule for node selection; and the linear, product, and most-fractional scoring rules for variable selection using strong branching. A point of clarification: the pathwise assumption is with respect to the numerical scores assigned to actions/nodes. The actual act of, for example, node selection, can depend on the entire tree. For example, consider the best-bound node selection rule in branch-and-cut, which chooses the node with the best LP estimate. Here, the scoring rule, which is the LP objective value itself, is pathwise, but ultimately the node that is selected depends on the LP bounds at every unexplored node of the tree. This is fine for our analysis. Similarly, the decision to fathom a node based on LP bounds is a decision that depends on the entire tree built so far, which is also captured by our analysis.

No one scoring rule is optimal across all application domains, and prior research on variable selection has shown that it can be advantageous to adapt the scoring rule to the application domain at hand Balcan et al. [2018a]. To this end, Algorithm 2 can be tuned by two parameters $\mu \in [0, 1]$ and $\lambda \in [0, 1]$ that control action selection and node selection, respectively. Given two fixed path-wise action-selection scores $\texttt{ascore}_1$ and $\texttt{ascore}_2$, we define a new score by

$$\texttt{ascore}_\mu(\mathcal{T}, Q) = \mu \cdot \texttt{ascore}_1(\mathcal{T}, Q) + (1 - \mu) \cdot \texttt{ascore}_2(\mathcal{T}, Q).$$

Similarly, given two path-wise node-selection scores $\texttt{nscore}_1$ and $\texttt{nscore}_2$, we define

$$\texttt{nscore}_\lambda(\mathcal{T}, Q, A) = \lambda \cdot \texttt{nscore}_1(\mathcal{T}, Q, A) + (1 - \lambda) \cdot \texttt{nscore}_2(\mathcal{T}, Q, A).$$

Then, if $\texttt{nscore}_\lambda$ and $\texttt{ascore}_\mu$ are used as the scores in Algorithm 2, we can view the behavior of tree search as a function of μ and λ. The choice to use a convex combination of scores is not new: prior research has shown that this idea can lead to dramatic improvements in the case of single-variable branching Balcan et al. [2018a]. Furthermore, the leading open source solver SCIP uses a hard-coded weighted sum of scoring rules to select cutting planes. More broadly, interpolating between two scores is a commonly-studied modeling choice in other machine learning topics such as clustering Balcan et al. [2017].

Finally, we assume there exists $b, k \in \mathbb{N}$ such that $|\texttt{actions}(\mathcal{T}, Q)| \leq b$ for any $Q \in \mathcal{T}$, and $|\texttt{children}(\mathcal{T}, Q, A)| \leq k$ for all Q, A.

4.4.2 Problem formulation

Let $\mathcal{Q}$ denote the domain of possible input root nodes Q to Algorithm 2 (for example, the set of all IPs with n variables and m constraints). As is common in prior research on algorithm configuration [Horvitz et al., 2001, Xu et al., 2008, 2011, Hutter et al., 2009, Leyton-Brown et al., 2009, Kadioglu et al., 2010, Sandholm, 2013], we assume there is some unknown distribution $\mathcal{D}$ over $\mathcal{Q}$. We are interested in bounding the sample complexity of classes of real valued functions $\mathcal{F} = \{f : \mathcal{Q} \to \mathbb{R}\}$. In the context of Algorithm 2, we study families of *tree-constant* cost functions. A cost function $\textsf{cost} : \mathcal{Q} \to \mathbb{R}$ is tree constant if $\textsf{cost}(Q)$ only depends on the tree built by Algorithm 2 on input Q (an example is tree size). Let $\textsf{cost}_{\mu,\lambda}(Q)$ denote this cost when Algorithm 2 is run using the scores $\texttt{ascore}_\mu = \mu \cdot \texttt{ascore}_1 + (1-\mu) \cdot \texttt{ascore}_2$ and $\texttt{nscore}_\lambda = \lambda \cdot \texttt{nscore}_1 + (1-\lambda) \cdot \texttt{nscore}_2$. We study the sample complexity of $\mathcal{F} = \{\textsf{cost}_{\mu,\lambda} : \mu, \lambda \in [0,1]\}$. We emphasize that we primarily interpret tree-constant functions as proxies for run-time/memory. In the context of integer programming, tree size is one such measure. A strength of these guarantees is that they apply no matter how the parameters are tuned: optimally or suboptimally, manually or automatically. For *any* configuration, these guarantees bound the difference between average performance over the training set and expected future performance on unseen IPs.

4.4.3 Generalization guarantees for tree search

In order to derive our sample complexity guarantees, we first prove a key structural property: the behavior of Algorithm 2 is piecewise constant as a function of the node-selection score parameter λ and the action-selection score parameter μ. We give a high-level outline of our approach. We first assume that the conditional checks $\texttt{fathom}(\mathcal{T}, Q, \cdot) = \texttt{true}$ (lines 4 and 8) are suppressed. Let $\mathcal{A}'$ denote Algorithm 2 without these checks (so $\mathcal{A}'$ fathoms a node if and only if the depth limit is reached or if the node has no children). The behavior of $\mathcal{A}'$ as a function of μ and λ can be shown to be piecewise constant using the same argument as in Claim 3.4 of Balcan et al. [2018a]. Given this, our first main technical contribution (Lemma 4.4.2) is a generalization of Claim 3.5 of Balcan et al. [2018a] that relates the behavior of $\mathcal{A}'$ to Algorithm 2. The argument in Balcan et al. [2018a] is specific to branching, but we are able to prove our result in a much more general setting. Our second main technical contribution (Lemma 4.4.4) is to establish piecewise structure when the node-selection score is controlled by $\lambda \in [0,1]$. The main reason for this auxiliary step of analyzing $\mathcal{A}'$ is due to the fact that $\texttt{fathom}$ is *not* necessarily a path-wise function, and can depend on the state of the entire tree.

Lemma 4.4.2. *Fix $\mu \in [0,1]$. Let $\mathcal{T}$ and $\mathcal{T}'$ be the trees built by Algorithm 2 and $\mathcal{A}'$, respectively, using the action-selection score $\mu \cdot \mathtt{ascore}_1 + (1-\mu) \cdot \mathtt{ascore}_2$. Let Q be any node in $\mathcal{T}$, and let $\mathcal{T}_Q$ be the path from the root of $\mathcal{T}$ to Q. Then, $\mathcal{T}_Q$ is a rooted subtree of $\mathcal{T}'$, no matter what node selection policy is used.*

Proof. Let t denote the length of the path $\mathcal{T}_Q$. Let $\mathcal{T}_Q$ be comprised of the sequence of nodes $(Q_1, \dots, Q_t)$ such that Q_1 is the root of $\mathcal{T}$, $Q_t = Q$, and for each τ, $Q_{\tau+1} \in \mathtt{children}(\mathcal{T}_{Q_\tau}, Q_\tau, A_\tau)$ where $A_\tau \in \mathtt{actions}(\mathcal{T}_{Q_\tau}, Q_\tau)$ is the action selected by Algorithm 2 at node Q_τ. We show that $(Q_1, \dots, Q_t)$ is a rooted path in $\mathcal{T}'$ as well.

Suppose for the sake of contradiction that this is not the case. Let $\tau \in \{2, \dots, t\}$ be the minimal index such that $(Q_1, \dots, Q_{\tau-1})$ is a rooted path in $\mathcal{T}'$, but there is no edge in $\mathcal{T}'$ from $Q_{\tau-1}$ to node Q_τ. There are two possible cases:

Case 1. $Q_{\tau-1}$ was fathomed by $\mathcal{A}'$. This case is trivially not possible since whenever $\mathcal{A}'$ fathoms a node, so does Algorithm 2 (recall $\mathcal{A}'$ was defined by suppressing fathoming conditions of Algorithm 2).

Case 2. $Q_\tau \notin \mathtt{children}(\mathcal{T}', Q_{\tau-1}, A'_{\tau-1})$ where $A'_{\tau-1}$ is the action taken by $\mathcal{A}'$ at node $Q_{\tau-1}$. In this case, if $\mathtt{children}(\mathcal{T}', Q_{\tau-1}, A'_{\tau-1}) = \emptyset$, then $Q_{\tau-1}$ would be fathomed by $\mathcal{A}'$, which cannot happen by the first case. Otherwise, if $\mathtt{children}(\mathcal{T}', Q_{\tau-1}, A'_{\tau-1}) \neq \emptyset$, we show that we arrive at a contradiction due to the fact that the scoring rules, action-set functions, and children functions are all path-wise. Let $A'_{\tau-1}$ denote the action taken by $\mathcal{A}'$ at $Q_{\tau-1}$, and let $A_{\tau-1}$ denote the action taken by Algorithm 2 at $Q_{\tau-1}$. Since $\mathtt{actions}$ is path-wise,

$$\mathtt{actions}(\mathcal{T}, Q_{\tau-1}) = \mathtt{actions}(\mathcal{T}_{Q_{\tau-1}}, Q_{\tau-1}) = \mathtt{actions}(\mathcal{T}', Q_{\tau-1}).$$

Since $\mathtt{ascore}_1$ and $\mathtt{ascore}_2$ are path-wise, we have

$$\begin{aligned}\mu \cdot \mathtt{ascore}_1(\mathcal{T}, &Q_{\tau-1}, A) + (1-\mu) \cdot \mathtt{ascore}_2(\mathcal{T}, Q_{\tau-1}, A)\\ &= \mu \cdot \mathtt{ascore}_1(\mathcal{T}_{Q_{\tau-1}}, Q_{\tau-1}, A) + (1-\mu) \cdot \mathtt{ascore}_2(\mathcal{T}_{Q_{\tau-1}}, Q_{\tau-1}, A)\\ &= \mu \cdot \mathtt{ascore}_1(\mathcal{T}', Q_{\tau-1}, A) + (1-\mu) \cdot \mathtt{ascore}_2(\mathcal{T}', Q_{\tau-1}, A).\end{aligned}$$

for all actions $A \in \mathtt{actions}(\mathcal{T}_{Q_{\tau-1}}, Q_{\tau-1})$. Therefore Algorithm 2 and $\mathcal{A}'$ choose the same action at node Q_{t-1}, that is, $A_{\tau-1} = A'_{\tau-1}$. Finally, since $\mathtt{children}$ is path-wise, we have

$$\mathtt{children}(\mathcal{T}, Q_{\tau-1}, A_{\tau-1}) = \mathtt{children}(\mathcal{T}_{Q_{\tau-1}}, Q_{\tau-1}, A_{\tau-1}) = \mathtt{children}(\mathcal{T}', Q_{\tau-1}, A_{\tau-1}).$$

Since $Q_\tau \in \mathtt{children}(\mathcal{T}, Q_{\tau-1}, A_{\tau-1})$, this is a contradiction, which completes the proof. □

We use the following generalization of Claim 3.4 of Balcan et al. [2018a] that shows the behavior of $\mathcal{A}'$ is piecewise constant. While their argument only applies to single-variable branching, our key insight is that the same reasoning can be readily adapted to handle any actions (including general branching constraints and cutting planes). The structure of our proof is identical, but is modified to work in our more general setting. This style of analysis is similar in spirit to Megiddo [1979].

Lemma 4.4.3. *Let $\mathtt{ascore}_1$ and $\mathtt{ascore}_2$ be two path-wise action-selection scores. Fix the input root node Q. There are $T \leq k^{\Delta(\Delta-1)/2} b^{\Delta}$ subintervals $I_1, \dots, I_T$ partitioning $[0,1]$ where for any subinterval I_t, the action-selection score $\mu \cdot \mathtt{ascore}_1 + (1-\mu) \cdot \mathtt{ascore}_2$ results in the same tree built by $\mathcal{A}'$ for all $\mu \in I_t$, no matter what node selection policy is used.*

Proof of Lemma 4.4.3. Let $\mathcal{T}$ denote the tree built by $\mathcal{A}'$. For $i \in [\Delta]$, let $\mathcal{T}[i]$ denote the restriction of $\mathcal{T}$ to nodes of depth at most i. Let $\texttt{ascore}_\mu = \mu \cdot \texttt{ascore}_1 + (1-\mu) \cdot \texttt{ascore}_2$. We prove the lemma by induction on i. In particular, we show that for each $i \in [\Delta]$, there are $k^{i(i-1)/2} b^i$ subintervals partitioning $[0,1]$ such that $\mathcal{T}[i]$ is invariant over all μ within any given subinterval. Since $\mathcal{T}[\Delta] = \mathcal{T}$, this implies the lemma statement. The base case of $i = 1$ is trivial since $\mathcal{T}[1]$ consists of only the root.

Now, suppose the statement holds for some $i \in \{1, \ldots, \Delta - 1\}$. That is, there are $T \leq k^{i(i-1)/2} b^i$ disjoint intervals $I_1 \cup \cdots \cup I_T = [0,1]$ such that $\mathcal{T}[i]$ is invariant over all μ within any given subinterval (our inductive hypothesis). Fix one of these subintervals I_t. We subdivide I_t into subintervals such that $\mathcal{T}[i+1]$ is invariant within each one of these smaller subintervals. Let Q be any leaf of $\mathcal{T}[i]$, and for $\mu \in I_t$ let $\mathcal{T}_\mu$ denote the state of the tree using $\texttt{ascore}_\mu$ at the point that Q is selected. Since $i < \Delta$, Q is not fathomed at line 4, regardless of μ. Next, since $\texttt{actions}$ is path-wise, the actions available at Q depend only on the path $\mathcal{T}_Q$ from the root of $\mathcal{T}$ to Q, which, by the inductive hypothesis, is invariant over all $\mu \in I_t$. That is, $\texttt{actions}(\mathcal{T}_\mu, Q) = \texttt{actions}(\mathcal{T}_Q, Q)$ for all $\mu \in I_t$. Then, $\texttt{ascore}_\mu$ with parameter μ will select action $A \in \texttt{actions}(\mathcal{T}_Q, Q)$ if and only if

$$\begin{aligned} A &= \operatorname*{argmax}_{A_0 \in \texttt{actions}(\mathcal{T}_Q, Q)} \mu \cdot \texttt{ascore}_1(\mathcal{T}_\mu, Q, A_0) + (1-\mu) \cdot \texttt{ascore}_2(\mathcal{T}_\mu, Q, A_0) \\ &= \operatorname*{argmax}_{A_0 \in \texttt{actions}(\mathcal{T}_Q, Q)} \mu \cdot \texttt{ascore}_1(\mathcal{T}_Q, Q, A_0) + (1-\mu) \cdot \texttt{ascore}_2(\mathcal{T}_Q, Q, A_0), \end{aligned}$$

where the second equality follows from the fact that $\texttt{ascore}_1$ and $\texttt{ascore}_2$ are path-wise. Thus, for a fixed A_0, $\texttt{ascore}_\mu$ is linear in μ, so for each A_0 there is at most one subinterval of $[0,1]$ such that for all μ in that subinterval, A_0 maximizes $\texttt{ascore}_\mu$. Thus, each leaf of $\mathcal{T}[i]$ contributes at most b subintervals such that for μ within a given subinterval, the action selected at each leaf of $\mathcal{T}[i]$ is invariant. $\mathcal{T}[i]$ consists of at most k^i leaves, so this is a total of at most $k^i b$ subintervals. Now, since the action A selected at each leaf Q of $\mathcal{T}[i]$ is invariant, the set of children $\texttt{children}(\mathcal{T}_\mu, Q, A) = \texttt{children}(\mathcal{T}_Q, Q, A)$ of Q added to the tree is also invariant, using the fact that $\texttt{children}$ is path-wise. This shows that within every subinterval, $\mathcal{T}[i+1]$ is invariant. The total number of subintervals is, by the induction hypothesis, at most $k^{i(i-1)/2} b^i \cdot k^i b = k^{(i+1)i/2} b^{i+1}$, as desired. □

We now prove our main structural result for Algorithm 2.

Lemma 4.4.4. *Let* $\texttt{ascore}_1$ *and* $\texttt{ascore}_2$ *be path-wise action-selection scores and let* $\texttt{nscore}_1$ *and* $\texttt{nscore}_2$ *be path-wise node-selection scores. Fix the input root node* Q. *There are* $T \leq k^{\Delta(9+\Delta)} b^\Delta$ *rectangles partitioning* $[0,1]^2$ *such that for any rectangle* R_t, *the node-selection score* $\lambda \cdot \texttt{nscore}_1 + (1-\lambda) \cdot \texttt{nscore}_2$ *and the action-selection score* $\mu \cdot \texttt{ascore}_1 + (1-\mu) \cdot \texttt{ascore}_2$ *result in the same tree built by Algorithm 2 for all* $(\mu, \lambda) \in R_t$.

Proof. By Lemma 4.4.3, there is a partition of $[0,1]$ into subintervals $I_1 \cup \cdots \cup I_T$ such that for all μ within a given subinterval, the tree built by $\mathcal{A}'$ is invariant (independent of the node-selection score). Fix a subinterval I_t of this partition. Let $\mathcal{T}$ denote the tree built by Algorithm 2. For each node $Q \in \mathcal{T}$, let $\mathcal{T}_Q$ denote the path from the root to Q in $\mathcal{T}$. Since $\texttt{nscore}_1$ is path-wise, for any tree $\mathcal{T}'$ containing $\mathcal{T}_Q$ as a rooted path, $\texttt{nscore}_1(\mathcal{T}', Q) = \texttt{nscore}_1(\mathcal{T}_Q, Q)$. The same holds for $\texttt{nscore}_2$. For every pair of nodes $Q_1, Q_2 \in \mathcal{T}$, let $\lambda(Q_1, Q_2) \in [0,1]$ denote the unique solution to

$$\begin{aligned}\lambda \cdot \texttt{nscore}_1(\mathcal{T}_{Q_1}, Q_1) + (1-\lambda) \cdot \texttt{nscore}_2(\mathcal{T}_{Q_1}, Q_1) \\ = \lambda \cdot \texttt{nscore}_1(\mathcal{T}_{Q_2}, Q_2) + (1-\lambda) \cdot \texttt{nscore}_2(\mathcal{T}_{Q_2}, Q_2),\end{aligned}$$

if it exists (if there are either (1) no solutions or (2) infinitely many solutions, set $\lambda(Q_1, Q_2) = 0$). The thresholds $\lambda(Q_1, Q_2)$ for every pair of nodes $Q_1, Q_2 \in \mathcal{T}$ partition $[0,1]$ into subintervals such that for all λ within a given subinterval, the total order over the nodes of $\mathcal{T}$ induced by $\texttt{nscore}_\lambda$ is invariant. In particular, this means that the node selected by each iteration of Algorithm 2 is invariant. Let $J_1 \cup \cdots \cup J_S$ denote these subintervals induced by the thresholds over all subinterval $I_t \in \{I_1, \ldots, I_T\}$ established in Lemma 4.4.3.

We now show that this implies that the tree built by Algorithm 2 is invariant over all (μ, λ) within a given rectangle $I_t \times J_s$. Fix some rectangle $I_t \times J_s$. We proceed by induction on the iterations (of the while loop) of Algorithm 2. For the base case (iteration 0, before entering the while loop), the tree consists of only the root, so the hypothesis trivially holds. Now, suppose the statement holds up until the jth iteration, for some j. We analyze each line of Algorithm 2 to show that the behavior of the $j+1$st iteration is independent of $(\mu, \lambda) \in I_t \times J_s$. First, since J_s determines the node selected at each iteration (as argued above), the node selected on the $j+1$st iteration (line 3) is fixed, independent of $(\mu, \lambda) \in I_t \times J_s$. Denote this node by Q. Thus, whether $\texttt{depth}(Q) = \Delta$ is independent of $(\mu, \lambda) \in I_t \times J_s$, and similarly whether $\texttt{fathom}(\mathcal{T}, Q, \texttt{None}) = \texttt{true}$ is independent of $(\mu, \lambda) \in I_t \times J_s$ (line 4). This implies that whether or not Q is fathomed at this stage is independent of $(\mu, \lambda) \in I_t \times J_s$. If Q was fathomed, we are done. Otherwise, we argue that the action selected at line 7 is invariant over $(\mu, \lambda) \in I_t \times J_s$. By Lemma 4.4.3, $\mathcal{A}'$ builds the same tree for all $\mu \in I_t$. Let $\mathcal{T}_Q$ denote the path from the root to Q in this tree. By Lemma 4.4.2, $\mathcal{T}_Q$ is the path from the root to Q in the tree built by Algorithm 2 as well. The action selected at Q by $\mathcal{A}'$ is invariant over $\mu \in I_t$ (by Lemma 4.4.3). Therefore, since $\texttt{actions}$, $\texttt{ascore}_1$, and $\texttt{ascore}_2$ are path-wise, the action A selected by Algorithm 2 at Q is invariant over $\mu \in I_t$. Finally, $\texttt{fathom}(\mathcal{T}, Q, A)$ and $\texttt{children}(\mathcal{T}, Q, A)$ are completely determined, so the execution of the remaining conditional statement (line 8 to line 13) is invariant over $(\mu, \lambda) \in I_t \times J_s$. Thus, the entire iteration of Algorithm 2 is invariant over $(\mu, \lambda) \in I_t \times J_s$, which completes the induction.

Finally, we count the total number of rectangles in our partition of $[0,1]^2$. For each interval I_t in the partition established in Lemma 4.4.3, we obtained a partition of $I_t \times [0,1]$ into rectangles induced by at most $\binom{|\mathcal{T}|}{2}$ thresholds, which consists of at most at most

$$1 + \binom{(k^{\Delta+1}-1)/(k-1)}{2} \leq 1 + \left(\frac{k^{\Delta+1}-1}{k-1}\right)^2 \leq k^{5\Delta}$$

subintervals. Accounting for every interval $I_t \in \{I_1, \ldots, I_T\}$ in the partition from Lemma 4.4.3, we get a total of $Tk^{5\Delta} \leq k^{\Delta(9+\Delta)/2} b^\Delta$ rectangles, as desired. □

We now derive generalization guarantees for the collection $\mathcal{F} = \{\mathsf{cost}_{\mu,\lambda} : (\mu, \lambda) \in [0,1]^2\}$ where cost is any tree-constant function, such as tree size. We do this by bounding the *pseudo-dimension* of $\mathcal{F}$.

Algorithm 3 Tree search with multiple actions

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 $\texttt{fathom}(\mathcal{T}, Q, \texttt{None}, \ldots, \texttt{None})$ **then**
        Fathom Q.
    else
        For $i = 1, \ldots, d$, take $A_i \in \texttt{actions}_i(\mathcal{T}, Q)$ that maximizes $\texttt{ascore}_i(\mathcal{T}, Q, A_i)$.
        if $\texttt{fathom}(\mathcal{T}, Q, A_1, \ldots, A_d)$ **then**
            Fathom Q.
        else if $\texttt{children}(\mathcal{T}, Q, A_1, \ldots, A_d) = \emptyset$ **then**
            Fathom Q.
        else
            Add all nodes in $\texttt{children}(\mathcal{T}, Q, A_1, \ldots, A_d)$ to $\mathcal{T}$ as children of Q.

Theorem 4.4.5. *Let* $\mathsf{cost}(Q)$ *be any tree-constant cost function, and let* $\mathsf{cost}_{\mu,\lambda}(Q)$ *be the cost of the tree built by Algorithm 2 on input root node Q using action-selection score parameterized by* μ *and node-selection score parameterized by* λ. *Then,* $\mathsf{Pdim}(\{\mathsf{cost}_{\mu,\lambda}\}) = O(\Delta^2 \log k + \Delta \log b)$.

Proof. By Lemma 4.4.4, there are at most $T = k^{\Delta(9+\Delta)} b^{\Delta}$ rectangles partitioning $[0,1]^2$ such that for a fixed input node Q, $\mathsf{cost}_{\mu,\lambda}(Q)$ is constant over each rectangle as a function of μ, λ. These T rectangles can be defined by T thresholds on $[0,1]$ corresponding to μ and T thresholds on $[0,1]$ corresponding to λ. Thus, the T rectangles can be identified by $T^2 = k^{2\Delta(9+\Delta)} b^{2\Delta}$ linear separators in $\mathbb{R}^2$. The VC dimension of linear separators in $\mathbb{R}^2$ is $O(1)$. The pseudo-dimension of the set of constant functions is also $O(1)$. Plugging these quantities into the main theorem of Balcan et al. [2021a] yields the theorem statement. □

Multiple actions Theorem 10.2.4 can be easily generalized to the case where there are multiple actions of different types taken at each node of Algorithm 2. Specifically, there are now d path-wise action-set functions $\texttt{actions}_1, \ldots, \texttt{actions}_d$, and at line 7 of Algorithm 2 we take one action of each type, that is, we select action $A_1 \in \texttt{actions}_1(\mathcal{T}, Q)$, $A_2 \in \texttt{actions}_2(\mathcal{T}, Q)$, and so on. The functions $\texttt{fathom}$ and $\texttt{children}$ then depend on all d actions taken at node Q. We assume that there are two scoring rules $\texttt{ascore}_1^i$ and $\texttt{ascore}_2^i$ for each action type $i = 1, \ldots, d$. Algorithm 2 can then be parameterized by $(\boldsymbol{\mu}, \lambda)$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ is a vector of parameters controlling each action, so the ith action is selected to maximize $\mu_i \cdot \texttt{ascore}_1^i + (1 - \mu_i) \cdot \texttt{ascore}_2^i$. Then, as long as $d = O(1)$, we get the same pseudo-dimension bound. We assume b is a uniform upper bound on the size of $\texttt{actions}_i$ for any i.

Let $\texttt{actions}_1, \ldots, \texttt{actions}_d$ be path-wise. The multi-action version of Algorithm 2 is given by Algorithm 3.

There are two scoring rules $\texttt{ascore}_1^i$ and $\texttt{ascore}_2^i$ for each action type $i \in [d]$. Algorithm 3 can then be parameterized by $(\boldsymbol{\mu}, \lambda)$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ is a vector of parameters controlling each action: the ith action is selected to maximize $\mu_i \cdot \texttt{ascore}_1^i + (1 - \mu_i) \cdot \texttt{ascore}_2^i$. As before, we assume there are $b, k \in \mathbb{N}$ such that $|\texttt{actions}_i(\mathcal{T}, Q)| \le b$ for any i and any $Q \in \mathcal{T}$, and $|\texttt{children}(\mathcal{T}, Q, A_1, \ldots, A_d)| \le k$ for all $Q, A_1, \ldots, A_d$.

Let $\mathcal{A}'$, as in the single-action setting, be Algorithm 3 with the evaluations of $\texttt{fathom}$ on line 3 and line 3 suppressed. Then, we may prove a slight generalization of lemma 4.4.3.

Lemma 4.4.6. *Let $\texttt{ascore}_1^i$ and $\texttt{ascore}_2^i$ be two path-wise action-selection scores, for each $i \in \{1, \ldots, d\}$. Fix the input root node Q. There are $T \le k^{d\Delta(\Delta-1)/2} b^{d\Delta}$ boxes of the form $R_t = I_1 \times \cdots \times I_d$ partitioning $[0,1]^d$ where for any box R_t, the action-selection scores $\mu_i \cdot \texttt{ascore}_1^i + (1-\mu_i) \cdot \texttt{ascore}_2^i$ results in the same tree built by $\mathcal{A}'$ for all $\boldsymbol{\mu} \in R_t$, no matter what node selection policy is used.*

Proof. Let $\mathcal{T}$ denote the tree built by $\mathcal{A}'$. For $i \in [\Delta]$, let $\mathcal{T}[i]$ denote the restriction of $\mathcal{T}$ to nodes of depth at most i. Let $\texttt{ascore}_{\mu_i}^i = \mu_i \cdot \texttt{ascore}_1^i + (1-\mu_i) \cdot \texttt{ascore}_2^i$. We prove the lemma by induction on i. In particular, we show that for each $i \in [\Delta]$, there are $k^{di(i-1)/2} b^{di}$ boxes partitioning $[0,1]^d$ such that $\mathcal{T}[i]$ is invariant over all $\boldsymbol{\mu}$ within any given box. Since $\mathcal{T}[\Delta] = \mathcal{T}$, this implies the lemma statement. The base case of $i = 1$ is trivial since $\mathcal{T}[1]$ consists of only the root, regardless of $\boldsymbol{\mu} \in [0,1]^d$.

Now, suppose the statement holds for some $i \in \{1, \ldots, \Delta - 1\}$. That is, there are $T \le k^{di(i-1)/2} b^{di}$ disjoint boxes $R_1 \cup \cdots \cup I_R = [0,1]^d$ such that $\mathcal{T}[i]$ is invariant over all $\boldsymbol{\mu}$ within any given boxes (our inductive hypothesis). Fix one of these boxes R_t. We subdivide R_t into sub-boxes such that $\mathcal{T}[i+1]$ is invariant within each one of these smaller boxes. Let Q be any leaf of $\mathcal{T}[i]$, and for $\boldsymbol{\mu} \in R_t$ let $\mathcal{T}_{\boldsymbol{\mu}}$ denote the state of the tree using $\texttt{ascore}_{\mu_i}^i$ for each i at the point that Q is selected. Since $i < \Delta$, Q is not fathomed at line 4, regardless of $\boldsymbol{\mu}$. Next, since $\texttt{actions}_i$ is path-wise for each i, the actions available at Q depend only on the path $\mathcal{T}_Q$ from the root of $\mathcal{T}$ to Q, which, by the inductive hypothesis, is invariant over all $\boldsymbol{\mu} \in R_t$. That is, for all i $\texttt{actions}_i(\mathcal{T}_\mu, Q) = \texttt{actions}_i(\mathcal{T}_Q, Q)$ for all $\boldsymbol{\mu} \in R_t$. Then, $\texttt{ascore}_{\mu_i}^i$ will select action $A_i \in \texttt{actions}_i(\mathcal{T}_Q, Q)$ if and only if

$$
\begin{aligned}
A_i &= \underset{A_0 \in \texttt{actions}_i(\mathcal{T}_Q, Q)}{\operatorname{argmax}} \mu \cdot \texttt{ascore}_1^i(\mathcal{T}_{\boldsymbol{\mu}}, Q, A_0) + (1-\mu_i) \cdot \texttt{ascore}_2^i(\mathcal{T}_{\boldsymbol{\mu}}, Q, A_0) \\
&= \underset{A_0 \in \texttt{actions}_i(\mathcal{T}_Q, Q)}{\operatorname{argmax}} \mu_i \cdot \texttt{ascore}_1^i(\mathcal{T}_Q, Q, A_0) + (1-\mu_i) \cdot \texttt{ascore}_2^i(\mathcal{T}_Q, Q, A_0),
\end{aligned}
$$

where the second equality follows from the fact that $\texttt{ascore}_1^i$ and $\texttt{ascore}_2^i$ are path-wise. Thus, for a fixed A_0, $\texttt{ascore}_{\mu_i}^i$ is linear in μ_i, so for each A_0 there is at most one subinterval of $[0,1]$ such that for all μ_i in that subinterval, A_0 maximizes $\texttt{ascore}_{\mu_i}^i$. Thus, each leaf of $\mathcal{T}[i]$ contributes at most b subintervals such that for μ_i within a given subinterval, the action of type i selected at each leaf of $\mathcal{T}[i]$ is invariant. $\mathcal{T}[i]$ consists of at most k^i leaves, so this is a total of at most $k^i b$ subintervals. Writing $R_t = I_1 \times \cdots I_d$, we have established that for each i, there are $k^i b$ subintervals partitioning I_i into subintervals such that as μ_i varies over each subinterval, the action of type i selected at every leaf of $\mathcal{T}[i]$ is invariant. These subintervals partition R_t into at most $(k^i b)^d$ boxes. As before, since the actions selected at each leaf Q of $\mathcal{T}[i]$ are invariant, the set of children $\texttt{children}(\mathcal{T}_{\boldsymbol{\mu}}, Q, A_1, \ldots, A_d) = \texttt{children}(\mathcal{T}_Q, Q, A_1, \ldots, A_d)$ of Q added to the tree is also invariant, using the fact that $\texttt{children}$ is path-wise. Therefore, within every sub-box of R_t, $\mathcal{T}[i+1]$ is invariant. The total number of boxes over each possible R_t is, by the induction hypothesis, at most $k^{di(i-1)/2} b^{di} \cdot k^{di} b^d = k^{d(i+1)i/2} b^{d(i+1)}$. $\square$

The proof of Lemma 4.4.2 is identical in the multi-action setting. The proof of Lemma 4.4.4 is also identical: here, we fix a box R in the partition established in Lemma 4.4.6, and get an identical partition of $R \times [0,1]$ such that the behavior of Algorithm 3 is invariant as λ varies in each subinterval of $[0,1]$. The number of boxes in the final partition of $[0,1]^{d+1}$ is $k^{d\Delta(\Delta-1)/2}b^{d\Delta} \cdot k^{5\Delta} \leq k^{d\Delta(9+\Delta)}b^{d\Delta}$. Our main pseudo-dimension bound for the multi-action setting follows from the same argument that exploits the framework of Balcan et al. [2021a].

Theorem 4.4.7. *Let $\mathsf{cost}(Q)$ be any tree-constant cost function, and let $\mathsf{cost}_{\boldsymbol{\mu},\lambda}(Q)$ be the cost of the tree built by Algorithm 2 on input root node Q using action-selection scores parameterized by $\boldsymbol{\mu} \in \mathbb{R}^d$, where $d = O(1)$, and node-selection score parameterized by λ. Then,* $\mathsf{Pdim}(\{\mathsf{cost}_{\boldsymbol{\mu},\lambda}\}) = O(d\Delta^2 \log k + d\Delta \log b)$.

When $d = O(1)$ we get the same pseudo-dimension bound as in the single-action setting: $\mathsf{Pdim}(\{\mathsf{cost}_{\boldsymbol{\mu},\lambda}\}) = O(\Delta^2 \log k + \Delta \log b)$.

4.4.4 Branch-and-cut for integer programming

We now instantiate our main results with the three main components of the B&C algorithm: branching, cutting planes, and node selection, used to solve IPs $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq 0, \boldsymbol{x} \in \mathbb{Z}^n\}$ where $\boldsymbol{c} \in \mathbb{R}^n$, $A \in \mathbb{Z}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$. The function $\texttt{fathom}(\mathcal{T}, Q, A)$ outputs $\texttt{true}$ if after having taken action A the LP relaxation at Q is integral, infeasible, or worse than the best integral solution found so far in $\mathcal{T}$. The function $\texttt{children}(\mathcal{T}, Q, A)$ outputs the two subproblems generated by the branching procedure on the IP at Q after having taken action A. For simplicity we refer only to IPs, but everything in our discussion applies to mixed IPs as well. In our model of tree search, node selection is controlled by λ. Cutting planes and branching are types of actions and controlled by μ.

Branching

In this section, we provide guarantees for branching. Throughout this section we assume $\Delta = O(n)$, as is the case with single-variable branching.

Multivariable branching constraints It is well known that allowing for more general generation of branching constraints can result in smaller B&C trees. Gilpin and Sandholm [2011] studied multivariable branches of the form $\sum_{i\in S} \boldsymbol{x}[i] \leq \left\lfloor \sum_{i\in S} \boldsymbol{x}^*_{\mathrm{LP}}[i] \right\rfloor$, $\sum_{i\in S} \boldsymbol{x}[i] \geq \left\lceil \sum_{i\in S} \boldsymbol{x}^*_{\mathrm{LP}}[i] \right\rceil$ where S is a subset of the integer variables such that $\sum_{i\in S} \boldsymbol{x}^*_{\mathrm{LP}}[i] \notin \mathbb{Z}$. Here, $\texttt{actions}(\mathcal{T}, Q) = 2^{[n]}$, so, $\mathsf{Pdim}(\{\mathsf{cost}_{\mu,\lambda}\}) = O(n^2)$. So our sample complexity bound for multivariable branching constraints is, surprisingly, only a constant factor worse than the bound for single-variable branching constraints.

We give a simple example where B&C using only single variable branches builds a tree of exponential size, while a single branch on the entire set of variables at the root yields two infeasible subproblems (and a B&C tree of size 3).

Theorem 4.4.8. *For any n, there is an IP with two constraints and n variables such that with only single variable branches, B&C builds a tree of size $2^{(n-1)/2}$, while with a suitable multivariable branch, B&C builds a tree of size three.*

Proof. Let n be an odd positive integer. Consider the infeasible IP $\max\{\sum_{i=1}^n x[i] : 2\sum_{i=1}^n x[i] = n, \boldsymbol{x} \in \{0,1\}^n\}$. Jeroslow [1974] proved that with only single-variable branches, B&C builds a tree with $2^{(n-1)/2}$ nodes to determine infeasibility. However, with a suitable multivariable branch, B&C will build a tree of constant size. The optimal solution to the LP relaxation of the IP is attained when all variables are set to $1/2$. A multivariable branch on all n variables produces the two subproblems with constraints $\sum_{i=1}^n x[i] \leq \lfloor n/2 \rfloor$ and $\sum_{i=1}^n x[i] \geq \lceil n/2 \rceil$, respectively. Since n is odd, $\lfloor n/2 \rfloor < n/2$ and $\lceil n/2 \rceil > n/2$, so the LP relaxations of both subproblems are infeasible. Thus, B&C builds a tree with three nodes. □

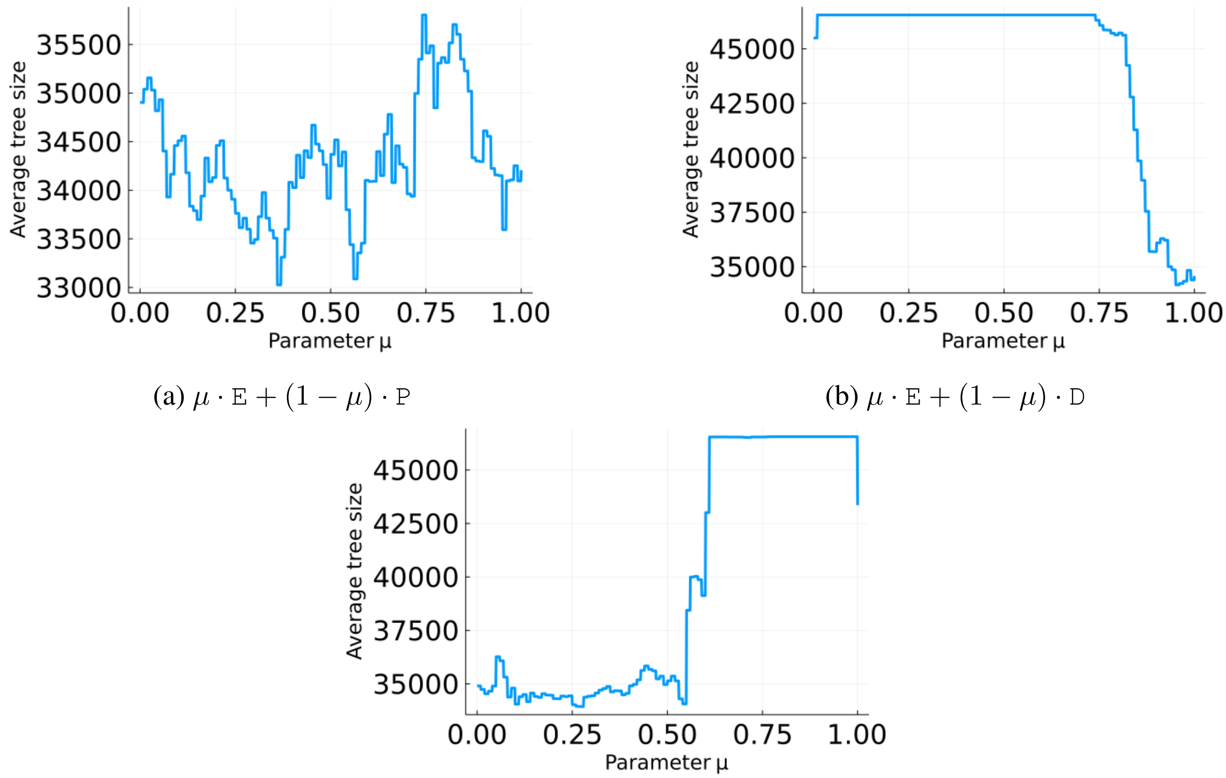
Yang et al. [2021] provide more examples of situations where multivariable branching yields dramatic improvements in tree size over single variable branching. They also perform a computational evaluation of a few different strategies for generating multivariable branching constraints. Yang et al. [2020] explore gradient-boosting for learning to mimic strong branching for multiple variables.

Branching on general disjunctions Branching constraints can be even more general than multivariable branches. Given any integer vector $\boldsymbol{\pi} \in \mathbb{Z}^n$ and any integer $\pi_0 \in \mathbb{Z}$ (jointly referred to as a *disjunction*), the constraints $\boldsymbol{\pi}^\top \boldsymbol{x} \leq \pi_0$ or $\boldsymbol{\pi}^\top \boldsymbol{x} \geq \pi_0 + 1$ represent a valid partition of the feasible region into subproblems. Owen and Mehrotra [2001] ran the first experiments demonstrating that branching on general disjunctions can lead to significantly smaller tree sizes. Subsequent works have posed different heuristics to select disjunctions to branch on [Fischetti and Lodi, 2002, Mahajan and Ralphs, 2009].

In practice it is known that additional IP constraints should not have coefficients that are too large. If C is a bound on the magnitude of the coefficient of any disjunction, then $\texttt{actions}(\mathcal{T}, Q) = \{-C, \ldots, C\}^{n+1}$, so $\mathsf{Pdim}(\{\mathsf{cost}_{\mu,\lambda}\}) = O(n^2 \log C)$. Karamanov and Cornuéjols [2011] conduct a computational evaluation of disjunctions derived from Gomory mixed-integer cuts. In this setting, $\texttt{actions}(\mathcal{T}, Q)$ is the set of m or fewer disjunctions corresponding to the m or fewer Gomory mixed-integer cuts derived from the simplex tableau from solving the LP relaxation of Q. In this case, $\mathsf{Pdim}(\{\mathsf{cost}_{\mu,\lambda}\}) = O(n^2 + n \log m)$.

Cutting planes

The action set can also correspond to cutting planes used to refine the feasible region of the IP at any stage of B&C. Here, $\texttt{actions}(\mathcal{T}, Q)$ is any set of cutting planes derived solely using the path from the root to the IP at Q. Examples include the set of Chvátal-Gomory (CG) derived from the simplex tableau [Gomory, 1958], and various combinatorial families of cutting planes such as clique cuts, odd-hole cuts, and cover cuts. The set $\texttt{actions}(\mathcal{T}, Q)$ can also consist of sequences of cutting planes, representing adding several cutting planes to the IP in waves. For example, the set of all sequences of w CG cuts generated from the simplex tableau for an IP with m constraints has size at most m^w (regardless of whether the LP is resolved after each cut). The number of such cutting planes provided by the LP tableau at any node in the tree is at most $O(m + nw)$ (the original IP has m constraints, and after at most n branches there are an additional n branching constraints and at most nw cutting planes), which means that $|\texttt{actions}(\mathcal{T}, Q)| \leq O(m + nw)^w$. Thus, $\mathsf{Pdim}(\{\mathsf{cost}_{\mu,\lambda}\}) = O(n^2 + nw \log(m + nw))$.



(a) $\mu \cdot \mathtt{E} + (1 - \mu) \cdot \mathtt{P}$

(b) $\mu \cdot \mathtt{E} + (1 - \mu) \cdot \mathtt{D}$

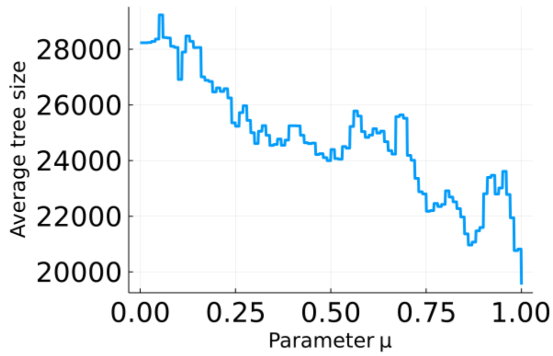
(c) $\mu \cdot \mathtt{D} + (1 - \mu) \cdot \mathtt{P}$

Figure 4.1: Chvátal distribution with 35 items and 2 knapsacks.

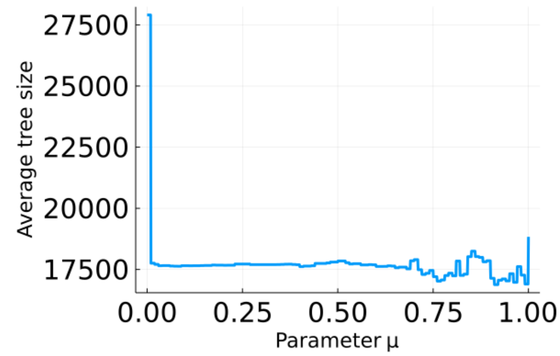
We can also handle arbitrary CG cuts (not just ones from the LP tableau). Balcan et al. [2021d] proved that given an IP with feasible region $\{\boldsymbol{x} \in \mathbb{Z}^n : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq 0\}$, even though there are infinitely many CG cut parameters, there are effectively only $O(w2^w \|A\|_{1,1} + 2^w \|\boldsymbol{b}\|_1 + nw)^{1+mw}$ distinct sequences of cutting planes that w CG cut parameters can produce. At any node in the B&C tree, the number of constraints is at most $O(m + nw)$. So, on the domain of IPs with $\|A\|_{1,1} \leq \alpha$ and $\|\boldsymbol{b}\|_1 \leq \beta$, $|\mathtt{actions}(\mathcal{T}, Q)| \leq O(w2^w\alpha + 2^w\beta + nw)^{1+w\cdot O(m+nw)}$. Thus, $\mathsf{Pdim}(\{\mathsf{cost}_{\mu,\lambda}\}) = O(n^2 w^3 m \log(\alpha + \beta + n))$.

Experiments on cover cuts for the multiple knapsack problem

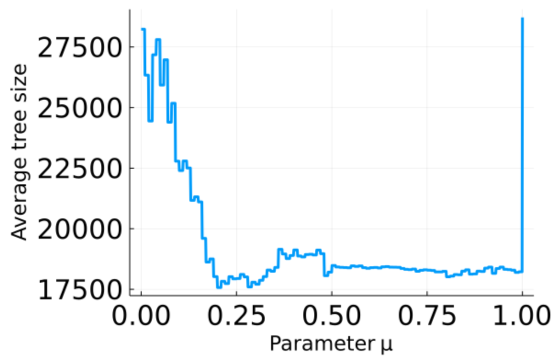
In this section, we demonstrate via experiments that tuning a convex combination of scoring rules to select cuts can lead to dramatically smaller branch-and-cut trees when done in a data-dependent manner. We study the classical NP-hard *multiple knapsack problem*: given a set N of items where each item $i \in N$ has a value $p_i \geq 0$ and a weight $w_i \geq 0$, and a set K of knapsacks where each knapsack $k \in K$ has a capacity $W_k \geq 0$, the goal is to find a feasible packing of the items into the knapsacks of maximum value. We assume, without loss of generality, that the items are labeled in descending order of weight, that is, $w_1 \geq w_2 \geq \cdots \geq w_{|N|}$. This problem



(a) $\mu \cdot \texttt{E} + (1-\mu) \cdot \texttt{P}$



(b) $\mu \cdot \texttt{E} + (1-\mu) \cdot \texttt{D}$



(c) $\mu \cdot \texttt{D} + (1-\mu) \cdot \texttt{P}$

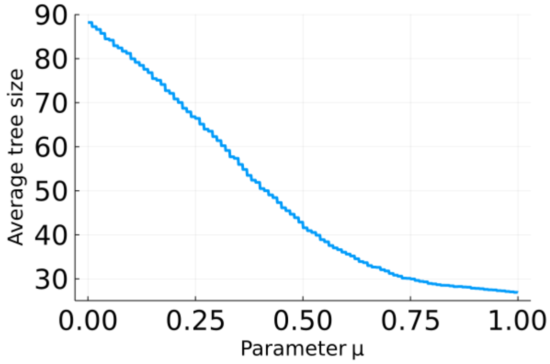
Figure 4.2: Chvátal distribution with 35 items and 3 knapsacks.

can be formulated as the following binary IP:

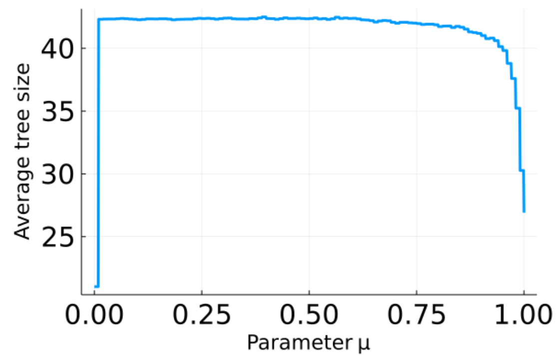
$$\begin{array}{ll} \text{maximize} & \sum_{i \in N} \sum_{k \in K} p_i x_{k,i} \\ \text{subject to} & \sum_{i \in N} w_i x_{k,i} \leq W_k \qquad \forall\, k \in K \\ & \sum_{k \in K} x_{k,i} \leq 1 \qquad\qquad \forall\, i \in N \\ & x_{k,i} \in \{0,1\} \qquad \forall\, i \in N, k \in K \end{array}$$

Recall from Chapter 3 that a subset $C \subseteq N$ of items is called a *cover* for knapsack $k \in K$ if $\sum_{i \in C} w_i > W_k$. If C is a cover, no feasible solution can have $x_{k,i} = 1$ for all $i \in C$, so $\sum_{i \in C} x_{k,i} \leq |C| - 1$ is a valid constraint—called a *cover cut*. When C is minimal (that is, $C \setminus \{i\}$ is not a cover for every $i \in C$), such cover cuts help tighten the knapsack IP by cutting off fractional LP solutions. We generate (a subset of all) cover cuts for each knapsack k as follows: for each $i \in N$, let $j > i$ be minimal such that $C = \{i, i+1, \ldots, j\}$ is a cover for k (if such a j exists). Since $w_i \geq w_j$ for $j > i$, C is a minimal cover, and moreover the *extended cover cut* $\sum_{i=1}^{j} x_i \leq |C| - 1$ is valid and dominates the minimal cover cut $\sum_{i \in C} x_i \leq |C| - 1$. Extended cover cuts generated from minimal covers are known to be facet defining for the integer hull under certain natural conditions [Conforti et al., 2014] (though these are in general a more limited/weaker family of cuts than those obtained via lifting in Prasad et al. [2024] covered in Chapter 3).

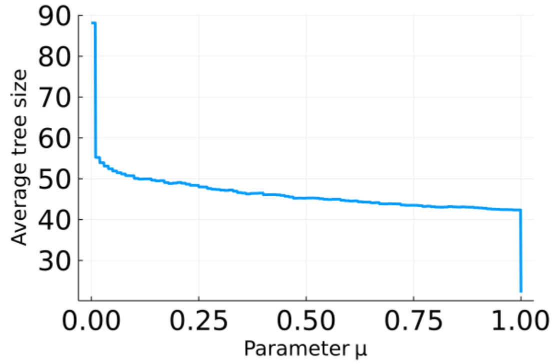
We investigate the relationship between three scoring rules for cutting planes. The first is



(a) $\mu \cdot \texttt{E} + (1-\mu) \cdot \texttt{P}$



(b) $\mu \cdot \texttt{E} + (1-\mu) \cdot \texttt{D}$



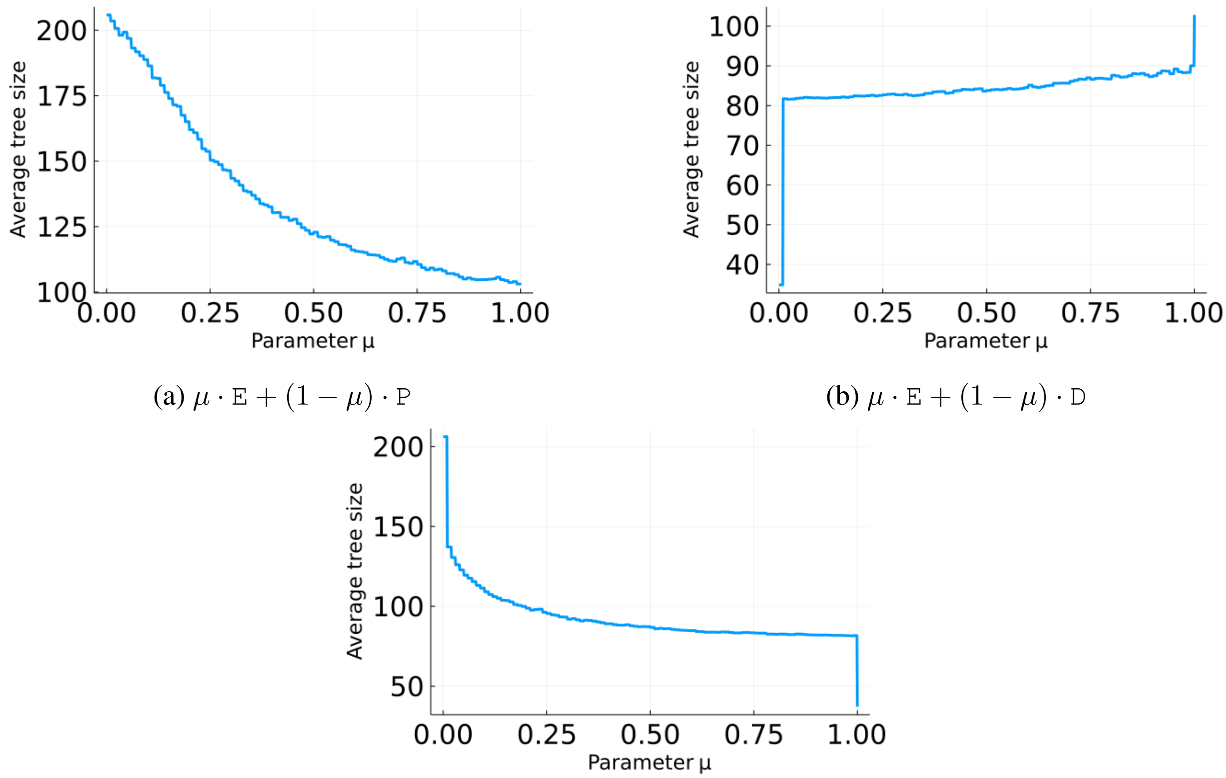
(c) $\mu \cdot \texttt{D} + (1-\mu) \cdot \texttt{P}$

Figure 4.3: Reverse Chvátal distribution with 100 items and 10 knapsacks.

efficacy ($\texttt{E}$), which is the perpendicular distance from the current LP solution to the cutting plane. The second is *parallelism* ($\texttt{P}$), which measures the angle between the objective and the normal vector to the cutting plane. The third is *directed cutoff* ($\texttt{D}$), which is the distance from the current LP solution to the cutting plane along the direction of the line segment connecting the LP solution to the current best incumbent integer solution. More details, including explicit formulas, can be found in Balcan et al. [2021d] and references therein.

We consider two specific instances of the multiple knapsack problem, which are loosely based on a class of knapsack problems introduced by Chvátal that are difficult to solve with vanilla branch-and-bound [Chvátal, 1980, Yang et al., 2021]. In the first, $p_i = w_i$ for all $i \in N$, and $W_k = \lfloor (\sum_{i \in N} w_i)/2|K| \rfloor + (k-1)$ for each $k = 1, \ldots, |K|$. In the second, $p_i = w_{|N|-i+1}$, so the most valuable item is the lightest and the least valuable item is the heaviest, and W_k is defined as in the first type. We call the first class of problems *Chvátal instances* and the second class *reverse Chvátal instances*. For a given N, K, we generate (reverse) Chvátal instances by drawing each weight independently as $w_i = \lfloor z_i \rfloor$, where $z_i \sim \mathcal{N}(50, 2)$, and sorting the items by weight in descending order.

In our experiments, we add (whenever possible) two extended cover cuts obtained in the aforementioned manner at every node of the B&C tree. The two cuts chosen are the two with the highest score $\mu \cdot \texttt{ascore}_1 + (1-\mu) \cdot \texttt{ascore}_2$ among all extended cover cuts that are violated by the current LP optimum, where $\texttt{ascore}_1, \texttt{ascore}_2 \in \{\texttt{E}, \texttt{D}, \texttt{P}\}$. Figures 4.1-4.4 display



(a) $\mu \cdot \texttt{E} + (1-\mu) \cdot \texttt{P}$

(b) $\mu \cdot \texttt{E} + (1-\mu) \cdot \texttt{D}$

(c) $\mu \cdot \texttt{D} + (1-\mu) \cdot \texttt{P}$

Figure 4.4: Reverse Chvátal distribution with 100 items and 15 knapsacks.

the average tree size over 1000 samples for different Chvátal and reverse Chvátal distributions as a function of μ, where the domain $[0, 1]$ of μ is discretized in increments of 0.01. We ran our experiments using the Python API of CPLEX 12.10 with default cut generation turned off. All other aspects of B&C (e.g. variable and node selection) are controlled by the default settings of CPLEX. The key takeaway of our plots is that tuning a convex combination of scoring rules can lead to significant savings in B&C tree size, and that this tuning must be done with the IP distribution in mind. No single parameter produces small trees for all the distributions we considered, and in fact a μ that minimizes tree size for one distribution can result in the largest trees for another (as in Figures 4.2b and 4.4b, for example). Furthermore, many of the plots display discernible trends (and in some cases are quite smooth), suggesting that the number of samples required to avoid overfitting in practice can be significantly smaller than our theoretical bounds.

4.4.5 Improved bounds for branch-and-cut

To allow node selection, branching, and cutting-plane selection to be tuned simultaneously, we apply Theorem 4.4.7. This allows us to bound the pseudo-dimension of the family of functions $\{\mathsf{cost}_{\mu_1,\mu_2,\lambda}\}$, where μ_1 controls branching, μ_2 controls cutting-plane selection, and λ controls node selection. Let $\texttt{actions}_1(\mathcal{T}, Q)$ denote the set of branching actions available at Q, and let

$\texttt{actions}_2(\mathcal{T}, Q)$ denote the set of cutting planes available at Q. Let $b_1, b_2 \in \mathbb{N}$ be such that $\texttt{actions}_1(\mathcal{T}, Q) \leq b_1$ and $\texttt{actions}_2(\mathcal{T}, Q) \leq b_2$ for all $\mathcal{T}$ and all $Q \in \mathcal{T}$. Fix two branching scores $\texttt{ascore}_1^1, \texttt{ascore}_2^1$, fix two cutting-plane selection scores $\texttt{ascore}_1^2, \texttt{ascore}_2^2$, and fix two node-selection scores $\texttt{nscore}_1, \texttt{nscore}_2$.

Theorem 4.4.9. *Let $\mathsf{cost}(Q)$ be any tree-constant cost function, and let $\mathsf{cost}_{\mu_1,\mu_2,\lambda}$ be the cost of the tree built by B&C using branching score $\mu_1 \cdot \texttt{ascore}_1^1 + (1-\mu_1) \cdot \texttt{ascore}_2^1$, cutting-plane selection score $\mu_2 \cdot \texttt{ascore}_1^2 + (1-\mu_2) \cdot \texttt{ascore}_2^2$, and node-selection score $\lambda \cdot \texttt{nscore}_1 + (1-\lambda) \cdot \texttt{nscore}_2$. Then, with $\Delta = O(n)$, $\mathsf{Pdim}(\{\mathsf{cost}_{\mu_1,\mu_2,\lambda}\}) = O(n^2 + n\log(b_1+b_2))$.*

Comparison to existing bounds

Balcan et al. [2021d], Vitercik [2021] give a pseudo-dimension bound for tree search with a linear dependence on a cap κ on the number of nodes allowed in any tree. Their pseudo-dimension bound in our setting is $\mathsf{Pdim}(\{\mathsf{cost}_{\mu_1,\mu_2,\lambda}\}) = O(\kappa\log\kappa + \kappa\log b_1 + \kappa\log b_2)$. While κ is treated as a constant, it can be a prohibitively large quantity. In fact, without explicitly enforcing a limit on the number of nodes expanded by B&C, Balcan et al. [2021d], Vitercik [2021] obtain a pseudo-dimension bound of $O(2^n(\log b_1 + \log b_2))$. Balcan et al. [2018a] use the path-wise property to prove that $\mathsf{Pdim}(\{\mathsf{cost}_\mu\}) = O(n^2)$ for single-variable branching, but for the case where branching is the only tunable component of B&C (and node selection is fixed).

4.4.6 Conclusions and future research

We presented a general model of tree search and proved sample complexity guarantees for this model that improve and generalize upon the recent sample complexity theory for configuring branch-and-cut. There are many interesting and open directions for future research. One compelling open question is to obtain pseudo-dimension bounds when action sets are infinite. Balcan et al. [2021d] alluded to this question in the case of cutting planes, and neither the techniques of their work nor the techniques of the present work can handle, for example, important infinite cutting-plane families such as the class of Gomory mixed-integer cuts, or the infinitely many valid disjunctions that could be branched on. Beyond integer programming, our model of tree search could potentially be applied to completely different problem domains that exhibit tree structure. Another direction is to extend our results to convex combinations of $\ell > 2$ scoring rules $\mu_1\texttt{score}_1 + \ldots \mu_\ell\texttt{score}_\ell$, as Balcan et al. [2021d] do in the special case of single-variable branching. However, their pseudo-dimension bound grows exponentially in the number of variables n in that special case; developing techniques that lead to a polynomial dependence on n remains a challenging open question.

4.5 Structural Analysis of Branch-and-Cut and the Learnability of Gomory Mixed-Integer Cuts

The class of *Gomory mixed-integer (GMI) cuts* are one of the most important family of cutting planes used in integer programming solvers. Bixby et al. [1999] and, more recently, Achterberg and Wunderling [2013] provide metrics reporting the outsized performance improvement of solvers due to GMI cuts. GMI cuts were furthermore the first class of general-purpose cuts that enabled a successful software implementation of branch-and-cut, by Balas et al. [1996b].

In this chapter, we study data-dependent GMI cut configuration. Figure 4.5 illustrates that tuning GMI cut selection parameters according to the instance distribution at hand can have a large impact on B&C's performance, and that for one distribution, the best parameters can be very different—in fact opposite—than the best parameters for another distribution.

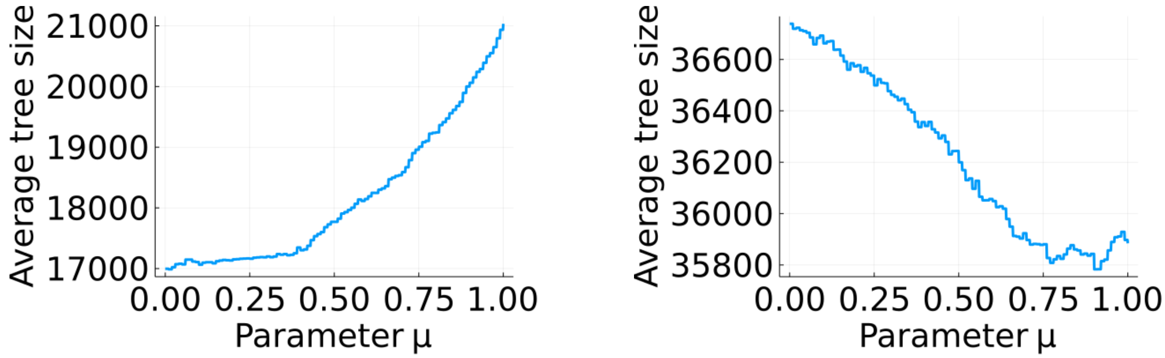
Prior sample complexity work (including primarily the work presented in the previous two chapters) has been unable to handle GMI cuts because there are an uncountably infinite number of different GMI cuts that one could add, whereas the techniques developed in those chapters were only able to handle cutting plane families of finite effective size. The current work closes this gap.

The key challenge is that an infinitesimal change to any GMI cut can completely change the entire course of B&C because a cut added at the root remains in the LP relaxations stored in each node all the way to the leaves. At its core, our analysis therefore involves understanding an intricate interplay between the continuous and discrete components of our problem. The first, continuous component requires us to characterize how an LP's solution changes as a function of its constraints. The optimum will move continuously through space until it jumps from one vertex of the polytope to another. We use this characterization to analyze how the B&C tree—a discrete, combinatorial object—varies as a function of its LP guidance, which allows us to prove our sample complexity bound.

Contributions

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.

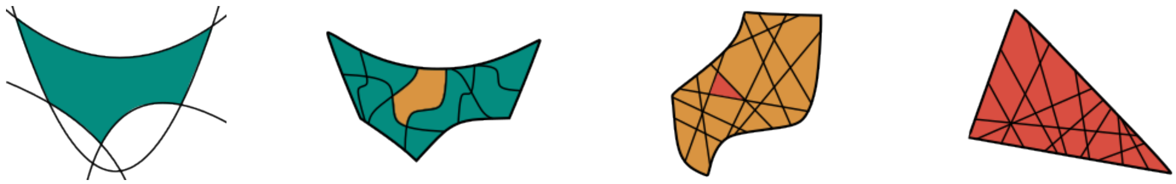


(a) Facility location with 40 locations and 40 clients; sampled by perturbing a base facility location IP.

(b) Facility location with 80 locations, 80 clients, and random Euclidean distance costs.

Figure 4.5: These figures illustrate the need for distribution-dependent policies for choosing cuts. We plot the average number of nodes B&C expands as a function of a parameter μ that controls a policy to add GMI cuts, detailed in Appendix B. In each figure, we draw a training set of facility location IPs from two different distributions. In Figure 4.5a, we define the distribution by starting with a uniformly random facility location instance and perturbing its costs. In Figure 4.5b, the costs are more structured: the facilities are located along a line and the clients have uniformly random locations. In Figure 4.5a, a smaller value of μ leads to small search trees, but in Figure 4.5b, a larger value of μ is preferable.

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 4.6:
   (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 4.6a.
   (b) We use this result to show that each region can be further partitioned (as illustrated in Figure 4.6b) 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 4.6b can be further partitioned into regions (illustrated in Figure 4.6c) 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 4.6d) 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



(a) LP optimum closed form

(b) Invariant branching

(c) Invariant LP integrality

(d) Invariant B&C execution

Figure 4.6: Our branch-and-cut analysis involves successive refinements to our partition of the parameter space.

insight is that the GMI cutting plane coefficients can be viewed as a mapping that embeds our polynomial partition from the previous step (Figure 4.6) 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.

Notation and Prerequisite Results

We consider *pure* integer programs given by objective $\boldsymbol{c} \in \mathbb{R}^n$, constraint matrix $A \in \mathbb{Z}^{m \times n}$, and constraint vector $\boldsymbol{b} \in \mathbb{Z}^m$, of the form

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{x} \in \mathbb{Z}^n\}. \tag{4.2}$$

The *linear programming (LP) relaxation* is formed by removing the integrality constraints: $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$. We denote the optimal solution to (4.2) by $\boldsymbol{x}^*_{\mathrm{IP}}$ and its LP-relaxation optimal solution by $\boldsymbol{x}^*_{\mathrm{LP}}$. Let $z^*_{\mathrm{LP}} = \boldsymbol{c}^\top \boldsymbol{x}^*_{\mathrm{LP}}$. If σ is a set of constraints, we let $\boldsymbol{x}^*_{\mathrm{IP}}(\sigma)$ denote the optimum of (4.2) subject to these additional constraints (similarly define $z^*_{\mathrm{LP}}(\sigma)$ and $\boldsymbol{x}^*_{\mathrm{LP}}(\sigma)$).

Polyhedra and polytopes. A set $\mathcal{P} \subseteq \mathbb{R}^n$ is a *polyhedron* if there exists an integer m, $A \in \mathbb{R}^{m \times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$ such that $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$. $\mathcal{P}$ is a *rational polyhedron* if there exists $A \in \mathbb{Z}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$ such that $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$. A bounded polyhedron is called a *polytope*. The feasible regions of all IPs considered in this chapter are assumed to be rational polytopes [1] of full dimension. Let $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^i\boldsymbol{x} \leq b_i, i \in M\}$ be a nonempty polyhedron. We assume the representation of $\mathcal{P}$ is *irredundant*, that is, $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^i\boldsymbol{x} \leq b_i, i \in M \setminus \{j\}\} \neq \mathcal{P}$ for all $j \in M$. For any $I \subseteq M$, the set $F_I := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^i\boldsymbol{x} = b_i, i \in I, \boldsymbol{a}^i\boldsymbol{x} \leq b_i, i \in M \setminus I\}$ is a *face* of $\mathcal{P}$. Conversely, if F is a nonempty face of $\mathcal{P}$, then $F = F_I$ for some $I \subseteq M$. Faces of dimension 1 are called *edges* and faces of dimension 0 are called *vertices*. A detailed reference on the polyhedral theory used in our arguments can be found in Conforti et al. [2014].

Given a set of constraints σ, let $\mathcal{P}(\sigma)$ denote the polyhedron that is the intersection of $\mathcal{P}$ with all inequalities in σ.

[1] This assumption is not a restrictive one. The Minkowski-Weyl theorem states that any polyhedron can be decomposed as the sum of a polytope and its recession cone. All results in this chapter can be derived for rational polyhedra by considering the corresponding polytope in the Minkowski-Weyl decomposition.

Gomory mixed-integer cuts. We denote a cutting plane in this chapter by the constraint $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$. Let $\mathcal{P}$ be the feasible region of the LP relaxation of (4.2) and $\mathcal{P}_{\mathrm{I}} = \mathcal{P} \cap \mathbb{Z}^n$ be the IP's feasible set. Recall a cut is *valid* if it is satisfied by every integer point in $\mathcal{P}_{\mathrm{I}}$: $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ for all $\boldsymbol{x} \in \mathcal{P}_{\mathrm{I}}$. A valid cut *separates* a point $\boldsymbol{x} \in \mathcal{P} \setminus \mathcal{P}_{\mathrm{I}}$ if $\boldsymbol{\alpha}^\top \boldsymbol{x} > \beta$. We refer to a cut both by its parameters $(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^{n+1}$ and the halfspace $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ in $\mathbb{R}^n$. An important family of valid cuts that we study in this chapter is the set of *Gomory mixed integer (GMI) cuts*. For decades, general-purpose cutting planes were thought to be unwieldy and useless for solving IPs quickly in practice. However, a seminal paper by Balas et al. [1996b] completely reversed this sentiment by showing that GMI cuts added throughout the B&C tree led to massive speedups. Today, GMI cuts are one of the most important components of state-of-the-art IP solvers.

Definition 4.5.1 (Gomory mixed integer cut)**.** Suppose the feasible region of the IP is in equality form $A\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \ge \boldsymbol{0}$ (which can be achieved by adding slack variables). For $\boldsymbol{u} \in \mathbb{R}^m$, let f_i denote the fractional part of $(\boldsymbol{u}^\top A)_i$ and let f_0 denote the fractional part of $\boldsymbol{u}^\top \boldsymbol{b}$. That is, $(\boldsymbol{u}^\top A)_i = (\lfloor \boldsymbol{u}^\top A \rfloor)_i + f_i$ and $\boldsymbol{u}^\top \boldsymbol{b} = \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor + f_0$. The *Gomory mixed integer (GMI) cut* parameterized by $\boldsymbol{u}$ is

$$\sum_{i: f_i \le f_0} f_i x_i + \frac{f_0}{1 - f_0} \sum_{i: f_i > f_0} (1 - f_i) x_i \ge f_0.$$

The form of the GMI cut is obtained via a slightly more nuanced rounding procedure than the one used to obtain the CG cut $\lfloor \boldsymbol{u}^\top A \rfloor \boldsymbol{x} \le \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor$. GMI cuts strictly dominate CG cuts. More details about GMI cuts can be found in the tutorial by Cornuéjols [2008].

Every step of B&C—including node and variable selection and the choice of whether or not to fathom—depends crucially on guidance from LP relaxations. Tighter LP relaxations provide more valuable LP guidance, highlighting the importance of cuts.

Polynomial arrangements in Euclidean space. Let $p \in \mathbb{R}[y_1, \dots, y_k]$ be a polynomial of degree at most d. The polynomial p partitions $\mathbb{R}^k$ into connected components that belong to either $\mathbb{R}^k \setminus \{(y_1, \dots, y_k) : p(y_1, \dots, y_k) = 0\}$ or $\{(y_1, \dots, y_k) : p(y_1, \dots, y_k) = 0\}$. When we discuss the connected components of $\mathbb{R}^k$ induced by p, we include connected components in both these sets. We make this distinction because previous work on sample complexity for data-driven algorithm design oftentimes only needed to consider the connected components of the former set. The number of connected components in both sets is $O(d^k)$ [Warren, 1968, Milnor, 1964, Thom, 1965].

The main distinction between our analysis in this chapter and the techniques used in the previous sections can be summarized as follows. Let $\boldsymbol{\mu}$ be a (potentially multidimensional) parameter controlling some aspect of the IP solver (e.g. a mixture parameter between branching rules or a cutting-plane parameter). In previous works, as $\boldsymbol{\mu}$ varied, there were only a finite number of states each node of branch-and-cut could be in. For example, in the case of branching/variable selection, $\boldsymbol{\mu}$ controls the additional branching constraint added to the IP at any given node of the search tree. There are only finitely many possible branching constraints, so there are only finitely many possible "child" IPs induced by $\boldsymbol{\mu}$. Similarly, if $\boldsymbol{\mu}$ represents the parameterization for Chvátal-Gomory cuts [Chvátal, 1973, Gomory, 1958], since Balcan et al. [2021d] (Section 4.3) showed that there are only finitely many distinct Chvátal-Gomory cuts for a given

IP, as $\boldsymbol{\mu}$ varies, there are only finitely many possible child IPs induced by $\boldsymbol{\mu}$ at any stage of the search tree. However, in many settings, this property does not hold. For example if $\boldsymbol{\mu} = (\boldsymbol{\alpha}, \beta)$ controls the normal vector and offset of an additional feasible constraint $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$, there are infinitely many possible IPs corresponding to the choice of $(\boldsymbol{\alpha}, \beta)$. Similarly, if $\boldsymbol{\mu}$ controls the parameterization of a GMI cut, there are infinitely many IPs corresponding to the choice of $\boldsymbol{\mu}$ (unlike Chvátal-Gomory cuts). in this chapter, we develop a new structural understanding of B&C that is significantly more involved than the structural results in prior work.

4.5.1 Linear programming sensitivity

Our main result in this section addresses a fundamental question in linear programming: how is an LP's optimal solution affected by the addition of new constraints? Later in this chapter, we use this result to prove sample complexity bounds for optimizing over the canonical family of GMI cuts.

More formally, fixing an LP with m constraints and n variables, if $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta) \in \mathbb{R}^n$ denotes the new LP optimum when the constraint $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ is added, we pin down a precise characterization of $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta)$ as a function of $\boldsymbol{\alpha}$ and β. We show that $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta)$ has a piece-wise closed form: there are surfaces partitioning $\mathbb{R}^{n+1}$ such that within each connected component induced by these surfaces, $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta)$ has a closed form. While the geometric intuition used to establish this piece-wise structure relies on the basic property that optimal solutions to LPs are achieved at vertices, the surfaces defining the regions are perhaps surprisingly nonlinear: they are defined by multivariate degree-2 polynomials in $\boldsymbol{\alpha}, \beta$.

The proof requires us to: (1) track the set of edges of the LP polytope intersected by the new constraint, and once those edges are fixed, (2) track which edge yields the vertex with the highest objective.

Let $M = [m]$ denote the set of m constraints. For $E \subseteq M$, let $A_E \in \mathbb{R}^{|E| \times n}$ and $\boldsymbol{b}_E \in \mathbb{R}^{|E|}$ denote the restrictions of A and $\boldsymbol{b}$ to E. For $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\beta \in \mathbb{R}$, and $E \subseteq M$ with $|E| = n - 1$, let $A_{E,\boldsymbol{\alpha}} \in \mathbb{R}^{n \times n}$ denote the matrix obtained by adding row vector $\boldsymbol{\alpha}$ to A_E and let $A^i_{E,\boldsymbol{\alpha},\beta} \in \mathbb{R}^{n \times n}$ be the matrix $A_{E,\boldsymbol{\alpha}}$ with the ith column replaced by $(\boldsymbol{b}_E, \beta)^\top$.

Theorem 4.5.2. *Let $(\boldsymbol{c}, A, \boldsymbol{b})$ be an LP with optimal solution $\boldsymbol{x}^*_{\mathrm{LP}}$. There are at most m^n hyperplanes and m^{2n} degree-2 polynomial hypersurfaces partitioning $\mathbb{R}^{n+1}$ into connected components such that for each component C, either: (1) $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta) = \boldsymbol{x}^*_{\mathrm{LP}}$, or (2) there is a set of constraints $E \subseteq M$ with $|E| = n - 1$ such that $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta)[i] = \det(A^i_{E,\boldsymbol{\alpha},\beta}) / \det(A_{E,\boldsymbol{\alpha}})$ for all $(\boldsymbol{\alpha}, \beta) \in C$.*

Proof. First, if $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ does not separate $\boldsymbol{x}^*_{\mathrm{LP}}$, then $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta) = \boldsymbol{x}^*_{\mathrm{LP}}$. The set of all such cuts is the halfspace given by $\{(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^{n+1} : \boldsymbol{\alpha}^\top \boldsymbol{x}^*_{\mathrm{LP}} \le \beta\}$. All other cuts separate $\boldsymbol{x}^*_{\mathrm{LP}}$ and thus pass through $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} \ge \boldsymbol{0}\}$, and the new LP optimum is achieved at a vertex created by the cut. We consider the new vertices formed by the cut, which lie on edges of $\mathcal{P}$. Each edge e of $\mathcal{P}$ can be identified with a subset $E \subset M$ of size $n - 1$ such that the edge is the set of all points $\boldsymbol{x}$ such that $\boldsymbol{a}_i^\top \boldsymbol{x} = b_i$ for all $i \in E$ and $\boldsymbol{a}_i^\top \boldsymbol{x} \le b_i$ for all $i \in M \setminus E$ where $\boldsymbol{a}_i$ is the ith row of A. If we drop the inequality constraints defining the edge, the equality constraints define a line in $\mathbb{R}^n$. The intersection of the cut $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ and this line is the solution to the system of n linear equations in n variables: $A_E \boldsymbol{x} = \boldsymbol{b}_E, \boldsymbol{\alpha}^\top \boldsymbol{x} = \beta$. By Cramer's rule, the unique solution $\boldsymbol{x} = (x_1, \ldots, x_n)$ to this system is given by $x_i = \det(A^i_{E,\boldsymbol{\alpha},\beta})/\det(A_{E,\boldsymbol{\alpha}})$. To ensure that the intersection point lies on the edge of the polytope, we stipulate that it satisfies the inequality constraints in $M \setminus E$. That is,

$$\sum_{j=1}^{n} a_{ij} \cdot \frac{\det(A^j_{E,\boldsymbol{\alpha},\beta})}{\det(A_{E,\boldsymbol{\alpha}})} \leq b_i \tag{4.3}$$

for every $i \in M \setminus E$ (if $\boldsymbol{\alpha}, \beta$ satisfy any of these constraints, it must be that $\det(A_{E,\boldsymbol{\alpha}}) \neq 0$, which guarantees that $A_E\boldsymbol{x} = \boldsymbol{b}_E, \boldsymbol{\alpha}^\top \boldsymbol{x} = \beta$ has a unique solution). Multiplying through by $\det(A_{E,\boldsymbol{\alpha}})$ shows that this constraint is a halfspace in $\mathbb{R}^{n+1}$, since $\det(A_{E,\boldsymbol{\alpha}})$ and $\det(A^i_{E,\boldsymbol{\alpha},\beta})$ are linear in $\boldsymbol{\alpha}$ and β. The collection of all the hyperplanes defining the boundaries of these halfspaces over all edges of $\mathcal{P}$ induces a partition of $\mathbb{R}^{n+1}$ into connected components such that for all $(\boldsymbol{\alpha}, \beta)$ within a given component, the (nonempty) set of edges of $\mathcal{P}$ that the hyperplane $\boldsymbol{\alpha}^\top \boldsymbol{x} = \beta$ intersects is invariant.

Now, consider a single connected component, denoted by C for brevity. Let $e_1, \ldots, e_k$ denote the edges intersected by cuts in C, and let $E_1, \ldots, E_k \subset M$ denote the sets of constraints that are binding at each of these edges, respectively. For each pair e_p, e_q, consider the surface

$$\sum_{i=1}^{n} c_i \cdot \frac{\det(A^i_{E_p,\boldsymbol{\alpha},\beta})}{\det(A_{E_p,\boldsymbol{\alpha}})} = \sum_{i=1}^{n} c_i \cdot \frac{\det(A^i_{E_q,\boldsymbol{\alpha},\beta})}{\det(A_{E_q,\boldsymbol{\alpha}})}. \tag{4.4}$$

Clearing the (nonzero) denominators shows this is a degree-2 polynomial hypersurface in $\boldsymbol{\alpha}, \beta$ in $\mathbb{R}^{n+1}$. This hypersurface is the set of all $(\boldsymbol{\alpha}, \beta)$ for which the LP objective values achieved at the vertices on edges e_p and e_q are equal. The collection of these surfaces for each p, q partitions C into further connected components. Within each component C', the edge containing the vertex that maximizes the objective is invariant. If this edge corresponds to binding constraints E, $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta)$ has the closed form $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta)[i] = \det(A^i_{E,\boldsymbol{\alpha},\beta})/\det(A_{E,\boldsymbol{\alpha}})$ for all $(\boldsymbol{\alpha}, \beta) \in C'$. We now count the number of surfaces in our decomposition. $\mathcal{P}$ has at most $\binom{m}{n-1} \leq m^{n-1}$ edges, and for each edge E, Equation (4.3) defines at most $|M \setminus E| \leq m$ hyperplanes for a total of at most m^n hyperplanes. Equation (4.4) defines a degree-2 polynomial hypersurface for every pair of edges, of which there are at most $\binom{m^n}{2} \leq m^{2n}$. □

In Section 4.5.4, we generalize Theorem 4.5.2 to understand $\boldsymbol{x}^*_{\mathrm{LP}}$ as a function of any K constraints. In this case, we show that the piecewise structure is given by degree-$2K$ multivariate polynomials.

Example in two dimensions

Consider the LP

$$\max\{x + y : x \leq 1, y \geq 0, y \leq x\}.$$

The optimum is at $(x^*, y^*) = (1, 1)$. Consider adding an additional constraint $\alpha_1 x + \alpha_2 y \leq 1$. Let h denote the hyperplane $\alpha_1 x + \alpha_2 y = 1$. We derive a description of the set of parameters (α_1, α_2) such that h intersects the hyperplanes $x = 1$ and $y = x$. The intersection of h and $x = 1$ is given by

$$(x, y) = \left(1, \frac{1 - \alpha_1}{\alpha_2}\right),$$

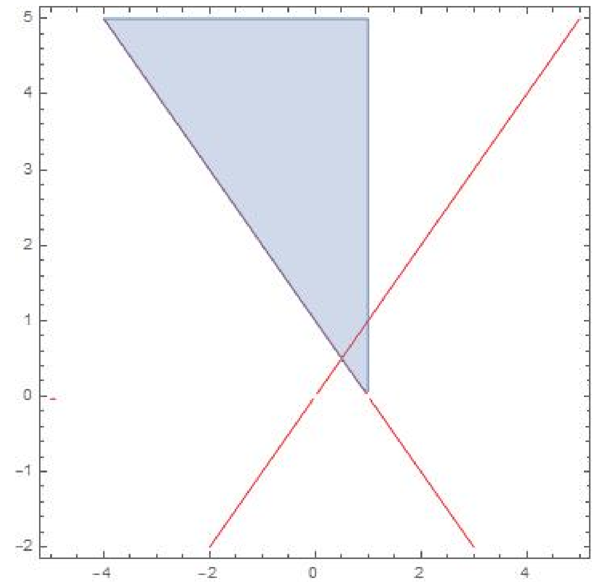


Figure 4.7: Decomposition of the parameter space: the blue region contains the set of (α_1, α_2) such that the constraint intersects the feasible region at $x = 1$ and $x = y$. The red lines consist of all (α_1, α_2) such that the objective value is equal at these intersection points. The red lines partition the blue region into two components: one where the new optimum is achieved at the intersection of h and $x = y$, and one where the new optimum is achieved at the intersection of h and $x = 1$.

which exists if and only if $\alpha_2 \neq 0$. This intersection point is in the LP feasible region if and only if $0 \leq \frac{1-\alpha_1}{\alpha_2} \leq 1$ (which additionally ensures that $\alpha_2 \neq 0$). Similarly, h intersects $y = x$ at

$$(x, y) = \left(\frac{1}{\alpha_1 + \alpha_2}, \frac{1}{\alpha_1 + \alpha_2}\right),$$

which exists if and only if $\alpha_1 + \alpha_2 \neq 0$. This intersection point is in the LP feasible region if and only if $0 \leq \frac{1}{\alpha_1+\alpha_2} \leq 1$. Now, we put down an "indifference" curve in (α_1, α_2)-space that represents the set of (α_1, α_2) such that the value of the objective achieved at the two aforementioned intersection points is equal. This surface is given by

$$\frac{2}{\alpha_1 + \alpha_2} = 1 + \frac{1 - \alpha_1}{\alpha_2}.$$

Since $\alpha_1 + \alpha_2 \neq 0$ and $\alpha_2 \neq 0$ (for the relevant α_1, α_2 in consideration), this is equivalent to $\alpha_1^2 - \alpha_2^2 - \alpha_1 + \alpha_2 = 0$, which is a degree-2 curve in α_1, α_2. The left-hand-side can be factored to write this as $(\alpha_1 - \alpha_2)(\alpha_1 + \alpha_2 - 1) = 0$. Therefore, this curve is given by the two lines $\alpha_1 = \alpha_2$ and $\alpha_1 + \alpha_2 = 1$. Figure 4.7 illustrates the resulting partition of (α_1, α_2)-space.

It turns out that when $n = 2$ the indifference curve can always be factored into a product of linear terms. Let the objective of the LP be (c_1, c_2), and let $s_1 x + s_2 y = u_1$ and $t_1 x + t_2 y = v_1$ be two intersecting edges of the LP feasible region. Let $\alpha_1 x + \alpha_2 y = \beta$ be an additional constraint.

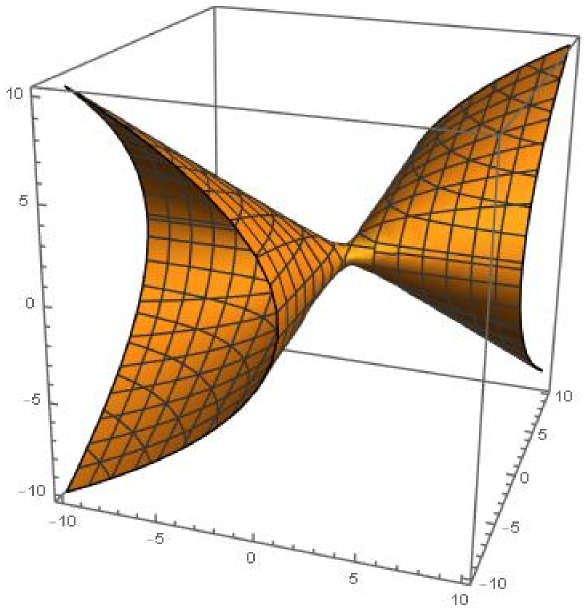


Figure 4.8: Indifference surface for two edges of the feasible region of an LP in three variables.

The intersection points of this constraint with the two lines, if they exist, are given by

$$\left(\frac{s_2\beta - u\alpha_2}{s_2\alpha_1 - s_1\alpha_2}, \frac{s_1\beta - u\alpha_1}{s_1\alpha_2 - s_2\alpha_1}\right) \text{ and } \left(\frac{t_2\beta - v\alpha_2}{t_2\alpha_1 - t_1\alpha_2}, \frac{t_2\beta - v\alpha_1}{t_1\alpha_2 - t_2\alpha_1}\right).$$

The indifference surface is thus given by

$$c_1\frac{s_2\beta - u\alpha_2}{s_2\alpha_1 - s_1\alpha_2} + c_2\frac{s_1\beta - u\alpha_1}{s_1\alpha_2 - s_2\alpha_1} = c_1\frac{t_2\beta - v\alpha_2}{t_2\alpha_1 - t_1\alpha_2} + c_2\frac{t_2\beta - v\alpha_1}{t_1\alpha_2 - t_2\alpha_1}.$$

For α_1, α_2 such that $s_2\alpha_1 - s_1\alpha_2 \neq 0$ and $t_2\alpha_1 - t_1\alpha_2 \neq 0$, clearing denominators and some manipulation yields

$$(c_1\alpha_2 - c_2\alpha_1)((ut_1 - vs_1)\alpha_2 - (ut_2 - vs_2)\alpha_1 + (s_2t_2 - t_1s_2)\beta) = 0.$$

This curve consists of the two planes $c_1\alpha_2 - c_2\alpha_1 = 0$ and $(ut_1 - vs_1)\alpha_2 - (ut_2 - vs_2)\alpha_1 + (s_2t_2 - t_1s_2)\beta = 0$.

If $n > 2$, the indifference surface need not decompose into linear terms. For example, consider an LP in three variables x, y, z with the constraints $x + y \leq 1, x + z \leq 1, x \leq 1, z \leq 1$. Writing out the indifference surface (assuming the objective is $\boldsymbol{c} = (1,1,1)^\top$) for the vertex on the intersection of $\{x + y = 1, x = 1\}$ and the vertex on $\{x + z = 1, z = 1\}$ yields

$$\alpha_1\alpha_2 - \alpha_2\beta - \alpha_3^2 + \alpha_3\beta = 0.$$

Setting $\beta = 1$, we can plot the resulting surface in $\alpha_1, \alpha_2, \alpha_3$ (Figure 4.8).

4.5.2 Structure and sensitivity of branch-and-cut

We now use Theorem 4.5.2 to answer a fundamental question about B&C: what is the structure of the B&C tree as a function of cuts at the root? Answering this question brings us one step closer toward providing sample complexity guarantees for GMI cuts. Said another way, we derive conditions on $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{R}^n$, $\beta_1, \beta_2 \in \mathbb{R}$, such that B&C behaves identically on the two IPs

$$\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \boldsymbol{x} \in \mathbb{Z}^n_{\ge 0}\} \text{ and } \max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{\alpha}_2^\top \boldsymbol{x} \le \beta_2, \boldsymbol{x} \in \mathbb{Z}^n_{\ge 0}\}.$$

We prove that the set of all cuts can be partitioned into a finite number of regions where by employing cuts from any one region, the B&C tree remains exactly the same. We also prove that the boundaries between regions are defined by low-degree polynomials. Figure 4.6 is a schematic diagram of our proof, which breaks the analysis of B&C into four main steps. Each step successively refines the partition obtained in the previous step, and uses the properties established in the previous step to analyze the next stage of B&C. We focus on a single cut added to the root and extend to multiple cuts in Section 4.5.4.

We use the following notation in this section. Given an IP, let $\tau = \lceil \max_{\boldsymbol{x} \in \mathcal{P}} \|\boldsymbol{x}\|_\infty \rceil$ be the maximum magnitude coordinate of any LP-feasible solution, rounded up. By Cramer's rule and Hadamard's inequality, $\tau \le a^n n^{n/2}$ where $a = \|A\|_{\infty,\infty}$. However, τ can be much smaller. For example, if A contains a row with only positive entries, then $\tau \le \|\boldsymbol{b}\|_\infty$. Let $\mathcal{BC} := \{\boldsymbol{x}[i] \le \ell, \boldsymbol{x}[i] \ge \ell\}_{0 \le \ell \le \tau, i \in [n]}$, which contains the set of all possible branching constraints. Let A_σ and $\boldsymbol{b}_\sigma$ denote A and b with the constraints in $\sigma \subseteq \mathcal{BC}$ added. For $E \subseteq M \cup \sigma$, let $A_{E,\sigma} \in \mathbb{R}^{|E| \times n}$ and $\boldsymbol{b}_E \in \mathbb{R}^{|E|}$ denote the restrictions of A_σ and $\boldsymbol{b}_\sigma$ to E. For $\boldsymbol{\alpha} \in \mathbb{R}^n, \beta \in \mathbb{R}$ and $E \subseteq M \cup \sigma$ with $|E| = n-1$, let $A_{E,\boldsymbol{\alpha},\sigma} \in \mathbb{R}^{n \times n}$ denote the matrix obtained by adding row vector $\boldsymbol{\alpha}$ to $A_{E,\sigma}$ and let $A^i_{E,\boldsymbol{\alpha},\beta,\sigma} \in \mathbb{R}^{n \times n}$ be the matrix $A_{E,\boldsymbol{\alpha},\sigma}$ with the ith column replaced by $(\boldsymbol{b}_{E,\sigma}, \beta)^\top$.

We require the following lemma which bounds the number of relevant subsets of $\mathcal{BC} := \{\boldsymbol{x}[i] \le \ell, \boldsymbol{x}[i] \ge \ell\}_{0 \le \ell \le \tau, i \in [n]}$ that define a possible node expanded by B&C. $\mathcal{BC}$ is a set of size $2n(\tau+1)$ so naïvely there are at most $2^{2n(\tau+1)}$ subsets of branching constraints. The following observation allows us to greatly reduce the number of sets we consider.

Lemma 4.5.3. *Fix an IP $(\boldsymbol{c}, A, \boldsymbol{b})$. Define an equivalence relation on pairs of branching-constraint sets $\sigma_1, \sigma_2 \subseteq \mathcal{BC}$, by $\sigma_1 \sim \sigma_2 \iff \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma_1) = \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma_2)$ for all possible cutting planes $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$. The number of equivalence classes of $\sim$ is at most τ^{3n}.*

Proof. Consider as an example $\sigma_1 = \{\boldsymbol{x}[1] \le 1, \boldsymbol{x}[1] \le 5\}$ and $\sigma_2 = \{\boldsymbol{x}[1] \le 1\}$. We have $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma_1) = \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma_2)$ for any cut $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$, because the constraint $\boldsymbol{x}[1] \le 5$ is redundant in σ_1. More generally, any $\sigma \subseteq \mathcal{BC}$ can be reduced by preserving only the tightest $\le$ constraint and tightest $\ge$ constraint without affecting the resulting LP optimal solutions. The number of such unique reduced sets is at most $((\tau+2)^2)^n < \tau^{3n}$ (for each variable, there are $\tau + 2$ possibilities for the tightest $\le$ constraint: no constraint or one of $\boldsymbol{x}[i] \le 0, \ldots, \boldsymbol{x}[i] \le \tau$, and similarly $\tau + 2$ possibilities for the $\ge$ constraint). □

Step 1: Understanding how the cut affects the LP optimum at any node of the B&C tree

Theorem 4.5.2 gives a (piecewise) closed form for the LP optimum $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta)$ at the root of the B&C tree as a function of coefficients $(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^{n+1}$ determining the cut. The first step is to extend this result to get a handle on the LP optimum at any node of any B&C tree. Suppose $\sigma \subseteq \mathcal{BC}$ is a set of branching constraints (any node of any B&C tree can be identified with some $\sigma \subseteq \mathcal{BC}$). We refine the partition of space obtained in Theorem 4.5.2 so that within a given region of the new partition, $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)$ has a closed form for all σ. This is illustrated by Figure 4.6a.

Lemma 4.5.4. *For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are at most $(m+2n)^n\tau^{3n}$ hyperplanes and at most $(m+2n)^{2n}\tau^{3n}$ degree-2 polynomial hypersurfaces partitioning $\mathbb{R}^{n+1}$ into connected components such that for each component C and every $\sigma \subset \mathcal{BC}$, either: (1) $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = \boldsymbol{x}^*_{\text{LP}}(\sigma)$ and $z^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = z^*_{\text{LP}}(\sigma)$, or (2) there is a set of constraints $E \subseteq M \cup \sigma$ with $|E| = n-1$ such that $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] = \frac{\det(A^i_{E,\boldsymbol{\alpha},\beta,\sigma})}{\det(A_{E,\boldsymbol{\alpha},\sigma})}$ for all $(\boldsymbol{\alpha}, \beta) \in C$.*

Proof. We carry out the same reasoning in the proof of Theorem 4.5.2 for each reduced σ. The number of edges of $\mathcal{P}(\sigma)$ is at most $\binom{m+|\sigma|}{n-1} \le (m+|\sigma|)^{n-1}$. For each edge E, we considered at most $|(M \cup \sigma) \setminus E| \le m + |\sigma|$ hyperplanes, for a total of at most $(m+|\sigma|)^n$ halfspaces. Then, we had a degree-2 polynomial hypersurface for every pair of edges, of which there are at most $\binom{(m+|\sigma|)^n}{2} \le (m+|\sigma|)^{2n}$. Summing over all reduced σ (of which there are at most τ^{3n}), combined with the fact that if σ is reduced then $|\sigma| \le 2n$, we get a total of at most $(m+2n)^n\tau^{3n}$ hyperplanes and at most $(m+2n)^{2n}\tau^{3n}$ degree-2 hypersurfaces, as desired. □

Step 2: Conditions for branching decisions to be identical

We next refine the decomposition obtained in Lemma 4.5.4 so that the branching constraints added at each step of B&C are invariant within a region, as in Figure 4.6b. For concreteness, we analyze the product scoring rule used by the leading open-source solver SCIP [Gamrath et al., 2020]. The high-level intuition is that we zoom in on a connected component in the partition of Lemma 4.5.4. Within this component, we may express $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)$ explicitly in terms of $\boldsymbol{\alpha}, \beta$, for all σ. This allows us to unravel the branching rule and derive conditions for invariance. We omit the details here, which can be found in the full version of the paper.

Lemma 4.5.5. *There is a set of of at most $3(m+2n)^n\tau^{3n}$ hyperplanes and $(m+2n)^{2n}\tau^{3n}$ degree-2 polynomial hypersurfaces partitioning $\mathbb{R}^{n+1}$ into connected components such that for any connected component C and any σ, the set of branching constraints $\{x_i \le \lfloor \boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] \rfloor, x_i \ge \lceil \boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] \rceil \mid i \in [n]\}$ is invariant across all $(\boldsymbol{\alpha}, \beta) \in C$.*

Proof. Fix a connected component C in the decomposition established in Lemma 4.5.4. By Lemma 4.5.4, for each σ, either $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = \boldsymbol{x}^*_{\text{LP}}(\sigma)$ or there exists $E \subseteq M \cup \sigma$ such that $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] = \frac{\det(A^i_{E,\boldsymbol{\alpha},\beta,\sigma})}{\det(A_{E,\boldsymbol{\alpha},\sigma})}$ for all $(\boldsymbol{\alpha}, \beta) \in C$. Fix a variable $i \in [n]$, which corresponds to two branching constraints

$$x_i \le \left\lfloor \boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] \right\rfloor \text{ and } x_i \ge \left\lceil \boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] \right\rceil. \tag{4.5}$$

If C is a component where $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = \boldsymbol{x}^*_{\text{LP}}(\sigma)$, then these two branching constraints are trivially invariant over $(\boldsymbol{\alpha}, \beta) \in C$. Otherwise, in order to further decompose C such that the right-hand-sides of these constraints are invariant for every σ, we add the two decision boundaries given by

$$k \leq \frac{\det(A^i_{E,\boldsymbol{\alpha},\beta,\sigma})}{\det(A_{E,\boldsymbol{\alpha},\sigma})} \leq k+1$$

for every i, σ, and every integer $k = 0, \dots, \tau - 1$, where $\tau = \max_{\boldsymbol{x} \in \mathcal{P} \cap \mathbb{Z}^n} \|\boldsymbol{x}\|_\infty$. This ensures that within every connected component of C induced by these boundaries (hyperplanes),

$$\lfloor \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma)[i] \rfloor = \left\lfloor \frac{\det(A^i_{E,\boldsymbol{\alpha},\beta,\sigma})}{\det(A_{E,\boldsymbol{\alpha},\sigma})} \right\rfloor \text{ and } \lceil \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma)[i] \rceil = \left\lceil \frac{\det(A^i_{E,\boldsymbol{\alpha},\beta,\sigma})}{\det(A_{E,\boldsymbol{\alpha},\sigma})} \right\rceil$$

are invariant, so the branching constraints from Equation (4.5) are invariant. For a fixed σ, there are two hyperplanes for every $E \subseteq M \cup \sigma$ corresponding to an edge of $\mathcal{P}(\sigma)$ and $i = 1, \dots, n$, for a total of at most $2n\binom{m+|\sigma|}{n-1} \leq 2n(m+|\sigma|)^{n-1}$ hyperplanes. Summing over all reduced σ, we get a total of $2n(m+2n)^{n-1}\tau^{3n} < 2(m+2n)^n\tau^{3n}$ hyperplanes. Adding these hyperplanes to the set of hyperplanes established in Lemma 4.5.4 yields the lemma statement. □

Lemma 4.5.6. *For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are at most $3(m+2n)^n\tau^{3n}$ hyperplanes, $3(m+2n)^{3n}\tau^{4n}$ degree-2 polynomial hypersurfaces, and $(m+2n)^{6n}\tau^{4n}$ degree-5 polynomial hypersurfaces partitioning $\mathbb{R}^{n+1}$ into connected components such that within each component, the branching constraints used at every step of B&C are invariant.*

Proof sketch. The proof is a careful analysis of the product scoring rule, combined with the previous lemma, which allows us to derive conditions ensuring that the branching variable selected is invariant. □

Step 3: When do nodes have an integral LP optimum?

We now move to the most critical phase of branch-and-cut: deciding when to fathom a node. The first reason a node might be fathomed is if the LP relaxation of the IP at that node has an integral solution. We derive conditions that ensure that nearby cuts have the same effect on the integrality of the IP at any node in the search tree. Recall $\mathcal{P}_{\mathrm{I}} = \mathcal{P} \cap \mathbb{Z}^n$ is the set of integer points in $\mathcal{P}$. Let $\mathcal{V} \subseteq \mathbb{R}^{n+1}$ denote the set of all valid cuts for the input IP $(\boldsymbol{c}, A, \boldsymbol{b})$. The set $\mathcal{V}$ is a polyhedron since it can be expressed as

$$\mathcal{V} = \bigcap_{\overline{\boldsymbol{x}} \in \mathcal{P}_{\mathrm{I}}} \{(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^{n+1} : \boldsymbol{\alpha}^\top \overline{\boldsymbol{x}} \leq \beta\},$$

and $\mathcal{P}_{\mathrm{I}}$ is finite as $\mathcal{P}$ is bounded. For cuts outside $\mathcal{V}$, we assume the B&C tree takes some special form denoting an invalid cut. Our goal now is to decompose $\mathcal{V}$ into connected components such that $\mathbf{1}\left[\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma) \in \mathbb{Z}^n\right]$ is invariant for all $(\boldsymbol{\alpha}, \beta)$ in each component.

Lemma 4.5.7. *For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are at most $3(m+2n)^n\tau^{4n}$ hyperplanes, $3(m+2n)^{3n}\tau^{4n}$ degree-2 polynomial hypersurfaces, and $(m+2n)^{6n}\tau^{4n}$ degree-5 polynomial hypersurfaces partitioning $\mathbb{R}^{n+1}$ into connected components such that for each component C and each $\sigma \subseteq \mathcal{BC}$, $\mathbf{1}\left[\boldsymbol{x}^*_{\mathrm{LP}}\left(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma\right) \in \mathbb{Z}^n\right]$ is invariant for all $(\boldsymbol{\alpha}, \beta) \in C$.*

Proof. Fix a connected component C in the decomposition that includes the facets defining $\mathcal{V}$ and the surfaces obtained in Lemma 4.5.6. For all $\sigma \in \mathcal{BC}$, $\boldsymbol{x}_{\rm I} \in \mathcal{P}_{\rm I}$, and $i = 1, \ldots, n$, consider the surface

$$\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] = \boldsymbol{x}_{\rm I}[i]. \tag{4.6}$$

This surface is a hyperplane, since by Lemma 4.5.4, either $\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] = \boldsymbol{x}^*_{\rm LP}(\sigma)[i]$ or $\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] = \frac{\det(A^i_{E,\boldsymbol{\alpha},\beta,\sigma})}{\det(A_{E,\boldsymbol{\alpha},\sigma})}$, where $E \subseteq M \cup \sigma$ is the subset of constraints corresponding to σ and C. Clearly, within any connected component of C induced by these hyperplanes, for every σ and $\boldsymbol{x}_{\rm I} \in \mathcal{P}_{\rm I}$, $\mathbf{1}[\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = \boldsymbol{x}_{\rm I}]$ is invariant. Finally, if $\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) \in \mathbb{Z}^n$ for some cut $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ within a given connected component, $\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = \boldsymbol{x}_{\rm I}$ for some $\boldsymbol{x}_{\rm I} \in \mathcal{P}_{\rm IH}(\sigma) \subseteq \mathcal{P}_{\rm I}$, which means that $\boldsymbol{x}^*_{\rm LP}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma) = \boldsymbol{x}_{\rm I} \in \mathbb{Z}^n$ *for all* cuts $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ in that connected component.

We now count the number of hyperplanes given by Equation 4.6. For each σ, there are $\binom{m+|\sigma|}{n-1} \le (m+2n)^{n-1}$ binding edge constraints $E \subseteq M \cup \sigma$ defining the formula of Lemma 4.5.4, and we have $n|\mathcal{P}_{\rm I}|$ hyperplanes for each E. Since $\tau = \max_{\boldsymbol{x} \in \mathcal{P}_{\rm I}} \|\boldsymbol{x}\|_\infty$, $|\mathcal{P}_{\rm I}| \le \tau^n$. So the total number of hyperplanes given by Equation 4.6 is at most $\tau^{3n}(m+2n)^{n-1} n \tau^n \le (m+2n)^n \tau^{4n}$. The number of facets defining $\mathcal{V}$ is at most $|\mathcal{P}_{\rm IH}| \le |\mathcal{P}_{\rm I}| \le \tau^n$. Adding these to the counts obtained in Lemma 4.5.6 yields the final tallies in the lemma statement. □

Lemma 4.5.7 is illustrated by Figure 4.6c. Next, suppose for a moment that B&C fathoms a node if and only if either the LP is infeasible or the LP optimal solution is integral—that is, the "bounding" of B&C is suppressed. In this case, the tree built by B&C is invariant within each component of the partition in Lemma 4.5.7. Equipped with this observation, we now analyze the full behavior of B&C.

Step 4: Pruning nodes with weak LP bounds

In this final step, we analyze the most important aspect of B&C: pruning nodes when the LP objective value is smaller than the best-known integral solution. Using the tools we have developed so far, expressing the question "is the LP value at a node smaller than the best-known integral solution?" becomes a simple matter of hyperplanes and halfspaces. This final step is illustrated by Figure 4.6d.

Theorem 4.5.8. *Given an IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there is a set of at most $O(14^n(m+2n)^{3n^2}\tau^{5n^2})$ polynomial hypersurfaces of degree ≤ 5 partitioning $\mathbb{R}^{n+1}$ into connected components such that the B&C tree built after adding the cut $\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta$ at the root is invariant over all $(\boldsymbol{\alpha}, \beta)$ within a given component.*

Proof. Fix a connected component C in the decomposition induced by the set of hyperplanes and degree-2 hypersurfaces established in Lemma 4.5.7. Let

$$Q_1, \ldots, Q_{i_1}, I_1, Q_{i_1+1}, \ldots, Q_{i_2}, I_2, Q_{i_2+1}, \ldots \tag{4.7}$$

denote the nodes of the tree branch-and-cut creates, in order of exploration, under the assumption that a node is pruned if and only if either the LP at that node is infeasible or the LP optimal solution is integral (so the "bounding" of branch-and-bound is suppressed). Here, a node is identified by the list σ of branching constraints added to the input IP. Nodes labeled by Q are either infeasible or have fractional LP optimal solutions. Nodes labeled by I have integral LP optimal solutions and are candidates for the incumbent integral solution at the point they are encountered. (The nodes are functions of $\boldsymbol{\alpha}$ and β, as are the indices $i_1, i_2, \ldots$.) By Lemma 4.5.7 and the observation following it, this ordered list of nodes is invariant over all $(\boldsymbol{\alpha}, \beta) \in C$.

Now, given an node index ℓ, let $I(\ell)$ denote the incumbent node with the highest objective value encountered up until the ℓth node searched by B&C, and let $z(I(\ell))$ denote its objective value. For each node Q_ℓ, let σ_ℓ denote the branching constraints added to arrive at node Q_ℓ. The hyperplane

$$z^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma_\ell) = z(I(\ell)) \tag{4.8}$$

(which is a hyperplane due to Lemma 4.5.4) partitions C into two subregions. In one subregion, $z^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma_\ell) \leq z(I(\ell))$, that is, the objective value of the LP optimal solution is no greater than the objective value of the current incumbent integer solution, and so the subtree rooted at Q_ℓ is pruned. In the other subregion, $z^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma_\ell) > z(I(\ell))$, and Q_ℓ is branched on further. Therefore, within each connected component of C induced by all hyperplanes given by Equation 4.8 for all ℓ, the set of node within the list (4.7) that are pruned is invariant. Combined with the surfaces established in Lemma 4.5.7, these hyperplanes partition $\mathbb{R}^{n+1}$ into connected components such that as $(\boldsymbol{\alpha}, \beta)$ varies within a given component, the tree built by branch-and-cut is invariant.

Finally, we count the total number of surfaces inducing this partition. Unlike the counting stages of the previous lemmas, we will first have to count the number of connected components induced by the surfaces established in Lemma 4.5.7. This is because the ordered list of nodes explored by branch-and-cut (4.7) can be different across each component, and the hyperplanes given by Equation 4.8 depend on this list. From Lemma 4.5.7 we have $3(m+2n)^n\tau^{4n}$ hyperplanes, $3(m+2n)^{3n}\tau^{4n}$ degree-2 polynomial hypersurfaces, and $(m+2n)^{6n}\tau^{4n}$ degree-5 polynomial hypersurfaces. To determine the connected components of $\mathbb{R}^{n+1}$ induced by the zero sets of these polynomials, it suffices to consider the zero set of the product of all polynomials defining these surfaces. Denote this product polynomial by p. The degree of the product polynomial is the sum of the degrees of $3(m+2n)^n\tau^{4n}$ degree-1 polynomials, $3(m+2n)^{3n}\tau^{4n}$ degree-2 polynomials, and $(m+2n)^{6n}\tau^{4n}$ degree-5 polynomials, which is at most

$$3(m+2n)^n\tau^{4n} + 2\cdot 3(m+2n)^{3n}\tau^{4n} + 5\cdot (m+2n)^{6n}\tau^{4n} < 14(m+2n)^{3n}\tau^{4n}.$$

By Warren's theorem, the number of connected components of $\mathbb{R}^{n+1} \setminus \{(\boldsymbol{\alpha}, \beta) : p(\boldsymbol{\alpha}, \beta) = 0\}$ is $O((14(m+2n)^{3n}\tau^{4n})^{n-1})$, and by the Milnor-Thom theorem, the number of connected components of $\{(\boldsymbol{\alpha}, \beta) : p(\boldsymbol{\alpha}, \beta) = 0\}$ is $O((14(m+2n)^{3n}\tau^{4n})^{n-1})$ as well. So, the number of connected components induced by the surfaces in Lemma 4.5.7 is $O(14^n(m+2n)^{3n^2}\tau^{4n^2})$. For every connected component C in Lemma 4.5.7, the closed form of $z^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma_\ell)$ is already determined due to Lemma 4.5.4, and so the number of hyperplanes given by Equation 4.8 is at most the number of possible $\sigma \subseteq \mathcal{BC}$, which is at most τ^{3n}. So across all connected components C, the total number of hyperplanes given by Equation 4.8 is $O(14^n(m+2n)^{3n^2}\tau^{5n^2})$. Finally, adding this to the surface-counts established in Lemma 4.5.7 yields the lemma statement. □

4.5.3 Sample complexity bounds for Gomory mixed integer cuts

In this section, we show how the results from Section 4.5.2 can be used to provide sample complexity bounds for GMI cuts (Definition 4.5.1), parameterized by $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^m$. We assume there is an unknown, application-specific distribution $\mathcal{D}$ over IPs. The learner receives a *training set* $\mathcal{S} \sim \mathcal{D}^N$ of N IPs sampled from this distribution. Formally, let $g_{\boldsymbol{u}}(\boldsymbol{c}, A, \boldsymbol{b})$ be the size of the tree B&C builds given the input $(\boldsymbol{c}, A, \boldsymbol{b})$ after applying the cut defined by $\boldsymbol{u}$ at the root. To derive our sample complexity guarantee, we bound the pseudo-dimension of $\mathcal{G} = \{g_{\boldsymbol{u}} : \boldsymbol{u} \in \mathcal{U}\}$.

So far, $\boldsymbol{\alpha}, \beta$ have been parameters that do not depend on the input instance $\boldsymbol{c}, A, \boldsymbol{b}$. Suppose now that they do: $\boldsymbol{\alpha}, \beta$ are functions of $\boldsymbol{c}, A, \boldsymbol{b}$ and a parameter vector $\boldsymbol{u}$ (as they are for GMI cuts). Despite the structure established in the previous section, if $\boldsymbol{\alpha}, \beta$ can depend on $(\boldsymbol{c}, A, \boldsymbol{b})$ in arbitrary ways, one cannot even hope for a finite sample complexity, illustrated by the following impossibility result.

Theorem 4.5.9. *There exist functions $\boldsymbol{\alpha}_{\boldsymbol{c},A,\boldsymbol{b}} : \mathcal{U} \to \mathbb{R}^n$ and $\beta_{\boldsymbol{c},A,\boldsymbol{b}} : \mathcal{U} \to \mathbb{R}$ such that $\mathsf{Pdim}(\{g_{\boldsymbol{u}} : \boldsymbol{u} \in \mathcal{U}\}) = \infty$, where $\mathcal{U}$ is any set with $|\mathcal{U}| = |\mathbb{R}|$.*

Proof. For a set $\mathcal{X}$, $\mathcal{X}^{<\mathbb{N}}$ denotes the set of finite sequences of elements from $\mathcal{X}$. There is a bijection between the set of IPs $(\boldsymbol{c}, A, \boldsymbol{b}) \in \mathcal{I} := \mathbb{R}^n \times \mathbb{Z}^{m \times n} \times \mathbb{Z}^m$ and $\mathbb{R}$, so IPs can be uniquely represented as real numbers (and vice versa). Now, consider the set of all finite sequences of pairs of IPs and ± 1 labels of the form $((\boldsymbol{c_1}, A_1, \boldsymbol{b}_1), \varepsilon_1), \ldots, ((\boldsymbol{c_N}, A_N, \boldsymbol{b}_N), \varepsilon_N), \varepsilon_1, \ldots, \varepsilon_N \in \{-1, 1\}$, that is, the set $(\mathcal{I} \times \{-1, 1\})^{<\mathbb{N}}$. There is a bijection between this set and $(\mathbb{R} \times \{-1, 1\})^{<\mathbb{N}}$, and in turn there is a bijection between $(\mathbb{R} \times \{-1, 1\})^{<\mathbb{N}}$ and $\mathbb{R}$. Hence, there exists a bijection between $\mathcal{U}$ and $(\mathcal{I} \times \{-1, 1\})^{<\mathbb{N}}$. Fix such a bijection $\varphi : \mathcal{U} \to (\mathcal{I} \times \{-1, 1\})^{<\mathbb{N}}$, and let $\varphi^{-1} : (\mathcal{I} \times \{-1, 1\})^{<\mathbb{N}} \to \mathcal{U}$ denote the inverse of φ, which is well defined and also a bijection.

Let n be odd. For $c \in \mathbb{R}$, let $\mathrm{IP}_c \in \mathcal{I}$ denote the IP

$$\begin{array}{ll} \text{maximize} & c \\ \text{subject to} & 2x_1 + \cdots + 2x_n = n \\ & \boldsymbol{x} \in \{0, 1\}^n. \end{array} \tag{4.9}$$

Since n is odd, IP_c is infeasible, independent of c. Jeroslow [1974] showed that without the use of cutting planes or heuristics, branch-and-bound builds a tree of size $2^{(n-1)/2}$ before determining infeasibility and terminating. The objective c is irrelevant, but is important in generating distinct IPs with this property. Consider the cut $x_1 + \cdots + x_n \leq \lfloor n/2 \rfloor$, which is a valid cut for IP_c (this is in fact a Chvátal-Gomory cut [Balcan et al., 2021d]). In particular, since n is odd, $x_1 + \cdots + x_n \leq \lfloor n/2 \rfloor \implies x_1 + \cdots + x_n \leq (n-1)/2 < n/2$, so the equality constraint of IP_c is violated by this cut. Thus, the feasible region of the LP relaxation after adding this cut is empty, and branch-and-bound will terminate immediately at the root (building a tree of size 1). Denote this cut by $(\boldsymbol{\alpha}^{(-1)}, \beta^{(-1)}) = (\mathbf{1}, \lfloor n/2 \rfloor)$. On the other hand, let $(\boldsymbol{\alpha}^{(1)}, \beta^{(1)}) = (\mathbf{0}, 0)$ be the trivial cut $0 \leq 0$. Adding this cut to the IP constraints does not change the feasible region, so branch-and-bound will build a tree of size $2^{(n-1)/2}$.

We now define $\boldsymbol{\alpha}_{\boldsymbol{c},A,\boldsymbol{b}}$ and $\beta_{\boldsymbol{c},A,\boldsymbol{b}}$. Let

$$(\boldsymbol{\alpha}_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{u}), \beta_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{u})) = \begin{cases} (\boldsymbol{\alpha}^{(1)}, \beta^{(1)}) & \text{if } ((\boldsymbol{c}, A, \boldsymbol{b}), 1) \in \varphi(\boldsymbol{u}) \text{ and } ((\boldsymbol{c}, A, \boldsymbol{b}), -1) \notin \varphi(\boldsymbol{u}) \\ (\boldsymbol{\alpha}^{(-1)}, \beta^{(-1)}) & \text{if } ((\boldsymbol{c}, A, \boldsymbol{b}), -1) \in \varphi(\boldsymbol{u}) \text{ and } ((\boldsymbol{c}, A, \boldsymbol{b}), 1) \notin \varphi(\boldsymbol{u}) \\ (\mathbf{0}, 0) & \text{otherwise} \end{cases}.$$

The choice to use $(\mathbf{0}, 0)$ in the case that either $((\boldsymbol{c}, A, \boldsymbol{b}), \varepsilon) \notin \varphi(\boldsymbol{u})$ for each $\varepsilon \in \{-1, 1\}$, or $((\boldsymbol{c}, A, \boldsymbol{b}), -1) \in \varphi(\boldsymbol{u})$ and $((\boldsymbol{c}, A, \boldsymbol{b}), 1) \in \varphi(\boldsymbol{u})$ is arbitrary and unimportant. Now, for any integer $N > 0$, constructing a set of N IPs and N thresholds that is shattered is almost immediate. Let $c_1, \ldots, c_N \in \mathbb{R}$ be distinct reals, and let $1 < r_1, \ldots, r_N < 2^{(n-1)/2}$. Then, the set $\{(\mathrm{IP}_{c_1}, r_1), \ldots, (\mathrm{IP}_{c_N}, r_N)\}$ can be shattered. Indeed, given a sign pattern $(\varepsilon_1, \ldots, \varepsilon_N) \in \{-1, 1\}^N$, let

$$\boldsymbol{u} = \varphi^{-1}\left((\mathrm{IP}_{c_1}, \varepsilon_1), \ldots, (\mathrm{IP}_{c_N}, \varepsilon_N)\right).$$

Then, if $\varepsilon_i = 1$, $(\boldsymbol{\alpha}_{\mathrm{IP}_{c_i}}(\boldsymbol{u}), \beta_{\mathrm{IP}_{c_i}}(\boldsymbol{u})) = (\boldsymbol{\alpha}^{(1)}, \beta^{(1)})$, so $g_{\boldsymbol{u}}(\mathrm{IP}_{c_i}) = 2^{(n-1)/2}$ and $\mathrm{sign}(g_{\boldsymbol{u}}(\mathrm{IP}_{c_i}) - r_i) = 1$. If $\varepsilon_i = -1$, $(\boldsymbol{\alpha}_{\mathrm{IP}_{c_i}}(\boldsymbol{u}), \beta_{\mathrm{IP}_{c_i}}(\boldsymbol{u})) = (\boldsymbol{\alpha}^{(-1)}, \beta^{(-1)})$, so $g_{\boldsymbol{u}}(\mathrm{IP}_{c_i}) = 1$ and $\mathrm{sign}(g_{\boldsymbol{u}}(\mathrm{IP}_{c_i}) - r_i) = -1$. So for any N there is a set of IPs and thresholds that can be shattered, which yields the theorem statement. □

However, in the case of GMI cuts (Def. 4.5.1), we show that the cutting plane coefficients parameterized by $\boldsymbol{u}$ are highly structured. Combining this structure with our analysis of B&C allows us to derive polynomial sample complexity bounds. We assume that $\boldsymbol{u} \in [-U, U]^m$ for some $U > 0$.

Let $\boldsymbol{\alpha} : [-U, U]^m \to \mathbb{R}^n$ denote the function taking GMI cut parameters $\boldsymbol{u}$ to the corresponding vector of coefficients determining the resulting cutting plane, and let $\beta : [-U, U]^m \to \mathbb{R}$ denote the offset of the resulting cutting plane. So (after multiplying through by $1 - f_0$),

$$\boldsymbol{\alpha}(\boldsymbol{u})[i] = \begin{cases} f_i(1 - f_0) & \text{if } f_i \le f_0 \\ f_0(1 - f_i) & \text{if } f_i > f_0 \end{cases}$$

and $\beta(\boldsymbol{u}) = f_0(1 - f_0)$ (f_0 and f_i are functions of $\boldsymbol{u}$, but we suppress this dependence for readability).

To understand the structure of B&C as a function of GMI cut parameters, we study the preimages of components $C \subseteq \mathbb{R}^{n+1}$ under the GMI coefficient maps $\boldsymbol{\alpha} : [-U, U]^m \to \mathbb{R}^n$, $\beta : [-U, U]^m \to \mathbb{R}$. If $C \subseteq \mathbb{R}^{n+1}$ (as in Theorem 4.5.8) is such that B&C (as a function of $\boldsymbol{\alpha}, \beta$) is invariant over C, then B&C (as a function of GMI parameter $\boldsymbol{u}$) is invariant over $D := \{\boldsymbol{u} : (\boldsymbol{\alpha}(\boldsymbol{u}), \beta(\boldsymbol{u})) \in C\}$. Our key structural insight for GMI cuts is that if C is the intersection of degree-d polynomial hypersurfaces in $\mathbb{R}^{n+1}$, then D is the intersection of degree-$2d$ polynomial hypersurfaces in $[-U, U]^m$. We provide the high-level intuition for this result below—the formal statements and proofs follow it.

Consider some degree-d polynomial p in variables $y_1, \ldots, y_{n+1}$ that defines C, which can be written as $\sum_{T \sqsubseteq [n+1], |T| \le d} \lambda_T \prod_{i \in T} y_i$ for some coefficients $\lambda_T \in \mathbb{R}$, where $T \sqsubseteq [n+1]$ means that T is a multiset of $[n+1]$. Evaluating at $(\boldsymbol{\alpha}(\boldsymbol{u}), \beta(\boldsymbol{u}))$, we get

$$\textstyle\sum_{|T| \le d} \lambda_T \prod_{i \in T \cap S \setminus \{n+1\}} f_i(1 - f_0) \prod_{i \in T \setminus S \setminus \{n+1\}} f_0(1 - f_i) \prod_{i \in T \cap \{n+1\}} f_0(1 - f_0).$$

Next, substitute $f_i = \boldsymbol{u}^\top \boldsymbol{a}_i - \lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor$ and $f_0 = \boldsymbol{u}^\top \boldsymbol{b} - \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor$. Restricted to $\boldsymbol{u}$ such that the floor terms round down to some fixed integers, the above expression is a polynomial in $\boldsymbol{u}$ of degree $\le 2d$. We run this procedure for every polynomial determining C, for every connected component C in the partition of $\mathbb{R}^{n+1}$ established in Theorem 4.5.8 to derive our main structural result for GMI cuts.

Lemma 4.5.10. *Consider the family of GMI cuts parameterized by $\boldsymbol{u} \in [-U, U]^m$. For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are at most $O(nU^2 \|A\|_1 \|\boldsymbol{b}\|_1)$ hyperplanes and $2^{O(n^2)}(m + 2n)^{O(n^3)}\tau^{O(n^3)}$ degree-10 polynomial hypersurfaces partitioning $[-U, U]^m$ into connected components such that the B&C tree built after adding the GMI cut defined by $\boldsymbol{u}$ is invariant over all $\boldsymbol{u}$ within a single component.*

To prove this formally, we establish two intermediate lemmas.

Lemma 4.5.11. *Consider the family of GMI cuts parameterized by $\boldsymbol{u} \in [-U, U]^m$. There is a set of at most $O(nU^2 \|A\|_1 \|\boldsymbol{b}\|_1)$ hyperplanes partitioning $[-U, U]^m$ into connected components such that $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor$, $\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor$, and $\mathbf{1}[f_i \leq f_0]$ are invariant, for every i, within each component.*

Proof. We have $f_i = \boldsymbol{u}^\top \boldsymbol{a}_i - \lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor$, $f_0 = \boldsymbol{u}^\top \boldsymbol{b} - \lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor$, and since $\boldsymbol{u} \in [-U, U]^m$, $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor \in [-U \|\boldsymbol{a}_i\|_1, U \|\boldsymbol{a}_i\|_1]$ and $\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor \in [-U \|\boldsymbol{b}\|_1, U \|\boldsymbol{b}\|_1]$. Now, for all i, $k_i \in [-U \|\boldsymbol{a}_i\|_1, U \|\boldsymbol{a}_i\|_1] \cap \mathbb{Z}$ and $k_0 \in [-U \|\boldsymbol{b}\|_1, U \|\boldsymbol{b}\|_1] \cap \mathbb{Z}$, put down the hyperplanes defining the two halfspaces

$$\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor = k_i \iff k_i \leq \boldsymbol{u}^\top \boldsymbol{a}_i < k_i + 1 \tag{4.10}$$

and the hyperplanes defining the two halfspaces

$$\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor = k_0 \iff k_0 \leq \boldsymbol{u}^\top \boldsymbol{b} < k_0 + 1. \tag{4.11}$$

In addition, consider the hyperplane

$$\boldsymbol{u}^\top \boldsymbol{a}_i - k_i = \boldsymbol{u}^\top \boldsymbol{b} - k_0 \tag{4.12}$$

for each i. Within any connected component of $\mathbb{R}^m$ determined by these hyperplanes, $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor$ and $\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor$ are constant. Furthermore, $\mathbf{1}[f_i \leq f_0]$ is invariant within each connected component, since if $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor = k_i$ and $\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor = k_0$, $f_i \leq f_0 \iff \boldsymbol{u}^\top \boldsymbol{a}_i - k_i \leq \boldsymbol{u}^\top \boldsymbol{b} - k_0$, which is the hyperplane given by Equation 4.12. The total number of hyperplanes of type 4.10 is $O(nU \|A\|_1)$, the total number of hyperplanes of type 4.11 is $O(U \|\boldsymbol{b}\|_1)$, and the total number of hyperplanes of type 4.12 is $nU^2 \|A\|_1 \|\boldsymbol{b}\|_1$. Summing yields the lemma statement. □

The next lemma allows us to transfer the polynomial partition of $\mathbb{R}^{n+1}$ from Theorem 4.5.8 to a polynomial partition of $[-U, U]^m$, incurring only a factor 2 increase in degree.

Lemma 4.5.12. *Let $p \in \mathbb{R}[y_1, \ldots, y_{n+1}]$ be a polynomial of degree d. Let $D \subseteq [-U, U]^m$ be a connected component from Lemma 4.5.11. Define $q : D \to \mathbb{R}$ by $q(\boldsymbol{u}) = p(\boldsymbol{\alpha}(\boldsymbol{u}), \beta(\boldsymbol{u}))$. Then q is a polynomial in $\boldsymbol{u}$ of degree $2d$.*

Proof. By Lemma 4.5.11, there are integers k_0, k_i for $i \in [n]$ such that $\lfloor \boldsymbol{u}^\top \boldsymbol{a}_i \rfloor = k_i$ and $\lfloor \boldsymbol{u}^\top \boldsymbol{b} \rfloor = k_0$ for all $\boldsymbol{u} \in D$. Also, the set $S = \{i : f_i \leq f_0\}$ is fixed over all $\boldsymbol{u} \in D$.

A degree-d polynomial p in variables $y_1, \ldots, y_{n+1}$ can be written as $\sum_{T \sqsubseteq [n+1], |T| \leq d} \lambda_T \prod_{i \in T} y_i$ for some coefficients $\lambda_T \in \mathbb{R}$, where $T \sqsubseteq [n+1]$ means that T is a multiset of $[n+1]$. Evaluating at $(\boldsymbol{\alpha}(\boldsymbol{u}), \beta(\boldsymbol{u}))$, we get

$$\sum_{|T| \leq d} \lambda_T \prod_{\substack{i \in T \cap S \\ i \neq n+1}} f_i(1 - f_0) \prod_{\substack{i \in T \setminus S \\ i \neq n+1}} f_0(1 - f_i) \prod_{\substack{i \in T \\ i = n+1}} f_0(1 - f_0).$$

Now, $f_i = \boldsymbol{u}^\top \boldsymbol{a}_i - k_i$ and $f_0 = \boldsymbol{u}^\top \boldsymbol{b} - k_0$ are linear in $\boldsymbol{u}$. The sum is over all multisets of size at most d, so each monomial consists of the product of at most d degree-2 terms of the form $f_i(1 - f_0)$, $f_0(1 - f_i)$, or $f_0(1 - f_0)$. Thus, $\deg(q) \leq 2d$, as desired. □

Proof of Lemma 4.5.10. Let $C \subseteq \mathbb{R}^{n+1}$ be a connected component in the partition established in Theorem 4.5.8, so C can be written as the intersection of at most $14^n(m+2n)^{3n^2}\tau^{5n^2}$ polynomial constraints of degree at most 5. Let $D \subseteq [-U,U]^m$ be a connected component in the partition established in Lemma 4.5.11. By Lemma 4.5.12, there are at most $14^n(m+2n)^{3n^2}\tau^{5n^2}$ polynomials of degree at most 10 partitioning D into connected components such that within each component, $\mathbf{1}[(\boldsymbol{\alpha}(\boldsymbol{u}),\beta(\boldsymbol{u}))\in C]$ is invariant. If we consider the overlay of these polynomial surfaces over all components C, we will get a partition of $[-U,U]^m$ such that *for every* C, $\mathbf{1}[(\boldsymbol{\alpha}(\boldsymbol{u}),\beta(\boldsymbol{u}))\in C]$ is invariant over each connected component of $[-U,U]^m$. Once we have this we are done, since all $\boldsymbol{u}$ in the same connected component of $[-U,U]^m$ will be sent to the same connected component of $\mathbb{R}^{n+1}$ by $(\boldsymbol{\alpha}(\boldsymbol{u}),\beta(\boldsymbol{u}))$, and thus by Theorem 4.5.8 the behavior of branch-and-cut will be invariant.

We now tally up the total number of surfaces. The number of connected components C was given by Warren's theorem and the Milnor-Thom theorem to be

$$O(14^{n(n+1)}(m+2n)^{3n^2(n+1)}\tau^{5n^2(n+1)}),$$

so the total number of degree-10 hypersurfaces is $14^n(m+2n)^{3n^2}\tau^{5n^2}$ times this quantity, which yields the lemma statement. □

Bounding $\mathsf{Pdim}(\{g_{\boldsymbol{u}} : \boldsymbol{u}\in[-U,U]^m\})$ is a direct application of the main theorem of Balcan et al. [2021a] along with standard results bounding the VC dimension of polynomial boundaries [Anthony and Bartlett, 1999].

Theorem 4.5.13. *The pseudo-dimension of the class of tree-size functions $\{g_{\boldsymbol{u}} : \boldsymbol{u}\in[-U,U]^m\}$ on the domain of IPs with $\|A\|_1 \le a$ and $\|\boldsymbol{b}\|_1 \le b$ is $O\left(m\log(abU) + mn^3\log(m+n) + mn^3\log\tau\right)$.*

We generalize the analysis of this section to multiple GMI cuts at the root of the B&C tree in the next section. We show that if K GMI cuts are sequentially applied at the root, the resulting partition of the parameter space is induced by polynomials of degree $O(K^2)$.

4.5.4 Extension to multiple cuts

Linear programming sensitivity for multiple constraints

Lemma 4.5.14. *Let $(\boldsymbol{c},A,\boldsymbol{b})$ be an LP and let M denote the set of its m constraints. Let $\boldsymbol{x}^*_{\mathrm{LP}}$ and z^*_{LP} denote the optimal solution and its objective value, respectively. For $F\subseteq M$, let $A_F\in\mathbb{R}^{|F|\times n}$ and $\boldsymbol{b}_F\in\mathbb{R}^{|F|}$ denote the restrictions of A and $\boldsymbol{b}$ to F. For $k\le n$, $\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_k\in\mathbb{R}^n$, $\beta_1,\dots,\beta_k\in\mathbb{R}$, and $F\subseteq M$ with $|F|=n-k$, let $A_{F,\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_k}\in\mathbb{R}^{n\times n}$ denote the matrix obtained by adding row vectors $\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_k$ to A_F and let $A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\dots,\boldsymbol{\alpha}_k,\beta_k}\in\mathbb{R}^{n\times n}$ be the matrix $A_{F,\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_k}\in\mathbb{R}^{n\times n}$ with the ith column replaced by $\begin{bmatrix}\boldsymbol{b}_F & \beta_1 & \cdots & \beta_k\end{bmatrix}^\top$. There is a set of at most K hyperplanes, nK^nm^n degree-K polynomial hypersurfaces, and nK^nm^{2n} degree-$2K$ polynomial hypersurfaces partitioning $\mathbb{R}^{K(n+1)}$ into connected components such that for each component C, one of the following holds: either (1) $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top\boldsymbol{x}\le\beta_1,\dots,\boldsymbol{\alpha}_K^\top\boldsymbol{x}\le\beta_K)=\boldsymbol{x}^*_{\mathrm{LP}}$, or (2) there is a subset of cuts indexed by $\ell_1,\dots,\ell_k\in[K]$ and a set of constraints $F\subseteq M$ with $|F|=n-k$ such that*

$$\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top\boldsymbol{x}\le\beta_1,\dots,\boldsymbol{\alpha}_K^\top\boldsymbol{x}\le\beta_K)=\left(\frac{\det(A^1_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\dots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k}})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\dots,\boldsymbol{\alpha}_{\ell_k}})},\dots,\frac{\det(A^n_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\dots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k}})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\dots,\boldsymbol{\alpha}_{\ell_k}})}\right),$$

for all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K) \in C$.

Proof. First, if none of $\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K$ separate $\boldsymbol{x}^*_{\mathrm{LP}}$, then

$$\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K) = \boldsymbol{x}^*_{\mathrm{LP}} \text{ and } z^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K) = z^*_{\mathrm{LP}}.$$

The set of all such cuts is given by the intersection of halfspaces in $\mathbb{R}^{K(n+1)}$ given by

$$\bigcap_{j=1}^{K} \left\{ (\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_k, \beta_k) \in \mathbb{R}^{K(n+1)} : \boldsymbol{\alpha}_j^\top \boldsymbol{x}^*_{\mathrm{LP}} \le \beta_j \right\}. \tag{4.13}$$

All other vectors of K cuts contain at least one cut that separates $\boldsymbol{x}^*_{\mathrm{LP}}$, and those cuts therefore pass through $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} \ge \boldsymbol{0}\}$. The new LP optimum is thus achieved at a vertex created by the cuts that separate $\boldsymbol{x}^*_{\mathrm{LP}}$. As in the proof of Theorem 4.5.2, we consider all possible new vertices formed by our set of K cuts. In the case of a single cut, these new vertices necessarily were on edges of $\mathcal{P}$, but now they may lie on higher dimensional faces.

Consider a subset of $k \le n$ cuts that separate $\boldsymbol{x}^*_{\mathrm{LP}}$. Without loss of generality, denote these cuts by $\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_k^\top \boldsymbol{x} \le \beta_k$. We now establish conditions for these k cuts to "jointly" form a new vertex of $\mathcal{P}$. Any vertex created by these cuts must lie on a face f of $\mathcal{P}$ with $\dim(f) = k$ (in the case that $k = n$, the relevant face f with $\dim(f) = n$ is $\mathcal{P}$ itself). Letting M denote the set of m constraints that define $\mathcal{P}$, each dimension-k face f of $\mathcal{P}$ can be identified with a (potentially empty) subset $F \subset M$ of size $n - k$ such that f is precisely the set of all points $\boldsymbol{x}$ such that

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{x} &= b_i \qquad \forall\, i \in F \\ \boldsymbol{a}_i^\top \boldsymbol{x} &\le b_i \qquad \forall\, i \in M \setminus F, \end{aligned}$$

where $\boldsymbol{a}_i$ is the ith row of A. Let $A_F \in \mathbb{R}^{n-k \times n}$ denote the restriction of A to only the rows in F, and let $\boldsymbol{b}_F \in \mathbb{R}^{n-k}$ denote the entries of $\boldsymbol{b}$ corresponding to the constraints in F. Consider removing the inequality constraints defining the face. The intersection of the cuts $\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_k^\top \boldsymbol{x} \le \beta_k$ and this unbounded surface (if it exists) is precisely the solution to the system of n linear equations

$$\begin{aligned} A_F \boldsymbol{x} &= \boldsymbol{b}_F \\ \boldsymbol{\alpha}_1^\top \boldsymbol{x} &= \beta_1 \\ &\vdots \\ \boldsymbol{\alpha}_k^\top \boldsymbol{x} &= \beta_k. \end{aligned}$$

Let $A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k} \in \mathbb{R}^{n \times n}$ denote the matrix obtained by adding row vectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ to A_F, and let $A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k} \in \mathbb{R}^{n \times n}$ denote the matrix $A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k}$ where the ith column is replaced by

$$\begin{bmatrix} \boldsymbol{b}_F \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} \in \mathbb{R}^n.$$

By Cramer's rule, the solution to this system is given by

$$\boldsymbol{x} = \left(\frac{\det(A^1_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})}, \ldots, \frac{\det(A^n_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})} \right),$$

and the value of the objective at this point is

$$\boldsymbol{c}^\top \boldsymbol{x} = \sum_{i=1}^n c_i \cdot \frac{\det(A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})}.$$

Now, to ensure that the unique intersection point $\boldsymbol{x}$ (1) exists and (2) actually lies on f (or simply lies in $\mathcal{P}$, in the case that $F = \emptyset$) , we stipulate that it satisfies the inequality constraints in $M \setminus F$. That is,

$$\sum_{j=1}^n a_{ij} \frac{\det(A^1_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})} \leq b_i \tag{4.14}$$

for every $i \in M \setminus F$. If $\boldsymbol{\alpha}_1, \beta_1 \ldots, \boldsymbol{\alpha}_k, \beta_k$ satisfies any of these constraints, it must be that $\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k}) \neq 0$, which guarantees that $A_F \boldsymbol{x} = \boldsymbol{b}_F, \boldsymbol{\alpha}_1^\top \boldsymbol{x} = \beta_1, \ldots, \boldsymbol{\alpha}_k^\top \boldsymbol{x} = \beta_k$ indeed has a unique solution. Now, $\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})$ is a polynomial in $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ of degree $\leq k$, since it is multilinear in each coefficient of each $\boldsymbol{\alpha}_\ell$, $\ell = 1, \ldots, k$. Similarly, $\det(A^1_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})$ is a polynomial in $\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_k, \beta_k$ of degree $\leq k$, again because it is multilinear in each cut parameter. Hence, the boundary each constraint of the form given by Equation 4.14 is a polynomial of degree at most k.

The collection of these polynomials for every k, every subset of $\{\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K\}$ of size k, and every face of $\mathcal{P}$ of dimension k, along with the hyperplanes determining separation constraints (Equation 4.13), partition $\mathbb{R}^{K(n+1)}$ into connected components such that for all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K)$ within a given connected component, there is a fixed subset of K and a fixed set of faces of $\mathcal{P}$ such that the cuts with indices in that subset intersect every face in the set at a common vertex.

Now, consider a single connected component, denoted by C. Let $f_1, \ldots, f_\ell$ denote the faces intersected by vectors of cuts in C, and let (without loss of generality) $1, \ldots, k$ denote the subset of cuts that intersect these faces. Let $F_1, \ldots, F_\ell \subset M$ denote the sets of constraints that are binding at each of these faces, respectively. For each pair f_p, f_q, consider the surface

$$\sum_{i=1}^n c_i \cdot \frac{\det(A^i_{F_p,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})}{\det(A_{F_p,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})} = \sum_{i=1}^n c_i \cdot \frac{\det(A^i_{F_q,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k})}{\det(A_{F_q,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k})},$$

which can be equivalently written as

$$\sum_{i=1}^n c_i \cdot \det(A^i_{F_p,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k}) \det(A_{F_q,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k}) = \sum_{i=1}^n c_i \cdot \det(A^i_{F_q,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k}) \det(A_{F_p,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k}). \tag{4.15}$$

This is a degree-$2k$ polynomial hypersurface in $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K) \in \mathbb{R}^{K(n+1)}$. This hypersurface is precisely the set of all cut vectors for which the LP objective achieved at the vertex on face f_p is equal to the LP objective value achieved at the vertex on face f_q. The collection of these surfaces for each p, q partitions C into further connected components. Within each of these connected components, the face containing the vertex that maximizes the objective is invariant, and the subset of cuts passing through that vertex is invariant. If $F \subseteq M$ is the set of binding constraints representing this face, and $\ell_1, \ldots, \ell_k \in [K]$ represent the subset of cuts intersecting this face, $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K)$ and $z^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K)$ have the closed forms:

$$\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K) = \left(\frac{\det(A^1_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k}})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k}})}, \ldots, \frac{\det(A^n_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k}})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k}})}\right),$$

and

$$z^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K) = \sum_{i=1}^n c_i \cdot \frac{\det(A^i_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k}})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k}})}.$$

for all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K)$ within this component. We now count the number of surfaces used to obtain our decomposition. First, we added K hyperplanes encoding separation constraints for each of the K cuts (Equation 4.13). Then, for every subset $S \subseteq K$ of size $\le n$, and for every face F of $\mathcal{P}$ with $\dim(F) = |S|$, we first considered at most $|M \setminus F| \le m$ degree-$\le K$ polynomial hypersurfaces representing decision boundaries for when cuts in S intersected that face (Equation 4.14). The number of k-dimensional faces of $\mathcal{P}$ is at most $\binom{m}{n-k} \le m^{n-k} \le m^{n-1}$, so the total number of these hypersurfaces is at most $(\binom{K}{0} + \cdots + \binom{K}{n})m^n \le nK^n m^n$. Finally, we considered a degree-$2K$ polynomial hypersurface for every subset of cuts and every pair of faces with degree equal to the size of the subset, of which there are at most $nK^n\binom{m^n}{2} \le nK^n m^{2n}$. □

B&C sensitivity with multiple cutting planes

We can similarly derive a multi-cut version of Lemma 4.5.4 that controls $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma)$ for any set of branching constraints. We use the following notation. Let $(\boldsymbol{c}, A, \boldsymbol{b})$ be an LP and let M denote the set of its m constraints. For $F \subseteq M \cup \sigma$, let $A_{F,\sigma} \in \mathbb{R}^{|F| \times n}$ and $\boldsymbol{b}_{F,\sigma} \in \mathbb{R}^{|F|}$ denote the restrictions of A_σ and $\boldsymbol{b}_\sigma$ to F. For $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k \in \mathbb{R}^n$, $\beta_1, \ldots, \beta_k \in \mathbb{R}$, and $F \subseteq M \cup \sigma$ with $|F| = n - k$, let $A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\sigma} \in \mathbb{R}^{n \times n}$ denote the matrix obtained by adding row vectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ to $A_{F,\sigma}$ and let $A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k,\sigma} \in \mathbb{R}^{n \times n}$ be the matrix $A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\sigma} \in \mathbb{R}^{n \times n}$ with the ith column replaced by $\begin{bmatrix} \boldsymbol{b}_{F,\sigma} & \beta_1 & \cdots & \beta_k \end{bmatrix}^\top$.

Corollary 4.5.15. *Fix an IP $(\boldsymbol{c}, A, \boldsymbol{b})$. There is a set of at most K hyperplanes, $nK^n(m + 2n)^n \tau^{3n}$ degree-K polynomial hypersurfaces, and $nK^n(m + 2n)^{2n}\tau^{3n}$ degree-$2K$ polynomial hypersurfaces partitioning $\mathbb{R}^{K(n+1)}$ into connected components such that for each component C and every $\sigma \subseteq \mathcal{BC}$, one of the following holds: either (1) $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma) = \boldsymbol{x}^*_{\text{LP}}(\sigma)$, or (2) there is a subset of cuts indexed by $\ell_1, \ldots, \ell_k \in [K]$ and a set of constraints $F \subseteq M \cup \sigma$ with $|F| = n - k$ such that*

$$\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma) = \left(\frac{\det(A^1_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k},\sigma})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\sigma})}, \ldots, \frac{\det(A^n_{F,\boldsymbol{\alpha}_{\ell_1},\beta_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\beta_{\ell_k},\sigma})}{\det(A_{F,\boldsymbol{\alpha}_{\ell_1},\ldots,\boldsymbol{\alpha}_{\ell_k},\sigma})}\right),$$

for all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K) \in C$.

Proof. The exact same reasoning in the proof of Lemma 4.5.14 applies. We still have K hyperplanes. Now, for each σ, for each subset $S \subseteq K$ with $|S| \leq n$, and for every face F of $\mathcal{P}(\sigma)$ with $\dim(F) = |S|$, we have at most m degree-K polynomial hypersurfaces. The number of k-dimensional faces of $\mathcal{P}(\sigma)$ is at most $\binom{m+|\sigma|}{n-k} \leq (m+2n)^{n-1}$, so the total number of these hypersurfaces is at most $nK^n(m+2n)^n\tau^{3n}$. Finally, for every σ, we considered a degree-$2K$ polynomal hypersurfaces for every subset of cuts and every pair of faces with degree equal to the size of the subset, of which there are at most $nK^n(m+2n)^{2n}\tau^{3n}$, as desired. ☐

We now refine the decomposition obtained in Lemma 4.5.4 so that the branching constraints added at each step of branch-and-cut are invariant within a region. For ease of exposition, we assume that branch-and-cut uses a lexicographic variable selection policy. This means that the variable branched on at each node of the search tree is fixed and given by the lexicographic ordering $x_1, \ldots, x_n$. Generalizing the argument to work for other policies, such as the product scoring rule, can be done as in the single-cut case.

Lemma 4.5.16. *Suppose branch-and-cut uses a lexicographic variable selection policy. Then, there is a set of of at most K hyperplanes, $3n^2K^n(m+2n)^n\tau^{3n}$ degree-K polynomial hypersurfaces, and $nK^n(m+2n)^{2n}\tau^{3n}$ degree-$2K$ polynomial hypersurfaces partitioning $\mathbb{R}^{n+1}$ into connected components such that within each connected component, the branching constraints used at every step of branch-and-cut are invariant.*

Proof. Fix a connected component C in the decomposition established in Corollary 4.5.15. Then, by Corollary 4.5.15, for each σ, either $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K, \sigma) = \boldsymbol{x}^*_{\mathrm{LP}}(\sigma)$ or there exists cuts (without less of generality) labeled by indices $1, \ldots, k \in [K]$ and there exists $F \subseteq M \cup \sigma$ such that

$$\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K, \sigma)[i] = \frac{\det(A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k,\sigma})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\sigma})}$$

for all $(\boldsymbol{\alpha}, \beta) \in C$ and all $i \in [n]$. Now, if we are at a stage in the branch-and-cut tree where σ is the list of branching constraints added so far, and the ith variable is being branched on next, the two constraints generated are

$$x_i \leq \lfloor \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K, \sigma)[i] \rfloor \text{ and } x_i \geq \lceil \boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K, \sigma)[i] \rceil,$$

respectively. If C is a component where $\boldsymbol{x}^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K, \sigma) = \boldsymbol{x}^*_{\mathrm{LP}}(\sigma)$, then there is nothing more to do, since the branching constraints at that point are trivially invariant over $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K) \in C$. Otherwise, in order to further decompose C such that the right-hand-side of these constraints are invariant for every σ and every $i = 1, \ldots, n$, we add the two decision boundaries given by

$$k \leq \frac{\det(A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k,\sigma})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\sigma})} \leq k+1$$

for every i, σ, and every integer $k = 0, \ldots, \tau - 1$, where $\tau = \lceil \max_{\boldsymbol{x} \in \mathcal{P}} \|\boldsymbol{x}\|_\infty \rceil$. This ensures that within every connected component of C induced by these boundaries (degree-K polynomial hypersurfaces),

$$\left\lfloor \boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] \right\rfloor = \left\lfloor \frac{\det(A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k,\sigma})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\sigma})} \right\rfloor$$

and

$$\left\lceil \boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \le \beta, \sigma)[i] \right\rceil = \left\lceil \frac{\det(A^i_{F,\boldsymbol{\alpha}_1,\beta_1,\ldots,\boldsymbol{\alpha}_k,\beta_k,\sigma})}{\det(A_{F,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\sigma})} \right\rceil$$

are invariant, so the branching constraints added by, for example, a lexicographic branching rule, are invariant. For a fixed σ, there are two hypersurfaces for every subset $S \subseteq [K]$, every $F \subseteq M \cup \sigma$ corresponding to a $|S|$-dimensional face of $\mathcal{P}(\sigma)$, and every $i = 1, \ldots, n$, for a total of at most $2n^2K^n\binom{m+|\sigma|}{|S|} \le 2n^2K^n(m+2n)^n$. Summing over all reduced σ, we get a total of $2n^2K^n(m+2n)^n\tau^{3n}$ hypersurfaces. Adding these hypersurfaces to the set of hypersurfaces established in Corollary 4.5.15 yields the lemma statement. $\square$

Now, as in the single-cut case, we consider the constraints that ensure that all cuts are valid. Let $\mathcal{V} \subseteq \mathbb{R}^{K(n+1)}$ denote the set of all vectors of valid K cuts. As before, $\mathcal{V}$ is a polyhedron, since we may write

$$\mathcal{V} = \bigcap_{k=1}^{K} \bigcap_{\boldsymbol{x}_{\text{IH}} \in \mathcal{P}_{\text{IH}}} \left\{ (\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_k) \in \mathbb{R}^{K(n+1)} : \boldsymbol{\alpha}_k^\top \boldsymbol{x}_{\text{IH}} \le \beta_k \right\}.$$

We now refine our decomposition further to control the integrality of the various LP solutions at each node of branch-and-cut.

Lemma 4.5.17. *Given an IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there is a set of at most $2K\tau^n$ hyperplanes, $4n^2K^n(m+2n)^n\tau^{4n}$ degree-K polynomial hypersurfaces, and $nK^n(m+2n)^{2n}\tau^{3n}$ degree-$2K$ polynomial hypersurfaces partitioning $\mathbb{R}^{K(n+1)}$ into connected components such that for each component C, and each $\sigma \subseteq \mathcal{BC}$,*

$$\mathbf{1}\left[\boldsymbol{x}^*_{\text{LP}}\left(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma\right) \in \mathbb{Z}^n\right]$$

is invariant for all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K) \in C$.

Proof. Fix a connected component C in the decomposition that includes the facets defining $\mathcal{V}$ and the surfaces obtained in Lemma 4.5.16. For all $\sigma \in \mathcal{BC}$, $\boldsymbol{x}_{\text{I}} \in \mathcal{P}_{\text{I}}$, and $i = 1, \ldots, n$, consider the surface

$$\boldsymbol{x}^*_{\text{LP}}\left(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma\right)[i] = \boldsymbol{x}_{\text{I}}[i]. \tag{4.16}$$

This surface is a polynomial hypersurface of degree at most K, due to Corollary 4.5.15. Clearly, within any connected component of C induced by these hyperplanes, for every σ and $\boldsymbol{x}_{\text{I}} \in \mathcal{P}_{\text{I}}$, $\mathbf{1}[\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma) = \boldsymbol{x}_{\text{I}}]$ is invariant. Finally, if $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma) \in \mathbb{Z}^n$ for some K cuts $\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K$ within a given connected component, $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma) = \boldsymbol{x}_{\text{I}}$ for some $\boldsymbol{x}_{\text{I}} \in \mathcal{P}_{\text{IH}}(\sigma) \subseteq \mathcal{P}_{\text{I}}$, which means that $\boldsymbol{x}^*_{\text{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K, \sigma) = \boldsymbol{x}_{\text{I}} \in \mathbb{Z}^n$ *for all* vectors of K cuts $\boldsymbol{\alpha}_1^\top \boldsymbol{x} \le \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \le \beta_K$ in that connected component.

We now count the number of hyperplanes given by Equation 4.16. For each σ, there are nK^n possible subsets of cut indices and at most $(m+2n)^{n-1}$ binding face constraints $F \subseteq$

$M \cup \sigma$ defining the formula of Corollary 4.5.15. For each subset-face pair, there are $n|\mathcal{P}_{\mathrm{I}}| \leq n\tau^n$ degree-K polynomial hypersurfaces given by Equation 4.16. So the total number of such hypersurfaces over all σ is at most $\tau^{3n}n^2K^n(m+2n)^{n-1}\tau^n$. The number of facets defining $\mathcal{V}$ is at most $K|\mathcal{P}_{\mathrm{I}}| \leq K\tau^n$. Adding these to the counts obtained in Lemma 4.5.16 yields the final tallies in the lemma statement. □

At this point, as in the single-cut case, if the bounding aspect of branch-and-cut is suppressed, our decomposition yields connected components over which the branch-and-cut tree built is invariant. We now prove our main structural theorem for B&C as a function of multiple cutting planes at the root.

Theorem 4.5.18. *Given an IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there is a set of at most $O(12^n n^{2n} K^{2n^2}(m+2n)^{2n^2}\tau^{5n^2})$ polynomial hypersurfaces of degree at most $2K$ partitioning $\mathbb{R}^{K(n+1)}$ into connected components such that the branch-and-cut tree built after adding the K cuts $\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_k^\top \boldsymbol{x} \leq \beta_k$ at the root is invariant over all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K)$ within a given component. In particular, $f_{\boldsymbol{c},A,\boldsymbol{b}}(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K)$ is invariant over each connected component.*

Proof. Fix a connected component C in the decomposition induced by the set of hyperplanes, degree-K hypersurfaces, and degree-$2K$ hypersurfaces established in Lemma 4.5.17. Let

$$Q_1, \ldots, Q_{i_1}, I_1, Q_{i_1+1}, \ldots, Q_{i_2}, I_2, Q_{i_2+1}, \ldots \tag{4.17}$$

denote the nodes of the tree branch-and-cut creates, in order of exploration, under the assumption that a node is pruned if and only if either the LP at that node is infeasible or the LP optimal solution is integral (so the "bounding" of branch-and-bound is suppressed). Here, a node is identified by the list σ of branching constraints added to the input IP. Nodes labeled by Q are either infeasible or have fractional LP optimal solutions. Nodes labeled by I have integral LP optimal solutions and are candidates for the incumbent integral solution at the point they are encountered. (The nodes are functions of $\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K$, as are the indices $i_1, i_2, \ldots$.) By Lemma 4.5.17, this ordered list of nodes is invariant for all $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_k) \in C$.

Now, given an node index ℓ, let $I(\ell)$ denote the incumbent node with the highest objective value encountered up until the ℓth node searched by B&C, and let $z(I(\ell))$ denote its objective value. For each node Q_ℓ, let σ_ℓ denote the branching constraints added to arrive at node Q_ℓ. The hyperplane

$$z^*_{\mathrm{LP}}\left(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_K^\top \boldsymbol{x} \leq \beta_K, \sigma_\ell\right) = z(I(\ell)) \tag{4.18}$$

(which is a hyperplane due to Corollary 4.5.15) partitions C into two subregions. In one subregion, $z^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_k^\top \boldsymbol{x} \leq \beta_k, \sigma_\ell) \leq z(I(\ell))$, that is, the objective value of the LP optimal solution is no greater than the objective value of the current incumbent integer solution, and so the subtree rooted at Q_ℓ is pruned. In the other subregion, $z^*_{\mathrm{LP}}(\boldsymbol{\alpha}_1^\top \boldsymbol{x} \leq \beta_1, \ldots, \boldsymbol{\alpha}_k^\top \boldsymbol{x} \leq \beta_k, \sigma_\ell) > z(I(\ell))$, and Q_ℓ is branched on further. Therefore, within each connected component of C induced by all hyperplanes given by Equation 4.18 for all ℓ, the set of node within the list (4.17) that are pruned is invariant. Combined with the surfaces established in Lemma 4.5.17, these hyperplanes partition $\mathbb{R}^{K(n+1)}$ into connected components such that as $(\boldsymbol{\alpha}_1, \beta_1 \ldots, \boldsymbol{\alpha}_K, \beta_K)$ varies within a given component, the tree built by branch-and-cut is invariant.

Finally, we count the total number of surfaces inducing this partition. Unlike the counting stages of the previous lemmas, we will first have to count the number of connected components induced by the surfaces established in Lemma 4.5.17. This is because the ordered list of nodes explored by branch-and-cut (4.17) can be different across each component, and the hyperplanes given by Equation 4.18 depend on this list. From Lemma 4.5.17 we have $6n^2K^n(m+2n)^{2n}\tau^{4n}$ polynomial hypersurfaces of degree $\leq 2K$. The set of all $(\boldsymbol{\alpha}_1, \beta_1, \ldots \boldsymbol{\alpha}_K, \beta_k) \in \mathbb{R}^{K(n+1)}$ such that $(\boldsymbol{\alpha}_1, \beta_1, \ldots, \boldsymbol{\alpha}_K, \beta_K)$ lies on the boundary of any of these surfaces is precisely the zero set of the product of all polynomials defining these surfaces. Denote this product polynomial by p. The degree of the product polynomial is the sum of the degrees of $6n^2K^n(m+2n)^{2n}\tau^{4n}$ polynomials of degree $\leq 2K$, which is at most $2K \cdot 6Kn^2K^n(m+2n)^{2n}\tau^{4n} = 12n^2K^{n+2}(m+2n)^{2n}\tau^{4n}$. By Warren's theorem, the number of connected components of $\mathbb{R}^{n+1} \setminus \{(\boldsymbol{\alpha}, \beta) : p(\boldsymbol{\alpha}, \beta) = 0\}$ is $O((12n^2K^{n+2}(m+2n)^{2n}\tau^{4n})^{n-1})$, and by the Milnor-Thom theorem, the number of connected components of $\{(\boldsymbol{\alpha}, \beta) : p(\boldsymbol{\alpha}, \beta) = 0\}$ is $O((12n^2K^{n+2}(m+2n)^{2n}\tau^{4n})^{n-1})$ as well. So, the number of connected components induced by the surfaces in Lemma 4.5.17 is $O(12^n n^{2n} K^{2n^2}(m+2n)^{2n^2}\tau^{4n^2})$. For every connected component C in Lemma 4.5.17, the closed form of $z^*_{\mathrm{LP}}(\boldsymbol{\alpha}^\top \boldsymbol{x} \leq \beta, \sigma_\ell)$ is already determined due to Corollary 4.5.15, and so the number of hyperplanes given by Equation 4.18 is at most the number of possible $\sigma \subseteq \mathcal{BC}$, which is at most τ^{3n}. So across all connected components C, the total number of hyperplanes given by Equation 4.18 is $O(12^n n^{2n} K^{2n^2}(m+2n)^{2n^2}\tau^{5n^2})$. Finally, adding this to the surface-counts established in Lemma 4.5.17 yields the theorem statement. $\square$

Multiple GMI cuts at the root

In this section we extend our results to allow for multiple GMI cuts at the root of the B&C tree. These cuts can be added simultaneously, sequentially, or in rounds. If GMI cuts $\boldsymbol{u}_1, \boldsymbol{u}_2$ are added simultaneously, both of them have the same dimension and are defined in the usual way. If GMI cuts $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ are added sequentially, $\boldsymbol{u}_2$ has one more entry than $\boldsymbol{u}_1$. This is because when cuts are added sequentially, the LP relaxation is re-solved after the addition of the first cut, and the second cut has a multiplier for all original constraints as well as for the first cut (this ensures that the second cut can be chosen in a more informed manner). If K cuts are made at the root, they can be added in sequential rounds of simultaneous cuts. In the following discussion, we focus on the case where all K cuts are added sequentially—the other cases can be viewed as instantiations of this (as in Section 4.3).

To prove an analogous result for multiple GMI cuts (in sequence, that is, each successive GMI cut has one more parameter than the previous), we combine the reasoning used in the single-GMI-cut case with some technical observations in Section 4.3.

Lemma 4.5.19. *Consider the family of K sequential GMI cuts parameterized by*

$$\boldsymbol{u}_1 \in [-U, U]^m, \boldsymbol{u}_2 \in [-U, U]^{m+1}, \ldots, \boldsymbol{u}_K \in [-U, U]^{m+K-1}.$$

For any IP $(\boldsymbol{c}, A, \boldsymbol{b})$, there are at most

$$O\left(nK(1+U)^{2K} \|A\|_1 \|\boldsymbol{b}\|_1\right)$$

degree-K polynomial hypersurfaces and

$$2^{O(n^2)} K^{O(n^3)} (m+2n)^{O(n^3)} \tau^{O(n^3)}$$

degree-$4K^2$ polynomial hypersurfaces partitioning $[-U,U]^m \times \cdots \times [-U,U]^{m+K-1}$ connected components such that the B&C tree built after sequentially adding the GMI cuts defined by $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K$ is invariant over all $(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K)$ within a single component.

Proof. We start with the setup used in Section 4.3 to prove similar results for sequential Chvátal-Gomory cuts. Let $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n \in \mathbb{R}^m$ be the columns of A. We define the following augmented columns $\widetilde{\boldsymbol{a}}_i^1 \in \mathbb{R}^m,\ldots,\widetilde{\boldsymbol{a}}_i^K \in \mathbb{R}^{m+K-1}$ for each $i \in [n]$, and the augmented constraint vectors $\widetilde{\boldsymbol{b}}^1 \in \mathbb{R}^m,\ldots,\widetilde{\boldsymbol{b}}^K \in \mathbb{R}^{m+K-1}$ via the following recurrences:

$$\widetilde{\boldsymbol{a}}_i^1 = \boldsymbol{a}_i$$
$$\widetilde{\boldsymbol{a}}_i^k = \begin{bmatrix} \widetilde{\boldsymbol{a}}_i^{k-1} \\ \boldsymbol{u}_{k-1}^\top \widetilde{\boldsymbol{a}}_i^{k-1} \end{bmatrix}$$

and

$$\widetilde{\boldsymbol{b}}^1 = \boldsymbol{b}$$
$$\widetilde{\boldsymbol{b}}^k = \begin{bmatrix} \widetilde{\boldsymbol{b}}^{k-1} \\ \boldsymbol{u}_{k-1}^\top \widetilde{\boldsymbol{b}}^{k-1} \end{bmatrix}$$

for $k = 2,\ldots,K$. In other words, $\widetilde{\boldsymbol{a}}_i^k$ is the ith column of the constraint matrix of the IP and $\widetilde{\boldsymbol{b}}^k$ is the constraint vector after applying cuts $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{k-1}$. A straightforward induction shows that for each $k \in [K]$,

$$\left\lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k \right\rfloor \in \left[-(1+U)^k \left\|\boldsymbol{a}_i\right\|_1, (1+U)^k \left\|\boldsymbol{a}_i\right\|_1 \right]$$

and

$$\left\lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k \right\rfloor \in \left[-(1+U)^k \left\|\boldsymbol{b}\right\|_1, (1+U)^k \left\|\boldsymbol{b}\right\|_1 \right].$$

Now, as in the single-GMI-cut setting, consider the surfaces

$$\left\lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k \right\rfloor = \ell_i \iff \ell_i \le \boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k < \ell_i + 1 \tag{4.19}$$

and

$$\left\lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k \right\rfloor = \ell_0 \iff \ell_i \le \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k < \ell_0 + 1 \tag{4.20}$$

for every i, k, and every integer $\ell_i \in [-(1+U)^k \left\|\boldsymbol{a}_i\right\|_1, (1+U)^k \left\|\boldsymbol{a}_i\right\|_1] \cap \mathbb{Z}$ and every integer $\ell_0 \in [-(1+U)^k \left\|\boldsymbol{b}\right\|_1, (1+U)^k \left\|\boldsymbol{b}\right\|_1] \cap \mathbb{Z}$. In addition, consider the surfaces

$$\boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k - \ell_i = \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k - \ell_0 \tag{4.21}$$

for each i, k, ℓ_i, ℓ_0. As observed in Section 4.3, $\boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k$ is a polynomial in

$$\boldsymbol{u}_1[1],\ldots,\boldsymbol{u}_1[m],\boldsymbol{u}_2[1],\ldots,\boldsymbol{u}_2[m+1],\ldots,\boldsymbol{u}_k[1],\ldots,\boldsymbol{u}_k[m+k-1]$$

of degree at most k (as is $\boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k$), so surfaces 4.19, 4.20, and 4.21 are all degree-K polynomial hypersurfaces for all i, k. Within any connected component of $[-U,U]^m \times \cdots \times [-U,U]^{m+K-1}$

induced by these hypersurfaces, $\lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k \rfloor$ and $\lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k \rfloor$ are constant. Furthermore $\mathbf{1}[f_i^k \leq f_0^k]$ is invariant for every i, k, where $f_i^k = \boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k - \lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{a}}_i^k \rfloor$ and $f_0^k = \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k - \lfloor \boldsymbol{u}_k^\top \widetilde{\boldsymbol{b}}^k \rfloor$.

Now, fix a connected component $D \subseteq [-U, U]^m \times \cdots \times [-U, U]^{m+K-1}$ induced by the above hypersurfaces, and let $C \subseteq \mathbb{R}^{K(n+1)}$ be the intersection of q polynomial inequalities of degree at most d. Consider a single degree-d polynomial inequality in $K(n+1)$ variables $y_1, \ldots, y_{K(n+1)}$, which can be written as

$$\sum_{\substack{T \sqsubseteq [K(n+1)] \\ |T| \leq d}} \lambda_T \prod_{j \in T} y_j = \sum_{\substack{T_1, \ldots, T_K \sqsubseteq [n+1] \\ |T_1| + \cdots + |T_K| \leq d}} \lambda_{T_1, \ldots, T_K} \prod_{j_1 \in T_1} y_{j_1} \cdots \prod_{j_K \in T_K} y_{j_K} \leq \gamma.$$

Now, the sets $S_1, \ldots, S_K$ defined by $S_k = \{i : f_i^k \leq f_0^k\}$ are fixed within D, so we can write this as

$$\sum_{\substack{T_1, \ldots, T_K \sqsubseteq [n+1] \\ |T_1| + \cdots + |T_K| \leq d}} \lambda_{T_1, \ldots, T_K} \prod_{k=1}^{K} \left[\prod_{\substack{j \in T_k \cap S_k \\ j \neq n+1}} f_j^k (1 - f_0^k) \prod_{\substack{j \in T_k \setminus S_k \\ j \neq n+1}} f_0^k (1 - f_j^k) \prod_{\substack{j \in T_k \\ j = n+1}} f_0^k (1 - f_0^k) \right] \leq \gamma.$$

We have that f_j^k and f_0^k are degree-k polynomials in $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$. Since the sum is over all multisets $T_1, \ldots, T_K$ such that $|T_1| + \cdots + |T_K| \leq d$, there are at most d terms across the products, each of the form $f_j^k (1 - f_0)^k$, $f_0^k (1 - f_j^k)$, or $f_0^k (1 - f_0)^k$. Therefore, the left-hand-side is a polynomial of degree at most $2dK$, and if $C \subseteq \mathbb{R}^{K(n+1)}$ is the intersection of q polynomial inequalities each of degree at most d, the set

$$\{(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in D : (\boldsymbol{\alpha}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K), \beta(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)) \in C\} \subseteq [-U, U]^m \times \cdots \times [-U, U]^{m+K-1}$$

can be expressed as the intersection of q degree-$2dK$ polynomial inequalities.

To finish, we run this process for every connected component $C \subseteq \mathbb{R}^{K(n+1)}$ in the partition established by Theorem 4.5.18. This partition consists of $O(12^n n^{2n} K^{2n^2} (m + 2n)^{2n^2} \tau^{5n^2})$ degree-$2K$ polynomials over $\mathbb{R}^{K(n+1)}$. By Warren's theorem and the Milnor-Thom theorem, these polynomials partition $\mathbb{R}^{K(n+1)}$ into $O(12^{n(n+1)} n^{2n(n+1)} K^{2n^2(n+1)} (m+2n)^{2n^2(n+1)} \tau^{5n^2(n+1)})$ connected components. Running the above argument for each of these connected components of $\mathbb{R}^{K(n+1)}$ yields a total of

$$\begin{aligned} &O\left(12^{n(n+1)} n^{2n(n+1)} K^{2n^2(n+1)} (m + 2n)^{2n^2(n+1)} \tau^{5n^2(n+1)}\right) \cdot O\left(12^n n^{2n} K^{2n^2} (m + 2n)^{2n^2} \tau^{5n^2}\right) \\ &\quad = 2^{O(n^2)} K^{O(n^3)} (m + 2n)^{O(n^3)} \tau^{O(n^3)} \end{aligned}$$

polynomials of degree $4K^2$. Finally, we count the surfaces of the form (4.19), (4.20), and (4.21). The total number of degree-K polynomials of type 4.19 is at most $O(nK(1 + U)^K \|A\|_1)$, the total number of degree-k polynomials of type 4.20 is $O(K(1 + U)^K \|\boldsymbol{b}\|_1)$, and the total number of degree-K polynomials of type 4.21 is $O(nK(1 + U)^{2K} \|A\|_1 \|\boldsymbol{b}\|)$. Summing these counts yields the desired number of surfaces in the lemma statement.

In any connected component of $[-U, U]^m$ determined by these surfaces, $\mathbf{1}[(\boldsymbol{\alpha}(\boldsymbol{u}), \beta(\boldsymbol{u})) \in C]$ is invariant for every connected component $C \subseteq \mathbb{R}^{K(n+1)}$ in the partition of $\mathbb{R}^{K(n+1)}$ established in Theorem 4.5.18. This means that the tree built by branch-and-cut is invariant, which concludes the proof. □

Finally, applying the main result of Balcan et al. [2021a] to Lemma 4.5.19, we get the following pseudo-dimension bound for the class of K sequential GMI cuts at the root of the B&C tree.

Theorem 4.5.20. *For*

$$\boldsymbol{u}_1 \in [-U, U]^m, \boldsymbol{u}_2 \in [-U, U]^{m+1}, \ldots, \boldsymbol{u}_K \in [-U, U]^{m+K-1},$$

let $g_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K}(\boldsymbol{c}, A, \boldsymbol{b})$ denote the number of nodes in the tree B&C builds given the input $(\boldsymbol{c}, A, \boldsymbol{b})$ after sequentially applying the GMI cuts defined by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ at the root. The pseudo-dimension of the set of functions

$$\left\{g_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K} : (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in [-U, U]^m \times \cdots \times [-U, U]^{m+K-1}\right\}$$

on the domain of IPs with $\|A\|_1 \leq a$ and $\|\boldsymbol{b}\|_1 \leq b$ is

$$O\left(mK^3 \log U + mn^3K^2 \log(mnK\tau) + mK^2 \log(ab)\right).$$

Part II

Mechanism Design with Side Information with Applications to Combinatorial Markets

Chapter 5

Multidimensional Mechanism Design with Side Information

Mechanism design is a high-impact branch of economics, computer science, and operations research that studies the implementation of socially desirable outcomes among strategic self-interested agents. Major real-world use cases include combinatorial auctions [Cramton et al., 2006] (*e.g.*, for strategic sourcing [Sandholm, 2013, Hohner et al., 2003] and radio spectrum auctions [Cramton, 2013, Bichler and Goeree, 2017, Leyton-Brown et al., 2017]), matching markets [Roth, 2018] (*e.g.*, for housing allocation and ridesharing), project fundraisers, and many more. The two most commonly studied objectives in mechanism design are *welfare* and *revenue*. In many settings, welfare maximization, or *efficiency*, is achieved by the classic Vickrey-Clarke-Groves (VCG) mechanism [Vickrey, 1961, Clarke, 1971, Groves, 1973]. Revenue maximization is a significantly more elusive problem that is only understood in very special cases. The seminal work of Myerson [1981] 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. It is known that welfare and revenue are generally at odds and optimizing one can come at a great expense of the other [Ausubel and Milgrom, 2006, Abhishek and Hajek, 2010, Anshelevich et al., 2016, Kleinberg and Yuan, 2013, Diakonikolas et al., 2012].

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. Machine learning approaches that exploit the proliferation of agent data have in particular witnessed a great deal of success both in theory [Conitzer and Sandholm, 2002, Likhodedov and Sandholm, 2004, Morgenstern and Roughgarden, 2016, Medina and Vassilvitskii, 2017, Balcan et al., 2018d, 2005] and in practice [Edelman et al., 2007, Sandholm, 2007, Walsh et al., 2008, Dütting et al., 2019, Sandholm, 2013]. 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 (though we show how to apply the techniques developed in this chapter to that setting as well). A nascent line of work (part of a larger agenda on *learning-augmented algorithms* [Mitzenmacher and Vassilvitskii, 2022]) has begun to examine the challenge of improving the performance of classical mechanisms with strategic participants when the designer has access to predictions about the agents, but only for fairly specific problem settings [Xu and Lu, 2022, Banerjee et al., 2022, Balkanski et al., 2023, Gkatzelis et al., 2022, Agrawal et al., 2022]. Algorithm and mechanism design with predictions takes a *beyond worst case* perspective on performance analysis, the primary motivation being the access to machine-learning predictions about the problem instance that can greatly improve performance (*e.g.,* run-time, memory, revenue, welfare, fairness, *etc.*) beyond what is possible in the worst case. This is in contrast to the *worst-case* perspective that has been traditional in classical algorithm design and theoretical computer science. We contribute to this budding area of *mechanism design with predictions* (also called *learning-augmented mechanism design*) with a new general side-information-dependent 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. A formal description of the model is in Section 10.2.1. *(1) An auction designer sets a bidder-specific item reserve price of $10 based on historical knowledge that the bidder is a high-spending bidder and the observation that all other bids received for that item are tightly clustered around $10. (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; we include many more such examples throughout to reinforce central concepts. Our methodology allows us to derive welfare and revenue guarantees under different assumptions on the veracity of the side information. Our model of side information involves a general and flexible language wherein nearly any claim of the form *"the joint type profile of the agents satisfies property P"* can be expressed and meaningfully used. We study some other forms of side information as well which we describe further next.

Our contributions

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

Prediction model, type spaces, and weakest-type VCG In Section 10.2.1, we introduce our model of predictors and formally define the components of multidimensional mechanism design. The abstraction of multidimensional mechanism design is a rich language that allows our theory to apply to many real-world settings including combinatorial auctions, matching markets, project fundraisers, and more—we expand on this list of examples further in Section 10.2.1. Our model of predictions is highly general and flexible, and it captures a number of information and knowledge formats in these diverse settings. One key aspect here is that the information conveyed about an agent *can depend on the revealed types of all other agents*. For example, a landowner who wants to sell mineral rights via an auction might not know the true market values of the natural resources present in his land, but might expect bids to be clustered around a high value or a low value based on the bidders' (who represent entities with domain expertise in, say, mining for rare materials) assessment of the resource quality. Then, based on a subset of revealed bids, the landowner can set informed reserve prices for other bidders to increase his revenue.

We then discuss an improvement to the VCG mechanism we call the *weakest-type VCG mechanism* (Section 5.1.1). While vanilla VCG charges an agent her externality measured relative to the welfare achievable by her non-participation, weakest-type VCG charges an agent her externality relative to the welfare achievable by the *weakest type* consistent with what the mechanism designer already knows about that agent. This idea is due to Krishna and Perry [1998], who showed that a Bayesian version of the weakest-type mechanism is revenue optimal among all efficient, Bayes incentive compatible, and Bayes individually rational mechanisms (assuming the mechanism designer has access to the prior distribution from which agents' values are drawn). Our weakest-type VCG mechanism adapts Krishna and Perry's mechanism to a prior-free setting and allows for a more general model of agent type spaces, and we prove that it is revenue optimal subject to efficiency, (ex-post) incentive compatibility, and (ex-post) individual rationality.

We show how the payment scheme implemented by the weakest-type VCG mechanism lends a natural interpretation in terms of information rents for the agents. Indeed, vanilla VCG is equivalent to a pay-as-bid/first-price payment scheme with agent discounts equal to the welfare improvement they create for the system. Weakest-type VCG is equivalent to a pay-as-bid scheme with agent discounts equal to the welfare improvement they create for the system *over the welfare created by their weakest type*. That difference in welfare is precisely an agent's *information rent*: the less private information she holds to "distinguish herself" from the weakest type, the smaller her discount.

Measuring prediction quality and weakest type computation In Section 5.2 we devise an appropriate error measure to quantify the quality of a prediction that is intimately connected to the weakest-type mechanism described above—it is precisely the delta in welfare created between an agent's true type and an agent's weakest type that is consistent with the information posited by the predictor (the key difference in the prediction setting is that the information that a prediction conveys about an agent need not be accurate). Remarkably, the ability of a prediction to boost revenue in our framework is largely unrelated to obvious measures of goodness. For example, one might expect that a good prediction should say something correct about an agent's true type—if it claims that an agent's type/valuation satisfies property P we would expect that her true type ought to satisfy property P for it to be a useful prediction. Rather counter-intuitively, even if the true type does not satisfy property P our framework can gain meaningful revenue mileage from that prediction. Our error measure completely characterizes the set of predictions that extract a given payment.

We also briefly discuss the computational complexity of finding weakest types (Section 5.2.1).

We show that weakest type computation can be formulated as a linear program with size equal to the size of the allocation space, which implies that weakest types can be found in time polynomial in the parameters defining the mechanism design environment. In environments where the size of the allocation space is prohibitively large (*e.g.,* in a combinatorial auction there are $(n+1)^m$ possible allocations of n items to m bidders and the seller), we show that the solution to the linear program can nonetheless be computed with polynomially many queries to an oracle for computing efficient allocations.

Prediction-augmented mechanisms and guarantees In Section 5.3.1 we present our main mechanism. It uses the information output by the predictors within the weakest-type VCG mechanism, but modifies that information via two tunable parameters. The first parameter allows for an initial modification to make the initial prediction more aggressive or more conservative. The second parameter controls a random relaxation of the modified prediction and serves to smooth out the behavior of the mechanism—we concretely demonstrate this via payment plots.

We prove that our mechanism achieves strong welfare and revenue guarantees that are parameterized by errors in the predictions and the quality of the parameter tuning. When its parameters are tuned well it achieves the efficient welfare OPT and revenue competitive with OPT, and its performance degrades gradually as both the tuning worsens and the quality of the predictions worsen. For a fixed default (untuned) parameter choice, it achieves welfare OPT and revenue $\approx \frac{\text{OPT}}{\log(\Delta^{\text{VCG}})}$ when the predictions are "perfect" (where Δ^{VCG} is a problem-dependent constant), and its performance degrades gradually as the quality of the predictions worsen (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.* We show that simplistic approaches that solely optimize for the *consistency* and *robustness* desiderata that have been studied in the mechanism design (and algorithm design more broadly) with predictions literature are brittle and overly sensitive to prediction errors. Our mechanism provides a more flexible, general, and robust alternative.

Other forms of side information In Section 5.4 we apply the weakest-type mechanism to three other formats of side information. First, in Section 5.4.1, we describe a more general model of predictions that can express arbitrary degrees of uncertainty over an agent's type. Here, we generalize the main randomized mechanism described previously. Second, in Section 5.4.2, we 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 (this is markedly different from our model of predictions and should be viewed as a restriction on the agent's type space itself). *For example, a real-estate agent might infer a buyers' relative property values based on value per square foot.* When each agent's true type is known to lie in a particular k-dimensional subspace of the ambient type space, we modify the weakest-type mechanism by choosing weakest types randomly from a careful discretization of the subspace, to obtain revenue at least $\Omega(\text{OPT}/k(\log H)^k)$ while simultaneously guaranteeing welfare at least $\text{OPT}/\log H$, where H is an upper bound on any agent's value. Third, in Section 5.4.3, we consider a textbook multidimensional mechanism design setup where the side information is in the form of a known prior distribution over agent types. Here, we show that the *revenue-optimal* (with no constraints other than incentive compatibility and individual rationality) Groves mechanism can be found by solving a separate single-parameter optimization problem for each agent. The optimization involves each agent's weakest type. Interestingly, our formulation recovers the optimal single-item auction of Myerson [1981] in a special case despite the fact that it is not globally revenue optimal in general.

Extension to affine-maximizer mechanisms Finally, we show how the weakest-type framework can be extended beyond VCG to the class of affine-maximizer (AM) mechanisms [Roberts, 1979]. We describe at least two attractive uses of a weakest-type AM in place of weakest-type VCG. First, it is well-known that in settings where agent types are drawn from a prior distribution, AMs can generate significantly more revenue than VCG. For example, work on *sample-based automated mechanism design* [Sandholm and Likhodedov, 2015, Balcan et al., 2018d, Curry et al., 2023] has shown that high-revenue AM parameters can be learned from data. Our techniques can then be appended as a post-processor to further improve the revenue of an already-tuned AM. Second, in many application domains it might make more sense to implement an allocation that maximizes a *weighted* version of welfare. For example, the mechanism designer might want to prioritize allocations that adequately reward small or minority-owned business (captured by multiplicative bidder weights). As another example, an auction designer might derive value from keeping some of the items for himself or from offering items for other uses outside the auction. In such cases, allocations that leave some items unsold might be prioritized if no bidder values those items competitively (captured by additive allocation boosts). We show how our main results extend to AMs, with appropriately redefined welfare and revenue benchmarks.

Related work

We survey related work and discuss work subsequent to our initial conference publication [Balcan et al., 2023].

Side information in mechanism design Various mechanism design settings have been studied under the assumption that some form of public side information is available. Medina and Vassilvitskii [2017] study single-item (unlimited supply) single-bidder posted-price auctions with bid predictions. Devanur et al. [2016] study the sample complexity of (single-parameter) auctions when the mechanism designer receives a distinguishing signal for each bidder. More generally, the active field of *algorithms with predictions* aims to improve the quality of classical algorithms when machine-learning predictions about the solution are available [Mitzenmacher and Vassilvitskii, 2022]. There have been recent explicit connections of this paradigm to settings with strategic agents [Agrawal et al., 2022, Gkatzelis et al., 2022, Balkanski et al., 2023, Banerjee et al., 2022]. Most related to our work, Xu and Lu [2022] study auctions for the sale of a (single copy of a) single item when the mechanism designer receives point predictions on the bidders' values. Unlike our approach, they focus on *deterministic* modifications of a second-price auction. An important drawback of determinism is that revenue guarantees do not decay continuously as prediction quality degrades. For agents with values in $[1, H]$ there is an error threshold after which, in the worst case, only a $1/H$-fraction of revenue can be guaranteed (a vacuous guarantee not even competitive with a vanilla second-price auction). Xu and Lu [2022] prove that such a revenue drop is unavoidable by deterministic mechanisms. Finally, our setting is distinct from, but similar to in spirit, work that uses public attributes for market segmentation to improve revenue [Balcan et al., 2005, 2020c].

Welfare-revenue tradeoffs in auctions Welfare and revenue relationships in Bayesian auctions have been widely studied since the seminal work of Bulow and Klemperer [1996]. Welfare-revenue tradeoffs for second-price auctions with reserve prices in the single item setting have been quantified [Hartline and Roughgarden, 2009, Daskalakis and Pierrakos, 2011], with some approximate understanding of the Pareto frontier [Diakonikolas et al., 2012]. Anshelevich et al. [2016] study welfare-revenue tradeoffs in large markets, Aggarwal et al. [2009] study the efficiency of revenue-optimal mechanisms, and Abhishek and Hajek [2010] study the efficiency loss of revenue-optimal mechanisms.

Weakest types The idea behind the weakest-type VCG mechanism is due to Krishna and Perry [1998] but the notion of a "worst-off" type was studied before that in the context of bilateral (and more general) trade by Myerson and Satterthwaite [1983] and Cramton et al. [1987]. That study identified the worst-off type in a trading environment and used that characterization to characterize individually rational trading mechanisms—the connection to individual rationality is similar to the one we draw in Theorem 5.1.1. The improvement to VCG via weakest types as first established by Krishna and Perry [1998] does not appear to have been further explored since then.

Distribution-free agent type models Our primary model of predictions is *distribution free* in that a prediction puts forward a postulate that an agent's true type belongs to some set, but conveys no further distributional information over that set. Such models of agent types have been previously studied, though in different contexts, by Hyafil and Boutilier [2004] for minimax optimal automated mechanism design, Holzman et al. [2004] to understand equilibria in combinatorial auctions, and Chiesa et al. [2015] for agents who have uncertainty about their own private types.

Constant-parameter mechanism design Revenue-optimal mechanism design for settings where each agent's type space is of a constant dimension has been studied previously in certain specific settings. Single-parameter mechanism design is a well-studied topic dating back to the seminal work of Myerson [1981], who (1) characterized the set of all truthful allocation rules and (2) derived the Bayesian optimal auction based on virtual values (a quantity that is highly dependent on knowledge of the agents' value distributions). Kleinberg and Yuan [2013] prove revenue guarantees for a variety of single-parameter settings that depend on distributional parameters. Financially constrained buyers with two-parameter valuations have also been studied [Malakhov and Vohra, 2009, Pai and Vohra, 2014].

Combinatorial auctions for limited supply Our mechanism when agent types lie on known linear subspaces can be seen as a generalization of the well-known logarithmic revenue approximation that is achieved by a second-price auction with a random reserve price in the single-item setting [Goldberg et al., 2001]. Similar revenue approximations have been derived in multi-item settings for various classes of bidder valuation functions such as unit-demand [Guruswami et al., 2005], additive [Sandholm and Likhodedov, 2015, Likhodedov and Sandholm, 2005, Babaioff et al., 2020], and subadditive [Balcan et al., 2008, Chakraborty et al., 2013]. To the best of our knowledge, no previous techniques handle agent types on low-dimensional subspaces. Furthermore, our results are not restricted to combinatorial auctions unlike most previous research.

Sample-based mechanism design Likhodedov and Sandholm [2004, 2005], Sandholm and Likhodedov [2015] introduced the idea of automatically tuning the parameters of a mechanism based on samples of valuations. They also formulated mechanism design as a search problem within a parameterized family. They developed custom hill-climbing methods for learning various high-revenue auction formats, including affine maximizers, which we study in this chapter. Balcan et al. [2005] were the first to apply tools from machine learning theory to establish theoretical guarantees on the *sample complexity* of mechanism design. Since then, there has been a large body of work on both theoretical and practical aspects of data-driven mechanism design, for example, studying the sample complexity of revenue maximization for various auction and mechanism families [Morgenstern and Roughgarden, 2016, Cole and Roughgarden, 2014, Balcan et al., 2018d] and using deep learning to design high-revenue mechanisms [Dütting et al., 2019, Duan et al., 2023, Curry et al., 2023].

Beyond mechanism design, the field of *data-driven algorithm design* [Balcan, 2020] establishes theoretical foundations for a strand of what was largely empirical work on sample-based algorithm configuration (that predates sample-based mechanism design, *e.g.,* Horvitz et al. [2001]). This line of work is thematically similar to ours, but the main focus of our paper (and of the mechanism design with predictions literature more broadly) is to quantify the performance improvement obtainable based on the quality of predictions (that might have been obtained through learning from data). Khodak et al. [2022] show how to "learn the predictions in algorithms with predictions", thus concretely connecting the two research strands.

Work subsequent to our initial publication After our initial conference publication [Balcan et al., 2023], a number of papers have continued to grow the area of mechanism design with predictions. Still, this literature has largely focused on single-parameter settings. Specific applications include online single-item auctions [Balkanski et al., 2024b], single-parameter clock auctions [Gkatzelis et al., 2025], and randomized single-item auctions [Caragiannis and Kalantzis, 2024]. Most closely related to the present paper is the work of Lu et al. [2024] who use the idea of randomly modifying weakest types, that we proposed in the original conference version of the present paper [Balcan et al., 2023], for error-tolerant design in a variety of single-parameter settings—such as digital good auctions and auctions for the sale of multiple copies of a single item.

Additionally, the subsequent chapters of this thesis explore further uses of weakest types—and more broadly the information conveyed by type spaces—to design new core-selecting combinatorial auctions, derive new characterizations of revenue-optimal efficient mechanisms for general type spaces, and more.

5.1 Problem Formulation, Example Applications, and Weakest-Type VCG

We consider a general multidimensional mechanism design setting with a finite allocation space Γ and n agents. Θ_i is the ambient type space of agent i. Agent i's true private type $\theta_i \in \Theta_i$ determines her value $v(\theta_i, \alpha)$ for allocation $\alpha \in \Gamma$. We will interpret Θ_i as a subset of $\mathbb{R}^\Gamma$, so $\theta_i[\alpha] = v(\theta_i, \alpha)$. We use $\boldsymbol{\theta} \in \times_{i=1}^n \Theta_i$ to denote a profile of types and $\boldsymbol{\theta}_{-i} \in \Theta_{-i} := \times_{j \neq i} \Theta_i$ to denote a profile of types excluding agent i. We now introduce our main model of side information which is the focus of Sections 5.2 and 5.3 (we discuss other models in Section 5.4). First, the *ambient type space* Θ_i is assumed to convey no information about each agent, that is, $\Theta_i = \mathbb{R}^\Gamma_{\geq 0}$ (this is the standard assumption in the mechanism design literature). The mechanism designer has access to a set-valued *predictor* $T_i : \times_{j \neq i} \Theta_j \to \mathcal{P}(\Theta_i)$ for each agent. Predictor T_i takes as input *the revealed types* $\boldsymbol{\theta}_{-i}$ *of all agents excluding* i and outputs a set $T_i(\boldsymbol{\theta}_{-i}) \subseteq \Theta_i$ that represents a prediction that agent i's true type lies in $T_i(\boldsymbol{\theta}_{-i})$. The mechanism designer has no apriori guarantees about the quality (or validity) of the prediction output by T_i. *We emphasize that a prediction about agent* i *can depend on the revealed types of all other agents.* This begets a rich and expressive language of prediction that can incorporate, for example, market insights by means of analyzing other agents' true types (a form of learning within an instance [Baliga and Vohra, 2003, Balcan et al., 2005, 2021c]), interlinked knowledge about multiple agents' types, *etc.* Importantly—as we will later demonstrate—allowing predictions to carry such a great level of expressive power poses no barrier to incentive compatibility of our mechanisms. Finally, we point out that the modeling assumption of a separate predictor for each agent does not in any way restrict the ability of side information to convey relationships between the types of different agents: indeed, the claim that "agent types $(\theta_1, \ldots, \theta_n)$ satisfy property P" induces predictors $T_1, \ldots, T_n$ where T_i is defined by $T_i(\boldsymbol{\theta}_{-i}) = \{\hat{\theta}_i : (\hat{\theta}_i, \boldsymbol{\theta}_{-i}) \text{ satisfies } P\}$.

A mechanism with predictors is specified by an allocation rule $\alpha(\boldsymbol{\theta}; T_1, \ldots, T_n) \in \Gamma$ and a payment rule $p_i(\boldsymbol{\theta}; T_1, \ldots, T_n) \in \mathbb{R}$ for each agent i. We assume agents have quasilinear utilities. A mechanism is *incentive compatible (IC)* if $\theta_i \in \operatorname{argmax}_{\theta_i' \in \Theta_i} \theta_i[\alpha(\theta_i', \boldsymbol{\theta}_{-i}; T_1, \ldots, T_n)] - p_i(\theta_i', \boldsymbol{\theta}_{-i}; T_1, \ldots, T_n)$ holds for all $i, \theta_i \in \Theta_i, \boldsymbol{\theta}_{-i} \in \Theta_{-i}$, that is, agents are incentivized to report their true type regardless of what other agents report (this definition is equivalent to the usual notion of dominant-strategy IC and simply stipulates that predictors ought to be used in an IC manner). A mechanism is *individually rational (IR)* if $\theta_i[\alpha(\theta_i, \boldsymbol{\theta}_{-i}; T_1, \ldots, T_n)] - p_i(\theta_i, \boldsymbol{\theta}_{-i}; T_1, \ldots, T_n) \geq 0$ holds for all $i, \theta_i, \boldsymbol{\theta}_{-i}$. We will analyze a variety of randomized mechanisms that randomize over IC and IR mechanisms. Such randomized mechanisms are thus IC and IR in the strongest possible sense (as supposed to Bayes-IC/IR which is weaker and holds only in expectation). *An important note: no assumptions are made on the veracity of* T_i*, and agent* i*'s misreporting space is the full ambient type space* Θ_i*.*

Given reported types $\boldsymbol{\theta}$, define $w(\boldsymbol{\theta}) = \max_{\alpha \in \Gamma} \sum_{i=1}^n \theta_i[\alpha]$ to be the *efficient* social welfare. Let $\alpha^* = \alpha^*(\boldsymbol{\theta}) = \operatorname{argmax}_\alpha \sum_{i=1}^n \theta_i[\alpha]$ denote the *efficient* allocation, that is, the allocation that achieves $w(\boldsymbol{\theta})$. Our benchmark for welfare and revenue is the prior-free efficient welfare $\text{OPT} = w(\boldsymbol{\theta})$, which is the strongest possible benchmark for both welfare and revenue.

Example applications

Our model of side information within the rich language of multidimensional mechanism design allows us to capture a variety of different problem scenarios where both welfare and revenue are desired objectives. We list a few examples of different multidimensional mechanism settings along with examples of different varieties of predictions.

- Combinatorial auctions: There are m indivisible items to be allocated among n bidders (or to no one). The allocation space Γ is the set of $(n+1)^m$ allocations of the items and $\theta_i[\alpha]$ is bidder i's value for the bundle of items she is allocated by α. Let X and Y denote two of the items for sale. The predictor $T_i(\boldsymbol{\theta}_{-i}) = \{\theta_i : \theta_i[XY] \geq 3/2 \cdot \theta_j[XY], \theta_i[XY] \geq \theta_i[X] + \theta_i[Y]\}$ represents the prediction that bidder i's value for the bundle XY is at least 50% greater than bidder j's value for the same bundle and that items X and Y are complements for her. Here, $T_i(\boldsymbol{\theta}_{-i})$ is the intersection of linear constraints.
- Matching markets: There are m items (*e.g.*, houses) to be matched to n buyers. The allocation space Γ is the set of matchings on the bipartite graph $K_{m,n}$ and $\theta_i[\alpha]$ is buyer i's value for the item α assigns her. Let $\alpha_1, \alpha_2, \alpha_3$ denote three matchings that match house 1, house 2, and house 3 to agent i, respectively. The *type space restriction* $\Theta_i = \{\theta_i : \theta_i[\alpha_1] = 2 \cdot \theta_i[\alpha_2] = 0.75 \cdot \theta_i[\alpha_3]\}$ represents the information that agent i values house 1 twice as much as house 2, and $3/4$ as much as house 3. Here, Θ_i is the linear space given by $\text{span}(\langle 1, 1/2, 4/3 \rangle)$ (this model is further studied in Section 5.4.2 and differs from our main prediction model).
- Fundraising for a common amenity: A multi-story office building that houses several companies is opening a new cafeteria on a to-be-determined floor and is raising construction funds. The allocation space Γ is the set of floors of the building and $\theta_i[\alpha]$ is the (inverse of the) cost incurred by building-occupant i for traveling to floor α. The set $T_i = \{\theta_i : \|\theta_i - \theta_i^*\|_p \leq k\}$ postulates that i's true type is no more than k away from θ_i^* in ℓ_p-distance, which might be derived from an estimate of the range of floors agent i works on based on the company agent i represents. Here, T_i is given by a (potentially nonlinear) distance constraint and has no dependence on the other agents' revealed types.
- Bidding for a shared outcome: A delivery service that offers multiple delivery rates (priced proportionally) needs to decide on a delivery route to serve n customers. The allocation space Γ is the set of feasible routes and $\theta_i[\alpha]$ is agent i's value for receiving her packages after the driving delay specified by α. Let α_t denote an allocation that imposes a driving delay of t on agent i. The set $T_i(\boldsymbol{\theta}_{-i}) = \{\theta_i : \theta_i[\alpha_0] \geq \max_{j \neq i} \theta_j[\alpha_0] + \$50, \theta_i[\alpha_{t+1}] \geq f_t(\theta_i[\alpha_t]) \; \forall t\}$ is the prediction that agent i is willing to pay \$50 more than any other agent to receive her package as soon as possible, and is at worst a time discounter determined by (potentially nonlinear) discount functions f_t. Here, the complexity of T_i is determined by the function f_t.

5.1.1 The weakest-type VCG mechanism

The classical Vickrey-Clarke-Groves (VCG) mechanism [Vickrey, 1961, Clarke, 1971, Groves, 1973] elicits agent types $\boldsymbol{\theta}$, implements the efficient allocation α^* and charges agent i a payment $p_i(\boldsymbol{\theta}) = w(0, \boldsymbol{\theta}_{-i}) - \sum_{j \neq i} \theta_j[\alpha^*]$ which is agent i's externality. VCG is generally highly suboptimal when it comes to revenue [Ausubel and Milgrom, 2006, Varian and Harris, 2014, Metz, 2015] (and conversely mechanisms that shoot for high revenue can be highly welfare suboptimal). However, if efficiency is enforced as a constraint of the mechanism design (in addition to IC and IR), then the following *weakest-type VCG mechanism*, first introduced by Krishna and Perry [1998] in a Bayesian form, is in fact revenue optimal (Krishna and Perry call it the generalized VCG mechanism). While VCG payments are based on participation externalities, weakest-type VCG payments are based on agents being replaced by their *weakest types* who have the smallest impact on welfare. This approach generates strictly more revenue than vanilla VCG. Krishna and Perry [1998] proved that the Bayesian version of weakest-type VCG is revenue optimal among all efficient, Bayes IC, and Bayes IR mechanisms. We next present the weakest-type VCG mechanism, which is a generalization of their mechanism in a prior-free setting.

To describe our weakest-type VCG mechanism, we depart slightly from the language of predictors and focus on what is already known to the mechanism designer as conveyed by the *joint type space* of the agents. That is, there is a joint type space $\boldsymbol{\Theta} \subseteq \times_{i=1}^n \mathbb{R}^\Gamma_{\geq 0}$ and the mechanism designer knows that the revealed type profile $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ belongs to $\boldsymbol{\Theta}$. The assumption of a joint type space with no further restrictions allows for a rich knowledge structure that can capture arbitrary relationships between agents; given revealed types $\boldsymbol{\theta}_{-i}$ for all agents excluding i, the mechanism designer knows that the revealed type of agent i belongs to the set $\{\hat{\theta}_i : (\hat{\theta}_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}\}$ (which is a *projection* of the joint type space $\boldsymbol{\Theta}$ onto agent i's space, given $\boldsymbol{\theta}_{-i}$). We now define the *weakest-type VCG mechanism* in this general setting.

Weakest-type VCG

Input: joint type space $\boldsymbol{\Theta} \subseteq \times_{i=1}^n \mathbb{R}^\Gamma_{\geq 0}$.

- Agents asked to reveal types $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$.
- The efficient allocation $\alpha^* = \text{argmax}_{\alpha \in \Gamma} \sum_{i=1}^n \theta_i[\alpha]$ is implemented and agent i is charged a payment of

$$p_i(\boldsymbol{\theta}) = \min_{\widetilde{\theta}_i : (\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) - \sum_{j \neq i} \theta_j[\alpha^*].$$

We call $\widetilde{\theta}_i$ in the above minimization agent i's *weakest type*. If $(0, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}$, $\widetilde{\theta}_i = 0$, and $p_i(\boldsymbol{\theta}) = w(0, \boldsymbol{\theta}_{-i}) - \sum_{j \neq i} \theta_j[\alpha^*]$ is the vanilla VCG payment. On the other extreme if $\{\hat{\theta}_i : (\hat{\theta}_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}\} = \{\theta_i\}$, that is, the the joint type space exactly conveys agent i's true type, $p_i(\boldsymbol{\theta}) = \theta_i[\alpha^*]$ and agent i's total value is extracted as payment. Krishna and Perry [1998] essentially prove a weaker form (they assume independent type spaces, that is, $\boldsymbol{\Theta} = \times_i \Theta_i$ has a product structure) of the following result in the Bayesian setting (that is, they only require the weaker constraints of Bayes IC and Bayes IR). We reproduce that here in a prior-free setting with our more general model of type spaces. We present two proofs: one based on the revenue equivalence theorem and one based on the result of Holmström [1979] concerning Groves mechanisms.

Theorem 5.1.1. *Let $\boldsymbol{\Theta}$ be a compact and convex joint type space. The weakest-type VCG mechanism is incentive compatible and individually rational. Furthermore, it is revenue-optimal among all efficient, incentive compatible, and individually rational mechanisms.*

Proof. Weakest-type VCG is incentive compatible since it is a Groves mechanism [Groves, 1973], that is, the *pivot* term $\min_{\widetilde{\theta}_i:(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}} w(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})$ has no dependence on agent i's revealed type θ_i. Furthermore, agent i's utility is $\sum_{j=1}^n \theta_j[\alpha^*] - \min_{\widetilde{\theta}_i:(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}}(\max_\alpha \sum_{j\neq i}\theta_j[\alpha] + \widetilde{\theta}_i[\alpha]) \geq \sum_{j=1}^n \theta_j[\alpha^*] - \max_\alpha \sum_{j=1}^n \theta_j[\alpha] = 0$, which proves individual rationality. The proof that weakest-type VCG is revenue optimal follows from the revenue equivalence theorem; see Vohra [2011, Theorem 4.3.1]. Let $p_i(\boldsymbol{\theta})$ be the weakest-type VCG payment rule, and let $p_i'(\boldsymbol{\theta})$ be any other payment rule that also implements the efficient allocation rule. By revenue equivalence, for each i, there exists $h_i(\boldsymbol{\theta}_{-i})$ such that $p_i'(\theta_i,\boldsymbol{\theta}_{-i}) = p_i(\theta_i,\boldsymbol{\theta}_{-i}) + h_i(\boldsymbol{\theta}_{-i})$. Suppose $\boldsymbol{\theta}$ is a profile of types such that p_i' generates strictly greater revenue than p_i, that is, $\sum_{i=1}^n p_i'(\boldsymbol{\theta}) > \sum_{i=1}^n p_i(\boldsymbol{\theta})$. Equivalently $\sum_{i=1}^n p_i(\theta,\boldsymbol{\theta}_{-i}) + h_i(\boldsymbol{\theta}_{-i}) > \sum_{i=1}^n p_i(\theta_i,\boldsymbol{\theta}_{-i})$. Thus, there exists i^* such that $h_{i^*}(\boldsymbol{\theta}_{-i^*}) > 0$. Let

$$\widetilde{\theta}_{i^*} = \underset{\theta'_{i^*}:(\theta'_{i^*},\boldsymbol{\theta}_{-i^*})\in\boldsymbol{\Theta}}{\operatorname{argmin}} w(\theta'_{i^*},\boldsymbol{\theta}_{-i^*})$$

be the *weakest type* with respect to $\boldsymbol{\theta}_{-i^*}$. If weakest-type VCG is run on the type profile $(\widetilde{\theta}_{i^*},\boldsymbol{\theta}_{-i^*})$, the agent with type $\widetilde{\theta}_{i^*}$ pays their value for the efficient allocation. In other words, the individual rationality constraint is binding for $\widetilde{\theta}_{i^*}$. Since $h_{i^*}(\boldsymbol{\theta}_{-i^*}) > 0$, p_i' violates individual rationality, which completes the proof.

We provide an alternate proof of the revenue optimality of weakest-type VCG via the result of Holmström [1979] that, for convex type spaces, any efficient and IC mechanism is a Groves mechanism. (Technically, Holmström [1979] assumes that agent type spaces are independent, that is $\boldsymbol{\Theta}$ has a product structure. We can circumvent this issue as follows. Let $\Theta_i(\boldsymbol{\theta}_{-i}) = \{\hat{\theta}_i : (\hat{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}\}$. We can then give $\boldsymbol{\Theta}$ product structure by writing $\boldsymbol{\Theta} = \Theta_1(\boldsymbol{\theta}_{-1})\times\cdots\times\Theta_n(\boldsymbol{\theta}_{-n})$. Each $\Theta_i(\boldsymbol{\theta}_{-i})$ is convex since $\boldsymbol{\Theta}$ itself is convex.) Then, the revenue-maximizing Groves payment scheme $h_i(\boldsymbol{\theta}_{-i})$ solves for each agent i

$$\max\left\{h_i(\boldsymbol{\theta}_{-i}) : \widetilde{\theta}_i[\alpha^*] - \left(h_i(\boldsymbol{\theta}_{-i}) - \sum_{j\neq i}\theta_j[\alpha^*]\right) \geq 0 \quad \forall\widetilde{\theta}_i \text{ s.t. } (\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}\right\},$$

which yields the maximum Groves payment subject to IR constraints for all possible types $\widetilde{\theta}_i$ such that $(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}$ (and has no dependence on agent i's revealed type θ_i). Rewrite the constraint as $h_i(\boldsymbol{\theta}_{-i}) \leq w(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\ \forall\,\widetilde{\theta}_i$ s.t. $(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}$. So, $h_i(\boldsymbol{\theta}_{-i}) \leq \min_{\widetilde{\theta}_i:(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})\in\boldsymbol{\Theta}} w(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})$ which is precisely weakest-type VCG. □

Remark. The weakest-type VCG mechanism is *not* equivalent to a mechanism that actually adds "fake" competitors into the mechanism environment (as is the case with mechanisms that use "phantom bidders" [Sandholm, 2013]). Indeed, consider the mechanism that, for each agent i, adds an agent $n+i$ who is the weakest type and then runs vanilla VCG over the augmented set of $2n$ agents. That mechanism is not necessarily efficient and is thus not equivalent to weakest-type VCG. As an example consider a two-item, two-bidder auction where each bidder only wants one item. Say $v_1(X) = v_1(Y) = 10$, $v_2(X) = 5, v_2(Y) = 4$. The efficient allocation gives

Y to 1 and X to 2 and has welfare 15. Suppose agent 1's type space is $\{v_1 : v_1(X) \geq 7\}$ and agent 2's type space is $\mathbb{R}^2_{\geq 0}$ (so type spaces are independent). So agent 1's weakest type is $\widetilde{v}_1(X) = 7, \widetilde{v}_1(Y) = 0$, and agent 2's weakest type is $\widetilde{v}_2(X) = \widetilde{v}_2(Y) = 0$. Running vanilla VCG among $\{v_1, \widetilde{v}_1, v_2, \widetilde{v}_2\}$ yields an efficient allocation that gives X to $\widetilde{1}$ and Y to 1. So, while an agent's weakest type will never take that agent's items away from them, they can win some other agent's items. In weakest-type VCG, the weakest types are not real agents and only serve to boost prices—they do not receive any utility from the allocation.

Vanilla VCG payments can be interpreted as implementing a first-price/pay-as-bid scheme with a discount for each agent equal to the welfare improvement they create by participating. Weakest-type VCG payments implement a pay-as-bid scheme with a discount equal to the welfare improvement an agent creates over the *weakest type* in their type space. The discount can be directly interpreted as an *information rent* (*e.g.,* Börgers [2015]) incurred by the agents. The more private information an agent has (measured by how far away her IR constraint in weakest-type VCG is from binding—we discuss this aspect formally in the subsequent section) relative to the information conveyed by the type space, the greater her discount. Said another way, the more private information an agent has to distinguish herself from the weakest type, the greater her discount. The weakest type is also the *most reluctant* type [Krishna and Perry, 1998] in that it receives zero utility from participation. Finally, weakest types can be viewed as a sophisticated form of reserve pricing (and any reserve pricing structure atop VCG can be expressed as a weakest-type mechanism).

We conclude our discussion of weakest-type VCG with the observation that the weakest-type VCG mechanism can loosely be interpreted as a prior-free analogue of the seminal Bayesian total-surplus-extraction mechanism of Crémer and McLean [1988] for correlated agents (generalized to infinite type spaces by McAfee and Reny [1992]). This is an interesting connection to explore further in future research.

In the following section we return to our model of predictors, which can now be interpreted as conveying the same kind of knowledge as a joint type space, but without the guarantee that the information conveyed about the agents is actually correct. That guarantee—which equivalently stipulates that agent i's *misreporting space* (that determines her IC and IR constraints) is restricted to the set $\{\hat{\theta}_i : (\hat{\theta}_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}\}$—was required in Theorem 5.1.1 to ensure individual rationality of weakest-type VCG. Our setting of mechanism design with predictors can thus be interpreted as removing this constraint on the misreporting space: predictors $\{T_i\}$ have no apriori guarantees on their veracity and we must use them in a way that respects IC and IR constraints for a misreporting space equal to the entire ambient type space.

5.2 Measuring Predictor Quality via Weakest Types

In this section we pin down what it means to be a high-quality predictor and completely characterize predictors that yield a certain level of payment in weakest-type VCG (including those that are too aggressive). Consider a direct use of the predictors in the weakest-type VCG mechanism: types $\boldsymbol{\theta}$ are elicited, the efficient allocation α^* is computed, and payments

$$p_i(\boldsymbol{\theta}) = \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) - \sum_{j \neq i} \theta_j[\alpha^*]$$

are computed. Writing

$$p_i = \theta_i[\alpha^*] - \left(w(\theta_i, \boldsymbol{\theta}_{-i}) - \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \right) \tag{5.1}$$

makes the connection between predictions and payment explicit—as mentioned previously we can interpret the payment as a pay-of-bid price minus a discount measuring the welfare gains created by θ_i over the weakest type in $T_i(\boldsymbol{\theta}_{-i})$. This way of writing the payment also makes explicit when a prediction $T_i(\boldsymbol{\theta}_{-i})$ is *too aggressive/competitive*: if $\min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) > w(\theta_i, \boldsymbol{\theta}_{-i})$, $p_i(\boldsymbol{\theta}) > \theta_i[\alpha^*]$, and agent i's IR constraint is violated (which any reasonable mechanism should recognize and rectify). Our main mechanisms in the subsequent sections will explicitly handle such predictors to ensure that IR is met for all agents.

Our measure of predictor error is precisely this delta in welfare: let

$$\Delta_i^{\text{err}} = \Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) = w(\theta_i, \boldsymbol{\theta}_{-i}) - \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}).$$

So, $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) < 0$ means the predictor is too aggressive, $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) > 0$ means the predictor is too conservative, and $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) = 0$ means the predictor exactly predicts agent i's welfare level. The following result is immediate from the definition of Δ_i^{err} and Theorem 5.1.1.

Theorem 5.2.1. *Let $T_1, \ldots, T_n$ be predictors such that $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) \geq 0$ for all agents i. The mechanism that implements the efficient allocation and prices given by Equation* (5.1) *is IC, IR, and extracts payment $p_i = \theta_i[\alpha^*] - \Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i})$ from each agent—and thus generates revenue equal to* $\text{OPT} - \sum_i \Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i})$.

As mentioned in Section 5.1.1, Δ_i^{err} can be directly interpreted as an information rent for agent i. The more private information she possesses relative to the weakest type, the greater her discount.

We describe an equivalent geometric characterization of predictors that generate a certain payment in the framework above. Let $L_w(\boldsymbol{\theta}_{-i}) = \left\{ \widetilde{\theta}_i : w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) = w \right\}$ be a *welfare level set* and let $L_{\geq k}(\boldsymbol{\theta}_{-i}) = \left\{ \widetilde{\theta}_i : w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \geq w \right\}$. The following propositions follow immediately from the definitions.

Proposition 5.2.2. *$L_{\geq w}(\boldsymbol{\theta}_{-i})$ is the union of axis-parallel halfspaces.*

Proof.

$$L_{\geq w}(\boldsymbol{\theta}_{-i}) = \left\{ \widetilde{\theta}_i : \bigvee_{\alpha \in \Gamma} \widetilde{\theta}_i[\alpha] \geq w - \sum_{j \neq i} \theta_j[\alpha] \right\} = \bigcup_{\alpha \in \Gamma} \left\{ \widetilde{\theta}_i : \widetilde{\theta}_i[\alpha] \geq w - \sum_{j \neq i} \theta_j[\alpha] \right\}.$$

□

Proposition 5.2.3. *A predictor T_i has error $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) = \Delta_i$ if and only if it intersects the level set $L_{w(\theta_i, \boldsymbol{\theta}_{-i}) - \Delta_i}$ and does not intersect any L_w with $w < w(\theta_i, \boldsymbol{\theta}_{-i})$. Predictor T_i satisfies $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) \geq 0$ if and only if $L_{w(\theta_i, \boldsymbol{\theta}_{-i})}(\boldsymbol{\theta}_{-i}) \subseteq L_{w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})}(\boldsymbol{\theta}_{-i})$ where $\widetilde{\theta}_i$ is the weakest type in $T_i(\boldsymbol{\theta}_{-i})$.*

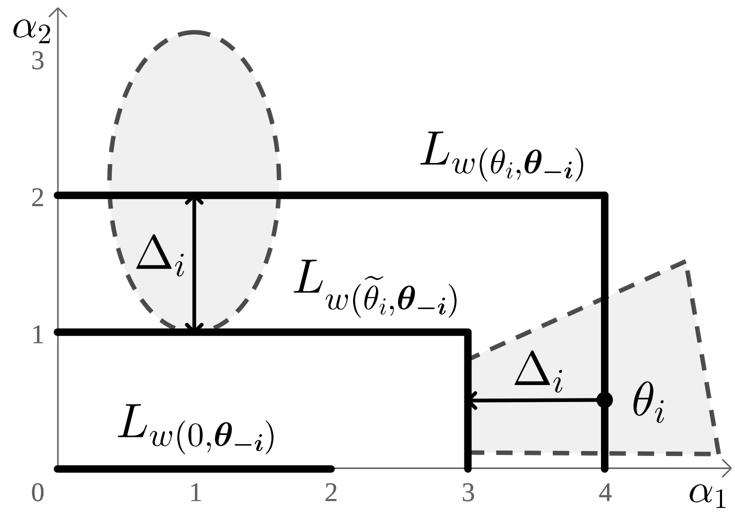


Figure 5.1: Two different predictions (the ellipse and polygon displayed with dashed boundaries) that are equivalent in the sense that their weakest types create the same amount of welfare $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) = w(\theta_i, \boldsymbol{\theta}_{-i}) - \Delta_i$ for the system and thus generate the same weakest-type payments for agent i, despite the fact that one prediction (the polygon) contains the true type and the other (the ellipse) completely misses the true type. Welfare level sets are depicted by the solid black lines.

One notable consequence is that if $T_i(\boldsymbol{\theta}_{-i})$ intersects $L_{w(0,\boldsymbol{\theta}_{-i})}$, T_i is a "useless" predictor in that the information it provides is not even strong enough to say that agent i generates any extra welfare to the system at all. More precisely, the weakest-type payment generated by T_i (Equation (5.1)) is equal to the vanilla VCG payment. On the other hand, $\Delta_i^{\text{err}} = 0$ is necessary and sufficient for a predictor to extract an agent's full value as payment. Such predictors need not even contain the true type, as illustrated further by the following example.

Example 5.2.4. Consider an allocation space $\Gamma = \{\alpha_1, \alpha_2\}$ with two outcomes as depicted in Figure 5.1. Suppose $\boldsymbol{\theta}_{-i}$ is such that $\sum_{j \neq i} \theta_j[\alpha_1] = 1$ and $\sum_{j \neq i} \theta_j[\alpha_2] = 3$, so $w(0, \boldsymbol{\theta}_{-i}) = 3$. Welfare level sets are depicted by the bold black lines. $L_{w(0,\boldsymbol{\theta}_{-i})}$ is the set of types $\widetilde{\theta}_i$ that do not create any additional welfare for the system which is precisely the set $\{\widetilde{\theta}_i : \widetilde{\theta}[\alpha_1] \leq 2 \wedge \widetilde{\theta}[\alpha_2] = 0\}$. The sets enclosed by the dotted lines represent the outputs of two different predictors. Each set's weakest type generates a welfare of $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) = 4$ as each prediction intersects L_4 and no lower level set. The true type of agent i, θ_i, generates welfare $w(\theta_i, \boldsymbol{\theta}_{-i}) = 5$. Therefore, both predictors have the same error of $\Delta_i^{\text{err}} = 1$ and generate equal weakest-type VCG payments (Equation (5.1)). This is in spite of the fact that only one of the predictions actually contains the true type (the polygon) while the other (the ellipse) is quite far away from the true type.

One remarkable property of this framework highlighted by the above example is that predictors need not know anything about the precise structure of types. As long as they provide a reasonable (under)estimate of agent i's competitive level given $\boldsymbol{\theta}_{-i}$ they can be used in the weakest-type mechanism to fruitfully boost revenue over vanilla VCG. Consider the setting of a combinatorial auction with m items and n bidders wherein a bidder's ambient valuation space is $\mathbb{R}_{\geq 0}^{2^m}$ since she can express a value for any bundle of items. In practice, the auction designer might set a cap on the number of distinct bundles any single bidder can submit bids for (as was done in the spectrum auctions held in the UK and Canada [Ausubel and Baranov, 2017]). In this case, a predictor need not reckon with the question of what bundles the bidder will bid on (which is ostensibly a very difficult prediction problem that would require unrealistic insight into a bidder's bidding strategy). It only needs to provide a good estimate of her competitive level as measured by the welfare she creates for the system, which we posit is a much more reasonable and practically plausible prediction task. Indeed, in high-stakes auctions, the auction designer might reasonably expect a certain bidder (who, say, represents a large conglomerate as is the case in sourcing and spectrum auctions) to win at least some items. That already provides the auction designer sufficient knowledge to implement a weakest-type pricing scheme that extracts greater revenues than VCG. To summarize, a predictor like the ellipse in Figure 5.1 that is completely wrong about the bundles bidder i bids on is still perfectly good in our framework, and therefore predictors have a good deal of flexibility in the information they convey.

The results of Chapter 7 give a formal framework to understand the types of richer side information structures needed to express the type of bundle uncertainty mentioned above. Here, side information can represent disjunctions, and therefore can capture more refined knowledge that depends on the exact realization of an agent's type. A generalization of the weakest-type mechanism characterizes the revenue-optimal efficient mechanism in this general setting.

Another way in which predictors have leeway is the fact that only the prediction error $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i})$ given the revealed types of the other agents affects payments. Predictor T_i could be wildly inaccurate on a different set of types, that is, $|\Delta_i^{\text{err}}(\boldsymbol{\theta}'_{-i})|$ might be huge, but as long as $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i})$ is small, it is a highly useful predictor.

5.2.1 Computational considerations

Before we present our main mechanism and its guarantees, we briefly discuss the computational complexity of computing weakest type payments. We consider the special case where the sets $T_i(\boldsymbol{\theta}_{-i})$ output by the predictors are polytopes. Let $\mathsf{size}(T_i)$ denote the max encoding size over all $\boldsymbol{\theta}_{-i}$ required to write down the constraints defining $T_i(\boldsymbol{\theta}_{-i})$.

Theorem 5.2.5. *Let $T_i(\boldsymbol{\theta}_{-i})$ be a polytope. The weakest type and its corresponding welfare $\min_{\widetilde{\theta}_i\in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$, and thus p_i in Equation* (5.1)*, can be computed in $poly(|\Gamma|, \mathsf{size}(T_i), n)$ time.*

Proof. Weakest type computation is a min-max optimization problem: $\min_{\widetilde{\theta}_i\in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) = \min_{\widetilde{\theta}_i\in T_i(\boldsymbol{\theta}_{-i})} \max_{\alpha\in\Gamma} \widetilde{\theta}_i[\alpha] + \sum_{j\neq i}\theta_j[\alpha]$. We can rewrite the min-max problem as a pure minimization problem by enumerating the set of allocations Γ and introducing an auxiliary scalar variable γ to replace the inner maximization. The weakest type in $T_i(\boldsymbol{\theta}_{-i})$ is therefore the solution $\widetilde{\theta}_i \in \mathbb{R}^\Gamma$ to the linear program

$$\min\left\{\gamma \ : \begin{array}{l} \widetilde{\theta}_i[\alpha] + \sum_{j\neq i}\theta_j[\alpha] \leq \gamma \ \ \forall \alpha\in\Gamma, \\ \widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i}), \gamma \geq 0 \end{array}\right\} \tag{5.2}$$

with $|\Gamma| + 1$ variables and $|\Gamma| + \mathsf{size}(T_i)$ constraints. Generating the first set of constraints requires the value of $\sum_{j\neq i}\theta_j[\alpha]$ for each $\alpha\in\Gamma$, which takes time $\leq n|\Gamma|$ to compute. □

More generally, the complexity of the above mathematical program is determined by the complexity of constraints needed to define $T_i(\boldsymbol{\theta}_{-i})$: for example, if $T_i(\boldsymbol{\theta}_{-i})$ is a convex set then they are convex programs. Naturally, a major caveat of Theorem 5.2.5 is that $|\Gamma|$ can be very large (for example, $|\Gamma|$ is exponential in combinatorial auctions). However, this issue can be circumvented as long as we have access to a practically-efficient routine for finding welfare-maximizing allocations, that is, for computing $w(\boldsymbol{\theta})$. For small allocation spaces that might amount to an exhaustive search for Γ. In large allocation spaces like in combinatorial auctions, that might involve integer programming or practically-efficient custom search techniques [Rothkopf et al., 1998, Sandholm et al., 2005].

Theorem 5.2.6. *Linear program* (5.2) *can be solved with polynomially many calls to* $w(\cdot)$ *and additional* $poly(\textsf{size}(T_i))$ *time.*

Proof. Let $\widetilde{\Gamma} \subseteq \Gamma$ denote the set of allocations that appear in the description of $T_i(\boldsymbol{\theta}_{-i})$ (*e.g.*, as a list of linear constraints on θ_i). Then, for any $\alpha \notin \widetilde{\Gamma}$, the weakest type $\widetilde{\theta}_i$ that minimizes $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$ over $\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})$ satisfies $\widetilde{\theta}_i[\alpha] = 0$ as $\widetilde{\theta}_i[\alpha]$ is unconstrained. Therefore, the number of variables in the linear program (5.2) is not $|\Gamma|$ but can be bounded by $|\widetilde{\Gamma}| \leq \textsf{size}(T_i)$. A separation oracle for the linear program, given as input to the Ellipsoid algorithm [Grotschel et al., 1993], can be implemented as follows: a candidate point $(\hat{\gamma}, \hat{\theta}_i)$ is feasible if and only if $w(\hat{\theta}_i, \boldsymbol{\theta}_{-i}) \geq \hat{\gamma}$ and $\hat{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})$; else the efficient allocation achieving welfare $w(\hat{\theta}_i, \boldsymbol{\theta}_{-i})$ represents a violated constraint. □

5.3 Main Mechanism and its Guarantees

We now present our main mechanism and analyze the total welfare and revenue it generates. Let $\Delta_i^{\texttt{VCG}}(\boldsymbol{\theta}_{-i}) = \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) - w(0, \boldsymbol{\theta}_{-i})$. (We have $\Delta_i^{\texttt{VCG}}(\boldsymbol{\theta}_{-i}) \leq \widetilde{\theta}_i[\widetilde{\alpha}]$, where $\widetilde{\theta}_i$ is the weakest type in $T_i(\boldsymbol{\theta}_{-i})$ and $\widetilde{\alpha}$ is the efficient allocation on $(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$.) Our mechanism $\mathcal{M}_{\zeta,\lambda}$ is parameterized by two tunable scalar parameters, ζ_i and $\lambda_i > 0$, per agent.

<u>Mechanism $\mathcal{M}_{\zeta,\lambda}$</u>

Input: predictors $T_i : \Theta_{-i} \to \mathcal{P}(\Theta_i)$ for each agent i.

- Agents asked to reveal types $\theta_1, \ldots, \theta_n$.
- Let $\alpha^* = \operatorname{argmax}_{\alpha \in \Gamma} \sum_{i=1}^n \theta_i[\alpha]$ be the efficient allocation. For each agent i let

  $$p_i = \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + \zeta_i - 2^{k_i}\lambda_i - \sum_{j \neq i} \theta_j[\alpha^*]$$

  where k_i is drawn uniformly at random from the set

  $$\left\{0, 1, \ldots, \left\lceil \log_2\left(\frac{\Delta_i^{\texttt{VCG}}(\boldsymbol{\theta}_{-i}) + \zeta_i}{\lambda_i}\right)\right\rceil\right\}.$$

- Let $\mathcal{I} = \{i : \theta_i[\alpha^*] - p_i \geq 0\}$. If agent $i \notin \mathcal{I}$, i is excluded and receives zero utility (zero value and zero payment).[1] If agent $i \in \mathcal{I}$, i enjoys allocation α^* and pays p_i.

[1]One practical consideration is that this step might require a more nuanced implementation of an "outside option"

$\mathcal{M}_{\zeta,\lambda}$ receives prediction $T_i(\boldsymbol{\theta}_{-i}) \subseteq \Theta_i$ for each agent i. Parameter ζ_i is added to the welfare $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$ of the weakest type in Equation 5.1 and allows for an initial modification to the prediction: $\zeta_i > 0$ makes the welfare more aggressive and $\zeta_i < 0$ makes the welfare less aggressive. In order to "smooth out" (more on this later) the final welfare level, the welfare is relaxed by subtracting a random loss of $2^{k_i}\lambda_i$ where k_i is chosen uniformly at random from a logarithmic discretization of the interval $[w(0, \boldsymbol{\theta}_{-i}), w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + \zeta_i]$. Critically, the discretization itself depends on the true types of all other agents $\boldsymbol{\theta}_{-i}$. Finally, if the final welfare level chosen by $\mathcal{M}_{\zeta,\lambda}$ for agent i (which has no dependence on agent i's revealed type) ends up being too aggressive, the mechanism has a final step that explicitly excludes that agent enforcing that they receive zero utility in order to prevent an IR violation due to charging that agent more than their value.

To summarize, $\mathcal{M}_{\zeta,\lambda}$ in essense performs a "doubling search" for the welfare created by agent i's true type, starting from the welfare $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$ of the given prediction. Parameter ζ_i is an "initial hop" that allows the mechanism designer to perform an initial modification to the prediction based on whether he thinks the prediction is likely to be too aggressive (in which case he should set $\zeta_i < 0$) or too conservative (in which case he should set $\zeta_i > 0$). Parameter λ_i controls how quickly the doubling search proceeds and covers the entire welfare range. Before proceeding to the analysis of $\mathcal{M}_{\zeta,\lambda}$, we briefly record that it is incentive compatible and individually rational.

Theorem 5.3.1. *$\mathcal{M}_{\zeta,\lambda}$ is IC and IR.*

Proof. $\mathcal{M}_{\zeta,\lambda}$ is IC for the same reason weakest-type VCG is IC (Theorem 5.1.1). It is IR by definition: all agents with potential IR violations (those not in $\mathcal{I}$) do not participate and receive zero utility. □

Finally, let $\mathcal{M}_{\zeta,0}$ denote—with a slight abuse of notation as it represents the limiting behavior of $\mathcal{M}_{\zeta,\lambda}$ as $\lambda \downarrow 0$—the deterministic mechanism that sets $p_i = \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) - \zeta_i - \sum_{j \neq i} \theta_j[\alpha^*]$. So, $\mathcal{M}_{0,0}$ uses the predicted sets output by each T_i without modification. $\mathcal{M}_{0,0}$ is essentially weakest-type VCG with a final step (that sacrifices welfare) to ensure that no agent's IR constraint is violated. Observe that if the mechanism designer knows the prediction errors $\{\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i})\}$, running $\mathcal{M}_{\zeta,0}$ with $\zeta_i = \Delta_i^{\text{err}}$ for each agent i achieves welfare and revenue equal to OPT. Of course this approach is extremely brittle to perturbations in the mechanism designer's estimate of Δ_i^{err}: if $\zeta_i > \Delta_i^{\text{err}}$ (that is, ζ_i makes the predictor too aggressive) the value and payment extracted from agent i both drop to zero (though observe that if $\zeta_i < \Delta_i^{\text{err}}$ welfare is unaffected and payment decreases linearly). We discuss the sensitivity and precise parameter dependence of $\mathcal{M}_{\zeta,\lambda}$ shortly—first we pin down its precise welfare and revenue guarantees.

5.3.1 Guarantees

We now state, prove, and discuss our main welfare and revenue guarantees on $M_{\zeta,\lambda}$. Define $\log_2^+ : \mathbb{R} \to \mathbb{R}_{\geq 0}$ by $\log_2^+(x) = 0$ if $x < 1$ and $\log_2^+(x) = \log_2(x)$ if $x \geq 1$. We abbreviate $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i})$ and $\Delta_i^{\text{VCG}}(\boldsymbol{\theta}_{-i})$ as Δ_i^{err} and Δ_i^{VCG}, respectively, for readability.

for agents to be indifferent between participating and being excluded versus not participating at all. (We do not pursue this highly application-specific issue in this work.) In auction and matching settings this step is standard: the agent simply receives no items.

Theorem 5.3.2 (Welfare guarantee)**.** *The expected value enjoyed by agent i under $\mathcal{M}_{\zeta,\lambda}$ is equal to*

$$\left(1-\frac{\lceil\log_2^+((\zeta_i-\Delta_i^{\text{err}})/\lambda_i)\rceil}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}\right)\theta_i[\alpha^*].$$

The expected welfare of $\mathcal{M}_{\zeta,\lambda}$ is equal to

$$\sum_{i=1}^n\left(1-\frac{\lceil\log_2^+((\zeta_i-\Delta_i^{\text{err}})/\lambda_i)\rceil}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}\right)\theta_i[\alpha^*]\geq\left(1-\max_i\frac{\lceil\log_2^+((\zeta_i-\Delta_i^{\text{err}})/\lambda_i)\rceil}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}\right)\text{OPT}\,.$$

Proof. Let $K_i=\lceil\log_2((\Delta_i^{\text{VCG}}(\boldsymbol{\theta}_{-i})+\zeta_i)/\lambda_i)\rceil$ and let k_i^* be the smallest $k\in\{0,\ldots,K_i\}$ such that $w(\theta_i,\boldsymbol{\theta}_{-i})\geq\min_{\widetilde{\theta}_i\in T_i(\boldsymbol{\theta}_{-i})}w(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})+\zeta_i-2^k\lambda_i$ (for brevity let $\widetilde{\theta}_i$ denote the weakest type), so

$$k_i^*=\left\lceil\log_2^+\left(\frac{\zeta_i-\Delta_i^{\text{err}}}{\lambda_i}\right)\right\rceil.$$

So, agent i's expected value is

$$\begin{aligned}\mathbb{E}\left[\theta_i[\alpha^*]\cdot\mathbf{1}\left[i\in\mathcal{I}\right]\right]&=\mathbb{E}\left[\theta_i[\alpha^*]\cdot\mathbf{1}\left[w(\theta_i,\boldsymbol{\theta}_{-i})\geq w(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})+\zeta_i-2^{k_i}\lambda_i\right]\right]\\&=\mathbb{E}\left[\theta_i[\alpha^*]\cdot\mathbf{1}\left[k_i\geq k_i^*\right]\right]\\&=\theta_i[\alpha^*]\cdot\Pr\left(k_i\geq k_i^*\right).\end{aligned}$$

and

$$\Pr\left(k_i\geq k_i^*\right)=1-\Pr\left(k_i<k_i^*\right)=1-\frac{k_i^*}{1+K_i}=1-\frac{\lceil\log_2^+((\zeta_i-\Delta_i^{\text{err}})/\lambda_i)\rceil}{1+\lceil\log_2((\zeta_i+\Delta_i^{\text{VCG}})/\lambda_i)\rceil}$$

as claimed. Finally, $\mathbb{E}[\text{welfare}]=\mathbb{E}\left[\sum_{i=1}^n\theta_i[\alpha^*]\cdot\mathbf{1}[i\in\mathcal{I}]\right]$ so summing the above over all agents yields the welfare bound. □

Theorem 5.3.3 (Revenue guarantee)**.** *The expected payment made by agent i in $\mathcal{M}_{\zeta,\lambda}$ is at least*

$$\left(1-\frac{\lceil\log_2^+((\zeta_i-\Delta_i^{\text{err}})/\lambda_i)\rceil}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}\right)(\theta_i[\alpha^*]-(\Delta_i^{\text{err}}-\zeta_i))-\frac{4(\Delta_i^{\mathit{VCG}}+\zeta_i)}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}.$$

The expected revenue of $\mathcal{M}_{\zeta,\lambda}$ is at least

$$\left(1-\max_i\frac{\lceil\log_2^+((\zeta_i-\Delta_i^{\text{err}})/\lambda_i)\rceil}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}\right)\left(\text{OPT}-\sum_{i=1}^n(\Delta_i^{\text{err}}-\zeta_i)\right)-\sum_{i=1}^n\frac{4(\Delta_i^{\mathit{VCG}}+\zeta_i)}{1+\lceil\log_2((\zeta_i+\Delta_i^{\mathit{VCG}})/\lambda_i)\rceil}.$$

Proof. Let k_i^* be defined as in Theorems 5.3.2 and let $\widetilde{\theta}_i$ be the weakest type in $T_i(\boldsymbol{\theta}_{-i})$. We have (as $k_i<k_i^*$ implies $w(\theta_i,\boldsymbol{\theta}_{-i})\geq w(\widetilde{\theta}_i,\boldsymbol{\theta}_{-i})+\zeta_i-2^{k_i}\lambda_i$ so agent i does not participate and pays

nothing)

$$
\begin{aligned}
\mathbb{E}[p_i] &= \sum_{k=k_i^*}^{K_i} \mathbb{E}[p_i | k_i = k] \cdot \Pr(k_i = k) \\
&= \frac{1}{1+K_i} \sum_{k=k_i^*}^{K_i} \left(\theta_i[\alpha^*] - (w(\theta_i, \boldsymbol{\theta}_{-i}) - (w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + \zeta_i - 2^k \lambda_i)) \right) \\
&= \frac{1}{1+K_i} \sum_{k=k_i^*}^{K_i} \left(\theta_i[\alpha^*] - (\Delta_i^{\text{err}} - \zeta_i + 2^k \cdot \lambda_i) \right) \\
&= \left(1 - \frac{k_i^*}{1+K_i} \right) (\theta_i[\alpha^*] - (\Delta_i^{\text{err}} - \zeta_i)) - \frac{\lambda_i}{1+K_i} \sum_{k=k_i^*}^{K_i} 2^k \\
&= \left(1 - \frac{k_i^*}{1+K_i} \right) (\theta_i[\alpha^*] - (\Delta_i^{\text{err}} - \zeta_i)) - \frac{\lambda_i 2^{K_i+1}}{1+K_i},
\end{aligned}
$$

where in the first line we have rewritten the price formula used in $\mathcal{M}_{\zeta,\lambda}$ as a pay-as-bid with discount, as discussed previously. We have $\lambda_i 2^{K_i+1} \leq \lambda_i(4(\Delta_i^{\text{VCG}} + \zeta_i)/\lambda_i) = 4(\Delta_i^{\text{VCG}} + \zeta_i)$. Substituting yields the desired per-agent payment bound and summing over agents yields the desired revenue bound. □

In the above bounds, the term $1 - \frac{\lceil \log_2^+((\zeta_i - \Delta_i^{\text{err}})/\lambda_i) \rceil}{1+\lceil \log_2((\zeta_i + \Delta_i^{\text{VCG}})/\lambda_i) \rceil}$ represents the probability that the modified weakest-type welfare is less than or equal to the true welfare created by agent i, that is, the probability that $i \in \mathcal{I}$. If $\zeta_i \leq \Delta_i^{\text{err}} + \lambda_i$, this probability is equal to 1, and so, as further described below, payment is linear in both ζ_i and Δ_i^{err} in that regime (and value/welfare is constant and optimal). The payment/revenue bounds suffer from an additional additive loss that has no dependence on Δ_i^{err}. This can be interpreted a penalty for how quickly the discretization covers the welfare interval $[w(0, \boldsymbol{\theta}_{-i}), w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + \zeta_i]$ (which enables more "smoothness" in payment degradation) controlled by parameter λ_i—it is decreasing in λ_i.

If the mechanism designer knows Δ_i^{err}, the optimal mechanism would be the deterministic mechanism $\mathcal{M}_{\zeta,0}$ where $\zeta_i = \Delta_i^{\text{err}}$, which obtains welfare and revenue equal to OPT. Furthermore, $\mathcal{M}_{\zeta,\lambda}$ with $\zeta_i = \Delta_i^{\text{err}}$ and $\lambda_i \leq O\left(\frac{\zeta_i + \Delta_i^{\text{VCG}}}{2^{(\zeta_i + \Delta_i^{\text{VCG}})/\varepsilon_i}}\right)$ achieves $\mathbb{E}[p_i] = \theta_i[\alpha^*] - \varepsilon_i$, so its welfare is equal to OPT and its revenue is equal to $\text{OPT} - \sum_i \varepsilon_i$. Of course it is unrealistic to assume that the mechanism designer knows Δ_i^{err} exactly, and this exact tuning leads to brittle performance if the mechanism designer overestimates Δ_i^{err} and sets $\zeta_i > \Delta_i^{\text{err}} + \lambda_i$. We illustrate how welfare and revenue degrade as the parameter tuning and/or the error of the predictors worsen. We plot agent value (Figure 5.2) and payment (Figure 5.3) as a function of ζ_i for different λ_i settings. The key takeaway is that if $\zeta_i \leq \Delta_i^{\text{err}} + \lambda_i$, the ζ_i-adjusted predictor is a conservative prediction, that is, its welfare $\zeta_i + \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \leq w(\theta_i, \boldsymbol{\theta}_{-i})$. In this case, agent value remains optimal (Figure 5.2) and payment degrades *linearly* in ζ_i and in Δ_i^{err} (Figure 5.3). If $\zeta_i > \Delta_i^{\text{err}} + \lambda_i$, value (Figure 5.2) and payment (Figure 5.3) degrade at a rate determined by λ_i. The larger λ_i is the more gradual the decay. Smaller values of λ_i yield greater payment extracted when $\zeta_i \leq \Delta_i^{\text{err}} + \lambda_i$, but lead to more drastic payment degradation for $\zeta_i > \Delta_i^{\text{err}} + \lambda_i$. So,

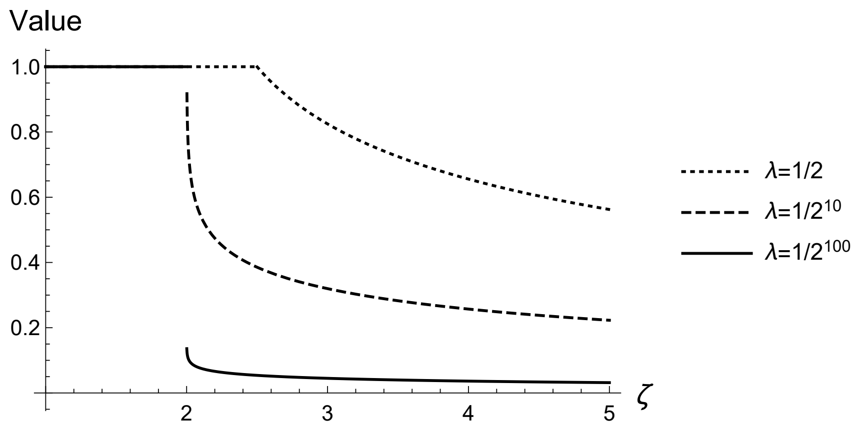


Figure 5.2: An agent's expected value (as a fraction of $\theta_i[\alpha^*]$) as a function of ζ_i for problem parameters $\Delta_i^{\text{VCG}} = 10$, $\Delta_i^{\text{err}} = 2$ (conservative prediction), varying $\lambda_i \in \{2^{-100}, 2^{-10}, 2^{-1}\}$.

the parameter λ_i represents a trade-off between performance in the best case and error tolerance, and is one that the mechanism designer must choose carefully based on his confidence in the prediction. Data-driven algorithm design [Balcan, 2020] provides a toolkit for parameter tuning with provable guarantees.

5.3.2 Consistency and robustness

We situate our results within the consistency-robustness framework studied by the algorithms-with-predictions literature. Furthermore, we retrospectively discuss the failures of some alternate approaches—and how $\mathcal{M}_{\zeta,\lambda}$ addresses those failures—that are solely concerned with consistency and robustness measures.

We say a mechanism is (a, b)-*consistent* and (c, d)-*robust* if when predictions are perfect (which, in our setting, means $\Delta_i^{\text{err}} = 0 \iff \min_{\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) = w(\theta_i, \boldsymbol{\theta}_{-i})$) it satisfies $\mathbb{E}[\text{welfare}] \geq a \cdot \text{OPT}$, $\mathbb{E}[\text{revenue}] \geq b \cdot \text{OPT}$, and satisfies $\mathbb{E}[\text{welfare}] \geq c \cdot \text{OPT}, \mathbb{E}[\text{revenue}] \geq d \cdot \mathsf{VCG}$ independent of the prediction quality (where VCG denotes the revenue of the vanilla VCG mechanism). Consistency demands near-optimal performance when the side information is perfect, and therefore we compete with the total social surplus OPT on both the welfare and revenue fronts. Robustness deals with the case of arbitrarily bad side information, in which case we would like our mechanism's performance to be competitive with vanilla VCG, which already obtains welfare equal to OPT. High consistency and robustness ratios are in fact trivial to achieve, and we will thus largely not be too concerned with these measures—our main goal is to design high-performance mechanisms that degrade gracefully as the prediction errors increase.

Failures of other approaches We discuss a trivial approach that obtains high consistency and robustness ratios, but suffers from huge discontinuous drops in performance even when predictions are nearly perfect, further illustrating the need for a well-tuned instantiation of $\mathcal{M}_{\zeta,\lambda}$.

Trust predictions completely One trivial way of using predictions is to trust them completely, that is, run weakest-type VCG with payments given by Equation (5.1) and exclude any

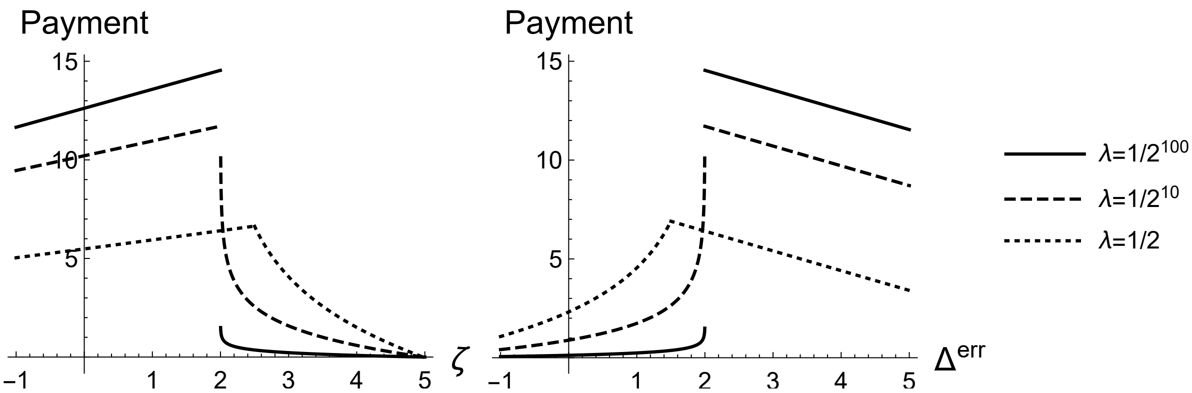


Figure 5.3: **Left:** Payment as a function of ζ_i for problem parameters $\theta_i[\alpha^*] = 15$, $\Delta_i^{\text{VCG}} = 10$, $\Delta_i^{\text{err}} = 2$ (conservative prediction), varying $\lambda_i \in \{2^{-100}, 2^{-10}, 2^{-1}\}$. **Right:** Payment as a function of Δ_i^{err} for problem parameters $\theta_i[\alpha^*] = 15$, $\Delta_i^{\text{VCG}} = 10$ and mechanism parameter $\zeta_i = 2$, varying $\lambda_i \in \{2^{-100}, 2^{-10}, 2^{-1}\}$.

agent who is charged too much (this mechanism is given by $\mathcal{M}_{0,0}$). This approach generates welfare $\sum_{i:\Delta_i^{\text{err}}\geq 0} \theta_i[\alpha^*]$ and revenue $\sum_{i:\Delta_i^{\text{err}}\geq 0} \theta_i[\alpha^*] - \Delta_i^{\text{err}}$ (Theorem 5.2.1). If predictions are perfect, that is, $\Delta_i^{\text{err}} = 0$ for all i, both welfare and revenue are equal to OPT. However, if all predictions are such that $\Delta_i^{\text{err}} < 0$, both welfare and revenue drop to 0. So this mechanism is $(1, 1)$-consistent and $(0, 0)$-robust.

Discard predictions randomly The issue with the above mechanism is that if all predictions are invalid, it generates no welfare and no revenue. We show how randomization can quell that issue. One trivial solution is to discard all predictions with probability β, and trust all predictions completely with probability $(1 - \beta)$. That is, with probability β charge each agent her vanilla VCG price and with probability $1-\beta$ charge each agent her weakest-type price (5.1) (and exclude any agent who is overcharged). This mechanism achieves strong consistency and robustness ratios. Indeed, its expected welfare is $\beta \cdot \text{OPT} + (1 - \beta) \cdot \sum_{i:\Delta_i^{\text{err}}\geq 0} \theta_i[\alpha^*]$ and its expected revenue is $\beta \cdot \texttt{VCG} + (1 - \beta) \cdot (\sum_{i:\Delta_i^{\text{err}}\geq 0} \theta_i[\alpha^*] - \Delta_i^{\text{err}})$. So, it is $(1, 1 - \beta)$-consistent and (β, β)-robust.

This approach suffers from a major issue: its revenue drops drastically the moment predictions are too aggressive, that is, $\Delta_i^{\text{err}} < 0$. In particular, if predictions are too aggressive, *but barely so*, this approach completely misses out on any payments from such agents and drops to the revenue of VCG (which can be drastically smaller than OPT). But, a tiny relaxation of these predictions would have sufficed to increase revenue significantly and perform competitively with OPT. One simple approach is to subtract a relaxation parameter η_i from the welfare of the weakest type consistent with each prediction with some probability, and discard the prediction with complementary probability. If $\eta_i + \Delta_i^{\text{err}} \geq 0$ for all i, then such a mechanism would perform well. The main issue with such an approach is that the moment $\eta_i + \Delta_i^{\text{err}} < 0$, our relaxation by η_i is still too aggressive and the performance drastically drops. Our main mechanism $\mathcal{M}_{\zeta,\lambda}$ essentially selects the η_i randomly from a suitable discretization of the ambient type space to "smooth out" this behavior and extract payments from each agent with reasonable probability.

Consistency and robustness of $\mathcal{M}_{\zeta,\lambda}$ We determine the consistency and robustness ratios for a fixed default tuning of our mechanism, namely with $\zeta_i = \lambda_i = 1$ for all agents. For these parameters, the starting element in the discretization computed in $\mathcal{M}_{1,1}$ uses the unmodified predictions. The remainder of the discretization is via a doubling search of the welfare interval $[w(0, \boldsymbol{\theta}_{-i}), w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})]$ with initial step size 1. In contrast to the trivial approach that either trusted the side information completely or discarded predictions completely, $\mathcal{M}_{1,1}$ does not suffer from large discontinuous drops in welfare nor revenue.

Theorem 5.3.4. *$\mathcal{M}_{1,1}$ is $\left(1, \frac{1}{1+\lceil \log_2(1+\Delta_i^{VCG})\rceil}\right)$-consistent and $\left(\frac{1}{1+\lceil \log_2(1+\Delta_i^{VCG})\rceil}, \frac{1}{1+\lceil \log_2(1+\Delta_i^{VCG})\rceil}\right)$-robust.*

Proof. Consistency: If $\Delta_i^{\text{err}} = 0$, Theorem 5.3.2 implies that for $\zeta_i = \lambda = 1$, the expected value enjoyed by agent i is (deterministically) equal to $\theta_i[\alpha^*]$. With the same notation as in the proofs of Theorems 5.3.2 and 5.3.3, we have $\Pr(i \in \mathcal{I}) \geq \Pr(k_i = k_i^*) = 1/(1 + \lceil \log_2(1 + \Delta_i^{\text{VCG}})\rceil)$ for $\zeta_i = \lambda_i = 1$, and $k_i^* = 0$. So

$$\begin{aligned}\mathbb{E}[p_i] &\geq \frac{1}{1 + \lceil \log_2(1 + \Delta_i^{\text{VCG}})\rceil} \left(\theta_i[\alpha^*] - (w(\theta_i, \boldsymbol{\theta}_{-i}) - (w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + 1 - 2^0))\right) \\ &= \frac{1}{1 + \lceil \log_2(1 + \Delta_i^{\text{VCG}})\rceil} \theta_i[\alpha^*].\end{aligned}$$

Robustness: Independent of Δ_i^{err} there always exists at least one k_i^* such that $w(\theta_i, \boldsymbol{\theta}_{-i}) \geq w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + \zeta_i - 2^{k_i^*}\lambda_i$, so the expected value enjoyed by agent i is at least $\frac{1}{1+\lceil \log_2(1+\Delta_i^{\text{VCG}})\rceil}\theta_i[\alpha^*]$. By the same reasoning, we can lower bound agent i's expected payment, independent of Δ_i^{err}, by $\frac{1}{1+\lceil \log_2(1+\Delta_i^{\text{VCG}})\rceil} \cdot p_i^{\text{VCG}}$, where p_i^{VCG} is agent i's vanilla VCG payment. □

Consistency and robustness, however, do not capture the important effect of how aggressive or how conservative a prediction is on the performance decay of our mechanism, as explained in Section 5.3.1 and the plots in Figures 5.2 and 5.3. The mechanism's performance can furthermore be greatly improved through high quality hyperparameter selection, which can, for example, be learned from data [Balcan, 2020, Khodak et al., 2022].

5.4 Other Forms of Side Information

We apply the weakest-type VCG mechanism to three other formats of side information distinctly different than the model of predictors used in the paper thus far. In each format, the weakest types are instantiated in a different way.

5.4.1 A more expressive prediction language for expressing uncertainty

In this subsection we establish an avenue for richer and more expressive side information languages. We show that the techniques we have developed so far readily extend to an even larger more expressive form of side information that allows one to express varying degrees of uncertainty. We now allow the output of $T_i(\boldsymbol{\theta}_{-i})$ to be an entire probability space $(\Theta_i, \mathcal{F}_i, \mu_i)$ where agent i's ambient type space Θ_i is the sample space, $\mathcal{F}_i$ is a σ-algebra on Θ_i, and μ_i is a probability measure.

$\mathcal{F}_i$ induces a partition of Θ_i into equivalence classes where $\theta_i \equiv \theta_i'$ if $\mathbf{1}[\theta_i \in A] = \mathbf{1}[\theta_i' \in A]$ for all $A \in \mathcal{F}_i$ (so the side-information structure does not distinguish between θ_i and θ_i'). Let $A(\theta_i) = \{\theta_i' : \theta_i \equiv \theta_i'\} \in \mathcal{F}_i$ be the equivalence class of θ_i. In this way the σ-algebra $\mathcal{F}_i$ determines the granularity of knowledge being conveyed by the predictor, and the probability measure $\mu_i : \mathcal{F}_i \to [0, 1]$ establishes uncertainty over this knowledge. Our model of side information in the form of a prediction set $T_i(\boldsymbol{\theta}_{-i}) = T_i$ considered previously in the paper corresponds to the σ-algebra $\mathcal{F}_i = \{\emptyset, T_i, \Theta_i \setminus T_i, \Theta_i\}$ with $\mu_i(\emptyset) = \mu_i(\Theta_i \setminus T_i) = 0$ and $\mu_i(T_i) = \mu_i(\Theta_i) = 1$.

We define the error of a predictor in the natural way. As usual, $\boldsymbol{\theta}$ denotes the agents' (true and) revealed type profile. Define random variable $X_i^{\text{err}} : \Theta_i \to \mathbb{R}_{\geq 0}$ by

$$X_i^{\text{err}}(\theta_i') = w(\theta_i, \boldsymbol{\theta}_{-i}) - \min_{\widetilde{\theta}_i \in A(\theta_i')} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}).$$

X_i^{err} is $\mathcal{F}_i$-measurable since it is (by definition) constant on all atoms of $\mathcal{F}_i$ (sets $A \in \mathcal{F}_i$ such that no nonempty $B \subset A$ is in $\mathcal{F}_i$). The error distribution on $\mathbb{R}_{\geq 0}$ is given by

$$\begin{aligned}\Pr(a \leq X_i^{\text{err}} \leq b) &= \mu_i\left(\left\{\theta_i' \in \Theta_i : a \leq w(\theta_i, \boldsymbol{\theta}_{-i}) - \min_{\widetilde{\theta}_i \in A(\theta_i')} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \leq b\right\}\right) \\ &= \mu_i\left(\bigcup\left\{A(\theta_i') : a \leq w(\theta_i, \boldsymbol{\theta}_{-i}) - \min_{\widetilde{\theta}_i \in A(\theta_i')} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \leq b\right\}\right).\end{aligned}$$

The generalized version of $M_{\zeta,\lambda}$ that receives as input a generalized predictor for each agent i given by $T_i(\boldsymbol{\theta}_{-i}) = (\Theta_i, \mathcal{F}_i, \mu_i)$ works as follows. It samples $\theta_i' \sim \Theta_i$ according to $(\mathcal{F}_i, \mu_i)$, sets $\widetilde{\theta}_i = \text{argmin}_{\hat{\theta}_i \in A(\theta_i')} w(\hat{\theta}_i, \boldsymbol{\theta}_{-i})$, and draws $k_i \sim_{\text{unif.}} \{0, \ldots, \lceil \log_2((\Delta_i^{\text{VCG}} + \zeta_i)/\lambda_i)\rceil\}$, where $\Delta_i^{\text{VCG}} = w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) - w(0, \boldsymbol{\theta}_{-i})$, as before. It then implements the efficient allocation α^* and computes a payment for agent i of $p_i = \theta_i[\alpha^*] - (w(\theta_i, \boldsymbol{\theta}_{-i}) - (w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) + \zeta_i - 2^{k_i}\lambda_i))$, excluding agents for which $p_i > \theta_i[\alpha^*]$. Applying Theorems 5.3.2 and 5.3.3 for a fixed θ_i' and then taking expectation over the draw of θ_i' yields the following guarantees.

Theorem 5.4.1. *The expected value enjoyed by agent i is equal to*

$$\mathop{\mathbb{E}}_{\widetilde{\theta}_i}\left[1 - \frac{\lceil \log_2^+((\zeta_i - X_i^{\text{err}})/\lambda_i)\rceil}{1 + \lceil \log_2((\zeta_i + \Delta_i^{VCG})/\lambda_i)\rceil}\right]\theta_i[\alpha^*]$$

and the expected payment made by agent i is at least

$$\mathop{\mathbb{E}}_{\widetilde{\theta}_i}\left[\left(1 - \frac{\lceil \log_2^+((\zeta_i - X_i^{\text{err}})/\lambda_i)\rceil}{1 + \lceil \log_2((\zeta_i + \Delta_i^{VCG})/\lambda_i)\rceil}\right)(\theta_i[\alpha^*] - (X_i^{\text{err}} - \zeta_i))\right] - \frac{4(\Delta_i^{VCG} + \zeta_i)}{1 + \lceil \log_2((\zeta_i + \Delta_i^{VCG})/\lambda_i)\rceil}.$$

Of course the above discussion assumes the existence of a routine for sampling from the abstract probability space specified by the predictors. We briefly discuss a concrete special form of generalized predictors for which this routine can be concretely described. For each agent i, $T_i(\boldsymbol{\theta}_{-i})$ outputs (i) a partition $(A_1^i, \ldots, A_m^i)$ of the ambient type space Θ_i into disjoint sets, (ii) probabilities $\mu_1^i, \ldots, \mu_m^i \geq 0; \sum_j \mu_j^i = 1$ corresponding to each partition element, and (iii) for each partition element an optional probability density function $f_j^i; \int_{A_j^i} f_j^i = 1$. The prediction represents (i) a belief over what partition element A_j^i the true type θ_i lies in and (ii) if a density is specified, the precise nature of uncertainty over the true type within A_j^i. Our model of side information in the form of a prediction set $T_i(\boldsymbol{\theta}_{-i}) = T_i \subseteq \Theta_i$ considered earlier in the paper corresponds to the partition $(T_i, \Theta_i \setminus T_i)$ with $\mu(T_i) = 1$ and no specified densities. The richer model allows side information to convey finer-grained beliefs; for example one can express quantiles of certainty, precise distributional beliefs, and arbitrary mixtures of these. Here, $M_{\zeta,\lambda}$ first samples a partition element A_j^i according to $(\mu_1^i, \ldots, \mu_m^i)$, and draws $k_i \sim_{\text{unif.}} \{0, \ldots, K_i\}$ where K_i is defined as before. If $f_j^i = \texttt{None}$, it uses weakest type $\widetilde{\theta}_i$ that minimizes $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$ over $\widetilde{\theta}_i \in A_j^i$. Otherwise, it samples $\widetilde{\theta}_i \sim f_j^i$ and uses that as the weakest type.

5.4.2 Constant-parameter agents: types on low-dimensional subspaces

In this section we show how the theory we have developed so far can be used to derive new revenue approximation results when the mechanism designer knows that each agent's type belongs to some low-dimensional subspace of her ambient type space $\mathbb{R}^\Gamma_{\geq 0}$ (these subspaces can be different for each agent).

This is a different setup from the previous sections. So far, we have assumed that $\Theta_i = \mathbb{R}^\Gamma_{\geq 0}$ for all i, that is, there is an ambient type space that is common to all the agents. Side information in the form of predictors $T_i : \times_{j \neq i} \mathbb{R}^\Gamma_{\geq 0} \to \mathbb{R}^\Gamma_{\geq 0}$ are given as input to the mechanism designer, with no assumptions on quality/correctness (and our guarantees in Section 5.3.1 were parameterized by the quality of the predictors). Here, we assume the side information that each agent's type lies in a particular subspace is guaranteed to be valid. Two equivalent ways of stating this setup are (1) that Θ_i is the corresponding subspace for agent i and the mechanism designer receives no additional predictor T_i or (2) $\Theta_i = \mathbb{R}^\Gamma_{\geq 0}$ and $T_i(\boldsymbol{\theta}_{-i}) = \mathbb{R}^\Gamma_{\geq 0} \cap U_i$ where U_i is a subspace of $\mathbb{R}^\Gamma$, and the mechanism designer has the additional guarantee that $\theta_i \in U_i$ (so T_i is a *correct* predictor in that the set it outputs actually contains the agent's true type). We shall use the language of the second interpretation.

In this setting, while the predictors are correct (which implies $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) \geq 0$), their errors Δ_i^{err} can be too large to meaningfully use our previous guarantees. More precisely, since T_i outputs an entire linear subspace of the ambient type space, it contains low welfare types—in particular it contains the zero type that creates welfare $w(0, \boldsymbol{\theta}_{-i})$. So, our randomized mechanism from Section 5.3 is not useful here as its revenue will be no better than vanilla VCG.

In this section we show how to fruitfully use the information provided by the subspaces $U_1, \ldots, U_n$ within the framework of our meta-mechanism. We assume $\Theta_i = [1, H]^\Gamma$, thereby imposing a lower bound of 1 (this choice of lower bound is not important, but the knowledge of some lower bound is needed) and an upper bound of H on agent values. The following more direct bound on p_i in terms of the weakest type's value will be needed.

Lemma 5.4.2. *Let T_i be a predictor such that $\Delta_i^{\text{err}}(\boldsymbol{\theta}_{-i}) \geq 0$. Its weakest-type VCG price p_i given in Equation* (5.1) *satisfies $p_i \geq \widetilde{\theta}_i[\alpha^*]$, where $\widetilde{\theta}_i$ is the weakest type that minimizes $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$ over*

$\widetilde{\theta}_i \in T_i(\boldsymbol{\theta}_{-i})$ *and* α^* *is the efficient allocation achieving* $w(\boldsymbol{\theta})$.

Proof. Let $\widetilde{\theta}_i$ be the weakest type in Θ_i with respect to $\boldsymbol{\theta}_{-i}$. The utility for agent i under $\mathcal{M}$ is

$$\begin{aligned}\theta_i[\alpha^*] - p_i &= \sum_{j=1}^n \theta_j[\alpha^*] - \min_{\theta_i' \in T_i(\boldsymbol{\theta}_{-i})}\left(\max_{\alpha\in\Gamma}\sum_{j\neq i}\theta_j[\alpha] + \theta_i'[\alpha]\right)\\ &= \sum_{j=1}^n \theta_j[\alpha^*] - \left(\max_{\alpha\in\Gamma}\sum_{j\neq i}\theta_j[\alpha] + \widetilde{\theta}_i[\alpha]\right)\\ &\le \sum_{j=1}^n \theta_j[\alpha^*] - \left(\sum_{j\neq i}\theta_j[\alpha^*] + \widetilde{\theta}_i[\alpha^*]\right)\\ &= \theta_i[\alpha^*] - \widetilde{\theta}_i[\alpha^*],\end{aligned}$$

so $p_i \ge \widetilde{\theta}_i[\alpha^*]$, as desired. □

We now describe the formal ingredients and present our mechanism. For each i, the mechanism designer knows that θ_i lies in a k-dimensional subspace $U_i = \text{span}(u_{i,1}, \ldots, u_{i,k})$ of $\mathbb{R}^\Gamma$ where each $u_{i,j} \in \mathbb{R}^\Gamma_{\ge 0}$ lies in the non-negative orthant and $\{u_{i,1}, \ldots, u_{i,k}\}$ is an orthonormal basis for U_i (U_i can depend on $\boldsymbol{\theta}_{-i}$). For simplicity, assume $H = 2^a$ for some positive integer a. Let $\mathcal{L}_{i,j} = \{\lambda u_{i,j} : \lambda \ge 0\} \cap [0, H]^\Gamma$ be the line segment that is the portion of the ray generated by $u_{i,j}$ that lies in $[0, H]^\Gamma$. Let $y_{i,j}$ be the endpoint of $\mathcal{L}_{i,j}$ with $\|y_{i,j}\|_\infty = H$ (the other endpoint of $\mathcal{L}_{i,j}$ is the origin). Let $z^1_{i,j} = y_{i,j}/2$ be the midpoint of $\mathcal{L}_{i,j}$, and for $\ell = 2, \ldots, \log_2 H$ let $z^\ell_{i,j} = z^{\ell-1}_{i,j}/2$ be the midpoint of $\overline{0z^{\ell-1}_{i,j}}$. So $\|z^{\log_2 H}_{i,j}\|_\infty = 1$. We terminate the halving of $\mathcal{L}_{i,j}$ after $\log_2 H$ steps due to the assumption that $\theta_i \in [1, H]^\Gamma$. For every k-tuple $(\ell_1, \ldots, \ell_k) \in \{1, \ldots, \log_2 H\}^k$, let

$$\widetilde{\theta}_i(\ell_1, \ldots, \ell_k) = \sum_{j=1}^k z^{\ell_j}_{i,j}.$$

Furthermore, let

$$W_\ell = \left\{(\ell_1, \ldots, \ell_k) \in \{1, \ldots, \log_2 H\}^k : \min_j \ell_j = \ell\right\}.$$

The sets $W_1, \ldots, W_{\log_2 H}$ form a partition of $\{1, \ldots, \log_2 H\}^k$ into levels, where W_ℓ is the set of points with ℓ_∞-distance $H/2^\ell$ from the origin.

Our mechanism is the following modification of weakest-type VCG, which we denote by $\mathcal{M}_k$.

Mechanism $\mathcal{M}_k$

Input: *correct* subspace predictions $U_1(\boldsymbol{\theta}_{-1}), \ldots, U_n(\boldsymbol{\theta}_{-n})$.

- Agents asked to reveal types $\theta_1, \ldots, \theta_n$.
- Let $\alpha^* = \text{argmax}_{\alpha\in\Gamma} \sum_{i=1}^n \theta_i[\alpha]$ be the efficient allocation. For each agent i let

$$p_i = w(\widetilde{\theta}_i(\ell_{i,1}, \ldots, \ell_{i,k}), \boldsymbol{\theta}_{-i}) - \sum_{j\neq i}\theta_j[\alpha^*]$$

where ℓ_i is chosen uniformly at random from the set $\{1,\ldots,\log_2 H\}$ and $(\ell_{i,1},\ldots,\ell_{i,k})$ is chosen uniformly at random from W_{ℓ_i}.

- Let $\mathcal{I} = \{i : \theta_i[\alpha^*] - p_i \geq 0\}$. If agent $i \notin \mathcal{I}$, i is excluded and receives zero utility (zero value and zero payment).

We now state and prove the welfare and revenue guarantees satisfied by $\mathcal{M}_k$. In the proof, we use the notation $\theta_i \succeq \theta_i'$ to mean $\theta_i[\alpha] \geq \theta_i'[\alpha]$ for all $\alpha \in \Gamma$. We have $\theta_i \succeq \theta_i' \implies w(\theta_i, \boldsymbol{\theta}_{-i}) \geq w(\theta_i', \boldsymbol{\theta}_{-i})$.

Theorem 5.4.3. *$\mathcal{M}_k$ satisfies $\mathbb{E}[$welfare$] \geq \frac{\text{OPT}}{\log_2 H}$ and $\mathbb{E}[$revenue$] \geq \frac{\text{OPT}}{2k(\log_2 H)^k}$.*

Proof. We have $\mathbb{E}[\text{welfare}] = \sum_{i=1}^n \theta_i[\alpha^*] \cdot \Pr(w(\theta_i, \boldsymbol{\theta}_{-i}) \geq w(\widetilde{\theta}(\ell_{i,1},\ldots,\ell_{i,k}), \boldsymbol{\theta}_{-i})) \geq \sum_{i=1}^n \theta_i[\alpha^*] \cdot \Pr(\ell_i = \log_2 H) = \frac{1}{\log_2 H} \cdot \text{OPT}$ (since $\theta_i \succeq \widetilde{\theta}_i(\log_2 H, \ldots, \log_2 H)$). The proof of the revenue guarantee relies on the following key claim: for each agent i, there exists $\ell_{i,1}^*, \ldots, \ell_{i,k}^* \in \{1,\ldots,\log_2 H\}$ such that $\widetilde{\theta}(\ell_{i,1}^*, \ldots, \ell_{i,k}^*) \succeq \frac{1}{2}\theta_i$. To show this, let θ_i^j denote the projection of θ_i onto u_j, so $\theta_i = \sum_{j=1}^k \theta_i^j$ since $\{u_{i,1},\ldots,u_{i,k}\}$ is an orthonormal basis. Let $\ell_{i,j}^* = \min\{\ell : \theta_i^j \succeq z_{i,j}^\ell\}$. Then, $z_{i,j}^{\ell_{i,j}^*} \succeq \frac{1}{2}\theta_i^j$, so

$$\widetilde{\theta}(\ell_{i,1}^*, \ldots, \ell_{i,k}^*) = \sum_{j=1}^k z_{i,j}^{\ell_{i,j}^*} \succeq \sum_{j=1}^k \frac{1}{2}\theta_i^j = \frac{1}{2}\theta_i.$$

We now bound the expected payment. Let $\ell_i^* = \min_j \ell_{i,j}^*$. We have

$$\begin{aligned}
\mathbb{E}[p_i] &\geq \mathbb{E}\left[p_i \mid (\ell_{i,1},\ldots,\ell_{i,k}) = (\ell_{i,1}^*,\ldots,\ell_{i,k}^*)\right] \cdot \Pr\left((\ell_{i,1},\ldots,\ell_{i,k}) = (\ell_{i,1}^*,\ldots,\ell_{i,k}^*)\right) \\
&= \frac{1}{|W_{\ell_i^*}| \log_2 H} \cdot \mathbb{E}\left[p_i \mid (\ell_{i,1},\ldots,\ell_{i,k}) = (\ell_{i,1}^*,\ldots,\ell_{i,k}^*)\right] \\
&\geq \frac{1}{\log_2 H((\log_2 H)^k - (\log_2 H - 1)^k)} \cdot \mathbb{E}\left[p_i \mid (\ell_{i,1},\ldots,\ell_{i,k}) = (\ell_{i,1}^*,\ldots,\ell_{i,k}^*)\right] \\
&\geq \frac{1}{k(\log_2 H)^k} \cdot \mathbb{E}\left[p_i \mid (\ell_{i,1},\ldots,\ell_{i,k}) = (\ell_{i,1}^*,\ldots,\ell_{i,k}^*)\right]
\end{aligned}$$

since the probability of obtaining the correct type $\widetilde{\theta}(\ell_{i,1}^*, \ldots, \ell_{i,k}^*)$ can be written as the probability of drawing the correct "level" $\ell_i^* \in \{1,\ldots,\log_2 H\}$ times the probability of drawing the correct type within the correct level $W_{\ell_i^*}$. We bound the conditional expectation with Lemma 5.4.2:

$$\mathbb{E}\left[p_i \mid (\ell_{i,1},\ldots,\ell_{i,k}) = (\ell_{i,1}^*,\ldots,\ell_{i,k}^*)\right] \geq \widetilde{\theta}_i(\ell_{i,1}^*,\ldots,\ell_{i,k}^*)[\alpha^*] \geq \frac{1}{2} \cdot \theta_i[\alpha^*].$$

Finally,

$$\mathbb{E}[\text{revenue}] = \sum_{i=1}^n \mathbb{E}[p_i] \geq \frac{1}{2k(\log_2 H)^k} \cdot \sum_{i=1}^n \theta_i[\alpha^*] = \frac{1}{2k(\log_2 H)^k} \cdot \text{OPT},$$

as desired. ∎

$\mathcal{M}_k$ can be viewed as a generalization of the $\log H$ revenue approximation in the single-item limited-supply setting that is achieved by a second-price auction with a uniformly random reserve price from $\{H/2, H/4, \ldots, 1\}$ [Goldberg et al., 2001] (and it recovers that guarantee when $k = 1$). Our results apply not only to auctions but to general multidimensional mechanism design problems such as the examples presented in Section 5.1.

5.4.3 Revenue-optimal Groves mechanisms given a known prior

In this section we consider a textbook mechanism design setup wherein the mechanism designer has access to a joint prior distribution over a joint type space $\Theta \subseteq \times_{i=1}^{n} \mathbb{R}_{\geq 0}^{\Gamma}$ over the agents. We formulate the design of the Groves mechanism that maximizes expected revenue over the prior subject to no other constraints other than IC and IR (in particular efficiency is no longer a constraint as in Section 10.2.1). We show that the problem reduces to n independent single-parameter optimization problems for each agent, though we do not pursue the issue of deriving a closed form/more explicit characterization. The optimization problem for each agent depends on that agent's weakest type.

For each agent i, the revealed type vector $\boldsymbol{\theta}_{-i}$ of all other agents induces a conditional distribution D_i over agent i's type. The mechanism designer can then optimize over that conditional distribution directly, and separately, for each agent. The payment-maximizing Groves mechanism can therefore be written as:

$$h_i(\boldsymbol{\theta}_{-i}) = \underset{w \geq \min_{\widetilde{\theta}_i \in \text{supp}(D_i)} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})}{\text{argmax}} \ \mathbb{E}_{\widehat{\theta}_i \sim D_i} \left[\left(\widehat{\theta}_i[\alpha^*(\widehat{\theta}_i, \boldsymbol{\theta}_{-i})] - w(\widehat{\theta}_i, \boldsymbol{\theta}_{-i}) + w \right) \mathbf{1} \left[w \leq w(\widehat{\theta}_i, \boldsymbol{\theta}_{-i}) \right] \right]$$

where $\text{supp}(D_i)$ is the support of D_i. It charges agent i a payment of $h_i(\boldsymbol{\theta}_{-i}) - \sum_{j \neq i} \theta_j[\alpha^*]$, excluding agents who are charged more than their value from the final allocation (that is, agents for which $h_i(\boldsymbol{\theta}_{-i}) > w(\boldsymbol{\theta})$). The weakest type in agent i's type space is inherently baked into the optimization to compute $h_i(\boldsymbol{\theta}_{-i})$. Indeed, the welfare contributed by agent i is lower bounded by $\min_{\widetilde{\theta}_i \in \text{supp}(D_i)} w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$. If D_i is supported on the entire ambient type space $\mathbb{R}_{\geq 0}^{\Gamma}$, the weakest type is the zero type, and welfare is lower bounded by $w(0, \boldsymbol{\theta}_{-i})$, so the expected payment extracted by $h_i(\boldsymbol{\theta}_{-i})$ is lower bounded by the expected vanilla VCG payment.

Finally, we remark that the single-parameter optimization problem to compute $h_i(\boldsymbol{\theta}_{-i})$ really only involves a two-dimensional joint distribution over $(\widehat{\theta}_i[\alpha^*(\widehat{\theta}_i, \boldsymbol{\theta}_{-i})], w(\widehat{\theta}_i, \boldsymbol{\theta}_{-i})) \in \mathbb{R}^2$, that is, the induced joint distribution over agent i's value in the efficient allocation and the welfare she creates (rather than the full type distribution which is $|\Gamma|$ dimensional).

Sale of a single indivisible item to multiple bidders In the sale-of-a-single-item setting with independent, symmetric, and regular (a distribution with cumulative density function $F : \mathbb{R} \to [0, 1]$ and continuous probability density function f is *regular* if $\varphi(v) = v - \frac{1-F(v)}{f(v)}$ is monotonically increasing in v) prior distributions over bidders' values, we show that the above approach recovers Myerson's revenue optimal auction [Myerson, 1981] which in this case is a second-price auction with reserve price $\varphi^{-1}(0)$. We denote bidder i's true and revealed value by v_i. In the single-item setting the efficient allocation gives the item to the highest bidder, so all other bidders receive and pay nothing. Let $i = 1$ be the index of the highest bidder and let $i = 2$ be the index of the second-highest bidder. Then, $w(\widehat{v}_1, \boldsymbol{v}_{-1}) = \max\{\widehat{v}_1, v_2\}$ and $\widehat{v}_1[\alpha^*(\widehat{v}_1, \boldsymbol{v}_{-1})] = \widehat{v}_1 \cdot \mathbf{1}[\widehat{v}_1 > v_2]$ (we abbreviate $\alpha^*(\widehat{v}_1, \boldsymbol{v}_{-1})$ as just α^* in the following). Let F denote the cumulative density function that is common to all bidders and let f be its probability density function. We have

$$\begin{aligned}
h_1(\boldsymbol{v}_{-1}) &= \operatorname*{argmax}_{w \geq v_2} \mathop{\mathbb{E}}_{\widehat{v}_1} \left[(\widehat{v}_1[\alpha^*] - w(\widehat{v}_1, \boldsymbol{v}_{-i}) + w) \cdot \mathbf{1}[w \leq w(\widehat{v}_1, \boldsymbol{v}_{-1})]\right] \\
&= \operatorname*{argmax}_{w \geq v_2} \mathop{\mathbb{E}}_{\widehat{v}_1} \left[(\widehat{v}_1[\alpha^*] - w(\widehat{v}_1, \boldsymbol{v}_{-i}) + w) \cdot \mathbf{1}[w \leq w(\widehat{v}_1, \boldsymbol{v}_{-1})] \mid \widehat{v}_1 \geq v_2\right] \cdot \Pr(\widehat{v}_1 \geq v_2) \\
&\qquad + \mathop{\mathbb{E}}_{\widehat{v}_1} \left[(\widehat{v}_1[\alpha^*] - w(\widehat{v}_1, \boldsymbol{v}_{-i}) + w) \cdot \mathbf{1}[w \leq w(\widehat{v}_1, \boldsymbol{v}_{-1})] \mid \widehat{v}_1 < v_2\right] \cdot \Pr(\widehat{v}_1 < v_2) \\
&= \operatorname*{argmax}_{w \geq v_2} \mathop{\mathbb{E}}_{\widehat{v}_1} \left[w \cdot \mathbf{1}[w \leq \widehat{v}_1] \mid \widehat{v}_1 \geq v_2\right] \cdot (1 - F(v_2)) + \underbrace{(w - v_2) \cdot \mathbf{1}[w \leq v_2] \cdot F(v_2)}_{= 0 \text{ as } w \geq v_2} \\
&= \operatorname*{argmax}_{w \geq v_2} w(1 - F(w))
\end{aligned}$$

which is achieved at $h_1(\boldsymbol{v}_{-1}) = \max\{v_2, \varphi^{-1}(0)\}$ where $\varphi(w) = w - \frac{1 - F(w)}{f(w)}$ is the virtual value function. This is precisely a second-price auction with reserve price $\varphi^{-1}(0)$, which is equivalent to Myerson's optimal auction in this setting (symmetric, regular, and independent bidder priors).

It is clear that in general settings this approach does not yield the revenue optimal auction. Indeed, it is well known that the revenue optimal mechanism in multi-item settings is randomized [Conitzer and Sandholm, 2002, Hart and Nisan, 2013, Hart and Reny, 2015] but the mechanism we present is deterministic. Furthermore, we limit ourselves to a subclass of Groves mechanisms which place a strong restriction on the set of allocations that can be realized—a bidder receives either her winning bundle in the efficient allocation or the empty bundle—while Myerson's revenue-optimal auction even in more general single-item settings can sell the item to a bidder other than the highest bidder.

5.5 Beyond VCG: Weakest-Type Affine-Maximizer Mechanisms

An *affine-maximizer* (AM) mechanism [Roberts, 1979] is a generalization of VCG that modifies the allocation and payments via agent-specific multipliers $\omega = (\omega_1, \ldots, \omega_n) \in \mathbb{R}_{\geq 0}$ and an allocation-based boost function $\tau : \Gamma \to \mathbb{R}_{\geq 0}$. We define the *weakest-type* AM in the following natural way (adopting the setup and notation of Section 5.1.1).

The weakest-type AM parameterized by ω, τ works as follows. Agents' types $\theta_1, \ldots, \theta_n$ are elicited, the allocation used is

$$\alpha_{\omega,\tau} = \operatorname*{argmax}_{\alpha \in \Gamma} \sum_{i=1}^{n} \omega_i \theta_i[\alpha] + \tau(\alpha),$$

and bidder i is charged payment

$$p_i = \frac{1}{\omega_i} \left(\min_{\widetilde{\theta}_i : (\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}} \left(\max_{\alpha \in \Gamma} \sum_{j \neq i} \omega_j \theta_j[\alpha] + \omega_i \widetilde{\theta}_i[\alpha] + \tau(\alpha) \right) - \left(\sum_{j \neq i} \omega_j \theta_j[\alpha_{\omega,\tau}] + \tau(\alpha_{\omega,\tau}) \right) \right).$$

In a vanilla AM there is no minimization and no $\omega_i\widetilde{\theta}_i[\alpha]$ term, so it is already clear that the weakest-type AM is a strict revenue improvement over the vanilla AM. We now generalize Theorem 5.1.1 to the affine-maximizer setting.

Theorem 5.5.1. *For any $\omega \in \mathbb{R}^n_{\geq 0}$ and $\tau : \Gamma \to \mathbb{R}_{\geq 0}$, the weakest-type AM parameterized by ω and τ is incentive compatible and individually rational. Furthermore, if $\mathbf{\Theta}$ is convex, it is revenue optimal among all incentive compatible and individually rational mechanisms that implement the allocation function $\boldsymbol{\theta} \mapsto \operatorname{argmax}_{\alpha\in\Gamma} \sum_{i=1}^n \omega_i\theta_i[\alpha] + \tau(\alpha)$.*

Proof. The proof involves an identical application of revenue equivalence as in the proof of Theorem 5.1.1. The key property required is that weakest-type AMA payment leaves the weakest type with zero utility, that is, the weakest type's IR constraint is binding, which is immediate from the payment formula. Therefore, any payment rule that generates strictly more revenue must violate individual rationality on some type profile of the form $(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$, where $\widetilde{\theta}_i$ minimizes $w(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i})$ over all $\widetilde{\theta}_i$ such that $(\widetilde{\theta}_i, \boldsymbol{\theta}_{-i}) \in \mathbf{\Theta}_i$. □

Let $\mathrm{OPT}(\omega, \lambda) = \sum_{i=1}^n \theta_i[\alpha_{\omega,\lambda}]$ be the welfare of the (ω, λ)-efficient allocation. All of the guarantees satisfied by $\mathcal{M}$ carry over to $\mathcal{M}(\omega, \lambda)$, the only difference being the modified benchmark of $\mathrm{OPT}(\omega, \lambda)$. Of course, $\mathrm{OPT}(\omega, \lambda) \leq \mathrm{OPT}$ is a weaker benchmark than the welfare of the efficient allocation. However, the class of affine maximizer mechanisms is known to achieve much higher revenue than the vanilla VCG mechanism. We leave it as a compelling open question to derive even stronger guarantees on mechanisms of the form $\mathcal{M}(\omega, \lambda)$ when the underlying affine maximizer is known to achieve greater revenue than vanilla VCG. In any case, if one has a tuned high-revenue AM on hand, our techniques (weakest-type AM and randomization) can be appended as a post-processor to further improve revenue.

5.6 Conclusions and Future Research

We developed a versatile new methodology for multidimensional mechanism design that incorporates side information about agent types with the bicriteria goal of generating high social welfare and high revenue simultaneously. We designed mechanisms for a variety of side information formats. Our starting point was the *weakest-type VCG mechanism*, which generalized the mechanism of Krishna and Perry [1998]. A randomized tunable version of that mechanism achieved strong welfare and revenue guarantees that were parameterized by errors in the side information, and could be tuned to boost its performance. We additionally applied the weakest-type mechanism to three other forms of side information: predictions that could express uncertainty, agent types known to lie on low-dimensional subspaces of the ambient type space, and a textbook mechanism design setting where the side information is in the form of a known prior distribution over agent types. Finally, we showed how to generalize our main results to affine-maximizer mechanisms.

There are many new research directions that stem from our work. For example, how far off are our mechanisms from the welfare-versus-revenue Pareto frontier? The weakest-type VCG mechanism is one extreme point, but what does the rest of the frontier look like? One possible approach here would be to extend our theory beyond VCG to the larger class of affine maximizers (which are known to contain higher-revenue mechanisms)—we provided some initial ideas in Section 5.5 but that is only a first step.

Computation and practical auction design: An important facet that we have only briefly discussed is computational complexity. The computations in our mechanism involving weakest types scale with the description complexity of $T_i(\boldsymbol{\theta}_{-i})$ (*e.g.*, the number of constraints, the complexity of constraints, and so on). An important question here is to understand the computational complexity of our mechanisms as a function of the differing (potentially problem-specific) language structures used to describe the predictors $T_i(\boldsymbol{\theta}_{-i})$. In particular, the kinds of side information that are accurate, natural/interpretable, and easy to describe might depend on the specific mechanism design domain. Expressive bidding languages for combinatorial auctions have been extensively studied with massive impact in practice [Sandholm, 2007, 2013]. Can a similar methodology be developed for side information? Chapter 6 takes the first steps along the computational vein, and Chapter 7 studies more expressive side information specifications. Another important direction here is the exploration of the weakest type idea in the realm of mechanism design problems with additional practically-relevant constraints. Examples include mechanism design with investment incentives [Akbarpour et al., 2021], obviously strategy-proof mechanisms [Li, 2017], and other concrete market design applications like sourcing [Sandholm, 2013], catch-share reallocation to prevent overfishing [Bichler et al., 2019], and spectrum auctions [Goetzendorff et al., 2015, Leyton-Brown et al., 2017].

Improved revenue when there is a known prior: Another direction is to improve the revenue of the Bayesian weakest-type VCG mechanism of Krishna and Perry when there is a known prior over agents' types. Here, the benchmark would be efficient welfare in expectation over the prior. Krishna and Perry's mechanism uses weakest types with respect to the prior's support to guarantee efficient welfare in expectation, but its revenue could potentially be boosted significantly by compromising on welfare as in our random expansion mechanism. We took some first steps in Section 5.4.3, but many open questions remain. For example, is there a closed form characterization of the revenue-maximizing Groves mechanism? Can those ideas be applied to revenue optimization of weakest-type affine maximizer mechanisms? Another direction here is to study the setting when the given prior might be inaccurate. Can our random expansion mechanism be used to derive guarantees that depend on the closeness of the given prior to the true prior? Such questions are thematically related to *robust mechanism design* [Bergemann and Morris, 2005]. Another direction along this vein is to generalize our mechanisms to depend on a known prior over prediction errors.

Chapter 6

Weakest Bidder Types and New Core-Selecting Combinatorial Auctions

The design of *combinatorial auctions (CAs)* is a complex task that requires careful engineering along several axes to best serve the application at hand. Just some of these axes are: taming cognitive and communication costs of eliciting and understanding bidders' inherently combinatorial valuations, tractable computation and optimization of economically efficient outcomes that allocate resources to those that value them the most, and determining prices that simplify bidders' incentives while generating acceptable revenues for the seller. These complexities are most evident in fielded applications of CAs including sourcing [Sandholm, 2013, Hohner et al., 2003, Sandholm et al., 2006], spectrum allocation [Cramton, 2013, Leyton-Brown et al., 2017], and others.

The focus of the present chapter is on better pricing rules for CAs. VCG is economically efficient and incentive compatible—a property of great practical importance since it levels the playing ground for bidders by making it worthless to strategize about their individual bids. But, the VCG auction has two major complementary issues (among others [Ausubel and Milgrom, 2006]) that prevent it from being practically viable: low revenue and prices that are not in the *core*. The latter means that some bidders might end up paying so little for their winnings that others who offered more for those same items would take issue. *Core-selecting CAs* fix this issue with prices that ensure no coalition of bidders plus the seller would want to renegotiate for a better deal, but these give up on incentive compatibility. Therefore, most core-selecting CA designs use core prices that minimize bidders' incentives to deviate from truthful bidding.

Core-selecting CAs have been used to auction licenses for wireless spectrum by a number of countries' governments including Australia, Canada, Denmark, Ireland, Mexico, the Netherlands, Portugal, Switzerland, the United Kingdom, and others, generating many billions of dollars in revenue [Cramton, 2013, Palacios-Huerta et al., 2024]. Ausubel et al. [2017] review some of the key design choices of the FCC incentive auction that was completed in the United States in 2017. They suggest that some instances of winners paying zero for certain packages despite losers bidding competitively [Ausubel and Baranov, 2023] could have been avoided with a core-selecting payment rule instead of the VCG rule adopted by the FCC (though a core-selecting rule would have introduced other practical difficulties in other stages of the auction). While the most prominent real-world deployment of core-selecting CAs is probably spectrum auctions, their use has been proposed for other important applications such as electricity markets [Karaca and Kamgarpour, 2019], advertisement markets [Goetzendorff et al., 2015, Niazadeh et al., 2022], and auctions for wind farm development rights [Ausubel and Cramton, 2011].

In this chapter we introduce a new class of core-selecting CAs that improve upon prior designs by taking advantage of bidder information available to the auction designer through constraints on the bidders' *type spaces*. Our starting point is the *weakest-type VCG (WT) auction*, which is a type-space-dependent improvement of VCG [Krishna and Perry, 1998, Balcan et al., 2023]. Our core-selecting CAs build upon the WT auction, and minimize the sum of bidders' incentives to deviate from truthful bidding. They generalize and improve upon the core-selecting CA designs that have been developed in the literature so far, some of which have been successfully used in spectrum auctions [Day and Raghavan, 2007, Day and Milgrom, 2010, Erdil and Klemperer, 2010, Day and Cramton, 2012].

Our Contributions

First, we show that information expressed by type spaces can overcome the following well-known impossibility result due to Othman and Sandholm [2010] and Goeree and Lien [2016]: under unrestricted type spaces either (i) VCG is not in the core in which case no IC core-selecting CA exists or (ii) VCG is the unique IC core-selecting CA. In general CAs where bidders' valuations exhibit complementarities (that is, the value of a bundle is more than the sum of its parts), VCG is typically not in the core. VCG is in the core only under strict conditions on bidder valuations that rule out complementarity (like buyer-submodularity or gross-substitutes [Ausubel and Milgrom, 2002]). We provide a revised and more general version of the impossibility result. Our result (Theorem 6.2.1) states that either (i) WT is not in the core in which case no IC core-selecting CA exists, (ii) WT is the unique IC core-selecting CA, or (iii) there are infinitely many IC core-selecting CAs including WT (and we characterize all such CAs). In particular, vanilla VCG has no bearing on the existence of IC core-selecting CAs (when type spaces are unrestricted VCG and WT are identical, so our result recovers the one by Othman and Sandholm [2010] and Goeree and Lien [2016] in that case).

Second, we devise a new family of type-space-dependent core-selecting CAs that minimize the sum of bidders' incentives to deviate from truthful bidding. Typical core-selecting CAs choose prices that lie on the minimum-revenue face—referred to as the *minimum-revenue core (MRC)*—of the core polytope [Parkes et al., 2001, Day and Raghavan, 2007, Day and Milgrom, 2010, Erdil and Klemperer, 2010, Day and Cramton, 2012]. Day and Milgrom [2010] show that MRC points minimize bidders' total incentive to deviate from truthful bidding (and therefore minimize incentives to deviate in a Pareto sense as well). Our new design chooses core prices that minimize revenue subject to the additional constraint that they lie above WT. We generalize Day and Milgrom's result (which hinges on the assumption of unrestricted typespaces), and show that our revised version of the minimum-revenue core provides optimal incentives for bidders.

Third, we develop new constraint generation routines for computing WT prices. We compare two linear programming formulations of WT price computation: one is due to Balcan et al. [2023] and the other is based on Bikhchandani and Ostroy [2002]. Both linear programs have an exponential number of constraints, so we develop constraint generation routines to solve them. In our experiments, the Balcan et al. [2023] formulation leads to significantly smaller constraint-generation solve times and iterations. On most instances, WT price computation via our constraint generation routine only adds a modest run-time overhead to the cost of winner determination.

Finally, we present proof-of-concept experiments that evaluate the incentive, revenue, and fairness properties of our new core-selecting CAs. We coin and implement three new core-selecting payment rules that select payments on our revised MRC. Our implementation uses the quadratic programming and core-constraint generation technique developed by Day and Cramton [2012].

Related Work

Weakest types The notion of a weakest type consistent with an agent's type space originates from the seminal works of Myerson and Satterthwaite [1983] and Cramton et al. [1987] in the context of efficient trade. It was first presented in an auction context by Krishna and Perry [1998], and later modified by Balcan et al. [2023] to derive revenue guarantees that depend on measures of informativeness of the type space. The weakest-type auction has found applications in other mechanism design settings (like digital goods auctions) as well [Lu et al., 2024].

Equilibrium bidding strategies in core-selecting CAs As core-selecting CAs are not generally incentive compatible, there is a sizable literature that studies bidding strategies and equilibrium outcomes in core-selecting CAs. Such work has generally been limited to very small CA instances with numbers of items and bidders both in the single digits. Goeree and Lien [2016] derive equilibrium strategies for the core-selecting CA of Day and Cramton [2012] and show that *revealed* core prices can be further away from the *true* core than VCG. Ausubel and Baranov [2020] are more optimistic and demonstrate the opposite phenomenon, providing more justification for the use of core-selecting CAs in practice. Bichler et al. [2013] run lab experiments to study bidding behavior and efficiency of the core-selecting combinatorial clock auction format. Ott and Beck [2013] study overbidding equilibria that can arise in core-selecting CAs.

Core-selecting CA design and computation Erdil and Klemperer [2010] introduce the idea of using "reference points" other than VCG [Day and Cramton, 2012] to find closest MRC prices. Bünz et al. [2022] perform a computational evaluation of different core-selecting payment rules that differ in their underlying reference point. Their focus is on computing equilibrium bidding strategies (using modern Bayes-Nash equilibrium solvers [Bosshard et al., 2017]) to evaluate true efficiency, and therefore their evaluation is limited to very small auction instances. These works study the properties of different core points that lie on the same minimum-revenue core. We redefine the minimum-revenue core to depend on the type space information known to the auction designer.

Bünz et al. [2015] provide improvements to the original core-constraint generation algorithms of Day and Raghavan [2007] and Day and Cramton [2012]. Niazadeh et al. [2022] develop non-exact algorithms that converge to core prices (though their experimental evaluation is in a not-fully-combinatorial advertising setting where winner determination is in P, in contrast to the general CA setting where winner determination is NP-complete). Generalizing their algorithms to take advantage of type space information is an interesting direction for future research. Goel et al. [2015], Markakis and Tsikiridis [2019] devise incentive compatible CAs that approximate the core revenue. A drawback of this line of work is that it sacrifices efficiency, which is one of the main tenets that motivates the need for core-selecting CAs in the first place. Goetzendorff et al. [2015] design new bidding languages for auctions with many items and respective techniques for core pricing; Moor et al. [2016] study core-selecting auctions when some items might no longer be available after the auction is run; Othman and Sandholm [2010] develop an iterative core-selecting CA that elicits bids over multiple rounds.

Core selection beyond CAs Some work has studied the design of core-selecting mechanisms in markets beyond auctions. Examples include combinatorial exchanges [Rostek and Yoder, 2015, Bichler and Waldherr, 2017], reallocative mechanisms like the FCC incentive auction [Rostek and Yoder, 2023], and markets with financially-constrained buyers [Batziou et al., 2022, Bichler and Waldherr, 2022].

6.1 Problem Formulation and Background on Core-Selecting CAs

In a combinatorial auction (CA) there is a set $M = \{1, \ldots, m\}$ of indivisible items to be auctioned off to bidders $N = \{1, \ldots, n\}$ who can submit bids for distinct bundles (or packages) of items. Bidder i reports to the auction designer her valuation $v_i : 2^M \to \mathbb{R}_{\geq 0}$ that encodes the maximum value $v_i(S)$ she is willing to pay for every distinct bundle of goods $S \subseteq M$. Let $\boldsymbol{v} = (v_1, \ldots, v_n)$ denote the valuation profile of all bidders, and let $\boldsymbol{v}_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ denote the profile of bids excluding bidder i. For $C \subseteq N$ let $\boldsymbol{v}_C = (v_j)_{j \in C}$ and let $\boldsymbol{v}_{-C} = (v_j)_{j \in N \setminus C}$. We assume bidders report their valuations in the XOR bidding language [Sandholm, 2002a, Nisan, 2000], under which a bidder can only win at most one of the bundles she explicitly placed a nonzero bid for. For bidder i, let $B_i \subseteq 2^M$ be the set of bundles she bid on (assume for notational convenience that each bidder i implicitly submits $v_i(\emptyset) = 0$). Let $\Gamma = \Gamma(B_1, \ldots, B_n) \subseteq B_1 \times \cdots \times B_n$ denote the set of feasible allocations, that is, the set of partitions $S_1, \ldots, S_n$ of M with $S_i \in B_i$ for each i and $S_i \cap S_j = \emptyset$ for each i, j. We use boldface $\boldsymbol{S} = (S_1, \ldots, S_n) \in \Gamma(B_1, \ldots, B_n)$ to denote a feasible allocation.

Before bids/valuations are submitted, bidder i's valuation v_i, also called her *type*, is her own private information. The auction designer might have some prior information about the bidders, and that is modeled by the *joint type space* of the bidders, denoted $\boldsymbol{\Theta} \subseteq \times_{i \in N} \mathbb{R}^{2^m}$. The auction designer knows that $\boldsymbol{v} \in \boldsymbol{\Theta}$. Given $\boldsymbol{v}_{-i}$, let $\Theta_i(\boldsymbol{v}_{-i}) = \{\hat{v}_i : (\hat{v}_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}\}$ be the projected type space of bidder i. So, after seeing the revealed bids $\boldsymbol{v}_{-i}$ of all other bidders, the auction designer knows $v_i \in \Theta_i(\boldsymbol{v}_{-i})$. This model of type spaces begets a rich and expressive language of bidder information available to the auction designer—$\boldsymbol{\Theta}$ can represent any statement of the form "the joint valuation profile $\boldsymbol{v}$ of all bidders satisfies property P" (Balcan et al. [2023] provide concrete examples). The typical assumption in mechanism design is an unrestricted type space $\Theta = \times_{i \in N} \mathbb{R}^{2^m}_{\geq 0}$ (what is usually assumed is the existence of a known prior distribution over the type space). In contrast, we will be concerned with explicit representations of the auction designer's knowledge via the type space and how that influences both practical computation and the auction design itself.

Efficient auctions An auction is determined by its allocation rule and its payment rule. In this chapter we are concerned with *efficient auctions*. An efficient auction selects the efficient (welfare-maximizing) allocation:

$$\boldsymbol{S}^* = (S_1^*, \ldots, S_n^*) = \operatorname*{argmax}_{\boldsymbol{S} \in \Gamma} \sum_{j \in N} v_j(S_j).$$

The *winner determination problem* of computing the efficient allocation is NP-complete (by a reduction from weighted set packing), but solving its integer programming formulation is generally a routine task for modern integer programming solvers. Let $w(\boldsymbol{v}) = \max_{S \in \Gamma} \sum_{j \in N} v_j(S_j)$ denote the efficient welfare.

We recall the definitions of the VCG and WT mechanisms from the previous chapter, rewritten here in the context of CAs.

Vickrey-Clarke-Groves (VCG) Auction The classical auction due to Vickrey [1961], Clarke [1971], and Groves [1973] (VCG) chooses the efficient allocation $\boldsymbol{S}^*$, and charges bidder i a payment of

$$p_i^{\mathtt{VCG}}(\boldsymbol{v}) = w(0, \boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(S_j^*).$$

Let $\boldsymbol{p}^{\mathtt{VCG}} = (p_1^{\mathtt{VCG}}, \ldots, p_n^{\mathtt{VCG}})$ denote the vector of VCG payments. VCG is incentive compatible and individually rational. Generally, to implement the VCG auction one must solve winner determination $n+1$ times—once to compute $w(\boldsymbol{v})$ and the efficient allocation, and once per bidder to compute $w(0, \boldsymbol{v}_{-i})$ in the formula for p_i.

Weakest-Type VCG (WT) Auction The weakest-type VCG (WT) auction [Krishna and Perry, 1998, Balcan et al., 2023] chooses the efficient allocation $\boldsymbol{S}^*$ achieving welfare $w(\boldsymbol{v})$ and charges bidder i a payment of

$$p_i^{\mathtt{WT}}(\boldsymbol{v}) = \min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(S_j^*). \tag{6.1}$$

In Equation (6.1), the bid vector $\widetilde{v}_i$ achieving the minimum is the *weakest type* in $\Theta_i(\boldsymbol{v}_{-i})$. Let $\boldsymbol{p}^{\mathtt{WT}} = (p_1^{\mathtt{WT}}, \ldots, p_n^{\mathtt{WT}})$ denote the vector of WT payments. WT is revenue maximizing among all efficient, IC, and IR auctions.

Core-Selecting CAs and the Minimum-Revenue Core Let $W = \{i \in N : S_i^* \neq \emptyset\}$ be the set of winning bidders in the efficient allocation $\boldsymbol{S}^*$. A combinatorial auction is in the *core* if (i) it chooses the efficient allocation $\boldsymbol{S}^*$ and (ii) prices $\boldsymbol{p}$ lie in the *core polytope*, defined by *core constraints* for every group of winning bidders and IR constraints:

$$\mathsf{Core}(\boldsymbol{v}) = \left\{ \boldsymbol{p} \in \mathbb{R}^W : \begin{array}{c} \displaystyle\sum_{i \in W \setminus C} p_i \geq w(\mathbf{0}, \boldsymbol{v}_C) - \sum_{j \in C} v_j(S_j^*) \ \forall C \subseteq N, \\ v_i(S_i^*) - p_i \geq 0 \ \ \forall i \in W \end{array} \right\}. \tag{6.2}$$

This formulation of the core gives rise to a direct interpretation of core prices as "group VCG prices": any set of winners must in aggregate pay the externality they impose on the other bidders (our formulation is not the typical formulation of the core, which is a notion originally from cooperative game theory, but is most convenient from an implementation/mathematical programming perspective as in Day and Raghavan [2007], Day and Cramton [2012], Bünz et al. [2015]). When $W \setminus C = \{i\}$ is a singleton, the core constraint reads $p_i \geq p_i^{\mathtt{VCG}}$.

The *minimum-revenue core (MRC)* is the set $\mathsf{MRC} = \operatorname{argmin}\{\|\boldsymbol{p}\|_1 : \boldsymbol{p} \in \mathsf{Core}\}$ that consists of all core prices of minimal revenue. Day and Raghavan [2007], Day and Milgrom [2010] show that the MRC captures exactly the set of core prices that minimize the sum of bidders' incentives to deviate from truthful bidding. The MRC is not unique and there can be (infinitely) many MRC prices. Some core-selecting CAs that select unique MRC points that have been proposed are VCG nearest [Day and Cramton, 2012], which finds the MRC point closest in Euclidean distance to VCG, and zero nearest [Erdil and Klemperer, 2010], which finds the MRC point closest in Euclidean distance to the origin.

Since core-selecting CAs are in general not IC, a core-selecting CA only guarantees that prices are in the *revealed* core with respect to reported bids. But, from a regulatory viewpoint, the revealed core is nonetheless a useful solution concept since core constraints prevent any group of bidders from lodging a meaningful complaint based on their actual bids [Bünz et al., 2022].

6.2 Impossibility of IC Core-Selecting CAs

We revisit the following dichotomy for core-selecting CAs when type spaces are unrestricted [Goeree and Lien, 2016, Othman and Sandholm, 2010]: either (i) VCG is not in the core which implies no IC core-selecting auction exists or (ii) VCG is in the core and is the unique IC core-selecting auction. That dichotomy relies on the assumption that Θ is unrestricted, that is, $\Theta = \mathbb{R}^{2^m}_{\geq 0}$. We revise and generalize that result to depend on bidders' type spaces. The proof relies on the revenue optimality of WT prices subject to efficiency, IC, and IR [Balcan et al., 2023, Krishna and Perry, 1998].

Theorem 6.2.1. *Let $\boldsymbol{\Theta}$ be compact and connected. Let $\boldsymbol{v}$ be the vector of bidders' true valuations. If $\boldsymbol{p}^{\mathtt{WT}}(\boldsymbol{v}) \notin \mathsf{Core}(\boldsymbol{v})$, no incentive compatible core-selecting CA exists. Otherwise, let $\mathfrak{C} \subseteq 2^N$ be the set of core constraints that $\boldsymbol{p}^{\mathtt{WT}}$ satisfies with equality. Let $\mathfrak{C}' = \{C' \subseteq N : C' \cap C = \emptyset\ \forall C \in \mathfrak{C}\}$ and for $C' \in \mathfrak{C}'$ let*

$$s(C') = \sum_{i \in W \setminus C'} p_i^{\mathtt{WT}} - w(\mathbf{0}, \boldsymbol{v}_{C'}) + \sum_{j \in C'} v_j(S_j^*)$$

be the slack of the C'-core constraint. Then for any $C' \in \mathfrak{C}'$ all prices in the set

$$\left\{\left(\boldsymbol{p}^{\mathtt{WT}}_{W \cap C'} - \boldsymbol{\varepsilon}, \boldsymbol{p}^{\mathtt{WT}}_{W \setminus C'}\right) : \|\boldsymbol{\varepsilon}\|_1 \leq s(C'), \boldsymbol{\varepsilon} \in \mathbb{R}^{W \cap C'}_{\geq 0}\right\}$$

are in the core and are attainable via an incentive compatible CA.

Proof. If $\boldsymbol{p}^{\mathtt{WT}} \notin \mathsf{Core}(\boldsymbol{v})$, it must be the case that for any $\boldsymbol{p} \in \mathsf{Core}(\boldsymbol{v})$ there exists i such that $p_i > p_i^{\mathtt{WT}}$. This means no IC core-selecting CA can exist because $\boldsymbol{p}^{\mathtt{WT}}$ is bidder-wise payment optimal subject to efficiency, IC, and IR [Balcan et al., 2023].

If $\boldsymbol{p}^{\texttt{WT}} \in \textsf{Core}(\boldsymbol{v})$, the price vector $(\boldsymbol{p}^{\texttt{WT}}_{W\cap C'} - \boldsymbol{\varepsilon}, \boldsymbol{p}^{\texttt{WT}}_{W\setminus C'})$ is also in the core for any $\boldsymbol{\varepsilon}$ with $\|\boldsymbol{\varepsilon}\|_1 \leq s(C')$ by construction. We now argue that there exists an IC auction that yields these prices. Consider the efficient Groves mechanism that uses pivot terms

$$h_i(\boldsymbol{v}_{-i}) = t_i \cdot w(0, \boldsymbol{v}_{-i}) + (1 - t_i) \cdot \min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$$

where $t_i \in [0, 1]$ is a parameter that does not depend on i's revealed type v_i. Such a Groves mechanism is IC and, since it produces payments between VCG and WT, IR. By continuity, there exist parameters $\boldsymbol{t} = ((t_i)_{i\in W\cap C'}, \boldsymbol{0})$ so that the Groves mechanism produces prices $(\boldsymbol{p}^{\texttt{WT}}_{W\cap C'} - \boldsymbol{\varepsilon}, \boldsymbol{p}^{\texttt{WT}}_{W\setminus C'})$. $\square$

Theorem 6.2.1 implies that if WT is in the core, there is a potential continuum of IC core-selecting payment rules obtained by decreasing WT prices along non-binding faces of the core. In particular, the existence of IC core-selecting CAs does not depend on VCG prices but on WT prices. WT and VCG coincide when type spaces do not convey sufficient information about the additional welfare created by a bidder: $p_i^{\texttt{WT}} = p_i^{\texttt{VCG}}$ if and only if $\min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) = w(0, \boldsymbol{v}_{-i})$, which says that the information conveyed by $\Theta_i(\boldsymbol{v}_{-i})$ about bidder i is so weak that it cannot even guarantee that i's presence adds any nonzero welfare to the auction. In this case, Theorem 6.2.1 recovers the impossibility result of Othman and Sandholm [2010] and Goeree and Lien [2016].

6.3 Our New Core-Selecting CAs and their Properties

In this section we introduce our new class of core-selecting CAs based on weakest types, and prove that it provides bidders with optimal incentives (by minimizing the sum of bidders' incentives to deviate, therefore providing optimal incentives in a Pareto sense as well) among all core-selecting CAs. Our result generalizes the result of Day and Milgrom [2010] which was in the setting of unrestricted type spaces (our result recovers theirs in the unrestricted case).

In Section 6.2 we have shown above that if WT is not in the core, then all core-selecting CAs necessarily violate incentive compatibility. To measure the incentive violations of a core-selecting CA, we borrow the notion of an incentive profile from Day and Milgrom [2010]. The *utility profile* (resp., *deviation profile*) of an efficient CA with payment rule $\boldsymbol{p}(\boldsymbol{v})$ is given by $\{\mu_i^{\boldsymbol{p}}(\boldsymbol{v})\}_{i\in W}$ (resp. $\{\delta_i^{\boldsymbol{p}}(\boldsymbol{v})\}_{i\in W}$), where

$$\mu_i^{\boldsymbol{p}}(\boldsymbol{v}) = \max_{\widehat{v}_i} \Big(v_i(\hat{S}_i) - p_i(\widehat{v}_i, \boldsymbol{v}_{-i}) \Big)$$

is bidder i's maximum obtainable utility from misreporting and

$$\delta_i^{\boldsymbol{p}}(\boldsymbol{v}) = \max_{\widehat{v}_i} \Big(v_i(\hat{S}_i) - p_i(\widehat{v}_i, \boldsymbol{v}_{-i}) \Big) - (v_i(S_i^*) - p_i(v_i, \boldsymbol{v}_{-i})) = \mu_i^{\boldsymbol{p}}(\boldsymbol{v}) - (v_i(S_i^*) - p_i(v_i, \boldsymbol{v}_{-i}))$$

is bidder i's maximum utility gain over truthful bidding ($\hat{\boldsymbol{S}}$ denotes the efficient allocation under reported bid profile $(\hat{v}_i, \boldsymbol{v}_{-i})$). Our goal is to define core-selecting payment rules $\boldsymbol{p}$ that minimize the sum of bidders' incentives to deviate, which is precisely $\sum_i \delta_i^{\boldsymbol{p}}(\boldsymbol{v})$. The quantity $\delta_i^{\boldsymbol{p}}$ can be viewed as a form of ex-post regret for truthful bidding for bidder i. Throughout this section, $\boldsymbol{v}$ denotes the true valuations of the bidders.

The following lemma generalizes Day and Raghavan [2007, Theorem 3.2]; its proof is identical to theirs.

Lemma 6.3.1. *Let $\hat{\boldsymbol{p}}$ be any payment rule that implements the efficient allocation such that $\hat{p}_i \geq p_i^{\mathtt{WT}}$. Then, $\mu_i^{\hat{\boldsymbol{p}}}(\boldsymbol{v}) \leq v_i(S_i^*) - p_i^{\mathtt{WT}}(\boldsymbol{v})$ and $\delta_i^{\hat{\boldsymbol{p}}}(\boldsymbol{v}) \leq \hat{p}_i(\boldsymbol{v}) - p_i^{\mathtt{WT}}(\boldsymbol{v})$. That is, the maximum utility winner i can obtain by misreporting under $\hat{\boldsymbol{p}}$ is no more than her utility under $\boldsymbol{p}^{\mathtt{WT}}$.*

Proof. Suppose for the sake of contradiction that there is a misreport v_i' for bidder i that gives her utility more than $v_i(S_i^*) - p_i^{\mathtt{WT}}(\boldsymbol{v})$, that is, $v_i(S_i') - \hat{p}_i(v_i', \boldsymbol{v}_{-i}) > v_i(S_i^*) - p_i^{\mathtt{WT}}(\boldsymbol{v})$ where $\boldsymbol{S}'$ is the efficient allocation for bid profile $(v_i', \boldsymbol{v}_{-i})$. Since $\hat{p}_i \geq p_i^{\mathtt{WT}}$, $v_i(S_i') - p_i^{\mathtt{WT}}(v_i', \boldsymbol{v}_{-i}) \geq v_i(S_i') - \hat{p}_i(v_i', \boldsymbol{v}_{-i})$, which, combined with the above, yields $v_i(S_i') - p_i^{\mathtt{WT}}(v_i', \boldsymbol{v}_{-i}) > v_i(S_i^*) - p_i^{\mathtt{WT}}(\boldsymbol{v})$. Incentive compatibility of WT is violated, a contradiction. □

The following result is an adaptation of Day and Milgrom [2010, Theorem 2]; the proof is similar to theirs.

Theorem 6.3.2. *Let $\hat{\boldsymbol{p}}$ be any IR payment rule that implements the efficient allocation such that $\hat{p}_i \geq p_i^{\mathtt{WT}}$. Let v_i' denote the misreport for winner i defined by $v_i'(S_i^*) = p_i^{\mathtt{WT}}(\boldsymbol{v})$, $v_i'(S) = 0$ for all $S \neq S_i^*$. Then, v_i' is a best response for i that gives her utility equal to $v_i(S_i^*) - p_i^{\mathtt{WT}}(\boldsymbol{v})$. That is, under $\hat{\boldsymbol{p}}$, winner i can always guarantee herself utility equal to what her utility would have been under $\boldsymbol{p}^{\mathtt{WT}}$.*

Proof. Reporting v_i' does not change the efficient allocation since $p_i^{\mathtt{WT}} \geq p_i^{\mathtt{VCG}}$ (and i's VCG price is her lowest possible misreport that preserves her winning bundle). So, the IR constraint for $\hat{\boldsymbol{p}}$ requires $v_i'(S_i^*) - \hat{p}_i(v_i', \boldsymbol{v}_{-i}) \geq 0$. Expanding the left-hand side yields $v_i'(S_i^*) - \hat{p}_i(v_i', \boldsymbol{v}_{-i}) = p_i^{\mathtt{WT}}(\boldsymbol{v}) - \hat{p}_i(v_i', \boldsymbol{v}_{-i}) = v_i(S_i^*) - (w(v_i, \boldsymbol{v}_{-i}) - \min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})) - \hat{p}_i(v_i', \boldsymbol{v}_{-i})$. So, $v_i(S_i^*) - \hat{p}_i(v_i', \boldsymbol{v}_{-i}) \geq w(v_i, \boldsymbol{v}_{-i}) - \min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$. The right-hand side is precisely i's utility under $\boldsymbol{p}^{\mathtt{WT}}$. By Lemma 6.3.1, this constitutes a best response. □

Theorem 6.3.2 allows us to characterize the subset of points that minimize the sum of bidders' incentives to deviate of any upwards closed region. They are exactly the set of points of minimal revenue. Given a price vector $\hat{\boldsymbol{p}} \in \mathbb{R}^W$ and any closed region $\mathcal{A} \subseteq \mathbb{R}^W$, let

$$\mathsf{MR}_{\mathcal{A}}(\hat{\boldsymbol{p}}) = \operatorname{argmin} \left\{ \|\boldsymbol{p}\|_1 : \boldsymbol{p} \in \mathcal{A}, \hat{\boldsymbol{p}} \leq \boldsymbol{p} \leq (v_i(S_i^*))_{i \in W} \right\}$$

be the set of IR price vectors in $\mathcal{A}$ of minimal revenue that lie above $\hat{\boldsymbol{p}}$.

Theorem 6.3.3. *Let $\mathcal{A} \subseteq \mathbb{R}^W$ be upwards closed. Then*

$$\mathsf{MR}_{\mathcal{A}}(\boldsymbol{p}^{\mathtt{WT}}) \subseteq \operatorname{argmin} \left\{ \sum_{i \in W} \delta_i^{\boldsymbol{p}}(\boldsymbol{v}) : \boldsymbol{p} \in \mathcal{A} \right\}.$$

Proof. Consider the map on pricing rules $\boldsymbol{p} \mapsto \boldsymbol{p}'$ defined by

$$p_i'(\boldsymbol{v}) = \begin{cases} p_i(\boldsymbol{v}) & p_i(\boldsymbol{v}) \geq p_i^{\mathtt{WT}}(\boldsymbol{v}) \\ p_i^{\mathtt{WT}}(\boldsymbol{v}) & p_i(\boldsymbol{v}) < p_i^{\mathtt{WT}}(\boldsymbol{v}). \end{cases}$$

This map satisfies the property that $\delta_i^{\boldsymbol{p}'}(\boldsymbol{v}) \leq \delta_i^{\boldsymbol{p}}(\boldsymbol{v})$ since if $p_i(\boldsymbol{v}) \geq p_i^{\texttt{WT}}(\boldsymbol{v})$, p_i is unchanged, and otherwise the WT price is used for which $\delta_i^{\boldsymbol{p}^{\texttt{WT}}}(\boldsymbol{v}) = 0$ due to incentive compatibility. So, for any price vector $\boldsymbol{p} \in \mathcal{A}$ such that $\boldsymbol{p} \not\geq \boldsymbol{p}^{\texttt{WT}}$, the described map produces $\boldsymbol{p}'$ such that $\boldsymbol{p}' \in \mathcal{A}$ (since $\mathcal{A}$ is upwards closed), $\boldsymbol{p}' \geq \boldsymbol{p}^{\texttt{WT}}$, and $\boldsymbol{p}'$ has deviation profile no worse than $\boldsymbol{p}$. It therefore suffices to consider the subset of $\mathcal{A}$ that lies above $\boldsymbol{p}^{\texttt{WT}}$ to find prices in $\mathcal{A}$ that minimize the sum of bidders' incentives to deviate. For $\boldsymbol{p} \geq \boldsymbol{p}^{\texttt{WT}}$ we have $\delta_i^{\boldsymbol{p}}(\boldsymbol{v}) = p_i(\boldsymbol{v}) - p_i^{\texttt{WT}}(\boldsymbol{v})$ by Theorem 6.3.2. So minimizing $\sum_i \delta_i^{\boldsymbol{p}}$ is equivalent to minimizing $\sum_i p_i$, which completes the proof. □

Let $\mathsf{MRC}(\widehat{\boldsymbol{p}}) = \mathsf{MR}_{\mathsf{Core}}(\widehat{\boldsymbol{p}})$ denote the minimum-revenue core above $\widehat{\boldsymbol{p}}$ (this is *not* the portion of the vanilla MRC that lies above $\widehat{\boldsymbol{p}}$; it is the minimum-revenue section of the subset of the core that lies above $\widehat{\boldsymbol{p}}$). Applying Theorem 6.3.3 yields:

Corollary 6.3.4. $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}}(\boldsymbol{v})) \subseteq \operatorname{argmin}\left\{\sum_{i \in W} \delta_i^{\boldsymbol{p}}(\boldsymbol{v}) : \boldsymbol{p} \in \mathsf{Core}(\boldsymbol{v})\right\}$.

Any payment rule $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}}(\boldsymbol{v}))$ is therefore incentive optimal in a Pareto sense as well: there is no other core-selecting $\boldsymbol{p}'$ such that $\delta_i^{\boldsymbol{p}'}(\boldsymbol{v}) \leq \delta_i^{\boldsymbol{p}}(\boldsymbol{v})$ for all i and $\delta_{i^*}^{\boldsymbol{p}'}(\boldsymbol{v}) \leq \delta_{i^*}^{\boldsymbol{p}}(\boldsymbol{v})$ for some i^*. Corollary 6.3.4 generalizes the results of Day and Raghavan [2007], Day and Milgrom [2010] since when $\Theta_i(\boldsymbol{v}_{-i})$ is unrestricted for each agent i, $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}}) = \mathsf{MRC}(\boldsymbol{p}^{\texttt{VCG}})$ which is the (unrestricted) minimum-revenue core they consider.

Corollary 6.3.4 gives strong theoretical justification for payment rules that lie on $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$. We expand on specific rules in Section 6.5, but as one concrete example one of the rules we coin—*WT nearest*—selects the price vector in $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ that minimizes Euclidean distance to $\boldsymbol{p}^{\texttt{WT}}$. WT nearest is the most direct generalization of the VCG nearest rule proposed by Day and Cramton [2012] that has been successfully used in spectrum auctions. In order to implement rules like WT nearest, we need algorithms for computing $\boldsymbol{p}^{\texttt{WT}}$. That is the topic of the next section (Section 6.4). We conclude this section with an example illustrating some of the key concepts introduced so far.

Example 6.3.5. Consider the CA with three items $\{a, b, c\}$ and 10 single-minded bidders who submit the following bids: $v_1(a) = 20$, $v_2(b) = 20$, $v_3(c) = 20$, $v_4(ab) = 28$, $v_5(ac) = 26$, $v_6(bc) = 23$, $v_7(a) = 10$, $v_8(b) = 10$, $v_9(c) = 10$, $v_{10}(abc) = 41$ (this a slight modification of an example from Day and Cramton [2012]). Bidders 1, 2, and 3 win in the efficient allocation and their VCG prices are $\boldsymbol{p}^{\texttt{VCG}} = (10, 10, 10)$. Say

$$\Theta_1 = \mathbb{R}_{\geq 0}, \Theta_2 = \{v_2(b) \geq 17\}, \Theta_3 = \{v_3(c) \geq 15\},$$

so $\boldsymbol{p}^{\texttt{WT}} = (10, 17, 15)$. The core constraints are given by

$$\{p_1, p_2, p_3 \geq 10, p_1 + p_2 \geq 28, p_1 + p_3 \geq 26, p_2 + p_3 \geq 23, p_1 + p_2 + p_3 \geq 41\}.$$

The vanilla VCG-nearest point of Day and Cramton [2012] on $\mathsf{MRC}(\boldsymbol{p}^{\texttt{VCG}})$ is $(14, 14, 13)$ and the WT-nearest point on $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ is $(11, 17, 15)$. Figure 6.1 is an illustration of this example.

6.4 Computing Weakest-Type Prices

In this section we develop techniques to compute $\boldsymbol{p}^{\texttt{WT}}$, which are needed as a subroutine for computing the payments of our new core-selecting CAs. Balcan et al. [2023] provide an initial

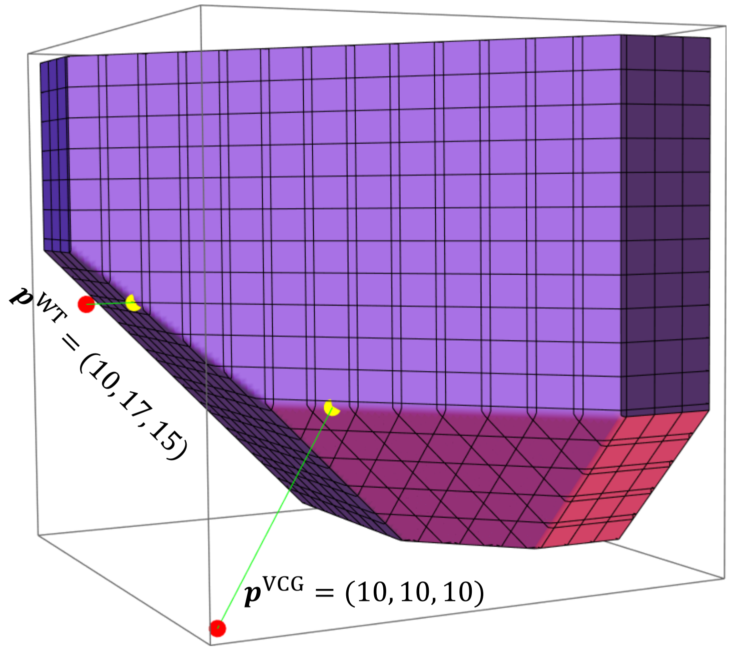


Figure 6.1: Price vectors $\boldsymbol{p}^{\text{VCG}}$ and $\boldsymbol{p}^{\text{WT}}$ (in red) and their nearest respective minimum-revenue core points (in yellow, connected by a green line) as derived in Example 6.3.5. $\mathsf{MRC}(\boldsymbol{p}^{\text{WT}})$ lies on a different face of the core than $\mathsf{MRC}(\boldsymbol{p}^{\text{VCG}})$ and is of higher revenue.

theoretical investigation of WT computation, and one of our approaches builds upon their formulation, but we are the first to develop practical techniques and evaluate them via experiments.

Recall $B_i \subseteq 2^M$ is the set of bundles bidder i bids on, so, for each $S \in B_i$, bidder j submits her value $v_i(S)$ which is the maximum amount she would be willing to pay to win bundle S. For $\boldsymbol{B} = (B_1, \ldots, B_n)$, $\Gamma(\boldsymbol{B})$ denotes the set of feasible selections of winning bids.

6.4.1 Background on Winner Determination Formulations

The standard integer programming formulation of winner determination involves variables $x_j(S)$ indicating whether bidder j is allocated bundle S:

$$w(\boldsymbol{v}) = \max \left\{ \sum_{j \in N} \sum_{S \in B_j} v_j(S) x_j(S) : \begin{array}{l} \displaystyle\sum_{j \in N} \sum_{S \in B_j, S \ni i} x_j(S) \leq 1 \ \ \forall i \in M \\ \displaystyle\sum_{S \in B_j} x_j(S) \leq 1 \ \ \forall j \in N \\ x_j(S) \in \{0, 1\} \ \ \forall j \in N, S \in B_j \end{array} \right\}.$$

The first set of constraints ensures that no item is over-allocated and the second set of constraints ensures that each bidder has at most one winning bid. This formulation is the de facto method for computing efficient allocations in practice, and is the formulation we use when solving winner determination problems in experiments.

Winner determination can also be formulated as the following linear program (LP) due to Bikhchandani and Ostroy [2002] (see also de Vries et al. [2007]). The linear program explicitly enumerates all possible feasible allocations of the items and has the property that its optimal solution is integral. It involves variables $x_j(S)$ indicating whether bundle S is allocated to bidder j and variables $\delta(\boldsymbol{S})$ indicating whether feasible allocation $\boldsymbol{S} \in \Gamma$ is chosen:

$$\max \left\{ \sum_{j \in N} \sum_{S \in B_j} v_j(S) x_j(S) : \begin{array}{l} x_j(S) \leq \displaystyle\sum_{\boldsymbol{S} \in \Gamma : S_j = S} \delta(\boldsymbol{S}) \;\; \forall j \in N, S \in B_j \;\; \boxed{p_j(S)} \\ \displaystyle\sum_{S \in B_j} x_j(S) \leq 1 \;\; \forall j \in N \;\; \boxed{\pi_j} \\ \displaystyle\sum_{\boldsymbol{S} \in \Gamma} \delta(\boldsymbol{S}) \leq 1 \;\; \boxed{\pi_s} \\ x_j(S) \geq 0 \;\; \forall j \in N, S \in B_j \\ \delta(\boldsymbol{S}) \geq 0 \;\; \forall \boldsymbol{S} \in \Gamma \end{array} \right\}.$$

The first set of constraints ensures that winning bids are consistent with the bundles in the efficient allocation, the second set of constraints ensures that each bidder has at most one winning bid, and the third constraint ensures that only one efficient allocation is chosen. The corresponding dual variables are boxed following the respective primal constraint. The dual LP is given by (with the corresponding primal variables boxed)

$$\min \left\{ \sum_{j \in N} \pi_j + \pi_s : \begin{array}{l} \pi_j \geq v_j(S) - p_j(S) \;\; \forall j \in N, S \in B_j \;\; \boxed{x_j(S)} \\ \pi_s \geq \displaystyle\sum_{j \in N} p_j(S_j) \;\; \forall \boldsymbol{S} \in \Gamma \;\; \boxed{\delta(\boldsymbol{S})} \end{array} \right\}$$

and has a constraint for every possible feasible allocation $\boldsymbol{S} \in \Gamma$. By strong duality, its optimal objective value is also $w(\boldsymbol{v})$. The dual variables $p_j(S)$ have the natural interpretation of non-additive non-anonymous bundle prices that support the efficient allocation computed by the primal, with π_j representing bidder j's utility and π_s the seller's revenue [Bikhchandani and Ostroy, 2002, de Vries et al., 2007], though in general these do not coincide with VCG prices (Bikhchandani et al. [2001] provide an in depth exploration of the connections between LP duality and VCG prices).

6.4.2 Formulations and Constraint Generation for WT Computation

Let $\widetilde{B}_i$ denote the set of bundles S_i such that $v_i(S_i)$ is constrained by $\Theta_i(\boldsymbol{v}_{-i})$ (so if $\Theta_i(\boldsymbol{v}_{-i})$ is explicitly represented as a list of constraints on v_i, $\widetilde{B}_i$ is the set of bundles S_i such that $v_i(S_i)$ appears in one of those constraints).

We consider two mathematical programming formulations of weakest-competitor VCG price computation, which is the min-max optimization problem

$$\min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} \max_{\boldsymbol{S} \in \Gamma(\widetilde{B}_i,, \boldsymbol{B}_{-i})} \widetilde{v}_i(S_i) + \sum_{j \in N \setminus i} v_j(S_j).$$

The first is due to Balcan et al. [2023] and the second is based on the dual of the winner determination LP (6.4.1) of Bikhchandani and Ostroy [2002]. Both formulations enumerate the set of feasible allocations Γ in their constraint set so they are too large to be written down explicitly. Instead, we develop constraint generation routines that dynamically add constraints as needed.

Formulation based on Balcan-Prasad-Sandholm LP

The mathematical program for computing the pivot term in bidder i's weakest-competitor VCG price $p_i^{\mathtt{WT}}$ due to Balcan et al. [2023] is:

$$\min \left\{ \gamma : \begin{array}{l} \widetilde{v}_i(S_i) + \sum_{j \neq i} v_j(S_j) \leq \gamma \ \ \forall \boldsymbol{S} \in \Gamma(\widetilde{B}_i, \boldsymbol{B}_{-i}), \\ \widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i}) \end{array} \right\}. \tag{BPS}$$

It turns the min-max problem into a pure minimization problem by enumerating the inner maximization terms and adding an auxiliary scalar variable γ to upper bound those terms. In constraint generation, we initialize the BPS program with some restricted set of constraints corresponding to feasible allocations $\Gamma_0 \subseteq \Gamma(\widetilde{B}_i, \boldsymbol{B}_{-i})$ and solve to get a candidate solution $\widehat{\gamma}, \widehat{v}_i$. Next, we find the most violated constraint not currently in Γ_0 by computing $w(\widehat{v}_i, \boldsymbol{v}_{-i})$ and comparing to $\widehat{\gamma}$. If $\widehat{\gamma} - w(\widehat{v}_i, \boldsymbol{v}_{-i}) < 0$ we have found a (most) violated constraint, and we add the constraint corresponding to the violating allocation $(\widehat{S}_1, \ldots, \widehat{S}_n)$ that solves $w(\widehat{v}, \boldsymbol{v}_{-i})$ to the restricted pricing LP (that is, $\Gamma_0 \leftarrow \Gamma_0 \cup \{\widehat{\boldsymbol{S}}\}$). The BPS program with the additional constraint is resolved and the process iterates. Otherwise if $\widehat{\gamma} - w(\widehat{v}_i, \boldsymbol{v}_{-i}) \geq 0$, all constraints of the unrestricted BPS program are satisfied and so $\widehat{\gamma}, \widehat{v}_i$ is an optimal solution to the BPS program and constraint generation terminates.

Formulation based on Bikhchandani-Ostroy LP

The mathematical program for computing the pivot term in bidder i's weakest-competitor VCG price $p_i^{\mathtt{WT}}$ based on the dual LP of the Bikhchandani and Ostroy [2002] formulation (6.4.1) is:

$$\min \left\{ \sum_{j \in N} \pi_j + \pi_s : \begin{array}{l} \pi_i \geq \widetilde{v}_i(S) - p_i(S) \ \ \forall S \in \widetilde{B}_i \\ \pi_j \geq v_j(S) - p_j(S) \ \ \forall j \in N \setminus i, S \in B_j \\ \pi_s \geq \sum_{j \in N} p_j(S_j) \ \ \forall \boldsymbol{S} \in \Gamma(\widetilde{B}_i, \boldsymbol{B}_{-i}) \\ \widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i}) \end{array} \right\}. \tag{BO}$$

It has $|\widetilde{B}_i| + \sum_{j \neq i} |B_j| + n + 1$ variables while the BPS formulation has $|\widetilde{B}_i| + 1$ variables. In constraint generation, we solve a restricted BO program over an initial set of feasible allocations Γ_0 (replacing the third set of constraints) and get a candidate solution $\widehat{v}_i$, $(\widehat{\pi}_j)$, $\widehat{\pi}_s$, $(\widehat{p}_j(S))$. To find the most violated constraint we solve winner determination where bidders' bids are given by the values of the supporting prices $\widehat{p}_j(S)$ and compare the value to the seller's revenue $\widehat{\pi}_s$. If $\widehat{\pi}_s - w(\widehat{\boldsymbol{p}}) < 0$ the constraint corresponding to the feasible allocation $\widehat{\boldsymbol{S}}$ that solves $w(\widehat{\boldsymbol{p}})$ is (most) violated. So, we add that constraint to the restricted BO program and iterate. Else if

$\widehat{\pi}_s - w(\widehat{\boldsymbol{p}}) \geq 0$ all BO constraints are satisfied, and our candidate solution is optimal so constraint generation terminates.

Remark An advantage of the BPS formulation is that it can generally be applied to any multi-dimensional mechanism design problem, and the constraint generation works as long as one can formulate and solve the separation problem (winner determination) in a tractable way. The BO formulation is specific to combinatorial auctions.

So far we have given an abstract presentation of the BPS and BO mathematical programs that make no reference to the structure of type spaces. The constraint generation routines we described work generally as long as one can solve the associated optimization problems, but the best approach to WT computation might differ based on the type space representation. For example, if $\Theta_i(\boldsymbol{v}_{-i})$ is a finite list of types, one would simply solve the winner determination problem for each type in the list and choose the smallest value. In our experiments type spaces are polyhedra described by linear constraints, so the BPS and BO formulations are linear programs for which constraint generation as we have described is a de facto approach. Extending these techniques to other forms of type spaces (convex sets, unions of polyhedra—in order to represent disjunctive statements, *etc.*) is a compelling direction for future work.

6.4.3 Comparison of the BPS and BO Formulations

We implemented constraint generation on both the BPS and BO formulations where the type spaces $\Theta_i(\boldsymbol{v}_{-i})$ were generated independently at random for each bidder. Each type space $\Theta_i(\boldsymbol{v}_{-i})$ was determined by 8 randomly generated linear constraints (so both the BPS and BO formulations were linear programs) that were consistent with bidder i's actual bids (we defer the specific details of how we generated CA instances and respective bidder type spaces to Section 6.5). For both formulations, we initialized the starting set of allocations Γ_0 with only the efficient allocation $\boldsymbol{S}^*$.

Run-times and total number of constraint generation iterations to compute $\boldsymbol{p}^{\mathtt{WT}}$ are reported in Table 6.1.

	BPS				BO			
Goods/Bids	Run-time (GM)	Run-time (GSD)	CG iters (GM)	CG iters (GSD)	Run-time (GM)	Run-time (GSD)	CG iters (GM)	CG iters (GSD)
64/250	4.0	2.3	25.6	2.4	7.3	2.3	58.2	2.5
64/500	9.4	2.9	45.8	3.1	20.8	2.8	102.5	2.6
128/250	8.5	3.2	42.2	2.5	17.6	2.9	110.3	2.4
128/500	46.9	6.3	59.8	2.9	110.6	5.8	164.2	2.4

Table 6.1: Geometric mean (GM) and standard deviation (GSD) of run-times (in seconds) and number of constraint generation (CG) iterations for the BPS and BO formulations, varying the number of goods and bids, averaged across 100 instances for each good/bid setting.

Constraint generation on the BPS formulation was significantly faster and required far fewer iterations than the BO formulation. On an instance-by-instance basis, the BPS formulation was faster and cheaper than the BO formulation on 100% of CA instances. In all experiments reported in the following section (Section 6.5), we therefore only ran constraint generation on the BPS formulation for all WT computations.

6.5 Experiments

We ran experiments to evaluate the revenue, incentive, fairness, and computational properties of our new core-selecting CAs. We describe the main components of the experimental setup below.

New and old core-selecting CAs

For a given CA instance we compare five different core-selecting payment rules, defined in the below bulleted list (the three new CAs we introduce in this work are bolded):

- Vanilla VCG nearest [Day and Cramton, 2012]: the point $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{VCG}})$ that minimizes $\|\boldsymbol{p} - \boldsymbol{p}^{\texttt{VCG}}\|_2^2$.
- Vanilla zero nearest [Erdil and Klemperer, 2010]: the point $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{VCG}})$ that minimizes $\|\boldsymbol{p}\|_2^2$.
- **WT nearest**: the point $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ that minimizes $\|\boldsymbol{p} - \boldsymbol{p}^{\texttt{WT}}\|_2^2$.
- **Zero nearest**: the point $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ that minimizes $\|\boldsymbol{p}\|_2^2$.
- **VCG nearest**: the point $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ that minimizes $\|\boldsymbol{p} - \boldsymbol{p}^{\texttt{VCG}}\|_2^2$.

The WT-nearest rule is the most direct generalization of the vanilla VCG-nearest rule proposed by Day and Cramton [2012] and the zero-nearest rule is the most direct generalization of the vanilla zero-nearest rule proposed by Erdil and Klemperer [2010].

Quadratic programming and core-constraint generation

Each of the five price vectors is computed via the quadratic programming and core-constraint generation technique developed by Day and Cramton [2012], which we describe here at a high level. Details can be found in Day and Cramton [2012]. Given an input *reference point* $\boldsymbol{p}^{\texttt{ref}}$, the goal is to find the point $\boldsymbol{p} \in \mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ that minimizes $\|\boldsymbol{p} - \boldsymbol{p}^{\texttt{ref}}\|_2^2$. Let QP($r$) denote the quadratic program

$$\min\left\{\left\|\boldsymbol{p} - \boldsymbol{p}^{\texttt{ref}}\right\|_2^2 : \boldsymbol{p} \in \mathsf{Core}, \boldsymbol{p} \geq \boldsymbol{p}^{\texttt{WT}}, \|\boldsymbol{p}\|_1 = r\right\}$$

and let LP denote the linear program

$$\min\left\{\|\boldsymbol{p}\|_1 : \boldsymbol{p} \in \mathsf{Core}, \boldsymbol{p} \geq \boldsymbol{p}^{\texttt{WT}}\right\}.$$

Core constraints make both formulations too large to represent explicitly, and hence they are solved with constraint generation. First, solve restricted LP with some initial set (possibly empty) of core constraints; let $\widehat{r}$ be the optimal objective. Then, solve restricted QP($\widehat{r}$) with the same initial core constraints, and let $\widehat{\boldsymbol{p}}$ be the optimal solution. To find the most violated core constraint, solve an auxiliary winner determination where all bids by winner i are reduced by their opportunity cost $v_i(S_i^*) - \widehat{p}_i$. If the optimal winner determination value/efficient welfare is more than the current QP revenue $\|\widehat{\boldsymbol{p}}\|_1$, the core constraint corresponding to the set of winners in the auxiliary winner determination is violated. Add that constraint to the restricted LP and QP, re-solve LP to get an updated $\widehat{r}$, solve QP($\widehat{r}$), and iterate. (The revenue-minimization LP is needed to ensure that we find the closest point to $\boldsymbol{p}^{\texttt{ref}}$ on $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$. Without that we might find a closer point, but it will be outside the minimum-revenue core and therefore not minimize the sum of incentives to deviate.)

Type space generation

For each CA instance, we generated synthetic bidder type spaces $\Theta_i(\boldsymbol{v}_{-i})$ determined by linear constraints (so the formulations for WT price computation from Section 6.4 are LPs). We generated $\Theta_i(\boldsymbol{v}_{-i})$ independently for each bidder by generating K random linear constraints according to parameter β as follows. Each constraint is of the form

$$\sum_{S_i \in B_i} X(S_i)c(S_i)\widetilde{v}_i(S_i) \geq \alpha \cdot \sum_{S_i \in B_i} X(S_i)c(S_i)v_i(S_i)$$

where $\widetilde{v}_i(S_i)$ are the variables representing bidder i's bids, each $X(S_i)$ is an independent Bernoulli $0/1$ random variable with success probability β, each $c(S_i)$ is drawn uniformly and independently from a decay distribution where $c(S_i)$ is initially equal to 1 and is repeatedly incremented with success probability 0.2 until failure, and α is drawn uniformly at random from $[1/2, 1]$. So, each such constraint is guaranteed to be satisfied by the actual bids, and α determines how close to tight the constraint is. Each of the K constraints per-bidder is generated this way independently.

6.5.1 Results

We used the Combinatorial Auction Test Suite (CATS) [Leyton-Brown et al., 2000] version 2.1 to generate CA instances. Like Day and Raghavan [2007] and Day and Cramton [2012], we generated each instance from a randomly chosen distribution from the seven available distributions meant to model real-world CA applications. Code for our experiments was written in C++ and we used Gurobi version 12.0.1, limited to 8 threads, to solve all linear programs, integer programs, and quadratic programs. All computations were done on a 64-core machine with 16GB of RAM allocated for each CA instance.

Run-time cost of WT computation

Table 6.2 records the effects of varying $\beta \in \{0.2, 0.5, 0.8\}$ (which controls the sparsity of type space constraints) on the run-time and number of CG iterations to compute $\boldsymbol{p}^{\texttt{WT}}$. We fixed the number of constraints $K = 8$, and for each β and each setting of goods in $\{64, 128\}$ and bids in $\{250, 500\}$ generated 100 instances (for a total of 400 instances). For these instances, the (geometric) mean run-time and worst-case run-time for $\boldsymbol{p}^{\texttt{VCG}}$ were 2.7 seconds and 608.5 seconds, respectively.

The worst run-time for WT computation was thus roughly $3.1\times$ the worst run-time for VCG computation. In general, increasing β, which increases the density of the type space constraints,

β	Run-time (GM)	CG iters (GM)	Run-time (Max)	CG iters (Max)
0.2	9.9	32.0	1515.1	424
0.5	13.2	56.4	1610.0	536
0.8	15.3	75.3	1896.0	567

Table 6.2: Run-times and constraint generation iterations for the BPS formulation as β varies, with number of goods varying in $\{64, 128\}$ and number of bids varying in $\{250, 500\}$, averaged over 100 instances for each β and each setting of goods/bids.

increases the cost of WT computation. The additional run-time cost for finding a $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ via core-constraint generation was in fact less expensive than the run-time of core-constraint generation to find vanilla MRC points. The geometric mean runtime of the vanilla VCG nearest rule of Day and Cramton [2012] on the above instances was 1.7 seconds, with a worst case run-time of 523.9 seconds. The geometric mean of our WT nearest rule on the same instances was 1.0 seconds, with a worst case run-time of 475.0 seconds. So, the main run-time cost of our new core-selecting CAs is in computing $\boldsymbol{p}^{\texttt{WT}}$.

Varying the number of type space constraints K did not have a significant impact on run-time nor number of constraint generation iterations for WT computation. Over all CA instances with number of goods in $\{64, 128\}$ number of bids in $\{250, 500, 1000\}$, and $\beta = 0.3$, the geometric mean of run-times over all K was 19.7 seconds and the geometric mean of constraint generation iterations was 42.7. The worst-case VCG run-time was 19545.2 seconds and the worst-case WT run-time was 47718.0 seconds ($2.44\times$ larger than VCG run-time). The significantly larger run-time relative to the experiment varying β was due to the inclusion of the the CATS instances with 1000 bids.

Incentive and revenue effects

We now discuss the impact of type space information on the sum of bidders' incentives to deviate from truthful bidding in a $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$-selecting CA. That is, we record the quantity $\sum_{i\in W} \delta_i^{\boldsymbol{p}}$ where $\boldsymbol{p}$ is any one of our new core-selecting CAs. By Theorem 6.3.2 this is equal to $\sum_{i\in W} p_i - p_i^{\texttt{WT}}$, that is, the difference in revenue between the $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$-selecting rule and WT. We track this quantity as the number of type space constraints K per bidder varies in $\{1, 2, 4, 8, 16\}$, and compare it to the sum of bidders' incentives to deviate in the vanilla unrestricted setting, which by Day and Milgrom [2010] is equal to the difference in revenue between a $\mathsf{MRC}(\boldsymbol{p}^{\texttt{VCG}})$-selecting rule and VCG. Each revenue difference recorded on the y-axis of Figure 6.2 is averaged over 100 CA instances each for goods in $\{64, 128\}$ and bids in $\{240, 500, 1000\}$, for a total of 600 CA instances and a total of $600 \times 5 = 3000$ type space instances/WT computations. We fixed the constraint sparsity parameter $\beta = 0.3$. Figure 6.2 shows a clear trend that more information about the bidders (in the form of more type space constraints) yields better core incentives—and vastly better incentives than a vanilla MRC-selecting rule.

On the revenue front, Figure 6.3 shows the impact of more informative type spaces on the revenues generated by our new core-selecting CAs (the experimental setup is the same as in the previous paragraph). The $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$-selecting rules are the clear winner, nearly closing half the

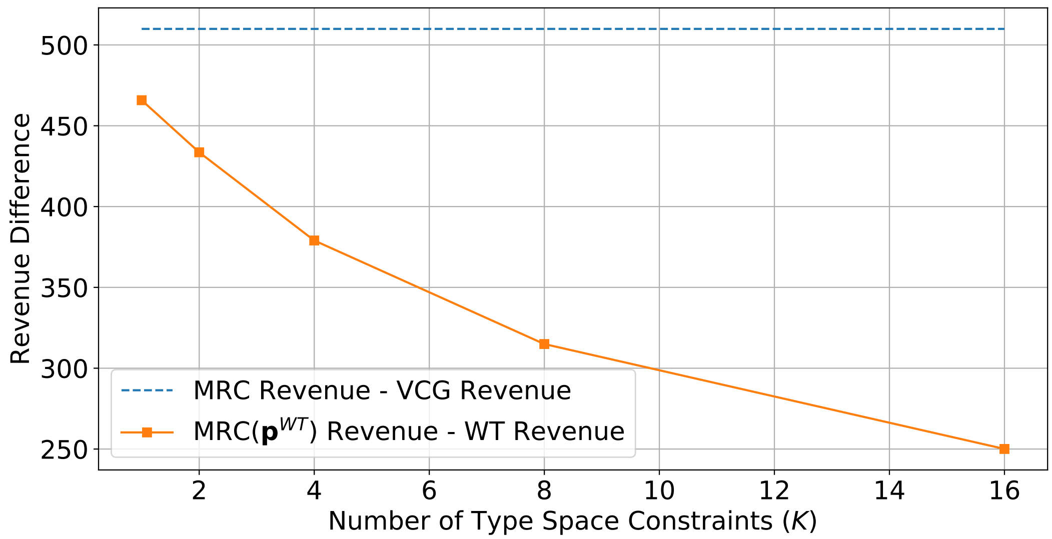


Figure 6.2: Incentive effects as type spaces convey more information (by varying the number of constraints $K \in \{1, 2, 4, 8, 16\}$, with number of goods varying in $\{64, 128\}$ and number of bids varying in $\{250, 500, 1000\}$, averaged over 100 instances for each K and each setting of goods/bids.

gap between MRC revenue and the efficient social welfare when type spaces are determined by $K = 16$ constraints. While the MRC($\boldsymbol{p}^{\mathtt{WT}}$) revenue is not significantly larger than the MRC revenue for $K \leq 8$, WT's revenue *is* much larger than VCG's, leading to much better incentives for the MRC($\boldsymbol{p}^{\mathtt{WT}}$) rule than the MRC rule in that regime despite similar revenues. So, a MRC($\boldsymbol{p}^{\mathtt{WT}}$)-selecting rule with revenue not much larger than a vanilla MRC($\boldsymbol{p}^{\mathtt{VCG}}$)-selecting rule can still provide significantly better incentives for bidders if $\|\boldsymbol{p}^{\mathtt{WT}}\|_1$ is much larger than $\|\boldsymbol{p}^{\mathtt{VCG}}\|_1$.

How often is WT in the core?

We now report on the frequency with which $\boldsymbol{p}^{\mathtt{WT}} \in \mathsf{Core}$. For CA instances with this property, all MRC($\boldsymbol{p}^{\mathtt{WT}}$)-selecting rules return $\boldsymbol{p}^{\mathtt{WT}}$ unmodified. The following table records the frequency as the number of type space constraints K varies (the setup is the same as those in Figures 6.2 and 6.3).

Number of type space constraints	1	2	4	8	16
Fraction of CAs where $\boldsymbol{p}^{\mathtt{VCG}} \notin \mathsf{Core}$ but $\boldsymbol{p}^{\mathtt{WT}} \in \mathsf{Core}$	2.00%	3.02%	3.36%	3.69%	4.18%

Table 6.3: Frequency with which WT is in the core but VCG is not, with number of goods varying in $\{64, 128\}$ and number of bids varying in $\{250, 500, 1000\}$; 100 instances for each K and each setting of goods/bids.

As the number of type space constraints increases, the likelihood that WT is in the core increases as well (which is in accordance with the trend in Figure 6.3 that more information through

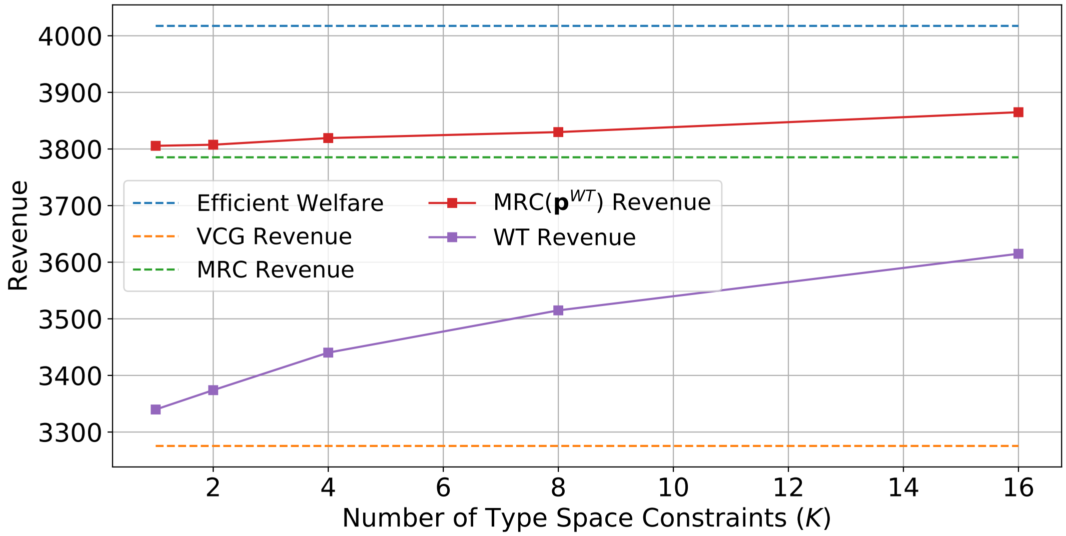


Figure 6.3: Revenue effects as type spaces convey more information (by varying the number of constraints $K \in \{1, 2, 4, 8, 16\}$, with number of goods varying in $\{64, 128\}$ and number of bids varying in $\{250, 500, 1000\}$, averaged over 100 instances for each K and each setting of goods/bids.

more type space constraints implies greater WT revenue). Additionally, 1.17% of all instances had the property that both VCG and WT were in the core, both generating nonzero revenue (this has no dependence on the type space since if VCG is in the core and generates nonzero revenue, so does WT). A fascinating phenomenon we observed was that 7.5% of instances had the property that all vanilla MRC-selecting rules (like vanilla VCG-nearest of Day and Cramton [2012] and vanilla zero-nearest of Erdil and Klemperer [2010]) generated zero revenue. In other words, VCG generates zero revenue *yet is in the core*. This is an *even worse* situation than the zero revenue cases described by Ausubel et al. [2017] and Ausubel and Baranov [2023] that a vanilla core-selecting rule is unable to fix. The WT auction is therefore indispensable to generate any revenue in these cases. [1] To our knowledge, no prior work discusses this phenomenon.

Who shoulders the core burden?

In Day and Cramton [2012], the impact of core pricing on the highest and lowest bidder is visualized. They show that on CATS instances with few bids (100 or less), their vanilla VCG nearest rule provides a more equitable apportionment of the core burden than the vanilla zero nearest rule of Erdil and Klemperer [2010]. That trend is less pronounced for the numbers of bids that we consider (250, 500, and 1000), and hence we present a slightly different visualization of

[1]In the context of Theorem 6.2.1, such situations arise when the core polytope is the box with diagonally opposite points given by the origin (which is equal to VCG) and the winning bid vector. The WT point is strictly in the interior of this box, and the infinitely many points on line segment connecting the origin to the WT point are attainable with an IC auction.

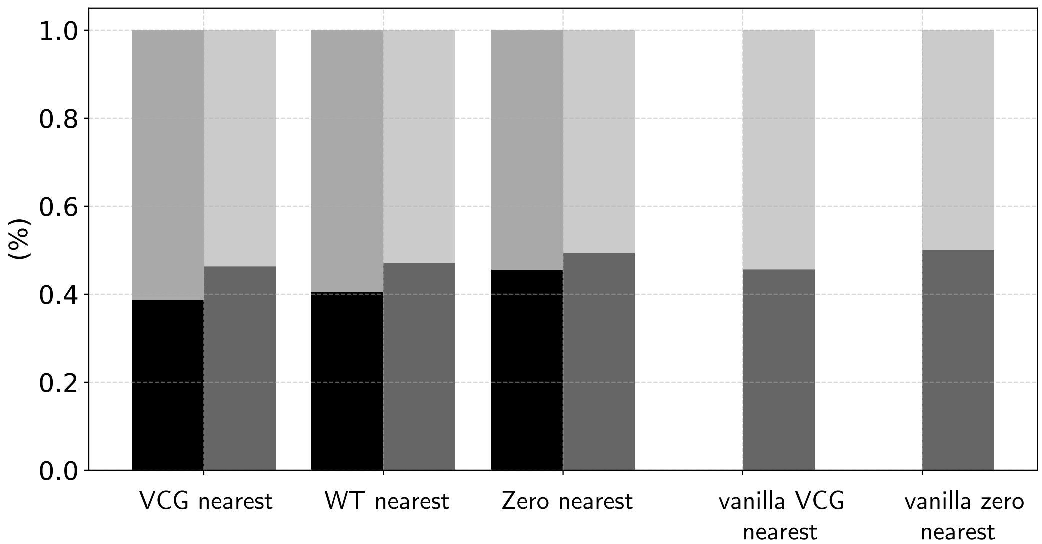


Figure 6.4: Core burdens shouldered by the lower and upper halves of bidders (measured by winning bid value). For the three $\mathsf{MRC}(\boldsymbol{p}^{\mathtt{WT}})$-selecting rules, the left bar displays the core burden split relative to WT, and the right bar displays the core burden split relative to VCG. For the two vanilla MRC-selecting rules, the bar displays the core burden split relative to VCG.

the splitting of the core burden.

For each CA instance, and each core-pricing rule $\boldsymbol{p}$, the core burden relative to WT (resp. VCG) of bidder i is the quantity $\frac{p_i - p_i^{\mathtt{WT}}}{\sum_{i \in W} p_i - p_i^{\mathtt{WT}}}$ (resp. $\frac{p_i - p_i^{\mathtt{VCG}}}{\sum_{i \in W} p_i - p_i^{\mathtt{VCG}}}$). We sorted the bidders in ascending order of winning bid $v_i(S_i^*)$, and summed up the total core burdens for the lower and higher halves of bidders. Figure 6.4 displays the splitting of core burdens between the lower and higher halves, averaged across all instances with $K = 8$. For VCG nearest, WT nearest, and zero nearest, the left bar displays core burdens relative to WT, with the solid black bottom representing the lower half of bidders and the gray top representing the upper half. The right bar displays core burdens relative to VCG, with the darker gray bottom representing the lower half of bidders and the lighter gray top representing the upper half. Only the core burden relative to VCG is displayed for the vanilla VCG nearest [Day and Cramton, 2012] and vanilla zero nearest [Erdil and Klemperer, 2010] since it would not make sense to compute core burdens relative to WT for these rules. Overall, there was not a significant difference between WT-nearest, zero-nearest, and VCG-nearest (and this was also the case in Day and Cramton [2012] in their comparison of vanilla VCG-nearest and vanilla zero-nearest on CATS instances with more than 250 bids). VCG-nearest placed the least core burden and zero-nearest placed the greatest core burden on the lower half of bidders, and all three rules are similar to the vanilla MRC rules in terms of core burdens relative to VCG. This fact provides further validation for our $\mathsf{MRC}(\boldsymbol{p}^{\mathtt{WT}})$-selecting rules as they do not unfairly skew the apportionment of the core burden.

6.5.2 Discussion of Alternate Rules

We conclude this section with a brief discussion of alternate core-selecting CAs that do not conform to the exact template that has been prescribed here and by prior work.

The previous discussion of equitable sharing of the core burden begets the question of whether there exist core-selecting rules that explicitly enforce how the core burden should be split. For example, is there a $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$-selecting CA $\boldsymbol{p}$ that enforces that each bidder pays a core burden in exact proportion to their winning bid, that is,

$$\frac{p_i - p_i^{\texttt{WT}}}{\sum_{i \in W} p_i - p_i^{\texttt{WT}}} \geq \alpha \cdot \frac{v_i(S_i^*)}{\sum_{i \in W} v_i(S_i^*)}$$

for some α? The answer is no due to the asymmetric information that can be conveyed about bidders by type spaces. For example, if $\Theta_1(\boldsymbol{v}_{-1}) = \{v_1\}$, so $p_i^{\texttt{WT}} = v_i(S_i^*)$, IR constraints force $p_i = p_i^{\texttt{WT}}$ for any core-selecting $\boldsymbol{p}$. So in such situations it could very well be the case that a low bidder is forced to shoulder a large majority of the core burden. A general rule of thumb here appears to be that the bidders with type spaces that convey the least information about them must pay most of the core burden. A formal investigation of this idea is an interesting direction for future research.

Schemes that do not select points on $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$ are also possible. For example, one could minimize the maximum payment over the subset of the core that lies above $\boldsymbol{p}^{\texttt{WT}}$. This would minimize the worst incentive of any bidder to deviate from truthful bidding (by the same argument as in Theorem 6.3.3) rather than the sum of bidders' incentives to deviate, and would still yield a core point that is incentive optimal in a Pareto sense. Parkes et al. [2001] proposed such rules in the context of budget balance in exchanges, but those rules have received limited empirical evaluation in the CA setting to date.

Finally, rules that minimize a weighted sum of squares (as studied by Bünz et al. [2022]) might be of particular relevance so that bidders i such that $p_i^{\texttt{WT}}$ is much larger than $p_i^{\texttt{VCG}}$ are made to pay less of the price difference in moving from $p_i^{\texttt{WT}}$ to $\mathsf{MRC}(\boldsymbol{p}^{\texttt{WT}})$.

6.6 Conclusions and Future Research

We presented a new family of core-selecting CAs that take advantage of bidder information known to the auction designer through bidders' type spaces. Our design built upon the WT auction, which boosts revenues beyond VCG by considering for each bidder the weakest type consistent with the auction designer's knowledge. We showed that sufficiently informative type spaces can overcome the well-known impossibility of core-selecting CAs, and gave a revised and generalized impossibility result that depends on whether or not the WT auction is in the core. We then showed that our new family of core-selecting CAs, defined by minimizing revenue on the section of the core above WT prices, minimizes the sum of bidders' incentives to deviate from truthful bidding. This result generalizes those of Day and Raghavan [2007] and Day and Milgrom [2010] which rely on unrestricted bidder type spaces. On the computational front, we developed new constraint generation techniques for computing WT prices. We compared two formulations, one due to Balcan et al. [2023] and a new one based on Bikhchandani and Ostroy

[2002] that is a contribution of this chapter. Finally, we evaluated our new core-selecting CAs on CATS instances, with synthetic generators for type space constraints. The revenue and incentive benefits of our new CAs, along with their manageable computational overhead, make them a useful addition to the auction design toolkit.

We conclude by discussing avenues for future research. Perhaps the most pressing direction is the development of realistic type space generators by incorporating the specific details of the application domain. Our new CAs display promise on our synthetically-generated type spaces, but to understand their viability in real-world auctions one must develop detailed models of auctioneer knowledge. Generalizing our techniques to type spaces that are not cleanly described by linear constraints is a prerequisite here.

A more thorough investigation is needed for the design of $\mathsf{MRC}(\boldsymbol{p}^{\mathtt{WT}})$-selecting rules. We introduced three specific ones in this chapter (WT nearest, zero nearest, and VCG nearest) that are natural generalizations of vanilla MRC-selecting rules, but as discussed in Section 6.5.2 there might be other more economically meaningful rules. A computational study extending Bünz et al. [2022] to $\mathsf{MRC}(\boldsymbol{p}^{\mathtt{WT}})$-selecting rules is relevant here as well. A promising direction along this vein is to use machine learning to design the reference point, weights, and amplifications of the parameterized rules in Bünz et al. [2022]. Explicit equilibrium analysis in the style of Goeree and Lien [2016] and Ausubel and Baranov [2020] is important as well.

Our formulations of WT computation were specific to the XOR bidding language. Extensions and modifications of our techniques are needed for other domains and other bidding languages such as those proposed for spectrum auctions [Bichler et al., 2023, Weiss et al., 2017], sourcing auctions [Sandholm, 2002b, 2013], and more general domain-independent use [Sandholm, 2002a, Nisan, 2000, Boutilier and Hoos, 2001, Boutilier, 2002]. The interaction between the bidding language of a CA and the language used to express type space knowledge is an unexplored area here as well.

Finally, an important direction within the research strand of *mechanism design with predictions* [Balcan et al., 2023, Balkanski et al., 2024a] is to relax the assumption that $\boldsymbol{v} \in \boldsymbol{\Theta}$, that is, that type spaces convey *correct* information about bidders. How can core-selecting CAs with strong incentive properties be designed using the techniques developed in this chapter when type spaces can have small errors? The techniques developed in [Balcan et al., 2023] in the general setting of revenue-maximizing multidimensional mechanism design will likely be useful here, and can also help shed light on better core selection in mechanism design settings beyond combinatorial auctions.

Chapter 7

Revenue-Optimal Efficient Mechanism Design with General Type Spaces

Efficient mechanism design—moving up in generality from the design of efficient CAs that was the topic of the previous chapter—is the science of implementing outcomes that maximize economic value among strategic self-interested agents. It is the cornerstone of prominent real-world market design applications including combinatorial auctions and Internet display advertisement auctions [Edelman et al., 2007, Varian, 2007]. Additional modern applications include proposed redesigns of financial exchanges [Budish et al., 2023], better incentive-aware recommender ecosystems [Prasad et al., 2023, Boutilier et al., 2024], and auctions for large language models [Dütting et al., 2024, Hajiaghayi et al., 2024].

Our focus in the present chapter is the design of *revenue-maximizing pricing rules* for efficient (general multidimensional) mechanism design. As in the previous chapters, our approach is to leverage information about the agents available to the mechanism designer through the agents' *type spaces*, which encode constraints on agents' private values (or *types*) that are known to the mechanism designer to hold before the private types are elicited. When type spaces are *connected*, Krishna and Perry [1998] and Balcan et al. [2023] show that the *weakest-type (WT) mechanism* is revenue optimal subject to efficiency, incentive compatibility (agents are best off reporting their true values to the mechanism designer), and individual rationality (no agent is charged more than their reported value for the chosen outcome).

However, connected type spaces are unable to express many natural constraints about agent types. For example, an auction designer might know that a bidder will bid for development rights in exactly one of two geographic regions, but not know which one. This kind of exclusivity constraint/disjunction can only be represented by a disconnected type space. Similarly, the auction designer might know that a bidder, if she submits a bid for a particular item, will bid at least $5 million for that item. The type space here is also inherently disconnected since it allows for either no ($0) bid or a bid exceeding $5 million, but nothing in between. Another cause of disconnectedness is discrete type expression. For example, the FCC has experimented with "click-box" bidding to prevent collusive bidding via bid signaling. Here, bids are placed by clicking on the desired spectrum licenses; the bid value is given by fixed increments which precludes the ability to bid any dollar amount [Cramton and Schwartz, 2000, Bajari and Yeo, 2009].

In this work we show that the WT mechanism generates suboptimal revenue when type spaces are disconnected, and derive the revenue-optimal efficient mechanism for general disconnected type spaces.

Our Contributions

We derive the revenue-optimal efficient mechanism for general agent type spaces. Prior work on efficient multidimensional mechanism design [Krishna and Perry, 1998, Balcan et al., 2023, 2025c] and efficient trade [Myerson and Satterthwaite, 1983, Cramton et al., 1987] has only considered connected type spaces. In Section 7.1 we set up the general backdrop of multidimensional mechanism design, review the current state of knowledge on revenue-optimal efficient mechanism design for connected type spaces, and present examples of natural constraints on agent types that are disconnected and therefore outside the scope of prior work. In Section 7.2 we present a simple example showing that the vanilla WT mechanism is suboptimal for disconnected type spaces. In Section 7.3.1 we derive the optimal efficient mechanism in terms of *allocation-wise Groves mechanisms*, a generalization of the classic Groves [1973] mechanism with a pricing scheme that depends more intimately on the efficient allocation. In Section 7.3.2 we provide an alternate characterization of the optimal efficient mechanism based on the decomposition of the type space into connected components, and *component-wise Groves mechanisms*. In Section 7.3.3 we illustrate our approach with a simple example.

Key to both our characterizations (in terms of allocations and connected components) is an underlying network flow structure to the optimal efficient mechanism that we establish. Either one of these characterizations could be more useful than the other depending on how the mechanism designer's knowledge about the agents' possible types is represented or learned.

Related Work

There is a small body of work on mechanism design with type spaces that do not conform to the usual assumptions—typically convexity, such as in the seminal work of Myerson [1981]—made in economics. Skreta [2006] derives the optimal mechanism for a single-parameter setting where agents' type spaces can be arbitrary measurable subsets of the real line. Monteiro [2009] studies incentive compatibility for general multidimensional type spaces. Lovejoy [2006] analyzes various characterizations of optimal mechanisms when agents have finite type spaces, and Mu'alem and Schapira [2008] provide characterizations of incentive compatibility in this setting. The mechanisms in these aforementioned works are *not* efficient.

Additionally, the design of pricing rules for *efficient* mechanism design has largely been left unexplored by the machine learning for mechanism design literature that we have surveyed in detail in previous chapters. The new characterization results we present here should serve as a timely launchpad for a new research strand along this vein.

7.1 Problem Formulation, Mechanism Design Background, and Examples of Disconnected Type Spaces

As introduced in previous chapters, we are in a multidimensional mechanism design setting with an abstract (finite) set Γ of outcomes or *allocations*. There are n agents, indexed by $i = 1, \ldots, n$, who have private values for each outcome; $v_i : \Gamma \to \mathbb{R}$. The *joint type space* of the agents is denoted by $\boldsymbol{\Theta} \subseteq \times_{i=1}^{n} \mathbb{R}^{\Gamma}$. Given revealed types $\boldsymbol{v}_{-i}$ of agents excluding i, Agent i's induced type space is $\Theta_i(\boldsymbol{v}_{-i}) = \{v_i : (v_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}\}$. Let $\boldsymbol{\Theta}_{-i} = \{\boldsymbol{v}_{-i} : \exists v_i \in \mathbb{R}^{\Gamma} \text{ s.t. } (v_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}\}$. We are interested in mechanisms that implement the efficient allocation $\alpha^{\texttt{eff}}(\boldsymbol{v}) = \text{argmax}_{\alpha \in \Gamma} \sum_{j=1}^{n} v_j(\alpha)$ (we assume a fixed tie-breaking rule so that the argmin is unique) via a pricing rule $\boldsymbol{p} = (p_1, \ldots, p_n)$ that is incentive compatible ($v_i(\alpha^{\texttt{eff}}(v_i, \boldsymbol{v}_{-i})) - p_i(v_i, \boldsymbol{v}_{-i}) \geq v_i(\alpha^{\texttt{eff}}(v_i', \boldsymbol{v}_{-i})) - p_i(v_i', \boldsymbol{v}_{-i})$ for all $(v_i, \boldsymbol{v}_{-i}), (v_i', \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}$) and individually rational ($v_i(\alpha^{\texttt{eff}}(v_i, \boldsymbol{v}_{-i})) - p_i(v_i, \boldsymbol{v}_{-i}) \geq 0$ for all $(v_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}$).

We next recall the definition of a Groves mechanism. Our main mechanisms in this chapter are even more general than Groves mechanisms.

Groves Mechanisms and Weakest Types A *Groves mechanism* is defined by functions $(h_i)_{i=1}^{n}$ where $h_i : \times_{j \neq i} \mathbb{R}^{\Gamma} \to \mathbb{R}$ does not depend on Agent i's revealed type v_i. It implements the efficient allocation $\alpha^{\texttt{eff}}(\boldsymbol{v})$ via payments $p_i(\boldsymbol{v}) = h_i(\boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(\alpha^{\texttt{eff}}(\boldsymbol{v}))$. All Groves mechanisms are efficient and IC. We recall that VCG and WT are given by the pricing rules $p_i^{\texttt{VCG}}(\boldsymbol{v}) = w(0, \boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(\alpha^{\texttt{eff}}(\boldsymbol{v}))$ and $p_i^{\texttt{WT}}(\boldsymbol{v}) = \inf_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(\alpha^{\texttt{eff}}(\boldsymbol{v}))$, respectively (we use the infimum here rather than the minimum as in previous chapters since we will make no assumptions about the existence of a minimizer). VCG and WT are both efficient, IC, and IR.

Our objective is to design efficient mechanisms with better pricing rules that yield better revenues. As detailed in the subsequent section, when type spaces are connected—a stringent assumption on the structure of agent types (but nonetheless the predominant assumption in the mechanism design literature)—one cannot beat WT.

7.1.1 Optimal Efficient Mechanism Design with Connected Type Spaces

We review the current state of knowledge on efficient mechanism design when type spaces are connected (which was briefly glossed over in Chapter 5). The first two results are about the uniqueness of Groves mechanisms as the only efficient and IC mechanisms. The third is about the revenue optimality of the weakest type mechanism.

Theorem 7.1.1 (Revenue Equivalence [Green and Laffont, 1977, Holmström, 1979]). *Suppose $\Theta_i(\boldsymbol{v}_{-i})$ is connected for every $\boldsymbol{v}_{-i}$. Let $\boldsymbol{p}$ be an IC pricing rule and let $\boldsymbol{p}'$ be any other pricing rule. Then, $\boldsymbol{p}'$ is IC if and only if there exist functions $h_i : \boldsymbol{\Theta}_{-i} \to \mathbb{R}$ such that $p_i'(\boldsymbol{v}) = p_i(\boldsymbol{v}) + h_i(\boldsymbol{v}_{-i})$ for all $\boldsymbol{v}$.* [1]

Nisan [2007] calls this result "uniqueness of prices" and provides a self-contained proof.

[1]This result holds more generally for any (not-necessarily-efficient) allocation function $f : \boldsymbol{\Theta} \to \Gamma$.

Theorem 7.1.2 (Uniqueness of Groves Mechanisms [Green and Laffont, 1977, Holmström, 1979]). *Suppose $\Theta_i(\boldsymbol{v}_{-i})$ is connected for every $\boldsymbol{v}_{-i}$ and let $\boldsymbol{p}$ be a pricing rule. Then, $\boldsymbol{p}$ is IC if and only if there exist functions $h_i : \boldsymbol{\Theta}_{-i} \to \mathbb{R}$ such that $p_i(\boldsymbol{v}) = h_i(\boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(\alpha^{eff}(\boldsymbol{v}))$. In other words, the only efficient and IC mechanisms on connected type spaces are Groves mechanisms.*

Remark. Theorems 7.1.1 and 7.1.2 are actually generalizations of the versions derived by Holmström [1979] and presented in Nisan [2007] (the proofs are identical so we omit them). Those versions do not allow Agent i's type space to vary based on the revealed types $\boldsymbol{v}_{-i}$ of the other agents; there is just a fixed type space Θ_i for each agent, and Θ_i needs to be connected. In contrast, we require that $\Theta_i(\boldsymbol{v}_{-i})$ is connected for each $\boldsymbol{v}_{-i}$. We present here a concrete example for which Theorems 7.1.1 and 7.1.2 apply but the original versions from Holmström [1979] do not. Consider an auction of a single item where the auctioneer does not know the quality of the item but knows that all bids will be clustered around either a high value or a low value—say $\boldsymbol{\Theta} = \{\boldsymbol{v} \in \mathbb{R}^n : v_i \in [H - \varepsilon, H + \varepsilon] \ \forall \ i\} \cup \{\boldsymbol{v} \in \mathbb{R}^n : v_i \in [L - \varepsilon, L + \varepsilon] \ \forall \ i\}$. While $\boldsymbol{\Theta}$ is disconnected, $\Theta_i(\boldsymbol{v}_{-i})$ is connected for every $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$ since revealed types $\boldsymbol{v}_{-i}$ determine whether v_i is a low bid or a high bid.

Theorem 7.1.3 (Optimality of Weakest Type [Krishna and Perry, 1998, Balcan et al., 2023]). *Suppose $\Theta_i(\boldsymbol{v}_{-i})$ is connected for every $\boldsymbol{v}_{-i}$. Let $\boldsymbol{p}$ be any IC and IR pricing rule. Then $p_i(\boldsymbol{v}) \leq p_i^{WT}(\boldsymbol{v})$ for all i and all $\boldsymbol{v} \in \boldsymbol{\Theta}$.*

7.1.2 Examples of Disconnected Type Spaces

Here we present three examples of natural constraints on agent types for which $\Theta_i(\boldsymbol{v}_{-i})$ is a disconnected set, illustrating the need for a more general theory than the current one.

- *Exclusivity constraints.* Constraints of the form "Bidder i will place a bid for development rights exceeding $\$10$ million in either San Francisco or New York City, but not both" correspond to disconnected type spaces of the form $\Theta_i = \{v_i : v_i(\text{SF}) \geq 10 \text{ million}, v_i(\text{NYC}) = 0\} \cup \{v_i : v_i(\text{SF}) = 0, v_i(\text{NYC}) \geq 10 \text{ million}\}$.
- *Conditionals.* Constraints of the form "if Bidder i bids on the bundle of items $\{A, B\}$, she will bid at least $\$5$ million" correspond to disconnected type spaces that looks like $\Theta_i = \{v_i : v_i(\{A, B\}) = 0 \text{ OR } v_i(\{A, B\}) \geq 5 \text{ million}\}$.

The above two kinds of constraints are natural in multi-item auctions since the auction designer likely does not know the specific items/packages a bidder will bid on, but has more refined knowledge about the value of any (hypothetical) bid.

- *Discrete types.* Type spaces of the form $\Theta_i = \{v_i : v_i(\alpha) \equiv 0 \pmod{1000}, v_i(A) \geq 5000\}$ convey that values for allocation α are expressed in increments of $\$1000$, starting at $\$5000$.[2]

Discretized type expression might be a natural part of an auction interface. For example, the FCC experimented with "click-box bidding" to prevent collusion wherein bidders can signal to other bidders via the numerical value of their bids [Cramton and Schwartz, 2000, Bajari and Yeo, 2009].

[2] Technically, this example is a restriction on the *reporting space* of the agents. In this section we will treat the reporting space and the type space identically, but treating them separately is an important nuance and a good future research direction.

7.2 Example Illustrating Sub-optimality of Vanilla Weakest Type

Consider the following example of a two-item auction where a bidder has a disconnected type space: there are two items A, B for sale, three bidders submit XOR bids $v_1(A) = 5$, $v_2(B) = 3$, and $v_3(A) = 1$ (under the XOR bidding language [Sandholm, 2002a] each bidder can only win a package they explicitly bid for, which effectively means bidders value packages they did not bid for at zero). The type space for Bidder 1 is

$$\Theta_1 = \{(v_1(A), 0, 0) : v_1(A) \geq 4\} \cup \{(0, v_1(B), 0) : v_1(B) \geq 4\}$$

which says that Bidder 1 wants either A or B, but not both, and will place a bid of at least $\$4$ on her desired item (the third coordinate represents $v_1(AB)$, which is zero). This is an example of an exclusivity constraint discussed previously. $\Theta_1 \subseteq \mathbb{R}^2$ (ignoring the third coordinate which is always zero) is a disjoint union of two rays with one on the $v_1(A)$ axis and one on the $v_1(B)$ axis. All other bidders' type spaces are unrestricted. Bidders 1 and 2 win items A and B, respectively, in the efficient allocation. VCG charges Bidder 1 $p_1^{\texttt{VCG}} = 4 - 3 = 1$. WT charges Bidder 1 $p_1^{\texttt{WT}} = \min\{7, 5\} - 3 = 2$. A better IC and IR payment scheme for Bidder 1 that is still efficient is: "if Bidder 1 wins either item she bid on, she pays $\$4$". That payment scheme extracts a payment of $\$4$ from Bidder 1, showing that WT is suboptimal here.

7.3 Characterization of the Optimal Efficient Mechanism

We now derive the optimal efficient mechanism. We provide two equivalent characterizations. The first (Section 7.3.1) is via a decomposition of the type space based on allocations. The second (Section 7.3.2) is based on the decomposition of the type space into connected components. Key to both our characterizations is an underlying network flow structure. Either one of these characterizations could be more useful than the other depending on how the mechanism designer's knowledge about the agents' possible types is represented or learned. Section 7.3.3 contains an illustrative example.

7.3.1 Allocational Characterization of the Optimal Efficient Mechanism

Let $\Theta_i^\alpha(\boldsymbol{v}_{-i}) = \{v_i \in \Theta_i(\boldsymbol{v}_{-i}) : \alpha^{\texttt{eff}}(v_i, \boldsymbol{v}_{-i}) = \alpha\}$ be the set of types v_i for Agent i leading to efficient allocation α. These sets form a partition of Agent i's type space: $\Theta_i(\boldsymbol{v}_{-i}) = \bigcup_{\alpha \in \Gamma} \Theta_i^\alpha(\boldsymbol{v}_{-i})$.

Allocation-wise Groves Mechanisms We define a large class of pricing rules, not all of which are IC, that contains all IC pricing rules. These generalize the vanilla Groves mechanisms. An *allocation-wise Groves mechanism* is defined by functions $h_i^\alpha : \boldsymbol{\Theta}_{-i} \to \mathbb{R}$ for every allocation $\alpha \in \Gamma$ and every agent i. It charges Agent i

$$p_i(\boldsymbol{v}) = h_i^{\alpha^{\texttt{eff}}(\boldsymbol{v})}(\boldsymbol{v}_{-i}) - \sum_{j \neq i} v_j(\alpha^{\texttt{eff}}(\boldsymbol{v})).$$

So, while the term h_i in a vanilla Groves mechanism cannot have any dependence on Agent i's revealed type, the corresponding term in an allocation-wise Groves mechanism can depend on the efficient allocation induced by Agent i's revealed type. Not all allocation-wise Groves mechanisms are IC, but, as the following lemma shows, it is a rich enough class to cover all IC mechanisms.

Lemma 7.3.1. *If $\boldsymbol{p}$ is IC, it is an allocation-wise Groves mechanism.*

Proof. For each α, partition $\Theta_i^\alpha(\boldsymbol{v}_{-i})$ as a disjoint union of its connected components: $\Theta_i^\alpha(\boldsymbol{v}_{-i}) = \bigcup_{C\in\mathcal{C}^\alpha} C$ ($\mathcal{C}$ need not be finite). When restricted to any connected component $C \in \mathcal{C}^\alpha$, $\boldsymbol{p}$ is a vanilla Groves mechanism (due to Theorem 7.1.2). That is, there exists h_i^C such that for all $v_i \in C$, $p_i(v_i, \boldsymbol{v}_{-i}) = h_i^C(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\alpha)$. It is a standard fact that a pricing rule is IC if and only if it prescribes identical payments for any two types leading to the same allocation. That is, $p_i(v_i, \boldsymbol{v}_{-i}) = p_i(v_i', \boldsymbol{v}_{-i})$ for any $v_i, v_i' \in \Theta_i^\alpha(\boldsymbol{v}_{-i})$ (*e.g.,* Proposition 1.27 of Nisan [2007]). Hence the functions h_i^C for each $C \in \mathcal{C}^\alpha$ are all identical; let h_i^α be this function. Then $\boldsymbol{p}$ is the allocation-wise Groves mechanism given by the h_i^α. □

We now characterize all IC and IR pricing rules that implement the efficient allocation.

Theorem 7.3.2. *A pricing rule $\boldsymbol{p}$ is IC and IR if and only if it is an allocation-wise Groves mechanism given, for each i, by $(h_i^\alpha)_{\alpha\in\Gamma}$ that satisfies*

$$\begin{aligned} &h_i^\alpha(\boldsymbol{v}_{-i}) \le \inf_{\widetilde{v}_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) \quad \forall\, \alpha \in \Gamma \\ &h_i^\alpha(\boldsymbol{v}_{-i}) - h_i^\beta(\boldsymbol{v}_{-i}) \le \inf_{\widetilde{v}_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - \left[\widetilde{v}_i(\beta) + \sum_{j\neq i} v_j(\beta)\right] \quad \forall\, \alpha, \beta \in \Gamma. \end{aligned} \tag{Constr.-Γ}$$

Proof. An allocation-wise Groves mechanism $(h_i^\alpha)_{\alpha\in\Gamma}$ is IR if and only if

$$\begin{aligned} &v_i(\alpha^{\mathtt{eff}}(v_i, \boldsymbol{v}_{-i})) - \left[h_i^{\alpha^{\mathtt{eff}}(v_i,\boldsymbol{v}_{-i})}(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\alpha^{\mathtt{eff}}(v_i, \boldsymbol{v}_{-i}))\right] \ge 0 \quad \forall\, (v_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta} \\ &\iff h_i^{\alpha^{\mathtt{eff}}(v_i,\boldsymbol{v}_{-i})}(\boldsymbol{v}_{-i}) \le w(v_i, \boldsymbol{v}_{-i}) \quad \forall\, (v_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta} \\ &\iff h_i^\alpha(\boldsymbol{v}_{-i}) \le w(v_i, \boldsymbol{v}_{-i}) \quad \forall\, \alpha \in \Gamma, \boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}, v_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i}) \\ &\iff h_i^\alpha(\boldsymbol{v}_{-i}) \le \inf_{\widetilde{v}_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) \quad \forall\, \alpha \in \Gamma, \boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}. \end{aligned}$$

It is IC if and only if for all $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$ and all $v_i, v_i' \in \Theta_i(\boldsymbol{v}_{-i})$, an agent of true type v_i has no incentive to misreport v_i' to the mechanism. Any allocation-wise Groves mechanism already satisfies these constraints for $v_i, v_i' \in \Theta_i^\alpha(\boldsymbol{v}_{-i})$, for every $\alpha \in \Gamma$, since when restricted to any Θ_i^α it is equivalent to the vanilla Groves mechanism given by h_i^α. So, for each $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$, it suffices to enforce IC constraints over all $v_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})$ and all $v_i' \in \Theta_i^\beta(\boldsymbol{v}_{-i})$, over every pair of differing allocations $\alpha, \beta \in \Gamma$. An allocation-wise Groves mechanism $(h_i^\alpha)_{\alpha\in\Gamma}$ is therefore IC if and only if (for all $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$)

$$v_i(\alpha) - \left[h_i^\alpha(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\alpha)\right] \ge v_i(\beta) - \left[h_i^\beta(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\beta)\right] \quad \forall\, \alpha, \beta \in \Gamma, v_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})$$

$$\iff h_i^\alpha(\boldsymbol{v}_{-i}) - h_i^\beta(\boldsymbol{v}_{-i}) \le w(v_i, \boldsymbol{v}_{-i}) - \left[v_i(\beta) + \sum_{j\neq i} v_j(\beta)\right] \quad \forall\, \alpha, \beta \in \Gamma, v_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})$$

$$\iff h_i^\alpha(\boldsymbol{v}_{-i}) - h_i^\beta(\boldsymbol{v}_{-i}) \le \inf_{\widetilde{v}_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - \left[\widetilde{v}_i(\beta) + \sum_{j\neq i} v_j(\beta)\right] \quad \forall\, \alpha, \beta \in \Gamma.$$

The theorem statement now follows from Lemma 7.3.1. □

Theorem 7.3.2 seems to leave open the possibility that there is actually a Parento frontier of undominated revenue-maximial allocation-wise Groves mechanisms. However, that is not the case. It turns out that the revenue-optimal allocation-wise Groves mechanism is unique, which we prove next.

Theorem 7.3.3. *The unique revenue-optimal mechanism subject to efficiency, IC, and IR is the allocation-wise Groves mechanism given by* $(h_i^\alpha)_{\alpha\in\Gamma}$ *that maximizes* $\sum_{\alpha\in\Gamma} h_i^\alpha$ *subject to constraints* (Constr.-Γ)*, for each* i*.*

We will prove Theorem 7.3.3 by interpreting the linear program

$$\max\left\{\sum_{\alpha\in\Gamma} h_i^\alpha \,:\, \text{(Constr.-}\Gamma\text{)}\right\} \qquad \text{(LP-}\Gamma\text{)}$$

from the perspective of network flow theory.

First, observe that the LP-Γ is always feasible. Indeed, vanilla WT is always a feasible solution: let $h_i^\alpha = \inf_{\widetilde{v}_i\in\Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ for all $\alpha \in \Gamma$. Constraints of the first form are clearly satisfied. Constraints of the second form are also clearly satisfied since the left-hand side is zero, and the right-hand side is always non-negative as α maximizes welfare for all $\widetilde{v}_i \in \Theta_i^\alpha(\boldsymbol{v}_{-i})$ (by definition of Θ_i^α). Vanilla VCG with $h_i^\alpha = w(0, \boldsymbol{v}_{-i})$ for all $\alpha \in \Gamma$ is also a feasible solution.

Proof of Theorem 7.3.3. Consider the directed graph $G = (V, E)$ with vertices $V = \{s\} \cup \Gamma$ (s is the source node), edges $E = (\{s\} \cup \Gamma) \times \Gamma$, and edge costs

$$\mathsf{cost}(s, \alpha) = \inf_{\widetilde{v}_i\in\Theta_i^\alpha(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) \quad \forall\, \alpha \in \Gamma$$

$$\mathsf{cost}(\beta, \alpha) = \inf_{\widetilde{v}_i\in\Theta_i^\alpha(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - \left[\widetilde{v}_i(\beta) + \sum_{j\neq i} v_j(\beta)\right] \quad \forall\, \alpha, \beta \in \Gamma.$$

In words, G is the complete directed graph on vertex set Γ with an additional source vertex s and directed edges from s to each vertex of Γ. Edges of the form (s, α) have cost equal to the welfare of the weakest type in $\Theta_i^\alpha(\boldsymbol{v}_{-i})$. Edges (β, α) have cost equal to the minimum welfare difference between allocations α and β over all types in $\Theta_i^\alpha(\boldsymbol{v}_{-i})$.

Linear program LP-Γ precisely solves the *single-source shortest paths* problem on G with source node s.[3] That is, the optimal solution $(h_i^\alpha)_{\alpha\in\Gamma}$ to LP-Γ has the property that, for every $\alpha \in \Gamma$, h_i^α is the cost of the shortest (minimum-cost) $s \to \alpha$ path in G (this is a standard fact from network flow theory; see, for example, Erickson [2017]). A minimum cost $s \to \alpha$ path in G can be equivalently computed as $\max\{h_i^\alpha \,:\, \text{(Constr.-}\Gamma\text{)}\}$, showing that LP-Γ yields the unique optimal efficient mechanism. □

[3]More accurately, it is the dual of the minimum-cost flow LP.

The network flow interpretation here is similar in spirit to those used to understand IC mechanisms in Vohra [2011]. Vohra's focus is on characterizing IC mechanisms in terms of the *existence* of bounded shortest paths (as witnessed by the no-negative-cycle condition) on graphs that are similar to ours. In contrast, our shortest path LPs are always feasible, and we use them to give a new insight into generalized Groves mechanisms. To our knowledge, this is the first time the network interpretation of incentive compatibility has been used to describe revenue-optimal efficient mechanisms.

7.3.2 Connected-Component Characterization of the Optimal Efficient Mechanism

In this section we give an equivalent characterization of the revenue-optimal efficient, IC, and IR mechanism based on decompositions of the type space into its connected components. The characterization does not make explicit reference to the underlying space of allocations Γ.

Given $\boldsymbol{v}_{-i}$, decompose $\Theta_i(\boldsymbol{v}_{-i})$ into a disjoint union of its connected components $\mathcal{C}(\boldsymbol{v}_{-i})$, that is, $\Theta_i(\boldsymbol{v}_{-i}) = \bigcup_{C\in\mathcal{C}(\boldsymbol{v}_{-i})} C$. We assume in this section that $\mathcal{C}(\boldsymbol{v}_{-i})$ is finite. Let $C(v_i, \boldsymbol{v}_{-i}) \in \mathcal{C}(\boldsymbol{v}_{-i})$ denote the connected component v_i lies in.

Component-wise Groves Mechanisms A *component-wise Groves mechanism* is defined by functions $h_i^C : \boldsymbol{\Theta}_{-i} \to \mathbb{R}$ for every connected component $C \in \mathcal{C}(\boldsymbol{v}_{-i})$ and for every agent i. It charges Agent i

$$p_i(\boldsymbol{v}) = h_i^{C(v_i,\boldsymbol{v}_{-i})}(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(\boldsymbol{v})).$$

Lemma 7.3.4. *If $\boldsymbol{p}$ is IC, it is a component-wise Groves mechanism.*

Proof. Let $\boldsymbol{p}$ be IC. When restricted to a connected component $C \in \mathcal{C}(\boldsymbol{v}_{-i})$, $\boldsymbol{p}$ is a Groves mechanism due to Theorem 7.1.2. That is, there exists h_i^C such that for all $v_i \in C$, $p_i(v_i, \boldsymbol{v}_{-i}) = h_i^C(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(\boldsymbol{v}))$. So $\boldsymbol{p}$ is a component-wise Groves mechanism given by $(h_i^C)_{C\in\mathcal{C}(\boldsymbol{v}_{-i})}$. □

Theorem 7.3.5. *A pricing rule $\boldsymbol{p}$ is IC and IR if and only if it is a component-wise Groves mechanism given, for each i and each $\boldsymbol{v}_{-i}$, by $(h_i^C)_{C\in\mathcal{C}(\boldsymbol{v}_{-i})}$ that satisfies*

$$h_i^C \le \inf_{\widetilde{v}_i^C\in C} w(\widetilde{v}_i^C, \boldsymbol{v}_{-i}) \;\forall\, C \in \mathcal{C}$$

$$h_i^C - h_i^D \le \inf_{\substack{\widetilde{v}_i^C\in C\\ \widetilde{v}_i^D\in D}} w(\widetilde{v}_i^C, \boldsymbol{v}_{-i}) - \left[\widetilde{v}_i^C(\alpha^{\mathrm{eff}}(\widetilde{v}_i^D, \boldsymbol{v}_{-i})) + \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(\widetilde{v}_i^D, \boldsymbol{v}_{-i}))\right] \;\forall\, C, D \in \mathcal{C}. \tag{Constr.-$\mathcal{C}$}$$

Proof. Component-wise Groves mechanism $(h_i^C)_{C\in\mathcal{C}(\boldsymbol{v}_{-i})}$ is IR if and only if

$$v_i(\alpha^{\mathrm{eff}}(v_i, \boldsymbol{v}_{-i})) - \left[h_i^{C(v_i,\boldsymbol{v}_{-i})}(\boldsymbol{v}_{-i}) - \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(v_i, \boldsymbol{v}_{-i}))\right] \ge 0 \quad \forall\, (v_i, \boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}$$

$$\iff h_i^{C(v_i,\boldsymbol{v}_{-i})}(\boldsymbol{v}_{-i}) \le w(v_i,\boldsymbol{v}_{-i}) \quad \forall\,(v_i,\boldsymbol{v}_{-i}) \in \boldsymbol{\Theta}$$
$$\iff h_i^{C}(\boldsymbol{v}_{-i}) \le w(v_i,\boldsymbol{v}_{-i}) \quad \forall\,\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}, C \in \mathcal{C}(\boldsymbol{v}_{-i}), v_i \in C$$
$$\iff h_i^{C}(\boldsymbol{v}_{-i}) \le \inf_{\widetilde{v}_i \in C} w(\widetilde{v}_i,\boldsymbol{v}_{-i}) \quad \forall\,\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}, C \in \mathcal{C}(\boldsymbol{v}_{-i}).$$

It is IC if and only if for all $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$ and all $v_i, v_i' \in \Theta_i(\boldsymbol{v}_{-i})$, an agent of true type v_i has no incentive to misreport v_i' to the mechanism. Any component-wise Groves mechanism already satisfies these constraints for $v_i, v_i' \in C$, for every $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$ and every $C \in \mathcal{C}(\boldsymbol{v}_{-i})$, since when restricted to any connected component $C \in \mathcal{C}(\boldsymbol{v}_{-i})$ it is equivalent to the vanilla Groves mechanism given by h_i^C. So, for each $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$, it suffices to enforce IC constraints over all $v_i \in C$ and all $v_i' \in D$, over every pair of differing connecting components $C, D \in \mathcal{C}(\boldsymbol{v}_{-i})$. A component-wise Groves mechanism $(h_i^C)_{C \in \mathcal{C}(\boldsymbol{v}_{-i})}$ is therefore IC if and only if (for all $\boldsymbol{v}_{-i} \in \boldsymbol{\Theta}_{-i}$; let $\mathcal{C} = \mathcal{C}(\boldsymbol{v}_{-i})$, $h_i^C = h_i^C(\boldsymbol{v}_{-i})$, $h_i^D = h_i^D(\boldsymbol{v}_{-i})$ for brevity)

$$v_i(\alpha^{\mathrm{eff}}(v_i,\boldsymbol{v}_{-i})) - \left[h_i^C - \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(v_i,\boldsymbol{v}_{-i}))\right]$$
$$\ge v_i(\alpha^{\mathrm{eff}}(v_i',\boldsymbol{v}_{-i})) - \left[h_i^D - \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(v_i',\boldsymbol{v}_{-i}))\right] \quad \forall\, C, D \in \mathcal{C}, v_i \in C, v_i' \in D$$
$$\iff h_i^C - h_i^D \le w(v_i,\boldsymbol{v}_{-i}) - \left[v_i(\alpha^{\mathrm{eff}}(v_i',\boldsymbol{v}_{-i})) + \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(v_i',\boldsymbol{v}_{-i}))\right]$$
$$\forall\, C, D \in \mathcal{C}, v_i \in C, v_i' \in D$$
$$\iff h_i^C - h_i^D \le \inf_{\substack{\widetilde{v}_i^C \in C\\ \widetilde{v}_i^D \in D}} w(\widetilde{v}_i^C,\boldsymbol{v}_{-i}) - \left[\widetilde{v}_i^C(\alpha^{\mathrm{eff}}(\widetilde{v}_i^D,\boldsymbol{v}_{-i})) + \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(\widetilde{v}_i^D,\boldsymbol{v}_{-i}))\right]$$
$$\forall\, C, D \in \mathcal{C}.$$

The theorem statement now follows from Lemma 7.3.4. □

Theorem 7.3.6. *The unique revenue-optimal mechanism subject to efficiency, IC, and IR is the component-wise Groves mechanism given by, for each agent i, $(h_i^C)_{C \in \mathcal{C}_{\boldsymbol{v}_{-i}}}$ that maximizes $\sum_{C \in \mathcal{C}(\boldsymbol{v}_{-i})} h_i^C$ subject to the constraints* (Constr.-$\mathcal{C}$).

Proof. The proof is similar to that of Theorem 7.3.3. The LP $\max\{\sum_{C \in \mathcal{C}(\boldsymbol{v}_{-i})} h_i^C$: (Constr.-$\mathcal{C}$)$\}$ solves the single-source shortest paths problem on the directed graph $G = (V, E)$ with vertices $V = \{s\} \cup \mathcal{C}$, edges $E = (\{s\} \cup \mathcal{C}) \times \mathcal{C}$, and edge costs

$$\mathsf{cost}(s, C) = \inf_{\widetilde{v}_i^C \in C} w(\widetilde{v}_i^C,\boldsymbol{v}_{-i}) \ \forall\, C \in \mathcal{C}$$
$$\mathsf{cost}(D, C) = \inf_{\substack{\widetilde{v}_i^C \in C\\ \widetilde{v}_i^D \in D}} w(\widetilde{v}_i^C,\boldsymbol{v}_{-i}) - \left[\widetilde{v}_i^C(\alpha^{\mathrm{eff}}(\widetilde{v}_i^D,\boldsymbol{v}_{-i})) + \sum_{j\neq i} v_j(\alpha^{\mathrm{eff}}(\widetilde{v}_i^D,\boldsymbol{v}_{-i}))\right] \ \forall\, C, D \in \mathcal{C}$$

in the sense that the optimal $h_i^C(\boldsymbol{v}_{-i})$ is the cost of the shortest $s \to C$ path in G. It follows that maximizing $\sum_C h_i^C$ yields the unique revenue optimal efficient mechanism. □

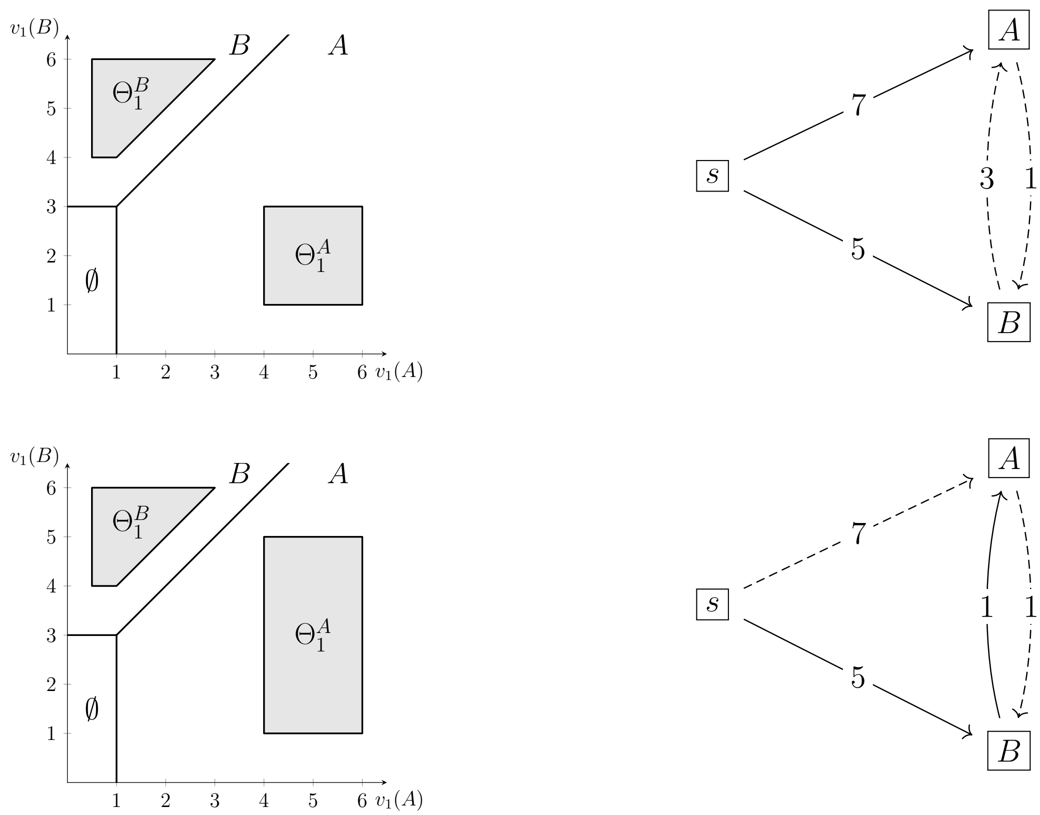


Figure 7.1: Examples of a disconnected type space $\Theta_1 = \Theta_1^A \cup \Theta_1^B$ and the corresponding graph G encoding the optimal efficient mechanism. The solid edges in G make up the tree of shortest paths.

Depending on the mechanism design setting, the connected component graph can be significantly smaller than the allocation graph from the previous section. For example, $|\Gamma|$ is exponential in combinatorial auctions so the graph from Section 7.3.1 is prohibitively large. If type spaces are represented as a union of K connected components, the graph in the present section has $K+1$ vertices and $K(K+1)$ edges, regardless of how large $|\Gamma|$ might be. The key disclaimer here is that we have not pursued the algorithmic question of how to compute the edge weights of these graphs—that is an important next step for future research.

7.3.3 Example

Consider the same example from Section 7.2 with $v_2(B) = 3$ and $v_3(A) = 1$. Figure 7.1 displays the partition of Bidder 1's ambient type space $\mathbb{R}^2$ into three regions labeled $\emptyset$, A, and B, in which she wins nothing, item A, or item B in the efficient allocation, respectively. Vanilla WT prescribes $h_i^A = h_i^B = \min\{5, 7\} = 5$, with a payment of 2 if $v_i \in \Theta_1^A$ and a payment of 4 if $v_i \in \Theta^B$. Suppose now that Bidder 1's type space is $\Theta_1^A \cup \Theta_1^B$ as displayed in the first row of Figure 7.1. Since Θ_1^A and Θ_1^B are each connected, the allocation-wise graph and the component-wise graph are identical. We have $\inf_{\widetilde{v}_i^A \in \Theta^A} w(\widetilde{v}_i^A, \boldsymbol{v}_{-i}) = 7$ and $\inf_{\widetilde{v}_i^B \in \Theta^B} w(\widetilde{v}_i^B, \boldsymbol{v}_{-i}) = 5$, attained by $\widetilde{v}_i^A = (4, 1)$ and $\widetilde{v}_i^B = (1, 4)$, respectively. We have $\mathsf{cost}(B, A) = 3$ with $\widetilde{v}_1^A = (4, 3)$ attaining the infimum and $\mathsf{cost}(A, B) = 1$ with $\widetilde{v}_1^B = (1, 4)$ attaining the infimum. The revenue-optimal mechanism thus sets $h_1^A = 7$ and $h_1^B = 5$, extracting a payment of 4 from Bidder 1 independent of which component her true type lies in. The second row of Figure 7.1 displays a similar situation with a larger Θ_1^A. The edge costs of (s, A), (s, B), and (A, B) remain the same, but now $\mathsf{cost}(B, A) = 1$ with $\widetilde{v}_1^A = (4, 5)$ attaining the infimum. The revenue-optimal mechanism here sets $h_1^A = 6$ and $h_1^B = 5$, extracting a payment of 3 from Bidder 1 if her true type is in Θ_1^A and 4 if it is in Θ_1^B. So, the "less precise" knowledge conveyed by the larger Θ_1^A results in lower payment extracted.

7.4 Conclusions and Future Research

We derived the revenue-optimal efficient mechanism when type spaces can be completely general. Our result significantly expands and generalizes the prior state of the art, the weakest type mechanism, that, while optimal for connected type spaces, is suboptimal for more general type spaces. Connected type spaces place a severe restriction on the kinds of knowledge structures that can be represented, and many natural informational constraints on agent types can only be described via a disconnected type space. We gave two characterizations of the optimal efficient mechanism, one via allocation-wise Groves mechanisms and one via component-wise Groves mechanisms. Both characterizations utilize the underlying network flow structure induced by incentive compatibility and individual rationality constraints.

Studying the computational aspects of our mechanisms is a pressing next research direction. Algorithms for computing the revenue-optimal efficient mechanism are an important next step to make our mechanisms practical. Such algorithms will be intrinsically tied to the *syntax* of agents' type spaces—how they are represented and learned will dictate how the optimal mechanism should be computed.

An important complementary research direction is to develop techniques to *learn* representations of agent type spaces. Due to the generality of knowledge that type spaces can represent, we envision that modern machine learning models can be especially useful. In our setting, the learning problem is in a sense decoupled from the actual mechanism design. For example, prior learning-based approaches to mechanism design such as Dütting et al. [2019], Wang et al. [2024] learned the allocation function and payment function by representing them as neural networks—in contrast, in our setting one would use a machine learning model to learn as much information about the agents as possible, and then use our mechanism as a post-processor. Learning supports of (mixtures of) distributions [Scott and Nowak, 2005, Dasgupta et al., 2005] and constraint learning [Fajemisin et al., 2024] are both relevant approaches for learning and representing type spaces from historical agent data. Extending the methodology here to other knowledge structures that incorporate distributional information and other forms of uncertainty is an interesting direction as well.

Chapter 8

Learning to Generate Artificial Competition

In this chapter, we introduce a new class of auctions that augment VCG prices with auctioneer-specified levels of *competition*. We show that this new auction class offers the flexibility and expressive power to meaningfully boost revenue under three different auctioneer knowledge models: (i) knowledge of full bidder valuation distributions, (ii) knowledge of bidder valuation quantiles, and (iii) knowledge of historical bidder valuation data.

Our primary research question in this chapter is: *how can the auction designer use additional knowledge to boost revenue via enhanced competition while striving to run an efficient auction?* If efficiency, IC, and IR are constraints of the auction design, the previous chapters have shown that the WT auction is revenue optimal (assuming connected type spaces, which we will do in this chapter). So, a more competitive auction that implements the efficient allocation and attempts to boost revenues beyond WT necessarily runs the risk of determining that a bidder should pay more than her winning bid price. Such a bidder could respond in one of two ways to that situation. She could decline the offer, which would force the auctioneer to keep her winning items unsold and result in an economically inefficient allocation. But if the overcharge is not too significant, she might accept the offer—violating her individual rationality constraint—leading to the efficient allocation to be realized. In both cases an economically desirable aspect of the auction design is compromised: either the auctioneer settles for a less-than-efficient allocation, or the auctioneer accepts that a bidder was overcharged (potentially eroding bidder trust and opening the door for further unwanted negotiation). We model this behavior, design new kinds of *competitive* auctions that are sensitive to this behavior, and show how those auctions can increase revenue without violating individual rationality nor efficiency too frequently.

Summary of Contributions and Related Work

Competitive VCG auctions We introduce a new family of auctions, $\boldsymbol{f}$-VCG auctions, that gives the auction designer the expressive ability to specify precisely, for each bidder, an artificial competitor to drive competitive prices. These auctions have the feature that the auction parameters for a bidder—her *competitor*—can depend on the revealed bids of all other bidders.

Bidder behavior and individual rationality We introduce a model of bidders who are amenable to being overcharged past their winning bid price: $a(p, \kappa)$ is the probability that a bidder who bid p for a particular bundle accepts a counteroffer for the same bundle at a price of $p + \kappa > p$. For example, a television company that bid \$10 million for broadcasting rights might be willing to pay an extra \$10000 to satisfy the competitive requirements of the auction and win the rights instead of dropping out altogether. In light of this bidder model we pose a weaker—but arguably more sensible from the auctioneer's perspective—individual rationality requirement: informally, an auction is (π, κ)-IR for a bidder if (i) the set of bidder types that the auction overcharges has probability mass $\leq 1 - \pi$ and (ii) no bidder type is ever overcharged by more than κ.

An alternate widely-studied relaxation of IR is *Bayesian-IR (B-IR)*, which demands that a bidder's utility is non-negative only in expectation over the other bidders' values. The revenue-optimal auction subject to efficiency, IC, and B-IR is the Bayesian weakest-type VCG auction of Krishna and Perry [1998]. We argue that our notion of (π, κ)-IR has several advantages over B-IR as an auction-design desideratum for at least the following reasons. First, the decision of whether or not to participate is made significantly easier for the bidders. A B-IR auction requires a bidder to understand the value distributions of other bidders, and that understanding should match the auctioneer's own understanding—a strong common knowledge assumption. In contrast, a (π, κ)-IR auction only requires bidders to reason about whether they are willing to accept an overcharge by $\$\kappa$ and thus provides bidders a greater degree of transparency. Second, a (π, κ)-IR auction is more favorable to risk-averse bidders than a B-IR auction which can lead to high overcharges (even if with low probability). B-IR auctions can indeed result in arbitrarily high overcharges (this is the case with the famous B-IR auction of Crémer and McLean [1988]; see Bikhchandani [2010]) while (π, κ)-IR auctions have an explicit cap κ on overcharge. Third, B-IR auctions can overcharge bidders with much higher frequency than (π, κ)-IR auctions which have an explicit cap $1 - \pi$ on overcharge frequency (Example 8.2.5). Fourth, (π, κ)-IR is a flexible enough participation model to capture forms of auctioneer knowledge other than a full value distribution. In Section 8.2 we study a knowledge model involving quantiles. Here, the appropriate participation constraint is a "robust" (π, κ)-IR constraint. B-IR, on the other hand, is incompatible with the quantile knowledge model.

Revenue-optimal efficient auctions When counteroffers are restricted to be close enough to the bid price so that bidders accept the overcharge, we derive the revenue-optimal auction subject to efficiency, IC, and (π, κ)-IR. The revenue-optimal auction belongs to our new family of $\boldsymbol{f}$-VCG auctions. We study two auctioneer knowledge models: full bidder value distributions and bidder value quantiles. We define the appropriate notion of (π, κ)-IR and derive the revenue-optimal efficient auction for both knowledge models.

Sample and computationally efficient learning The third auctioneer knowledge model we study is sample access to historical bidder data. We derive a general learning framework to find revenue-maximizing $\boldsymbol{f}$-VCG auctions when bidder behavior is prescribed by their overcharge acceptance probability. When overcharges are sufficiently small such that efficiency can be ensured, our learning algorithms output nearly globally revenue-optimal efficient auctions subject to (π, κ)-IR. We then show how to learn high-revenue, probably-efficient $\boldsymbol{f}$-VCG auctions subject to ex-post IR when bidders *never accept overcharges* (the standard bidder assumption in auction design). In both of these important settings we show how our learning algorithms can be efficiently implemented with a winner determination oracle.

An important and unique feature of our learning framework for competition is that the algorithms are *instance adaptive* and parallelize across bidders. In all prior work, the auction parameter optimization is done based on the training data before the test instance is drawn. In our approach, the auction parameters for a particular bidder are chosen *based on the test-time revealed bids of all other bidders*, and parameter optimization across bidders can be done in parallel.

Related work In Section 8.2 bidder types can be arbitrarily correlated, and the revenue-optimal choice of competitor for each bidder depends heavily on the revealed types of all other bidders. The interdependent values model [Milgrom and Weber, 1982] is thematically similar in that one bidder's private value can be influenced by the others. In our setting bidders themselves have no inherent uncertainty about their private values—it is the auctioneer who can refine his knowledge about a bidder after seeing the revealed types of everybody else. In fact, efficiency might be impossible to achieve when a bidder's own understanding of her value is correlated to other bidders [Dasgupta and Maskin, 2000, Jehiel et al., 2006].

Increasing competition (and thus revenue) by recruiting additional bidders has been studied starting with Bulow and Klemperer [1996]. Our approach gives the auction designer the flexibility to express artificial competition. In high-stakes applications like sourcing or spectrum recruiting additional bidders might not be possible. Another class of auctions that, indirectly, boost revenue while maintaining efficiency are core-selecting auctions [Day and Raghavan, 2007]. However, such auctions are not IC [Goeree and Lien, 2016, Othman and Sandholm, 2010, Prasad et al., 2025b]. Our auctions are IC and relax IR (Section 8.2) and sometimes efficiency (Section 8.3). Finally, Sandholm [2013] used *phantom bids* in sourcing/procurement auctions to optimize the decision of what items to procure through other means—a form of competition that is different from our approach since it directly affects the final allocation. Our competitive auctions solely drive prices.

Finally, our learning algorithms in Section 8.3 are *instance adaptive* and parallelize across bidders unlike prior approaches to data-driven auction design. We situate our work within that literature in Section 8.3.

8.1 Problem Formulation, f-VCG Auctions, and Our Bidder Model

$\boldsymbol{f}$-VCG auctions

We now define our new auction family: $\boldsymbol{f}$-VCG auctions. For a tuple of functions $\boldsymbol{f} = (f_1, \ldots, f_n)$, $f_i : \times_{j \neq i} \mathbb{R}^{2^m} \to \mathbb{R}^{2^m}$, the $\boldsymbol{f}$-VCG auction (1) elicits bidders' types $\boldsymbol{v} = (v_1, \ldots, v_n)$, (2) selects the efficient allocation $\boldsymbol{S}^*$ achieving welfare $w(\boldsymbol{v})$, and (3) offers bidder i her winning bundle S_i^* for a price of $p_i^{\boldsymbol{f}}(\boldsymbol{v}) = w(f_i(\boldsymbol{v}_{-i}), \boldsymbol{v}_{-i}) - w(\boldsymbol{v}|N \setminus i)$. In step (3) if $v_i(S_i^*) \geq p_i^{\boldsymbol{f}}(\boldsymbol{v})$, bidder

i is required to pay $p_i^{\boldsymbol{f}}(\boldsymbol{v})$ (this prevents equilibria other than truthful bidding where low bidders overbid). Otherwise if $v_i(S_i^*) < p_i^{\boldsymbol{f}}(\boldsymbol{v})$, bidder i can choose to accept the higher payment (violating her IR constraint) or exit the auction altogether (leading to an inefficient allocation with S_i^* unsold). All $\boldsymbol{f}$-VCG auctions are IC since they are Groves mechanisms (that is, the *pivot* term $w(f_i(\boldsymbol{v}_{-i}), \boldsymbol{v}_{-i})$ has no dependence on bidder i's revealed type), and have the natural interpretation of $f_i(\boldsymbol{v}_{-i})$ outputting an artificial *competitor* for bidder i. The $\boldsymbol{f}$-auctions $f_i = 0$, $f_i = \operatorname{argmin}_{\widetilde{v}_i \in \Theta_i} \mathbb{E}_{\boldsymbol{v}_{-i}}[w(\widetilde{v}_i, \boldsymbol{v}_{-i})]$, and $f_i(\boldsymbol{v}_{-i}) = \operatorname{argmin}_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ are vanilla VCG, Bayesian WT [Krishna and Perry, 1998], and WT [Balcan et al., 2023], respectively. Let $p_i^{\widetilde{v}_i} = w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - w(\boldsymbol{v}|N \setminus i)$ be the price when the competitor $\widetilde{v}_i$ is directly specified.

Overcharges, competition, bidder behavior

An $\boldsymbol{f}$-VCG auction risks incurring an *overcharge* of $o_i^{\boldsymbol{f}}(\boldsymbol{v}) := p_i^{\boldsymbol{f}}(\boldsymbol{v}) - v_i(S_i^*) > 0$ for bidder i. Let $a(p, \kappa) \in [0, 1]$ be the probability that a bidder who wins bundle S_i^* with bid price $p = v_i(S_i^*)$ accepts a counteroffer for the same bundle at price $p + \kappa$. We say bidder i is *overcharged* by κ if $o_i^{\boldsymbol{f}}(\boldsymbol{v}) = \kappa > 0$, regardless of whether she accepts or not. We have $o_i^{\boldsymbol{f}}(\boldsymbol{v}) = w(f(\boldsymbol{v}_{-i}), \boldsymbol{v}_{-i}) - w(v_i, \boldsymbol{v}_{-i})$, so bidder i is overcharged if and only if $w(v_i, \boldsymbol{v}_{-i}) < w(f(\boldsymbol{v}_{-i}), \boldsymbol{v}_{-i})$, that is, she is not competitive enough. Let $\mathsf{pay}_i^{\boldsymbol{f}}(\boldsymbol{v}) = p_i^{\boldsymbol{f}}(\boldsymbol{v})(\mathbf{1}[o_i^{\boldsymbol{f}}(\boldsymbol{v}) \le 0] + a(v_i(S_i^*), o_i^{\boldsymbol{f}}(\boldsymbol{v}))\mathbf{1}[o_i^{\boldsymbol{f}}(\boldsymbol{v}) > 0])$ be bidder i's expected payment in the $\boldsymbol{f}$-VCG auction. Let $o_i^{\widetilde{v}_i}(\boldsymbol{v}) = p_i^{\widetilde{v}_i}(\boldsymbol{v}) - v_i(S_i^*) = w(\widetilde{v}_i, \boldsymbol{v}_{-i}) - w(v_i, \boldsymbol{v}_{-i})$ and $\mathsf{pay}_i^{\widetilde{v}_i}(\boldsymbol{v}) = p_i^{\widetilde{v}_i}(\boldsymbol{v})(\mathbf{1}[o_i^{\widetilde{v}_i}(\boldsymbol{v}) \le 0] + a(v_i(S_i^*), o_i^{\widetilde{v}_i}(\boldsymbol{v}))\mathbf{1}[o_i^{\widetilde{v}_i}(\boldsymbol{v}) > 0])$ be the overcharge and expected payment, respectively, when the competitor $\widetilde{v}_i$ is directly specified. Finally, let $C(\widetilde{v}_i; \boldsymbol{v}_{-i}) = \{v_i \in \Theta_i(\boldsymbol{v}_{-i}) : w(v_i, \boldsymbol{v}_{-i}) \ge w(\widetilde{v}_i, \boldsymbol{v}_{-i})\}$ be the set of types competitive with $\widetilde{v}_i$ given $\boldsymbol{v}_{-i}$.

8.2 Revenue-Optimal Efficient Auctions

We study two sources of additional bidder information available to the auction designer: knowledge of a full value distribution and knowledge of value quantiles consistent with an unknown value distribution. In this section we assume overcharges are small enough to always be accepted. This allows us to guarantee efficiency of our auctions and derive revenue-optimal efficient auctions subject to relaxed IR.

Definition 8.2.1. Fix $\boldsymbol{v}_{-i}$. We say κ is an *acceptable* overcharge for bidder i if $a(v_i(S_i), \kappa) = 1$ for all $v_i \in \Theta_i(\boldsymbol{v}_{-i}), S_i \in B_i, S_i \neq \emptyset$. Let v_i^{κ}, which we call the *κ-competitor*, denote a bidder type such that $w(v_i^{\kappa}, \boldsymbol{v}_{-i}) = \kappa + \min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$.

If an $\boldsymbol{f}$-VCG auction only generates acceptable overcharges, $\mathsf{pay}_i^{\boldsymbol{f}}(\boldsymbol{v}) = p_i^{\boldsymbol{f}}(\boldsymbol{v})$ and the auction is efficient.

To situate our results, we first restate the revenue optimality result of Balcan et al. [2023] (covered in Chapter 5) in terms of $\boldsymbol{f}$-VCG auctions. In all results, D is a Borel probability distribution on Θ.

Theorem 8.2.2 (Balcan et al. [2023]). *Let Θ be compact and connected. Let D be any distribution on Θ. The $\boldsymbol{f}$-VCG auction $f_i(\boldsymbol{v}_{-i}) = \operatorname{argmin}_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ maximizes $\mathbb{E}_{\boldsymbol{v} \sim D}[\mathsf{pay}_i]$ for each i, and is thus revenue optimal, subject to efficiency, IC, and IR.*

8.2.1 Knowledge Model 1: Bidder Value Distributions

We formally define our IR relaxation, (π, κ)-IR, in the distributional knowledge model where the auction designer knows the bidder valuation distribution D over Θ.

Definition 8.2.3 ((π,κ)-IR, full value distribution)**.** An auction is (π, κ)-*IR* with respect to D if for each bidder i $\Pr_{\boldsymbol{v}\sim D}[o_i(\boldsymbol{v}) > 0] \le 1-\pi$ and $o_i(\boldsymbol{v}) \le \kappa$ for all $\boldsymbol{v} \in \Theta$.

We now characterize the revenue-optimal auction subject to efficiency, IC, and (π, κ)-IR. It can be written as an $\boldsymbol{f}$-VCG auction.

Theorem 8.2.4. *Let Θ be a compact and connected type space. Let D be a distribution supported on Θ and let μ be the corresponding probability measure. Let $\mu_{\boldsymbol{v}_{-i}}$ be the conditional measure over $v_i \in \Theta_i(\boldsymbol{v}_{-i})$. Let $v_i^\pi \in \Theta_i(\boldsymbol{v}_{-i})$ be such that $\mu_{\boldsymbol{v}_{-i}}(C(v_i^\pi; \boldsymbol{v}_{-i})) = \pi$ and let κ be an acceptable overcharge. The $\boldsymbol{f}$-VCG auction defined by*

$$f_i(\boldsymbol{v}_{-i}) = \begin{cases} v_i^\pi & \textit{if } w(v_i^\pi, \boldsymbol{v}_{-i}) \le w(v_i^\kappa, \boldsymbol{v}_{-i}) \\ v_i^\kappa & \textit{otherwise} \end{cases}$$

maximizes $\mathbb{E}_{\boldsymbol{v}\sim D}[\mathsf{pay}_i]$ for each i, and is thus revenue optimal, subject to efficiency, IC, and (π,κ)-IR w.r.t. D.

Proof. For $\pi \in (0, 1]$ let

$$L_\pi(\boldsymbol{v}_{-i}) = \operatorname*{argmax}_{L \subseteq \Theta_i(\boldsymbol{v}_{-i})} \left\{ w(\widetilde{v}_i, \boldsymbol{v}_{-i}) : \begin{array}{l} \mu_{\boldsymbol{v}_{-i}}(L) = \pi, \\ \widetilde{v}_i = \operatorname{argmin}_{\hat{v}_i \in L} w(\hat{v}_i, \boldsymbol{v}_{-i}) \end{array} \right\},$$

that is, $L_\pi(\boldsymbol{v}_{-i}) \subseteq \Theta_i(\boldsymbol{v}_{-i})$ is the set of probability mass π with the strongest weakest type. For a candidate weakest type $\hat{v}_i$, consider the set $C(\hat{v}_i; \boldsymbol{v}_{-i}) = \{v_i \in \Theta_i(\boldsymbol{v}_{-i}) : w(v_i, \boldsymbol{v}_{-i}) \ge w(\hat{v}_i, \boldsymbol{v}_{-i})\}$. $C(\hat{v}_i, \boldsymbol{v}_{-i})$ is precisely the set of types v_i in $\Theta_i(\boldsymbol{v}_{-i})$ that are *competitive with* $\hat{v}_i$, that is, types v_i that are not overcharged by $p_i^{\hat{v}_i}(\cdot, \boldsymbol{v}_{-i})$. There are three steps to the proof. First, we show there exists a type $\widetilde{v}_i$ such that $C(\widetilde{v}_i; \boldsymbol{v}_{-i})$ has measure π. Next, we show that $L_\pi(\boldsymbol{v}_{-i}) = C(\widetilde{v}_i; \boldsymbol{v}_{-i})$, which explicitly characterizes $L_\pi(\boldsymbol{v}_{-i})$ in terms of competitive types (this is an alternate characterization to the one in the theorem statement that was solely based on competitive sets). Finally, an application of revenue equivalence in the style of Krishna and Perry [1998], Balcan et al. [2023] allows us to establish payment optimality.

Let $\underline{v}_i = \operatorname{argmin}_{\hat{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\hat{v}_i, \boldsymbol{v}_{-i})$ and $\overline{v}_i = \operatorname{argmax}_{\hat{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\hat{v}_i, \boldsymbol{v}_{-i})$ be the weakest and strongest competitors in $\Theta_i(\boldsymbol{v}_{-i})$, respectively (both exist due to compactness of $\Theta_i(\boldsymbol{v}_{-i})$). We have $C(\underline{v}_i; \boldsymbol{v}_{-i}) = \Theta_i(\boldsymbol{v}_{-i})$ so $\mu_{\boldsymbol{v}_{-i}}(C(\underline{v}_i; \boldsymbol{v}_{-i})) = 1$. We now argue that $\mu_{\boldsymbol{v}_{-i}}(C(\overline{v}_i; \boldsymbol{v}_{-i})) = 0$. For $S_i \in B_i$ let S_{-i} be the allocation restricted to $N \setminus i$ that maximizes welfare subject to the constraint that bidder i wins S_i. We have

$$\begin{aligned} C(\overline{v}_i; \boldsymbol{v}_{-i}) &= \{v_i : w(v_i, \boldsymbol{v}_{-i}) = w(\overline{v}_i, \boldsymbol{v}_{-i})\} \\ &= \bigcup_{S_i^* \in B_i} \left\{ v_i : \begin{array}{l} v_i(S_i^*) + \sum_{j \ne i} v_j(S_j^*) \ge v_i(S_i') + \sum_{j\ne i} v_j(S_j') \;\forall\; S_i' \in B_i \setminus S_i^*, \\ v_i(S_i^*) + \sum_{j\ne i} v_j(S_j^*) = w(\overline{v}_i, \boldsymbol{v}_{-i}) \end{array} \right\} \end{aligned}$$

where each set in the (finite) union is of measure zero since the second constraint demands the zero probability event that $v_i(S_i^*)$ take on the particular fixed value of $w(\overline{v}_i, \boldsymbol{v}_{-i}) - \sum_{j \ne i} v_j(S_j^*)$.

So $C(\overline{v}_i; \boldsymbol{v}_{-i})$ is itself of measure zero. As $\Theta_i(\boldsymbol{v}_{-i})$ is convex (and thus connected), continuity of $\mu_{\boldsymbol{v}_{-i}}(C(\cdot; \boldsymbol{v}_{-i}))$ and the intermediate value theorem imply the existence of $\widetilde{v}_i$ with $\mu_{\boldsymbol{v}_{-i}}(C(\widetilde{v}_i; \boldsymbol{v}_{-i})) = \pi$. Fix this $\widetilde{v}_i$. We claim that $L_\pi(\boldsymbol{v}_{-i}) = C(\widetilde{v}_i; \boldsymbol{v}_{-i})$. For the sake of contradiction, suppose that $L_\pi(\boldsymbol{v}_{-i}) = L \neq C(\widetilde{v}_i; \boldsymbol{v}_{-i})$, and let v'_i be the weakest type of L. So $\mu_{\boldsymbol{v}_{-i}}(L) = \pi$ and $w(v'_i, \boldsymbol{v}_{-i}) > w(\widetilde{v}_i, \boldsymbol{v}_{-i})$. Since the weakest type v'_i of L generates strictly more welfare than $\widetilde{v}_i$, the set of types competitive with v'_i is a strict subset of the set of types competitive with $\widetilde{v}_i$, that is, $C(v'_i; \boldsymbol{v}_{-i}) \subset C(\widetilde{v}_{-i}; \boldsymbol{v}_{-i})$, which means $\mu_{\boldsymbol{v}_{-i}}(C(v'_i; \boldsymbol{v}_{-i})) < \pi$. But as $L \subseteq C(v'_i; \boldsymbol{v}_{-i})$, this is a contradiction.

We now use the revenue equivalence theorem and the above characterization to prove payment optimality. The key intuition is that $f_i(\boldsymbol{v}_{-i})$ outputs a competitor that makes either the κ-constraint or the π-constraint of (π, κ)-IR tight. Therefore, greater payment cannot be obtained without violating relaxed-IR. Formally, suppose $p'_i(\boldsymbol{v})$ is an alternate payment rule that implements the efficient allocation, is IC, and $\mathbb{E}_{\boldsymbol{v}\sim D}[p'_i(\boldsymbol{v})] > \mathbb{E}_{\boldsymbol{v}\sim D}[p^{\boldsymbol{f}}_i(\boldsymbol{v})]$. By revenue equivalence [Vohra, 2011, Theorem 4.3.1], there exists a function $h_i(\boldsymbol{v}_{-i})$ such that $p'_i(\boldsymbol{v}) = p^{\boldsymbol{f}}_i(\boldsymbol{v}) + h_i(\boldsymbol{v}_{-i})$. So

$$\mathop{\mathbb{E}}_{\boldsymbol{v}\sim D}[p^{\boldsymbol{f}}_i(\boldsymbol{v}) + h_i(\boldsymbol{v}_{-i})] > \mathop{\mathbb{E}}_{\boldsymbol{v}\sim D}[p^{\boldsymbol{f}}_i(\boldsymbol{v})],$$

which means there exists a particular $\boldsymbol{v}_{-i}$ such that $h_i(\boldsymbol{v}_{-i}) > 0$. Fix this $\boldsymbol{v}_{-i}$, and let $\widetilde{v}_i = f_i(\boldsymbol{v}_{-i})$ (so $\widetilde{v}_i \in \{v^\kappa_i, v^\pi_i\}$). If $\widetilde{v}_i = v^\kappa_i$, the weakest type $\underline{v}_i$ of $\Theta_i(\boldsymbol{v}_{-i})$ is overcharged by exactly κ by $p^{\boldsymbol{f}}_i(\underline{v}_i, \boldsymbol{v}_{-i})$. That weakest type is therefore overcharged by more than κ by p'_i, violating the κ-constraint in (π, κ)-IR. Else if $\widetilde{v}_i = v^\pi_i$, v^π_i's utility is zero (that is, her IR constraint is tight) under $p^{\boldsymbol{f}}_i(v^\pi_i, \boldsymbol{v}_{-i})$. Therefore, v^π_i is overcharged by $p'_i(v^\pi_i, \boldsymbol{v}_{-i})$, and more importantly by continuity of $p^{\boldsymbol{f}}_i(\cdot, \boldsymbol{v}_{-i})$ there is a sufficiently-small open ball centered at v^π_i such that all types in that ball are overcharged by p'_i. Since the measure of types less competitive than v^π_i is exactly $1 - \pi$, a set of types of measure $> 1 - \pi$ is overcharged by p'_i. So p'_i violates the π-constraint of (π, κ)-IR in this case. □

We show via an example that B-IR auctions, specifically the Bayesian weakest-type auction of Krishna and Perry [1998], can overcharge with high frequency, giving further credence to our approach of optimal efficient auction design subject to an overcharge frequency cap.

Example 8.2.5. Consider an auction with two items A and B and two bidders $i \in \{1, 2\}$. The type space is $\Theta = \Theta_1 \times \Theta_2$ with $\Theta_i = \{(v_i(A), v_i(B)) \in \mathbb{R}^2_{\geq 0} : v_i(A) + v_i(B) = 1\}$ for both bidders (so, implicitly, $v_i(AB) = 0$ for both bidders). Suppose both bidders' valuations are distributed uniformly and independently over the type space. The Bayesian weakest types prescribed by Krishna and Perry [1998] are chosen before true values are revealed. The Bayesian weakest type for bidder 1 (and identically for bidder 2) is the valuation $\widetilde{v}_1 = (\widetilde{v}_1(A), \widetilde{v}_1(B))$ that minimizes $\mathbb{E}_{v_2}[w(\widetilde{v}_1, v_2)]$, which is $\widetilde{v}_1 = (1/2, 1/2)$.

Indeed, Let E denote the event that the weakest type $\widetilde{v}_1$ wins item A, so $E = \{\widetilde{v}_1(A) \geq v_2(A)\}$ and $\Pr(E) = \widetilde{v}_1(A)$. By definition of the type space, the weakest type wins B if and only if event E does not occur. For a given $\widetilde{v}_1$,

$$\begin{aligned}\mathop{\mathbb{E}}_{v_2}[w(\widetilde{v}_1, v_2)] &= \mathop{\mathbb{E}}_{v_2}[\widetilde{v}_1(A) \cdot \mathbf{1}[E] + \widetilde{v}_1(B) \cdot (1 - \mathbf{1}[E]) + v_2(A) \cdot (1 - \mathbf{1}[E]) + v_2(B) \cdot \mathbf{1}[E]] \\ &= \widetilde{v}_1(A)^2 + (1 - \widetilde{v}_1(A))^2 + (1 - \widetilde{v}_1(A)) \cdot \frac{\widetilde{v}_1(A) + 1}{2} + \frac{\widetilde{v}_1(A)^2}{2}\end{aligned}$$

which is minimized at $\widetilde{v}_1(A) = 1/2$, as claimed.

Suppose now that the realized type of bidder 2 is $(v_2(A), v_2(B)) = (1, 0)$, so bidder 2 wins item A and bidder 1 wins item B. According to the Bayesian weakest type, bidder 1 is charged $(1 + 1/2) - 1 = 1/2$, so whenever $v_1(B) < 1/2$, bidder 1 is overcharged. So, there is a 50% probability that bidder 1 is overcharged.

Looking to Theorem 8.2.4, the auction designer chooses the competitor v_i^π for bidder 1 after having seen bidder 2's revealed type of $(1, 0)$. For an overcharge probability of $1-\pi$, that weakest type is $v_i^\pi = (1 - \pi, \pi)$. The Bayesian weakest type is $v_i^{1/2}$ which induces an impractically high overcharge rate of 50%.

8.2.2 Knowledge Model 2: Bidder Value Quantiles

We now study a knowledge model where the auctioneer has less knowledge than a full bidder value distribution. In the quantile knowledge model, we assume some underlying *unknown* value distribution, but the auction designer knows quantiles corresponding to the distribution. Formally, for each bidder i, the auctioneer possesses a sequence of sets (that can depend on the revealed types of the other bidders) $\{\Theta_i^\pi(\boldsymbol{v}_{-i})\}_{0<\pi\leq 1}$ with $\Theta_i^\pi(\boldsymbol{v}_{-i}) \supseteq \Theta_i^{\pi'}(\boldsymbol{v}_{-i})$ for any $\pi \geq \pi'$ and $\Theta_i^1(\boldsymbol{v}_{-i}) = \Theta_i(\boldsymbol{v}_{-i})$. This sequence of *quantiles* represents the knowledge that $v_i \in \Theta_i^\pi(\boldsymbol{v}_{-i})$ with probability π given the bid profile $\boldsymbol{v}_{-i}$ of all other bidders. A distribution D over Θ is *consistent* with the quantiles if $\Pr_{\hat{\boldsymbol{v}}\sim D}[\hat{v}_i \in \Theta_i^\pi(\boldsymbol{v}_{-i}) | \hat{\boldsymbol{v}}_{-i} = \boldsymbol{v}_{-i}] = \pi$. The notion of (π, κ)-IR in the quantile knowledge model is a robust version of the distributional version.

Definition 8.2.6 ((π, κ)-IR, quantiles)**.** An auction is (π, κ)-*IR* with respect to $\{\Theta_i^\pi\}$ if for each bidder i $\sup_{\hat{D} \text{ consistent with } \{\Theta_i^\pi\}} \Pr_{\boldsymbol{v}\sim\hat{D}}[o_i(\boldsymbol{v}) > 0] \leq 1 - \pi$ and $o_i(\boldsymbol{v}) \leq \kappa$ for all $\boldsymbol{v} \in \Theta$.

Theorem 8.2.7. *Let Θ be a compact and connected type space. Let $\{\Theta_i^\pi(\boldsymbol{v}_{-i})\}$ be a sequence of quantiles such that (i) the set-valued function $\pi \mapsto \Theta_i^\pi(\boldsymbol{v}_{-i})$ is continuous and (ii) the map $\pi \mapsto \min_{\widetilde{v}_i\in\Theta_i^\pi(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ is decreasing in π. Let D be any distribution supported on Θ consistent with $\{\Theta_i^\pi(\boldsymbol{v}_{-i})\}$. Let $v_i^\pi = \operatorname{argmin}_{\widetilde{v}_i\in\Theta_i^\pi} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ be the weakest type in quantile $\Theta_i^\pi(\boldsymbol{v}_{-i})$ and let κ be an acceptable overcharge. The $\boldsymbol{f}$-VCG auction defined by*

$$f_i(\boldsymbol{v}_{-i}) = \begin{cases} v_i^\pi & \text{if } w(v_i^\pi, \boldsymbol{v}_{-i}) \leq w(v_i^\kappa, \boldsymbol{v}_{-i}) \\ v_i^\kappa & \text{otherwise} \end{cases}$$

maximizes $\mathbb{E}_{\boldsymbol{v}\sim D}[\mathsf{pay}_i]$ for each i, and is thus revenue optimal, subject to efficiency, IC, and (π, κ)-IR w.r.t. $\{\Theta^\pi\}$.

Proof sketch. Suppose there exists a prior distribution D consistent with the quantiles and an alternate payment rule p_i' that generates strictly more payment than the $\boldsymbol{f}$-VCG auction defined in the theorem statement, that is, $\mathbb{E}_{\boldsymbol{v}\sim D}[p_i'(\boldsymbol{v})] > \mathbb{E}_{\boldsymbol{v}\sim D}[p_i^{\boldsymbol{f}}(\boldsymbol{v})]$, and p_i' implements the efficient allocation and is IC. We will show that there exists a distribution $\hat{D}$ consistent with the quantiles such that under p_i', $\Pr_{\boldsymbol{v}\sim\hat{D}}[o_i'(\boldsymbol{v})] > 1 - \pi$. First, by revenue equivalence [Vohra, 2011, Theorem 4.3.1], there exists a function $h_i(\boldsymbol{v}_{-i})$ such that $p_i'(\boldsymbol{v}) = p_i^{\boldsymbol{f}}(\boldsymbol{v}) + h_i(\boldsymbol{v}_{-i})$. So we have $\mathbb{E}_{\boldsymbol{v}\sim D}[p_i^{\boldsymbol{f}}(\boldsymbol{v}) + h_i(\boldsymbol{v}_{-i})] > \mathbb{E}_{\boldsymbol{v}\sim D}[p_i^{\boldsymbol{f}}(\boldsymbol{v})]$, which means there must exist a particular $\boldsymbol{v}_{-i}$ such that $h_i(\boldsymbol{v}_{-i}) > 0$. Fix this $\boldsymbol{v}_{-i}$. We next construct the promised worst-case measure $\hat{\mu}$.

The construction of the worst-case measure $\hat{\mu}$ is simple: it is supported on a set of weakest types of the form[1]

$$\left\{ v_i^\pi : v_i^\pi = \underset{\widetilde{v}_i \in \Theta_i^\pi(\boldsymbol{v}_{-i})}{\operatorname{argmin}} \; w(\widetilde{v}_i, \boldsymbol{v}_{-i}), \pi \in (0,1] \right\}$$

and is defined to be consistent with the quantiles as $\hat{\mu}(\{v_i^\pi : \pi \in [\pi_1, \pi_2]\}) = \pi_2 - \pi_1$ for all $0 < \pi_1 < \pi_2 \leq 1$. The key property of this distribution is that if $\pi_1 < \pi_2$, $w(v_i^{\pi_1}, \boldsymbol{v}_{-i}) > w(v_i^{\pi_2}, \boldsymbol{v}_{-i})$, that is, the weakest type in quantile π_2 cannot compete with the weakest type in quantile π_1, so the $\hat{\mu}$-measure of types that cannot compete with v_i^π is precisely $1 - \pi$ (this shows that $\hat{\mu}$ achieves the supremum in the definition of (π, κ)-IR for any distribution D consistent with the quantiles). Another key fact is that the map $\omega : (0,1] \to \mathbb{R}_{\geq 0}$ defined by $\omega(\pi) = w(v_i^\pi, \boldsymbol{v}_{-i})$ is continuous (this is a consequence of Berge's Maximum Theorem and the fact that the set-valued function $\pi \mapsto \Theta_i^\pi$ is continuous).

Now, as in the proof of Theorem 8.2.4, if $\hat{\mu}(C(v_i^\kappa; \boldsymbol{v}_{-i})) \geq \pi$, v_i^κ is the optimal competitor since it satisfies (π, κ)-IR and we cannot overcharge by more than κ. Otherwise if $\hat{\mu}(C(v_i^\kappa; \boldsymbol{v}_{-i})) < \pi$, that is, the overcharge probability is $> 1 - \pi$, we must pick a weaker competitor to reduce the probability of overcharge. That competitor is the v_i^π such that $\hat{\mu}(C(v_i^\pi; \boldsymbol{v}_{-i})) = \pi$, which is precisely the weakest type v_i^π that minimizes $w(v_i^\pi, \boldsymbol{v}_{-i})$ over $v_i^\pi \in \Theta_i^\pi(\boldsymbol{v}_{-i})$.

Finally, consider the alternate payment rule p_i', and let $\widetilde{v}_i = f_i(\boldsymbol{v}_{-i})$ (so $\widetilde{v}_i \in \{v_i^\pi, v_i^\kappa\}$). If $\widetilde{v}_i = v_i^\kappa$, the weakest type $\underline{v}_i = v_i^1$ of $\Theta_i(\boldsymbol{v}_{-i})$ is overcharged by exactly κ by $p_i^{\boldsymbol{f}}(\underline{v}_i, \boldsymbol{v}_{-i})$. That weakest type is therefore overcharged by more than κ by p_i', violating the κ-constraint in (π, κ)-IR. Else if $\widetilde{v}_i = v_i^\pi$, v_i^π's IR constraint is tight when using payment rule $p_i^{\boldsymbol{f}}$. So p_i' overcharges v_i^π, and more importantly, by continuity of ω, there exists ε sufficiently small such that for all $\pi' \in (\pi - \varepsilon, \pi + \varepsilon)$, $v_i^{\pi'}$ is overcharged by p_i'. So the $\hat{\mu}$-probability mass of types being overcharged is more than $1 - \pi$, so p_i' violates the π-constraint of (π, κ)-IR. □

Let us emphasize that (for both knowledge models) in a $(99\%, \kappa)$-IR auction, only 1% of bidder types ever have to deal with issues of overcharge and participation. 99% of the time the auction is perfectly efficient, IC, IR, and enjoys improved revenues. The auctioneer sets π and κ to strike a balance between risk of overcharging weak bidders and enjoying increased revenue from the large majority of bidders.

We derived the globally revenue optimal efficient auction for *acceptable* overcharges. Acceptability ensured that a (π, κ)-IR auction remained efficient. Otherwise it is unlikely that a concise global revenue optimality guarantee exists since, without an efficiency constraint, that would solve revenue-optimal multi-item auction design—a major open question—as a special case. In Section 8.3 we use a data-driven approach to design $\boldsymbol{f}$-VCG auctions that are nearly revenue-optimal for the class of $\boldsymbol{f}$-VCG auctions (but not globally revenue optimal) for general overcharges. Before that, we discuss how our results generalize beyond auctions.

[1] How ties are broken in the argmin is irrelevant. What is important is continuity of the induced welfare function which is a consequence of Berge's theorem of the maximum.

8.2.3 Beyond Auctions: General Mechanism Design

Our results so far are not specific to combinatorial auctions and hold in a more general multidimensional mechanism design setting as in Balcan et al. [2023]. In that setting, Γ is a finite set of outcomes and an agent's type is a vector $v_i \in \mathbb{R}^\Gamma$ indexing her value for each outcome. The chief issue that must be addressed when applying our methodology to other settings is non-participation due to overcharge. What does non-participation mean, and what are its consequences, in the mechanism design setting of interest? In auctions, a non-participating agent receives no items. In other settings, for example public projects where the final outcome involves a resource shared by agents, non-participation might not be as naturally implementable.

8.3 Learning to Generate Competition

In the previous section we studied two different knowledge models for the auction designer: knowledge of the bidders' value distributions and knowledge of quantiles consistent with the bidders' value distributions. In practice, access to an exact prior is unrealistic, and fine-grained knowledge of quantiles as in the continuity requirement in Theorem 8.2.7 might be impractical. In this section we study a third, realistic, knowledge model: access to historical bidder data.

First we establish the formal setting. Our setup mirrors how combinatorial auctions are run in practice. We then prove our main learning guarantees for independently distributed bidder values (this is the standard assumption in mechanism design; we discuss challenges to extending our approach to correlated bidders) and provide learning algorithms. We then study the computational complexity of the learning algorithms. Throughout, we situate our results within the broader context of data-driven auction design.

Bidder valuations In practice a full valuation vector cannot be communicated due to its exponential length. Instead, the auction designer alleviates this issue by placing one of two restrictions on bidder valuations: (i) bidder i is restricted to submit bids on a set $B_i \subseteq 2^M$ of predetermined bundles or (ii) bidder i is allowed to submit bids on at most b bundles of her choice. Let $\mathsf{supp}(v_i) = \{S \subseteq M : v_i(S) > 0\}$ denote the *supported bids* of a valuation vector. We refer to valuation functions supported on B_i as B_i*-valuations* and valuation functions with support size $\leq b$ as b*-valuations*. In this section, for simplicity, we assume that bidder i submits a B_i-valuation function where B_i is set by the auction designer (we handle b-valuations in the full version of the paper). This is a practical requirement in combinatorial auctions to alleviate communication costs and the computational cost of winner determination (*e.g.,* spectrum auctions in the UK and Canada employed the XOR language with explicit bid limits of 4000 and 500, respectively [Ausubel and Baranov, 2017]).

Data-driven auction design The auction designer in our setting has access to K independently and identically distributed (IID) samples $V = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(K)}\}$ drawn from an unknown distribution D supported on Θ. We assume bidders' type spaces and type distributions are independent, that is, $\Theta = \Theta_1 \times \cdots \times \Theta_n$ and $D = D_1 \times \cdots \times D_n$ have product structures. So, $\Theta_i = \Theta_i(\boldsymbol{v}_{-i})$ is independent of the revealed types of the other agents and the conditional distribution over bidder

i's type given $\boldsymbol{v}_{-i}$ is just D_i. As discussed above, D_i is a distribution over B_i-valuations, that is, the type space of bidder i is of the form $\Theta_i \subseteq \{v_i \in [0, H]^{2^m} : \mathsf{supp}(v_i) = B_i\}$ where H is an upper bound on any bid.

Overcharge acceptance probability We assume that the probability of accepting an overcharge only depends on the overcharge: $a(\kappa) = a(p, \kappa)$. For example, if there are known appraisal values on the items being auctioned, it might be reasonable to assume some bid-independent probability of overcharge acceptance. This (stylized) assumption is solely for technical ease of exposition; without it our bounds would only change slightly to depend on the structure of $a(p, \kappa)$.

8.3.1 Learning Guarantees and Algorithms for Independent Bidder Types

Even with independent bidder types, our learning algorithms choose a competitor for bidder i that is highly dependent on $\boldsymbol{v}_{-i}$. For a dataset $V = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(K)}\}$ of type profiles, define $V_i = \{v_i^{(1)}, \ldots, v_i^{(K)}\}$ to be the dataset of bidder-i types. Since bidders are independently distributed, each V_i is an IID dataset from D_i. Let $\mathrm{OPT}_i^{\boldsymbol{f}}(\pi, \kappa)$ denote the optimal payment $\mathbb{E}_{\boldsymbol{v}\sim D}[\mathsf{pay}_i]$ of any (π, κ)-IR $\boldsymbol{f}$-VCG auction and let $\mathrm{OPT}_i(\pi, \kappa)$ denote the globally optimal payment of any efficient, IC, and (π, κ)-IR mechanism (achieved by the $\boldsymbol{f}$-VCG auction of Theorem 8.2.4 for acceptable κ).

We now present our main learning guarantees. Let $\mathcal{F}^{\mathsf{price}}(B_i) = \{p_i^{\widetilde{v}_i} : \Theta \to [0, H] : \mathsf{supp}(\widetilde{v}_i) = B_i\}$ and $\mathcal{F}^{\mathsf{pay}}(B_i) = \{\mathsf{pay}_i^{\widetilde{v}_i} : \Theta \to [0, H] : \mathsf{supp}(\widetilde{v}_i) = B_i\}$ be the collection of price and payment functions, respectively, parameterized by B_i-competitor $\widetilde{v}_i$. We bound the intrinsic complexity as measured by *pseudodimension* of these function families in order to prove our learning guarantees.

Lemma 8.3.1. *$\mathsf{Pdim}(\mathcal{F}^{\mathsf{price}}(B_i))$ and $\mathsf{Pdim}(\mathcal{F}^{\mathsf{pay}}(B_i))$ are at most $O(|B_i| \log |B_i|)$.*

Proof. We prove the bounds for $\mathcal{F}^{\mathsf{price}}(B_i)$ and $\mathcal{F}^{\mathsf{pay}}(B_i)$ first. Fix $\boldsymbol{v}$. For each $S_i \in B_i$, let $S_{-i} = (S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_n)$ denote the allocation that maximizes welfare subject to the constraint that bidder i wins S_i. Over all $S_i \in B_i$, consider the set of halfspaces in $\widetilde{v}_i \in \mathbb{R}^{B_i}$:

$$\widetilde{v}_i(S_i) + \sum_{j \neq i} v_j(S_j) \geq \widetilde{v}_i(S_i') + \sum_{j \neq i} v_j(S_j') \;\forall\; S_i' \in B_i \setminus S_i$$

where S_{-i}' denotes the welfare maximizing allocation subject to the constraint that i wins S_i'. This set of $\leq |B_i|^2$ hyperplanes corresponding to those halfspaces partitions $\mathbb{R}^{B_i}$ into regions such that within each region, the overall efficient allocation $\boldsymbol{S}$ is fixed. Thus, within each region,

$$p_i^{\widetilde{v}_i}(\boldsymbol{v}) = \widetilde{v}_i(S_i) + \sum_{j \neq i} v_j(S_j) - \sum_{j \neq i} v_j(S_j^*)$$

is linear in $\widetilde{v}_i$. An application of the main result of Balcan et al. [2025d] proves the pseudodimension bound for $\mathcal{F}^{\mathsf{price}}(B_i)$. To understand the structure of $\mathsf{pay}_i^{\widetilde{v}_i}$, consider the same set of halfspaces as above along with the following set of B_i additional halfspaces:

$$\widetilde{v}_i(S_i) + \sum_{j \neq i} v_j(S_j) \geq \sum_{j=1}^{n} v_j(S_j^*) \;\forall\; S_i \in B_i.$$

In each region in the previous decomposition where some fixed allocation $\boldsymbol{S}$ was efficient over all $\widetilde{v}_i$ in that region, the new halfspace creates two additional regions: in one $\widetilde{v}_i$ is less competitive than v_i and so $o_i^{\widetilde{v}_i}(\boldsymbol{v}) = 0 \implies \mathsf{pay}_i^{\widetilde{v}_i}(\boldsymbol{v}) = p_i^{\widetilde{v}_i}(\boldsymbol{v})$ and in the other $\widetilde{v}_i$ is more competitive than v_i so $\mathsf{pay}_i^{\widetilde{v}_i}(\boldsymbol{v}) = p_i^{\widetilde{v}_i}(\boldsymbol{v}) \cdot a(\kappa)$. In both cases pay is linear within each region. So in total, $O(|B_i|^2)$ hyperplanes partition $\mathbb{R}^{B_i}$ into regions such that within each region, $\mathsf{pay}_i^{\widetilde{v}_i}(\boldsymbol{v})$ is linear as a function of $\widetilde{v}_i$. The pseudodimension bound follows from Balcan et al. [2025d]. □

Let $\varepsilon(K, \delta) = O(H\sqrt{(|B_i| \log |B_i| + \ln(1/\delta))/K})$. The following corollary, which is a consequence of standard results from learning theory, shows that ε controls the error between empirical payment and expected payment uniformly over all possible competitors.

Corollary 8.3.2. *Fix $\boldsymbol{v}$. With probability $\geq 1-\delta$ over the draw of dataset $V = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(K)}\}$, the following quantities are at most $\varepsilon(K, \delta/n)$ for all i and all B_i-valuations $\widetilde{v}_i$.*

- $|\frac{1}{K}\sum_{\ell=1}^{K} p_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i}) - \mathbb{E}_{v_i\sim D_i}[p_i^{\widetilde{v}_i}(v_i, \boldsymbol{v}_{-i})]|$
- $|\frac{1}{K}\sum_{\ell=1}^{K} \mathsf{pay}_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i}) - \mathbb{E}_{v_i\sim D_i}[\mathsf{pay}_i^{\widetilde{v}_i}(v_i, \boldsymbol{v}_{-i})]|$
- $|\frac{|\{\ell : o_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i})>0\}|}{K} - \Pr_{v_i\sim D_i}[o_i^{\widetilde{v}_i}(v_i, \boldsymbol{v}_{-i}) > 0]|$

The above uniform convergence bounds are, for each bidder i, over a transformed training set of the form $(v_i^1, \boldsymbol{v}_{-i}), \ldots, (v_i^K, \boldsymbol{v}_{-i})$ for each bidder i. This is a form of *instance-adaptive* learning since we use the test-time revealed bids $\boldsymbol{v}_{-i}$ to (i) define the training set for bidder i and (ii) optimize the auction parameters, namely the competitor $\widetilde{v}_i$, for bidder i (as we show in Theorems 8.3.3 and 8.3.4). This is markedly different from prior approaches to data-driven auction design, for example by Balcan et al. [2025d] and references within, where in order for the learned auction to be IC, the auction parameters are set before the test instance is seen. Some prior work tackles the *unlimited supply* setting by learning prices "within an instance" from other bidders' revealed bids [Baliga and Vohra, 2003, Balcan et al., 2005], but limited supply (our setting) is more challenging [Balcan et al., 2021c].

We now translate these generalization guarantees into concrete learning algorithms. The most general result for any overcharge acceptance function $a(\kappa)$ is Theorem B.2 in the full version of the paper. It outputs an empirical-payment-maximizing competitor subject to empirical overcharge constraints. Here, we present algorithms for two pertinent cases. The first case is for acceptable κ—here the revenue-optimal efficient (π, κ)-IR auction is given by Theorem 8.2.4. The second case is for bidders who *do not accept overcharges*, that is, $a(\kappa) = 0$ for all κ. Here we obtain high-revenue learned auctions that are exactly ex-post IR and probably efficient.

Acceptable overcharges: nearly revenue-optimal efficient auctions

The learning algorithm defining $f_i(\boldsymbol{v}_{-i})$ outputs a competitor either from the dataset V_i or defaults to a κ-competitor v_i^{κ} based on empirical overcharge frequency.

Theorem 8.3.3. *Let κ be acceptable and let the underlying type space Θ be compact and con-*

nected. Given an IID dataset $V = \{\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(K)}\}$ *define the following* $\boldsymbol{f}$*-VCG auction:*

$$
\begin{aligned}
f_i(\boldsymbol{v}_{-i}) = \underset{\widetilde{v}_i \in V_i \cup \{v_i^\kappa\}}{\operatorname{argmax}} \; & w(\widetilde{v}_i, \boldsymbol{v}_{-i}) \\
\textit{s.t.} \quad & \tfrac{|\{\ell : o_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i}) > 0\}|}{K} \le 1 - \pi + \varepsilon(K, \delta/n) \\
& w(\widetilde{v}_i, \boldsymbol{v}_{-i}) \le w(v_i^\kappa, \boldsymbol{v}_{-i}).
\end{aligned}
$$

The resulting auction is efficient and, with probability $\ge 1-\delta$ *over the draw of* V, $\mathbb{E}_{\boldsymbol{v}\sim D}[\mathsf{pay}_i^{\boldsymbol{f}}(\boldsymbol{v})] \ge \mathrm{OPT}_i(\pi,\kappa) - 2\varepsilon(K,\delta/n)$ *for all* i *and is thus nearly revenue-optimal, and is* $(\pi - 2\varepsilon(K,\delta/n), \kappa)$*-IR.*

High-revenue probably-efficient $\boldsymbol{f}$-VCG auctions

We apply our techniques to the setting where bidders do not accept overcharges ($a(\kappa) = 0$ for all $\kappa > 0$). In other words, bidders' IR constraints must be satisfied. In this case, the only way to increase revenue is to sacrifice efficiency. As discussed previously, this is the standard model of bidders in auction design. We learn revenue-maximizing auctions within the class of $\boldsymbol{f}$-VCG auctions subject to a efficiency constraint: let $\mathrm{OPT}_i^{\boldsymbol{f}}(\pi)$ denote the optimal payment $\mathbb{E}_{\boldsymbol{v}\sim D}[\mathsf{pay}_i^{\boldsymbol{f}}] = \mathbb{E}_{\boldsymbol{v}\sim D}[p_i^{\boldsymbol{f}} \cdot \mathbf{1}[o_i^{\boldsymbol{f}}(\boldsymbol{v}) \le 0]]$ of any $\boldsymbol{f}$-VCG auction such that $\Pr_{\boldsymbol{v}\sim D}[o_i^{\boldsymbol{f}}(\boldsymbol{v}) < 0] < 1 - \pi$ (so π is the probability bidder i is sold her winning bundle in the efficient allocation). We no longer need to consider a κ-competitor since there are no IR violations.

Theorem 8.3.4. *Assume bidders do not accept overcharges. Given* $V = \{\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(K)}\}$ *an IID dataset, define the following* $\boldsymbol{f}$*-VCG auction:* $f_i(\boldsymbol{v}_{-i})$ *outputs*

$$
\begin{aligned}
\underset{\widetilde{v}_i \in V_i}{\operatorname{argmax}} \; & \tfrac{1}{K} \textstyle\sum_{\ell=1}^K p_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i}) \mathbf{1}[o_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i}) \le 0] \\
\textit{s.t.} \quad & \tfrac{|\{\ell : o_i^{\widetilde{v}_i}(v_i^{(\ell)}, \boldsymbol{v}_{-i}) > 0\}|}{K} \le 1 - \pi + \varepsilon(K, \delta/n).
\end{aligned}
$$

The resulting auction is IR and, with probability $\ge 1 - \delta$ *over the draw of* V*, for all bidders* i*:* $\mathbb{E}_{\boldsymbol{v}\sim D}[\mathsf{pay}_i^{\boldsymbol{f}}(\boldsymbol{v})] \ge \mathrm{OPT}_i^{\boldsymbol{f}}(\pi) - 2\varepsilon(K,\delta/n)$ *and* i *is sold her winning bundle in the efficient allocation with probability* $\ge \pi - 2\varepsilon(K,\delta)$.

Setting $\pi = 0$ corresponds to "pure" revenue maximization subject to IC and IR within the class of $\boldsymbol{f}$-VCG auctions with no other constraints. On the other hand, $\pi = 0.99$ corresponds to the revenue-maximizing $\boldsymbol{f}$-VCG auction that retains each bidder with probability at least 99%.

Challenges posed by correlation in bidder types The assumption of independent bidder types is critical to the above empirical payment maximization algorithms; each V_i is independent and consists of IID draws of bidder i's type and therefore we can optimize over the dataset $(v_i^{(1)}, \boldsymbol{v}_{-i}), \dots, (v_i^{(K)}, \boldsymbol{v}_{-i})$ for each bidder without introducing any correlation. Without independence, the dataset can be completely uninformative about bidder i's test-time type. To illustrate, consider the extreme scenario where the type space for bidder i implied by the test-time revealed types of all other bidders $\boldsymbol{v}_{-i}$ is completely disjoint from the samples, that is, $\Theta_i(\boldsymbol{v}_{-i}) \cap V_i = \emptyset$. Then, V_i gives the auction designer absolutely no information about the conditional distribution over v_i given $\boldsymbol{v}_{-i}$. Tackling this challenge, possibly via out-of-distribution learning [Ben-David et al., 2010], is a compelling direction for future work since most real-world settings involve correlation.

8.3.2 Computational Considerations

A feature of the $\boldsymbol{f}$-VCG auctions above is that the competitor $f_i(\boldsymbol{v}_{-i})$ is determined via a search over the set V_i of historical bids for i and the κ-competitor. Furthermore they are sample efficient: the number of samples required to meet a prescribed error bound ε is $O(\frac{H^2}{\varepsilon^2}(|B_i|\log|B_i| + \ln(n/\delta)))$. This is in stark contrast with other combinatorial auction formats (*e.g.,* affine maximizer auctions [Roberts, 1979]) for which empirical revenue maximization requires exponentially many samples and is computationally intractable [Balcan et al., 2025d] (an approach via hyperplane arrangements has been explored in some restricted settings [Balcan et al., 2021c, 2022a]).

We determine the computational complexity of our learning algorithms given a *winner-determination oracle* that on input $\boldsymbol{v}$ outputs $w(\boldsymbol{v})$ and the efficient allocation $\boldsymbol{S}^*$. Winner determination is NP-complete but can be efficiently implemented in practice via custom search algorithms [Sandholm et al., 2005, Sandholm, 2006] or by integer programming. First, assuming type spaces described by linear constraints, we show how to compute a κ-competitor.

Theorem 8.3.5. *Given as input $\boldsymbol{v}_{-i}$ and a polynomial number of linear constraints defining $\Theta_i(\boldsymbol{v}_{-i})$, a κ-competitor v_i^κ with $w(v_i^\kappa, \boldsymbol{v}_{-i}) = \kappa + \min_{\widetilde{v}_i \in \Theta_i(\boldsymbol{v}_{-i})} w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ can be computed with a polynomial number of calls to a winner-determination oracle and additional polynomial run time.*

Proof. In Chapters 5 and 6 we gave linear programs to compute *weakest types* (zero-competitors in our terminology). The LP enumerates all feasible allocations Γ in its constraint set. A separation oracle for that LP can be implemented with a single call to a winner determination oracle. So the weakest type $\widetilde{v}_i$ that minimizes $w(\widetilde{v}_i, \boldsymbol{v}_{-i})$ can be computed via the ellipsoid algorithm [Grotschel et al., 1993]. Extending to a κ-competitor is straightforward. □

To find the empirical-payment-maximizing competitor $f_i(\boldsymbol{v}_{-i})$ one needs to call the winner determination oracle to compute $w(v_i^{(\ell)}, \boldsymbol{v}_{-i})$ for each $v_i^{(\ell)} \in V_i$.

Theorem 8.3.6. *The competitor $f_i(\boldsymbol{v}_{-i})$ in Theorems 8.3.3 and 8.3.4 can be computed with polynomial calls to a winner determination oracle and additional polynomial run time.*

Finally, observe that the $\boldsymbol{f}$-VCG auction computation can be parallelized across (independent) bidders. The empirical-payment-maximization algorithm to compute $f_i(\boldsymbol{v}_{-i})$ for different bidders uses completely disjoint portions of the dataset, and is an independent computation for each bidder. This has not been the case even in modern approaches to data-driven auction design via, for example, deep learning [Dütting et al., 2019, Curry et al., 2023, Duan et al., 2023].

8.4 Conclusions and Future Research

We showed how to inject artificial competition into combinatorial auctions to accomplish greater revenue when efficiency is a constraint of the auction design. While the weakest-type VCG mechanism of Balcan et al. [2023] (Krishna and Perry [1998]) poses a revenue barrier for efficient, IC, and IR (B-IR) auctions, we showed that under a relaxed participation model for bidders we can nonetheless make fruitful progress. We derived the revenue optimal auction subject to efficiency, IC, and a relaxed notion of IR that involved auctioneer-set caps on overcharge frequency and magnitude, for different auctioneer knowledge models. Our new auction class, $\boldsymbol{f}$-VCG auctions, provided a unified language of artificial competition and contained the revenue optimal auctions in all the above settings. Finally, we gave sample and computationally efficient *instance-adaptive* learning algorithms that parallelize across bidders in a data-driven auction design setting.

There are a number of important theoretical and practical extensions needed to develop a more complete landscape of competitive efficient auctions. First, extensions of and more realistic versions of our bidder participation model are needed. While our (stylized) model takes a first step towards understanding how a bidder would respond to competitive prices, a more nuanced model that ties together bidder uncertainty, rationality, and attitudes towards risk is needed. Consolidation with the results of Chapter 7 [Prasad et al., 2025a] is also important. Finally, the broader idea of auction parameter optimization that *separates* across bidders and uses the revealed types of other bidders merits deeper investigation. Current combinatorial auctions do not have this property and a more thorough understanding of when it can be exploited might lead to new and better designs.

Part III

Other Models of Learning for Mechanism Design

Chapter 9

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 [1971] 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 [1941], 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* Conitzer and Sandholm [2002]. 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 [Likhodedov and Sandholm, 2004, 2005, Sandholm and Likhodedov, 2015]. Balcan et al. [2018d] 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. 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. Our algorithms also have the obvious practical uses in designing TPTs and menus thereof.

9.1 Problem Formulation

In our model, the seller has K units of a good to sell among n buyers $j \in \{1, \ldots, n\}$ via a menu of TPTs. Each buyer is described by his valuation function $v_j : \{1, \ldots, K\} \to \mathbb{R}$ over the K units. So, $v_j(q)$ is the value that buyer j assigns to getting q units of the item. (We implicitly assume that each buyer's value for getting nothing is zero.) We assume that buyers act in a utility maximizing manner: when presented with a menu $((p_1^1, p_2^1), \ldots, (p_1^L, p_2^L))$ of TPTs, buyer j with valuation function $v_j : \{1, \ldots, K\} \to \mathbb{R}$ will choose to buy q units priced by tariff r to maximize $v_j(q) - (p_1^r + q \cdot p_2^r)$, buying 0 units if there are no values of q and r that make the above expression non-negative. Given one sample of buyers $v = (v_1, \ldots, v_n)$, when faced with menu $\boldsymbol{p}$, say buyer j purchases quantity q_j of tariff r_j. Then the revenue of $\boldsymbol{p}$ with respect to v, denoted $\mathsf{Rev}_v(\boldsymbol{p})$, is $\sum_{j=1}^n \mathbf{1}(q_j \geq 1) \cdot (p_1^{r_j} + q_j \cdot p_2^{r_j})$.[1]

However, the model above allows for the possibility that the total quantity $\sum_{j=1}^n q_j$ is larger than K. To deal with this issue, we will usually stipulate that the seller offers a menu $\boldsymbol{p}$ that is *feasible* (that is, the total quantity purchased is at most K) for each sample he sees (and we show that doing so ensures with high probability that the menu $\boldsymbol{p}$ is also feasible on a freshly drawn potential future sample).

We also study the case where each buyer belongs to one of M markets, in which case a TPT pricing scheme is of the form $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)$, where buyers in market m are offered menu $\boldsymbol{p}_m$. Revenue is defined similarly, which we denote by $\mathsf{Rev}_v(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)$. For a set of samples $S = \{v^1, \ldots, v^N\}$, the empirical revenue of $\boldsymbol{p}$ with respect to S is denoted by $\widehat{\mathsf{Rev}}_S(\boldsymbol{p}) = \frac{1}{N} \sum_{i=1}^N \mathsf{Rev}_{v^i}(\boldsymbol{p})$, and similarly for the market-segmented case.

We now state the formal generalization guarantee of Balcan et al. [2018d] for the mechanism class of length-L menus of TPTs for selling K units to n buyers partitioned into M markets. Let $\mathcal{D}$ be some unknown distribution over n-tuples of buyer valuations and markets. For any $0 < \varepsilon, \delta < 1$, there exists an $N_{TPT}(\varepsilon, \delta) \in \mathbb{N}$ such that for all $N \geq N_{TPT}(\varepsilon, \delta)$, it holds with probability at least $1 - \delta$ over the draw of $S = \{v^1, \ldots, v^N\} \sim \mathcal{D}^N$ that for every M-tuple $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)$ of length L menus of TPTs,

$$|\widehat{\mathsf{Rev}}_S(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M) - \mathop{\mathbb{E}}_{v \sim \mathcal{D}}[\mathsf{Rev}_v(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m)| \leq \varepsilon.$$

The sample complexity $N_{TPT}(\varepsilon, \delta)$ is at most $O_{\varepsilon,\delta}(ML \log(nKL))$, where we have hidden the dependence on ε and δ as is typical in learning theory. This follows from the piecewise structure of the class of revenue functions: there is a partition of the TPT parameter space $\mathbb{R}^{2LM}$ by hyperplanes into not-too-numerous regions such that empirical revenue is linear over each region (this notion is formalized in the main result of Balcan et al. [2018d]).

[1]We use boldface $\boldsymbol{p} \in \mathbb{R}^{2L}$ to abbreviate a menu of L TPTs. It is understood, then, that p_1^r and p_2^r denote $\boldsymbol{p}_{2r-1}$ and $\boldsymbol{p}_{2r}$, respectively.

The overarching goal of our paper is to efficiently find TPT pricing schemes that maximize empirical revenue over a set of samples—which by the above uniform convergence result is highly likely to be nearly optimal in terms of expected revenue as well. The number of samples needed to guarantee generalization only depends at most linearly in the problem parameters, so computationally efficient algorithms for empirical revenue maximization in this setting will be sample efficient as well.

Summary of contributions

In Section 9.2 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 $L = 1$ in the single buyer setting, that generalizes to an $O(n^3N^3K^3)$ algorithm in the multi-buyer setting (Section 9.2.1). We then give an $(NK)^{O(L)}$ algorithm for the setting where $L \geq 1$ (Section 9.2.2). 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.

In Section 9.3 we 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 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.

Additional Related Research

(Menus of) two-part tariffs have been studied in economics [Feldstein, 1972, Ng and Weisser, 1974, Leland and Meyer, 1976, Murphy, 1977, Maskin and Riley, 1984, Wilson, 1993, Armstrong and Vickers, 2001, Sundararajan, 2004, Shi et al., 2009]. The approach taken by much of the economic literature on this topic is rather different from the perspective we pursue: most work aims to find closed-form solutions for revenue maximizing two-part tariff menus, and in attempting to do so often places various (strong) restrictions on the setting. For example, Kolay and Shaffer [2003] derive closed forms for the profit-maximizing length-two menu of two-part tariffs when there are exactly two types of buyers. Bagh and Bhargava [2013] derive further closed-form results when valuations come from a finite discrete distribution. They moreover consider three-part tariffs—which has an additional quantity allowance after which the per-unit price takes effect. Schlereth et al. [2010] study some algorithmic aspects of finding revenue-maximizing TPTs. They cast the revenue-maximization problem as a mixed integer linear program and compare the performance of a few different heuristic solution algorithms. We can also write a mixed integer linear program to solve revenue maximization in our setting, but the algorithms we pose are more efficient. Other works consider two-part tariff pricing in relation to, for example, uncertainty [Lambrecht et al., 2007, Png and Wang, 2010], opportunism [Marx and

Shaffer, 2004], and other practical buyer behavior [Narayanan et al., 2007, Iyengar et al., 2008]. To our knowledge, all prior work in economics considers continuous models, where quantity purchased is a continuous parameter and valuations are continuous and differentiable functions of quantity. Our setting considers a discrete and finite model, which is what gives rise to the interesting algorithmic challenges we tackle. In addition, the various examples of TPT pricing in the real world previously mentioned involve discrete quantities of goods, so our model is arguably a more realistic description of TPT pricing. TPTs have received some recent attention in computer science as well. Chawla and Miller [2016] study a form of TPTs (that is different from ours) in the context of finding simple mechanisms that yield (multiplicative) approximations to optimal revenue. However, they assume that the seller knows the distribution over buyers' values, and the mechanism design is tuned to that distribution. Notions of menu complexity and market segmentation have also been studied by computer scientists, though in different contexts [Babaioff et al., 2017, Hart and Nisan, 2019, Cummings et al., 2020].

The only prior work that studied the model of TPTs that we address is that of Balcan et al. [2018d]. However, they only studied sample complexity rather than algorithms. We take this a step further and solve the learning problem efficiently in terms of computation.

Subsequent work Since the publication of the work covered in this chapter in IJCAI 2020, the techniques introduced here have been improved upon and extended to obtain algorithms that are more computationally efficient and sample efficient [Balcan et al., 2024, Balcan and Beyhaghi, 2024].

9.2 Algorithms for Optimal TPT Structures

In this section we study the computation of TPT structures that maximize empirical revenue over the given set of samples. We are given a set of samples $S = \{v^1, \ldots, v^N\}$, where each sample $v^i = (v_1^i(1), \ldots, v_1^i(K)), \ldots, (v_n^i(1), \ldots, v_n^i(K))$. That is, each sample gives a value for each buyer for each number of units bought. In the sample-based mechanism design literature, it is standard to assume a complete valuation draw like this in each sample. We also use the shorthand $v_j^i = (v_j^i(1), \ldots, v_j^i(K))$. In the first subsection we discuss computation of a single TPT and in the next subsection computation of a menu of multiple TPTs. In both sections we discuss the single-buyer case for simplicity, and then in the third subsection we present the generalization to the multi-buyer case.

9.2.1 An Efficient Algorithm for a Single TPT

In this subsection we give a polynomial-time algorithm to solve the empirical revenue maximization problem in the case where we can offer only one two-part tariff, that is, the menu length $L = 1$. Because in this section we are presenting the single-buyer case for simplicity, we do not include the buyer subscript in the valuations. So, our input is $S = \{v^1, \ldots, v^N\} = \{(v^1(1), \ldots, v^1(K)), \ldots, (v^N(1), \ldots, v^N(K))\}$.

We observe the following, which is key for our algorithm.

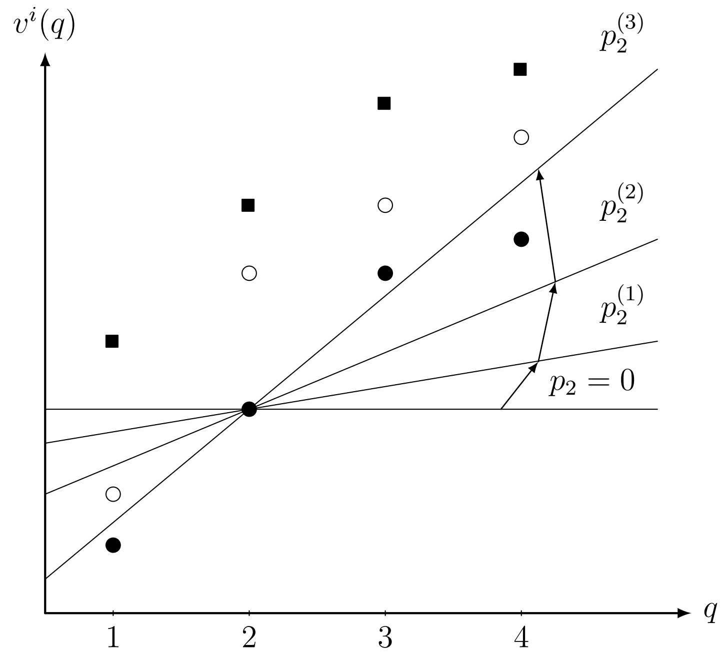


Figure 9.1: Three iterations of the single tariff algorithm from a given hinge point. The points displayed represent the valuations of three buyers (differentiated by the rendering style of the points) over four units. If, for example, $p_2', p_2'' \in (p_2^{(1)}, p_2^{(2)})$, then the quantities purchased by each buyer remain the same for the tariffs with slopes p_2' and p_2'' hinged at the given point.

Lemma 9.2.1. *Suppose (p_1, p_2) is a TPT that maximizes empirical revenue over S. Then, the line with y-intercept p_1 and slope p_2 passes through a point $(q, v^i(q))$ for some $i \in \{1, \ldots, N\}$ and $q \in \{1, \ldots, K\}$.*

Proof. Consider a TPT line with y-intercept p_1 and slope p_2 that does not pass through any such point. Let $d > 0$ be minimal such that the TPT line with y-intercept $p_1 + d$ and slope p_2 passes through such a point. For any buyer j,

$$\operatorname*{argmax}_q v^j(q) - (p_1 + p_2 q) = \operatorname*{argmax}_q v^j(q) - (p_1 + d + p_2 q),$$

so any buyer who was purchasing a nonzero quantity q when faced with (p_1, p_2) continues to purchase quantity q when faced with $(p_1 + d, p_2)$. Any buyer who purchased nothing when faced with (p_1, p_2) clearly continues to purchase nothing. Hence the TPT $(p_1 + d, p_2)$ generates strictly more revenue than (p_1, p_2). $\square$

Now, for each point $(q, v^i(q))$, we demonstrate that to find the revenue-maximizing tariff line it suffices to search over a set of at most NK tariff lines passing through $(q, v^i(q))$. Suppose we have a tariff line passing through $(q_0, v^{i_0}(q_0))$, at which buyer i buys quantity q_i. Let us compute the slope p_2 of such a tariff line at which i prefers to buy quantity q over q_i. We have

$$v^i(q) - (v^{i_0}(q_0) + p_2(q - q_0)) \geq v^i(q_i) - (v^{i_0}(q_0) + p_2(q_i - q_0)) \iff p_2 \geq \frac{v^i(q_i) - v^i(q)}{q_i - q}.$$

Algorithm 4 Single TPT for a Single Buyer

Input: Set of samples $S = \{v^1, \ldots, v^N\}$
Output: TPT (p_1, p_2) maximizing empirical revenue over S

Initialize $\text{MaxRev} \leftarrow 0$.
for $(i, q) \in \{1, \ldots, N\} \times \{1, \ldots, K\}$ **do**
  $p_2 \leftarrow 0$
  for $j \in \{1, \ldots, N\}$ **do**
    if $v^j(K) \geq v^i(q)$ **then**
      $q_j \leftarrow K$ ▷ This makes the typical assumption that each buyer's valuation is nondecreasing in quantity. If that is not the case, one can in this line simply loop over quantities for each buyer separately in order to find the utility-maximizing quantity for that buyer.
    else
      $q_j \leftarrow 0$
  repeat
    $p_2 \leftarrow \min_{j,q'} \alpha_{i,q}(j, q_j, q')$
    $(j^*, q^*) \leftarrow \arg\min_{j,q'} \alpha_{i,q}(j, q_j, q')$
    $q_{j^*} \leftarrow q^*$
    $p_1 \leftarrow v^i(q) - q \cdot p_2$
    $\text{MaxRev} \leftarrow \max\{\text{MaxRev}, \widehat{\mathsf{Rev}}_S(p_1, p_2)\}$
  until no update is found
return revenue maximizing (p_1, p_2)

For i, q, q' let $\alpha_{i_0,q_0}(i, q, q') = \frac{v^i(q) - v^i(q')}{q - q'}$ denote the slope where buyer i switches preferences between quantity q and q' (this has no dependence on i_0 and q_0, but for notational consistency we leave the subscript). We must also account for the case where buyer i switches preferences between quantity q and purchasing nothing, for which a similar calculation gives $\alpha_{i_0,q_0}(i, q, 0) = \frac{v^i(q) - v^{i_0}(q_0)}{q - q_0}$.

The algorithm works by "hinging" a TPT at every point of the form $(q, v^i(q))$, and then increases the slope of the tariff line at increments determined by the α values—these increments represent thresholds at which some buyer starts buying a different quantity. As the slope of the hinged tariff line increases, the quantity purchased by a given buyer changes at most K times, and hence there are at most NK slope thresholds to check—between any two thresholds the average revenue is linear in the TPT since the quantities purchased are constant, and so the maximum revenue must be attained at one of these thresholds. See Figure 9.1 for an illustration. We write down the precise algorithm as Algorithm 4.

Theorem 9.2.2. *Algorithm 4 finds the single tariff (p_1, p_2) that maximizes empirical revenue over a sample set of size N in $O(N^3K^3)$ time.*

Proof. That the algorithm finds the maximum revenue TPT follows from the fact that if $p_2 < p_2'$ are two consecutive slopes checked by the algorithm, the average revenue is linear as the slope varies between p_2 and p'_2 (revenue is linear since, by construction, the quantities purchased by each buyer are constant for slopes varying between p_2 and p'_2) and hence it suffices to compute revenue at the endpoints. Lemma 9.2.1 shows that the empirical revenue maximizer passes through a point $(q, v^i(q))$, and since Algorithm 4 checks all slope thresholds for TPTs passing through each such point, we are guaranteed to find the TPT yielding the maximum empirical revenue.

We now count the number of steps taken by Algorithm 4. Line 2 involves NK iterations, Line 4 involves N iterations, Line 9 involves at most NK iterations (since each buyer starts by purchasing K units, and can change quantities at most K times as the slope p_2 increases), computing the minimization in Lines 10 and 11 requires at most NK steps, and computing revenue in Line 14 requires N steps. So the total run time is $O(NK(N + NK(NK + N))) = O(N^3K^3)$. □

We ran Gurobi (the fastest general-purpose mixed integer program solver) to find the revenue-maximizing single TPT for a single buyer (after formulating this problem as an integer program), and Algorithm 4 beat it dramatically. For example, averaged over 10 runs on randomly generated instances with $K = 5$ units and $N = 600$ samples, our algorithm returned the revenue-maximizing TPT in under 23 minutes while Gurobi took over 3.5 hours.

9.2.2 An Algorithm for Multiple TPTs

In this subsection we give an algorithm for optimizing a menu of two-part tariffs. In most applications, for practical reasons, the length of the menu cannot be very long, so L is a small constant (typically 2 or 3). We present an algorithm that is exponential in L but still polynomial in N and K. It can be viewed as a generalization of the single tariff algorithm. The geometric structure of the problem is the same as in Balcan et al. [2018d], but we exploit it to get algorithms while they use it to prove sample complexity bounds.

Theorem 9.2.3. *There is an algorithm that finds the empirical revenue maximizing length L menu of tariffs over a sample set of size N in $(NK)^{O(L)}$ time.*

Proof. For input valuations $v^1, \ldots, v^N$, let $H_i(q, q', r, r')$ denote the hyperplane

$$v^i(q) - (p_1^r + q \cdot p_2^r) = v^i(q') - (p_1^{r'} + q' \cdot p_2^{r'}),$$

where if q (or q') is 0, the LHS (or RHS) is replaced by 0. Consider the collection of hyperplanes $\mathcal{H}$ consisting of these (at most $N(KL)^2$) hyperplanes for each $q, q' \in \{0, \ldots, K\}$, $r, r' \in \{1, \ldots, L\}$, $i \in \{1, \ldots, N\}$. It is a basic combinatorial fact that $\mathcal{H}$ partitions $\mathbb{R}^{2L}$ into at most $|\mathcal{H}|^{2L} \leq N^{2L}K^{4L}$ regions (each region is a connected component of

$$\mathbb{R}^{2L} \setminus \cup_{i,q,q',r,r'} H_i(q, q', r, r')$$

and is an intersection of at most $|\mathcal{H}|$ halfspaces). The average revenue over the set of samples is linear within each such region, since the quantity purchased by each buyer remains constant within each region, so the maximum revenue within a region C can be found by solving the following linear program: if buyer i purchases quantity $q_i(C) \in \{0, \ldots, K\}$ of tariff $r_i(C) \in \{1, \ldots, L\}$ within C, the maximum revenue in C is

$$\max_{\boldsymbol{p} \in C} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(q_i(C) \geq 1) \cdot (p_1^{r_i(C)} + q_i(C) \cdot p_2^{r_i(C)}).$$

Each linear program involves $2L$ real variables, and $|\mathcal{H}| \leq NK^2$ constraints. So, it can be solved in $poly(N, K, L)$ time.

Moreover, there is a simple algorithm with run time $poly(N^{2L}K^{4L})$ which outputs a representation of each region determined by $\mathcal{H}$ as a $0/1$ vector of length $|\mathcal{H}|$, where the kth entry determines on which side of the kth hyperplane of $\mathcal{H}$ the region lies on. The high-level idea is to sequentially add each hyperplane to the list of regions maintained so far (starting with the entire Euclidean space $\mathbb{R}^{2L}$), iterating over the current regions and checking whether the added hyperplane intersects each region—updating the list of regions if so. See Xu [2020] for a more detailed description of enumerating the regions formed by a collection of hyperplanes (in a totally different context). Our algorithm solves the aforementioned linear program for every such region and picks the solution that yields highest empirical revenue. □

In the single TPT ($L = 1$) case, Algorithm 4 is more efficient than the algorithm presented in Theorem 9.2.3. This is because the former is a direct combinatorial algorithm that does not require solving LPs.

9.2.3 Generalization to Multiple Buyers

While our algorithms in the two subsections above were presented in the single buyer setting, they directly extend to the multi-buyer setting as follows. Algorithm 4 generalizes by feeding in not just N valuations but all nN valuations. The only change to the algorithm itself is that in Step 12 we check that the allocation is feasible, that is, not more than the total quantity K is bought among the buyers in any sample $i \in \{1, \ldots, N\}$; if more is bought, then the assignment on that line is skipped because that pricing solution is infeasible. (Another nuance is that the initialization in Lines 2–8 might not be feasible, but that is fine.)

Similarly, in our multi-TPT algorithm, whenever we are about to solve an LP corresponding to some region determined by the set of hyperplanes, we first check that the region is feasible in the sense that the total quantity bought by buyers in any one sample is at most K.

Remark. Suppose buyers have additive valuations, that is, $v^i(q_1 + q_2) = v^i(q_1) + v^i(q_2)$ for any quantities $q_1 + q_2 \leq K$. Then, the revenue maximization problems considered in this section become trivial. In particular, the run time dependence on K and L vanishes. This can be seen due to the fact that $\text{price}(q) := \min_r p_1^r + q \cdot p_2^r$ is a piece-wise linear increasing concave function. An additive buyer's valuation function is simply a line with positive slope passing through the origin, as $v^i(q) = v^i(1 + \cdots + 1) = q \cdot v^i(1)$. Hence, the difference $v^i(q) - \text{price}(q)$ is always maximized when $q = K$, that is, buyers are always only interested in the entire K-unit bundle. Thus, revenue is determined by a single price, that of the entire bundle, and the seller simply can try every possible price in $\{v^1(K), \ldots, v^N(K)\}$, due to Lemma 9.2.1.

9.3 Market Segmentation

We now consider a setting in which each buyer belongs to one of M markets $\mathcal{X}_1, \dots, \mathcal{X}_M$—determined by attributes such as geographic location, income level, etc. The seller sets M length L-menus of TPTs $(\boldsymbol{p}_1, \dots, \boldsymbol{p}_M)$, where buyers in market m are allowed to purchase according to $\boldsymbol{p}_m$.

The seller wants to offer a TPT menu for each market so that the overall solution across markets is feasible, that is, that the sum of the demands of the markets does not exceed K. We show that any solution that is feasible for each sample in a large enough sample set is with high probability a feasible solution for any future sample.

Proposition 9.3.1. *Let $N \geq N_{TPT}(\varepsilon, \delta)$. With probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^N$, if $\boldsymbol{p}_1, \dots, \boldsymbol{p}_M$ is feasible for S,*

$$\Pr_{v \sim \mathcal{D}}[(\boldsymbol{p}_1, \dots, \boldsymbol{p}_M) \textit{ is feasible for } v] \geq 1 - \varepsilon.$$

Proof. Consider the class of $0/1$ valued indicator functions $\{f_v(\boldsymbol{p}_1, \dots, \boldsymbol{p}_M)\}$ indicating whether $(\boldsymbol{p}_1, \dots, \boldsymbol{p}_M)$ is feasible for v. For a single sample i, consider the set of hyperplanes of the form

$$v_j^i(q) - (p_{1,m(i,j)}^r + q \cdot p_{2,m(i,j)}^r) = v_j^i(q') - (p_{1,m(i,j)}^{r'} + q' \cdot p_{2,m(i,j)}^{r'})$$

for each j, q, q', r, r', where $m(i,j)$ is the market to which buyer j in sample i belongs. These hyperplanes partition the tariff space $\mathbb{R}^{2LM}$ into at most $(n(KL)^2)^{2LM}$ regions such that the indicator is constant within each region. Thus, by Balcan et al. [2018d], for a sample set S of size at least $N_{TPT}(\varepsilon, \delta)$, it holds with probability at least $1 - \delta$ that for all $\boldsymbol{p}_1, \dots, \boldsymbol{p}_M$,

$$\left| \widehat{f}_S(\boldsymbol{p}_1, \dots, \boldsymbol{p}_M) - \mathop{\mathbb{E}}_{v \sim \mathcal{D}}[f_v(\boldsymbol{p}_1, \dots, \boldsymbol{p}_M) \right| \leq \varepsilon.$$

So, if $(\boldsymbol{p}_1^S, \dots, \boldsymbol{p}_M^S)$ is any feasible solution for S, that is, $\widehat{f}_S(\boldsymbol{p}_1^S, \dots, \boldsymbol{p}_M^S) = 1$, we have

$$\mathbb{E}[f_v(\boldsymbol{p}_1^S, \dots, \boldsymbol{p}_M^S)] = \Pr[f_v(\boldsymbol{p}_1^S, \dots, \boldsymbol{p}_M^S) = 1] \geq 1 - \varepsilon,$$

with probability at least $1 - \delta$ over the draw of S and v. □

It turns out that market segmentation introduces substantial computational hurdles to revenue maximization. Even when bidders are additive (which removes the parameters L and K from the problem as remarked in the previous section), the problem of setting a feasible price for each market in an empirical revenue maximizing way is NP hard. Since additive buyers either purchase the entire bundle of K units or nothing, each menu is reduced to a single price, so the seller's problem is to set prices $p_1, \dots, p_M$ for each market. Any solution must be feasible for the set of samples, which means at most one buyer can purchase the full K units in each sample.

Theorem 9.3.2. *Consider a set of samples $S = \{v^1, \dots, v^N\}$ where each buyer belongs to one of M markets. Even if all buyers have additive valuations, there is no algorithm that finds feasible prices $p_1, \dots, p_M$ that maximize empirical revenue over S in time polynomial in M and N, unless $\mathsf{P} = \mathsf{NP}$.*

Proof. We reduce from Maximum Weight Independent Set. Given an instance $G = (V, E)$ of Maximum Weight Independent Set (without loss of generality assume G has no isolated vertices), label the vertices $V = \{v_1, \ldots, v_n\}$, and let $w_i = \text{weight}(v_i)$. Let $p_i = \frac{w_i |E|}{\deg(v_i)}$. We will have n markets $\mathcal{X}_1, \ldots, \mathcal{X}_n$, corresponding to the vertices of G.

For each $(v_i, v_j) \in E$, we introduce a sample consisting of a buyer in market $\mathcal{X}_i$ with value p_i and a buyer in market $\mathcal{X}_j$ with value p_j (ensuring that no feasible pricing solution can simultaneously offer p_i to market $\mathcal{X}_i$ and p_j to market $\mathcal{X}_j$). So we have a total of $|E|$ samples. Clearly, any feasible revenue maximizing solution involves offering market $\mathcal{X}_i$ either price p_i, or something higher than p_i (so that no buyer in market $\mathcal{X}_i$ across any of the samples makes a purchase).

Our construction yields a one-to-one correspondence between independent sets in G and feasible n-tuples of prices: an independent set $I \subseteq V$ with weight $W = \sum_{v_i \in I} w_i$ corresponds to a pricing solution where if $v_i \in I$, market $\mathcal{X}_i$ is offered p_i, and if $v_i \notin I$, market $\mathcal{X}_i$ is offered something higher than p_i. For a vertex v_i in the independent set, there are precisely $\deg(v_i)$ samples containing a buyer in market $\mathcal{X}_i$ who makes a purchase at price p_i, so the average revenue obtained by the pricing solution corresponding to I is

$$\frac{1}{|E|} \sum_{v_i \in I} \deg(v_i) \cdot p_i = W,$$

by the choice of p_i. This completes the (clearly polynomial time) reduction. □

Remark. This hardness is inherent to the limited supply setting. If the seller has unlimited supply, and K is instead the maximum quantity any buyer is willing to purchase, we can find the empirical revenue maximizing market-segmented solution in $M(nNK)^{O(L)}$ time simply by running the procedure described in the previous section restricted to each market in turn. This finds the empirical optimum over each market, and without capacity constraints, this is a feasible solution and thus the optimal market-segmented solution as well.

9.3.1 Buyers with Identically Distributed Valuations

To circumvent the hardness of feasible empirical revenue maximization over worst case instances, we now study a setting where each buyer's valuation vector is drawn from the same distribution. Each market is of a certain prescribed size, and buyers are indistinguishable across markets. An example of a natural real-world market segmentation that potentially satisfies this is segmentation based on geographic location. For example, there may be no reason to believe that the average buyer in San Francisco values a gym-membership plan any differently than the average buyer in Pittsburgh. We show that under certain conditions, it is optimal to treat every buyer equally—regardless of whether they come from a large market or a small market. This immediately yields a simple algorithm in which we solve the non-segmented version of the problem, and reuse the solution for the segmented version.

Suppose there are a total of n buyers across markets, and an α_m fraction of these buyers belong to market $\mathcal{X}_m$. For simplicity, we assume that the seller receives zero revenue on instances on which the chosen solution is infeasible. Suppose the optimal solution $(\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_M^*) = \text{argmax}_{(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)} \mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)]$ satisfies the property that in expectation, buyers from market $\mathcal{X}_m$ contribute an α_m fraction of the total revenue. That is, $\mathbb{E}_v[\mathsf{Rev}_{v|\mathcal{X}_m}(\boldsymbol{p}_m^*)] = \alpha_m \cdot \mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_M^*)]$ for each m. In this case, we can reuse the non-segmented solution. For a randomly drawn v, let $\mathcal{F}$ denote the event that $f_v(\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_M^*) = 1$. We have that

$$\begin{aligned}
\mathop{\mathbb{E}}_v[\mathsf{Rev}_{v|\mathcal{X}_m}(\boldsymbol{p}_m^*)] &= \mathop{\mathbb{E}}_v\left[\mathsf{Rev}_{v|\mathcal{X}_m}(\boldsymbol{p}_m^*) \mid \mathcal{F}\right] \cdot \Pr_v[\mathcal{F}] \\
&= \mathop{\mathbb{E}}_v[\textstyle\sum_{v_m \in \mathcal{X}_m} \mathsf{Rev}_{v_m}(\boldsymbol{p}_m^*) \mid \mathcal{F}] \cdot \Pr_v[\mathcal{F}] \\
&= \alpha_m n \mathop{\mathbb{E}}_v[\mathsf{Rev}_{v_m}(\boldsymbol{p}_m^*) \mid \mathcal{F}] \cdot \Pr_v[\mathcal{F}],
\end{aligned}$$

so $\mathbb{E}_v[\mathsf{Rev}_{v_m}(\boldsymbol{p}_m^*) \mid \mathcal{F}] = \frac{1}{n}\mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_M^*) \mid \mathcal{F}]$ for each m. Thus, we can set $\boldsymbol{p}_1^* = \cdots = \boldsymbol{p}_m^*$, and hence we only need to search for an optimal solution in the non-segmented case that we then offer to every market. The empirical revenue maximizing menu of TPTs $\boldsymbol{p}$ in the non-segmented case can be computed in $(nNK)^{O(L)}$ time, as in the previous section. Finally, we provide the generalization guarantee for using the unsegmented solution in the market-segmented case.

Theorem 9.3.3. *Let $N \geq N_{TPT}(\varepsilon, \delta)$. In the above setting, with probability at least $1 - \delta$ over the draw of $S \sim \mathcal{D}^N$,*

$$\left|\widehat{\mathsf{Rev}}_S(\boldsymbol{p}, \ldots, \boldsymbol{p}) - \max_{(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)} \mathop{\mathbb{E}}_v[\mathsf{Rev}_v(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M)]\right| \leq 2\varepsilon,$$

where $\boldsymbol{p}$ is the empirical revenue maximizing solution when all the markets are combined.

Proof. Let $\boldsymbol{p}^*$ denote the expected revenue maximizer in the non-segmented case, so by the previous discussion, $(\boldsymbol{p}^*, \ldots, \boldsymbol{p}^*)$ is also the expected revenue maximizer in the segmented case. By Balcan et al. [2018d], $|\widehat{\mathsf{Rev}}_S(\boldsymbol{p}) - \mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p})]|, |\widehat{\mathsf{Rev}}_S(\boldsymbol{p}^*) - \mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p}^*)| \leq \varepsilon$ with probability at least $1 - \delta$ over the draw of S. As $\widehat{\mathsf{Rev}}_S(\boldsymbol{p}) \geq \widehat{\mathsf{Rev}}_S(\boldsymbol{p}^*)$ and $\mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p}^*)] \geq \mathbb{E}_v[\mathsf{Rev}_v(\boldsymbol{p})]$, applying the triangle inequality yields the result. □

Chapter 10

Within-Instance Learning for Auction Design

In this chapter we present some new approaches to the elusive problem of designing high-revenue (limited supply) multi-item, multi-bidder auctions when no additional information is available. While Chapter 9 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 10.1, 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 10.2, 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?

10.1 Learning Within an Instance for Designing High-Revenue Combinatorial Auctions

The setting here is a limited-supply combinatorial auction where a seller has m indivisible items to allocate among a set S of n bidders—this is the same setup as in Chapter 6 of this thesis.

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 [Goldberg et al., 2001, Balcan et al., 2005, Alaei et al., 2009].

However, 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 [Balcan et al., 2007], or by placing combinatorial constraints on the environment [Devanur and Hartline, 2009, Devanur et al., 2015].

In this section we circumvent these issues by applying auction formats that generalize the classical VCG auction 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) [Roberts, 1979], virtual-valuations combinatorial auctions (VVCAs) [Likhodedov and Sandholm, 2004, 2005, Sandholm and Likhodedov, 2015], λ-auctions [Jehiel et al., 2007], mixed-bundling auctions [Jehiel et al., 2007], and mixed-bundling auctions with reserve prices [Tang and Sandholm, 2012]. 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. Suppose we randomly partition the set of bidders into two groups S^1 and S^2, and apply the optimal mechanism for S^1 to S^2. Consider learning a second-price auction with a reserve in the case of selling a single item. Suppose there is one bidder who values the item at 10 and the remaining buyers' values are in $[0, 9]$. The high bidder is in S^1 with probability $1/2$. So with probability $1/2$, the optimal reserve price for S^1 is 10, and the revenue obtained from S^2 is 0. More generally, since we study large parameterized auction classes, the optimal auction for S^1 potentially overfits to a small number of bidders. Another adaption along the lines of vanilla random sampling to prevent overfitting would be to partition the set S of bidders into N groups, use the first $N-1$ groups to learn a high-revenue auction, and then apply that auction to the Nth group. The issue with this approach is that generalization guarantees would require N large. Thus the final mechanism only sells items to a tiny fraction of bidders, incurring a large revenue loss.

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* Conitzer and Sandholm [2002], Sandholm [2003].

Setup and the Main Mechanism

In our model, the seller has m indivisible items to allocate among a set S of n bidders/buyers. Each buyer is described by her valuation function $v_i : 2^{\{1,\ldots,m\}} \to \mathbb{R}_{\geq 0}$ over bundles of the m items. (We implicitly assume that each buyer's value for getting the empty bundle is zero.) We do not assume that $b \subseteq b'$ implies $v(b) \leq v(b')$ (a common assumption called free disposal). For an allocation α, $v_i(\alpha)$ denotes the value buyer i assigns to the bundle she receives according to α (we assume that buyers valuations are independent of what other buyers' receive). For an allocation α, $W(\alpha) = \sum_{i=1}^{n} v_i(\alpha)$ denotes the welfare of α, and $W_{-i}(\alpha) = \sum_{j \neq i} v_j(\alpha)$ denotes the welfare of α when bidder i is absent. For a set of bidders S, $W(S) = \max_\alpha W(\alpha)$ denotes the welfare of an efficient allocation. (We depart slightly from the notation of Part II for convenience.) The auctions we study in this section are parameterized generalizations of the

VCG auction that modify the welfare function by applying boosts to specific allocations with the aim of increasing revenue. For an auction M and a set of bidders $S' \subseteq S$, we denote by $\mathsf{Rev}_M(S')$ the sum of the payments made by bidders in S' when the seller runs M among bidders in S'. We write $S' \sim_p S$ to denote a subset S' being sampled from S by including each bidder in S' independently with probability p.

We now present the main mechanism of this section.

Learning-within-an-instance mechanism (LWI) Parameters: p, q, N

1. Draw a group of participatory buyers $S_{par} \sim_p S$.
2. Draw learning groups of buyers $S_1, \ldots, S_N \sim_q S \setminus S_{par}$.
3. Find the mechanism $\widehat{M} \in \mathcal{M}$ that maximizes empirical revenue $\frac{1}{N}\sum_{t=1}^{N} \mathsf{Rev}_M(S_t)$ over the learning groups.
4. Apply mechanism $\widehat{M}$ to S_{par}.

When $\mathcal{M}$ is a class of incentive-compatible mechanisms, LWI is incentive-compatible since $\widehat{M}$ does not depend on the valuations of the bidders in S_{par}.

Summary of Contributions

First, we provide the main guarantees satisfied by our LWI framework. The guarantees are derived using learning-theoretic techniques. Informally, they provide (high probability) lower bounds on the performance of LWI of the form $\mathsf{Rev}_{\widehat{M}}(S_{par}) \geq W(S)(L_{\mathcal{M}} - \varepsilon_{\mathcal{M}}(N, \delta)) - \tau_{\mathcal{M}}$, where $L_{\mathcal{M}}$ measures the revenue loss incurred by allocating items only to participatory bidders, $\varepsilon_{\mathcal{M}}$ is a standard learning-theoretic error term that depends on the intrinsic complexity of $\mathcal{M}$, and $\tau_{\mathcal{M}}$ is an additional error term we coin *partition discrepancy*. Partition discrepancy is also a measure of the intrinsic complexity of $\mathcal{M}$, but is simultaneously a measure of the level of uniformity in the set S of bidders. We provide examples and a general bound to illustrate properties of partition discrepancy.

We next 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. We show that under certain conditions LWI on the class of bundling-boosted auctions yields an $(O(p) - \varepsilon)$-approximation with high probability.

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. Finally, we 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.

Additional Related Research

There have been various alternate approaches to revenue maximization for limited supply. Balcan et al. [2008] obtain a $O(2^{\sqrt{\log m \log\log m}})$-approximation for bidders with subadditive valuations, which was improved to a $O(\log^2 m)$-approximation by Chakraborty et al. [2013]. Both these works studied item-pricing mechanisms. Likhodedov and Sandholm [2005], Sandholm and Likhodedov [2015] obtain a $(2 + 2\log(h/l))$-approximation when bidders have additive valuations, where l and h are lower and upper bounds on the valuation of any bidder for any item. Our results significantly improve upon these existing results in various situations. For example, for $W(S)$ sufficiently large, we prove that our LWI mechanism run on the class of bundling-boosted auctions yields an $(O(p) - \varepsilon)$-approximation. In addition, previous approximations are on expected revenue, while we give the much stronger guarantee of *high-probability* revenue approximation. Furthermore, our results do not require restrictions on valuation functions, giving them very broad applicability.

A recent line of work studies learning revenue-maximizing auctions for limited supply *across instances* [Mohri and Medina, 2014, Morgenstern and Roughgarden, 2015, Balcan et al., 2018d]. These works laid down the framework for understanding learning-theoretic quantities related to auctions in order to prove generalization guarantees. Our paper studies the significantly tougher and unsolved problem of learning from a single instance for limited supply. We extend the techniques of Balcan et al. [2005] (that can be viewed as learning within an instance for unlimited supply) and show that learning theory combined with the power of parameterized auctions provides a way to meaningfully learn within an instance in the more challenging setting of limited supply.

10.1.1 Main Guarantees of our Framework

In this section we present the main guarantees satisfied by LWI in terms of structural properties of the auction class and the set of bidders. Our guarantees are in terms of partition discrepancy, delineability, and the following quantity that controls the revenue loss incurred by selling only to bidders in S_{par}. For $S' \subseteq S$, let $\mathrm{OPT}_{\mathcal{M}}(S') = \sup_{M\in\mathcal{M}} \mathsf{Rev}_M(S')$ and let $L_{\mathcal{M}}(S') = \mathrm{OPT}_{\mathcal{M}}(S')/W(S)$.

For a given participatory set of bidders S_{par}, *partition discrepancy* measures the worst-case deviation in an auction class between the revenue on S_{par} versus the expected revenue on a set of bidders sampled from $S \setminus S_{par}$. For $0 < q < 1$ and $S_{par} \subset S$, partition discrepancy is defined as

$$\tau_{\mathcal{M}}(q, S_{par}) = \sup_{M\in\mathcal{M}} \left| \mathsf{Rev}_M(S_{par}) - \mathop{\mathbb{E}}_{S_0\sim_q S\setminus S_{par}} [\mathsf{Rev}_M(S_0)] \right|.$$

Partition discrepancy is a measure of both the intrinsic complexity of the class $\mathcal{M}$ and the amount of uniformity in the set S of bidders. We now present general guarantees for LWI in terms of partition discrepancy. The guarantees follow from uniform convergence results, and depend on the expected Rademacher complexity $R_{\mathcal{M}}(N; S\setminus S_{par})$ of $\mathcal{M}$ with respect to $S\setminus S_{par}$ and the pseudodimension $\mathsf{Pdim}(\mathcal{M})$ of $\mathcal{M}$. Recall that *Empirical Rademacher complexity* is defined as

$$R_{\mathcal{M}}(S_1, \ldots, S_N) = \mathop{\mathbb{E}}_{\sigma} \left[\sup_{M\in\mathcal{M}} \frac{1}{N} \sum_{t=1}^{N} \sigma_t \mathsf{Rev}_M(S_t) \right],$$

where σ is chosen uniformly at random from $\{-1, 1\}^N$. Expected Rademacher complexity is defined as

$$R_{\mathcal{M}}(N; S_{lrn}) = \mathop{\mathbb{E}}_{S_1,\ldots,S_N \sim S_{lrn}} [R_{\mathcal{M}}(S_1, \ldots, S_N)].$$

Our LWI mechanism satisfies a standard uniform convergence guarantee since each learning group S_t is sampled independently and identically from $S_{lrn} := S \setminus S_{par}$ [Anthony and Bartlett, 1999].

Theorem 10.1.1. *Let S_{lrn} denote the learning pool of bidders chosen by a run of LWI. With probability at least $1-\delta$ over the draw of $S_1, \ldots, S_N \sim S_{lrn}$, every mechanism $M \in \mathcal{M}$ satisfies $\mathbb{E}_{S_0 \sim S_{lrn}}[\mathsf{Rev}_M(S_0)] \le \frac{1}{N}\sum_{t=1}^N \mathsf{Rev}_M(S_t) + 2R_{\mathcal{M}}(N; S_{lrn}) + W(S)\sqrt{\frac{\ln(1/\delta)}{2N}}$.*

Rademacher complexity and pseudodimension are related via the following bound due to Dudley [1987]:

$$R_{\mathcal{M}}(N; S_{lrn}) \le 60 \cdot W(S)\sqrt{\frac{\mathsf{Pdim}(\mathcal{M})}{N}}.$$

Delineability and pseudodimension are related via the main result of Balcan et al. [2018d]: if $\mathcal{M}$ is (d,h)-delineable, $\mathsf{Pdim}(\mathcal{M}) \le 9d\ln(4dh)$.

$\widehat{M}$ denotes the empirical-revenue-maximizing mechanism used by LWI.

Theorem 10.1.2. *Let S_{par} denote the participatory set of bidders chosen by a run of LWI. Then, with probability $\ge 1-2\delta$ over the draw of $S_1, \ldots, S_N \sim_q S \setminus S_{par}$, (a) $\mathsf{Rev}_{\widehat{M}}(S_{par}) \ge W(S)\Big(L_{\mathcal{M}}(S_{par}) - 4R_{\mathcal{M}}(N; S \setminus S_{par}) - \sqrt{2\ln(1/\delta)/N}\Big) - 2\tau_{\mathcal{M}}(q, S_{par})$ and (b) $\mathsf{Rev}_{\widehat{M}}(S_{par}) \ge W(S)\Big(L_{\mathcal{M}}(S_{par}) - 240\sqrt{\mathsf{Pdim}(\mathcal{M})/N} - \sqrt{2\ln(1/\delta)/N}\Big) - 2\tau_{\mathcal{M}}(q, S_{par})$.*

Proof. Let $\varepsilon = \varepsilon_{\mathcal{M}}(\delta, N) = 2R_{\mathcal{M}}(N; S_{lrn}) + \sqrt{\frac{\ln(1/\delta)}{2N}}$. By uniform convergence, it holds with probability at least $1-\delta$ that for all $M \in \mathcal{M}$,

$$\mathop{\mathbb{E}}_{S_0 \sim S_{lrn}}[\mathsf{Rev}_M(S_0)] \le \frac{1}{N}\sum_{t=1}^N \mathsf{Rev}_M(S_t) + \varepsilon W(S),$$

and symmetrically it holds with probability at least $1-\delta$ that for all $M \in \mathcal{M}$,

$$\frac{1}{N}\sum_{t=1}^N \mathsf{Rev}_M(S_t) \le \mathop{\mathbb{E}}_{S_0 \sim S_{lrn}}[\mathsf{Rev}_M(S_0)] + \varepsilon W(S).$$

Hence, the probability of both events is at least $1-2\delta$. Let $M^* = \operatorname{argmax}_{M \in \mathcal{M}} \mathsf{Rev}_M(S_{par})$. For brevity, let $\tau = \tau_{\mathcal{M}}(q, S_{par})$. Then,

$$\begin{aligned}
\mathsf{Rev}_{\widehat{M}}(S_{par}) &\ge \mathop{\mathbb{E}}_{S_0 \sim S_{lrn}}[\mathsf{Rev}_{\widehat{M}}(S_0)] - \tau \\
&\ge \frac{1}{N}\sum_{t=1}^N \mathsf{Rev}_{\widehat{M}}(S_t) - \varepsilon W(S) - \tau \\
&\ge \frac{1}{N}\sum_{t=1}^N \mathsf{Rev}_{M^*}(S_t) - \varepsilon W(S) - \tau
\end{aligned}$$

$$
\begin{aligned}
&\geq \mathop{\mathbb{E}}_{S_0 \sim S_{lrn}} [\mathsf{Rev}_{M^*}(S_{par})] - 2\varepsilon W(S) - \tau \\
&\geq \mathsf{Rev}_{M^*}(S_{par}) - 2\varepsilon W(S) - 2\tau \\
&= W(S)(L_{\mathcal{M}}(p, S_{par}) - 2\varepsilon) - 2\tau,
\end{aligned}
$$

as desired. Part (b) is a consequence of the bound $R_{\mathcal{M}}(N; S_{lrn}) \leq 60W(S)\sqrt{\frac{\mathsf{Pdim}(\mathcal{M})}{N}}$ [Dudley, 1987]. □

If $\mathcal{M}$ has finite pseudodimension (this is not necessarily the case if we only have a bound on Rademacher complexity), we can give an equivalent sample-complexity version of the guarantee. Let $N(\varepsilon, \delta, \mathsf{Pdim}(\mathcal{M})) = 480^2 \mathsf{Pdim}(\mathcal{M}) \ln\left(\frac{1}{\delta}\right)/\varepsilon^2$.

Corollary 10.1.3. *Let S_{par} denote the participatory set of bidders chosen by a run of LWI with parameters p, q, N, where $N \geq N(\varepsilon, \delta, \mathsf{Pdim}(\mathcal{M}))$. Then, with probability $\geq 1 - 2\delta$ over the draw of $S_1, \ldots, S_N \sim_q S \setminus S_{par}$, $\mathsf{Rev}_{\widehat{M}}(S_{par}) \geq W(S)(L_{\mathcal{M}}(S_{par}) - \varepsilon) - 2\tau_{\mathcal{M}}(q, S_{par})$.*

To understand the pseudodimension of various mechanism classes, Balcan et al. [2018d] introduced the notion of *delineability*. A class of mechanisms $\mathcal{M}$ is (d,h)-*delineable* if (1) every $M \in \mathcal{M}$ can be parameterized by a vector $\theta \in \mathbb{R}^d$, and (2) for every set S of bidder valuations, there are at most h hyperplanes partitioning $\mathbb{R}^d$ such that $\mathsf{Rev}_S(\theta) := \mathsf{Rev}_\theta(S)$ is linear in θ over each connected component of $\mathbb{R}^d$ determined by the hyperplanes. The way we have stated delineability requires h to be independent of the number of bidders in S. We include an analysis of the case where h is allowed to be a function of n in the appendix. The following example illustrates delineability in a simple case. Balcan et al. [2018d] provide more examples and a more detailed discussion.

Example 10.1.4 (Second-price auctions with a reserve price)**.** The class of second-price auctions with reserve prices for selling a single item is $(1,2)$-delineable. Indeed, if v_1 and v_2 are highest and second-highest values for the item, respectively, then for $r < v_2$ the revenue of a second-price auction with reserve r is v_2, for $v_2 \leq r \leq v_1$ it is r, and for $r > v_1$ it is 0.

Rademacher complexity, pseudodimension, and delineability are connected through the following relations: $R_{\mathcal{M}}(N; S_{lrn}) \leq 60W(S)\sqrt{\mathsf{Pdim}(\mathcal{M})/N}$ [Dudley, 1987] and if $\mathcal{M}$ is (d,h)-delineable, $\mathsf{Pdim}(\mathcal{M}) \leq 9d\ln(4dh)$ [Balcan et al., 2018d].

We present our main guarantee in terms of delineability:

Theorem 10.1.5. *Suppose $\mathcal{M}$ is (d,h)-delineable. Let S_{par} denote the participatory set of bidders chosen by a run of LWI with parameters p, q, N, where $N \geq N(\varepsilon, \delta, 9d\ln(4dh))$. Then, with probability $\geq 1 - 2\delta$ over the draw of $S_1, \ldots, S_N \sim_q S \setminus S_{par}$, $\mathsf{Rev}_{\widehat{M}}(S_{par}) \geq W(S)(L_{\mathcal{M}}(S_{par}) - \varepsilon) - 2\tau_{\mathcal{M}}(q, S_{par})$.*

We provide analogous guarantees for mechanism classes that satisfy a version of delineability that is dependent on the number of bidders in the appendix.

10.1.2 Partition Discrepancy

In this section we develop a further understanding of partition discrepancy. We first provide two examples illustrating structural properties of partition discrepancy. We then provide a general-purpose high-probability bound on partition discrepancy based on pseudodimension of the mechanism class.

The first example relates the failure of vanilla random sampling to large partition discrepancy using the scenario given in the introduction. We show how LWI alleviates that issue.

Example 10.1.6 (LWI versus random sampling)**.** Consider the example from the introduction where a single item is for sale and $\mathcal{M}$ is the class of second-price auctions with reserve. There is one bidder with value 10, and all remaining bidders' values are in $[0, 9]$. Suppose LWI is run with parameters $p = 1/2, q = 1$ (which corresponds to vanilla random sampling). Then, for any participatory set S_{par}, $\tau_{\mathcal{M}}(1, S_{par}) = 10 = W(S)$, achieved by setting a reserve price of 10. If instead LWI was run with parameters $p = q < 1$, the high bidder is in $S \setminus S_{par}$ with probability $1 - p$, and in this case $\tau_{\mathcal{M}}(q, S_{par}) = 10q$ If, for example, $p = q = 1/20$, this is a small additive loss in the overall revenue guarantee.

The next example involves replica economies, where the set of bidders is composed of several copies of a ground set of bidders. Replica economies have been studied extensively in economics (and recently from an algorithmic viewpoint) in the context of convergence to equilibria [Debreu and Scarf, 1963, Aumann, 1964, Barman and Echenique, 2020].

Example 10.1.7 (Replica economies)**.** Suppose $S_0 = \{v_1, v_2, v_3\}$, and S consists of n_0 replicas of S_0. Let $\mathcal{M}$ be any auction class that can be parameterized in a way that does not depend on the number of bidders (for example, any of the auctions we define in Section 3). Any $M \in \mathcal{M}$ can be identified by the overall allocation it uses, and the at most m allocations it uses to determine payments. Hence, M can be encoded as a vector of length at most $(m + 1)^2$ (there are at most $m + 1$ allocations, and each sells to at most m buyers). Over all $M \in \mathcal{M}$, the number of such vectors is at most $3^{(m+1)^2}$, since bidders of the same type are indistinguishable. Suppose LWI is run with $p = 1/2, q = 1$. Due to Chernoff bounds, for n_0 large, it holds with exceedingly high probability that the number of bidders of each type (v_1, v_2, or v_3) in S_{par} is in $[n_0/2 - 10\sqrt{n_0}, n_0/2 + 10\sqrt{n_0}]$ (and likewise for $S \setminus S_{par}$). So if n_0 is sufficiently large, both S_{par} and $S \setminus S_{par}$ contain enough bidders of each type to form each of the at most $3^{(m+1)^2}$ auction vectors with extremely high probability, in which case $\mathsf{Rev}_M(S_{par}) = \mathsf{Rev}_M(S \setminus S_{par})$ for every $M \in \mathcal{M}$ and so $\tau_{\mathcal{M}}(1, S_{par}) = 0$.

We now present a general bound on partition discrepancy in terms of the learning-theoretic complexity of $\mathcal{M}$ when LWI is run with parameters $p = 1/3$ and $q = 1/2$. For each bidder i, let $\tilde{v}_i = \max_{|S'| \geq n/3 - 5\sqrt{n}} \sup_{M \in \mathcal{M}} |\mathsf{Rev}_M(S' \cup \{i\}) - \mathsf{Rev}_M(S')|$ and let $\tilde{v} = (\tilde{v}_1, \ldots, \tilde{v}_n) \in \mathbb{R}^n$. These terms measure how sensitive the mechanism class is to the addition of a single bidder to an already large set of bidders. In the following results on partition discrepancy, we condition on the (probability $\geq 1 - e^{-25}$) event that $|S_{par}| \geq n/3 - 5\sqrt{n}$.

Theorem 10.1.8. *With probability $\geq 1 - \delta$ over the draw of $S_{par} \sim_{1/3} S$, $\tau_{\mathcal{M}}(1/2, S_{par}) \leq ||\tilde{v}||_2 \sqrt{2n\mathsf{Pdim}(\mathcal{M}) \ln \frac{4e^2\mathsf{Pdim}(\mathcal{M})W(S)}{\delta}}$.*

Proof. Let $S = \{v_1, \ldots, v_n\}$ and let $Y_i = 1$ with probability $1/3$ and $Y_i = 0$ with probability $2/3$. Fix a single mechanism $M \in \mathcal{M}$. Let $g(Y_1, \ldots, Y_n) = \mathsf{Rev}_M(S_{par}) - \mathbb{E}_{S_0 \sim S_{lrn}}[\mathsf{Rev}_M(S_0)]$ where $S_{par} = \{v_i : Y_i = 1\}$. Let c_i be an upper bound for $|g(Y_1, \ldots, Y_n) - g(Y_1, \ldots, Y_i', \ldots, Y_n)|$. We have $\sum_i c_i^2 \leq ||\tilde{v}||_2^2$. By symmetry and the fact that each bidder has an equal probability of $1/3$ of being in S_{par} or in S_0, $\mathbb{E}_{S_{par} \sim S, S_0 \sim S_{lrn}}[\mathsf{Rev}_M(S_{par}) - \mathsf{Rev}_M(S_0)] = 0$. McDiarmid's inequality therefore yields

$$\Pr_{S_{par}\sim S}(\tau_M(1/2,S_{par})\geq t)=\Pr_{S_{par}\sim S}\left(\left|\mathsf{Rev}_M(S_{par})-\mathop{\mathbb{E}}_{S_0\sim S_{lrn}}[\mathsf{Rev}_M(S_0)]\right|\geq t\right)\leq e^{-2t^2/\sum c_i^2}.$$

Fix t. Let $\widetilde{\mathcal{M}}\subseteq\mathcal{M}$ be a subset of mechanisms that forms a $t/2^{n+2}$ L_1-cover of $\{(\mathsf{Rev}_M(S'))_{S'\subseteq S}: M\in\mathcal{M}\}\subset\mathbb{R}^{2^n}$. Now, for each $M\in\mathcal{M}$, there is $\widetilde{M}\in\widetilde{\mathcal{M}}$ such that for any S_{par}, with $n_1=|S_{par}|$, $n_2=|S_{lrn}|$, we have

$$\begin{aligned}\tau_M(1/2,S_{par})&=\left|\mathsf{Rev}_M(S_{par})-\frac{1}{2^{n_2}}\sum_{S_0\subseteq S_{lrn}}\mathsf{Rev}_M(S_0)\right|\\&\leq|\mathsf{Rev}_M(S_{par})-\mathsf{Rev}_{\widetilde{M}}(S_{par})|+\frac{1}{2^{n_2}}\sum_{S_0\subseteq S_{lrn}}|\mathsf{Rev}_M(S_0)-\mathsf{Rev}_{\widetilde{M}}(S_0)|\\&\qquad+\left|\mathsf{Rev}_{\widetilde{M}}(S_{par})-\frac{1}{2^{n_2}}\sum_{S_0\subseteq S_{lrn}}\mathsf{Rev}_{\widetilde{M}}(S_0)\right|\\&\leq\frac{t}{4}+\frac{t}{2^{n_2+2}}+\tau_{\widetilde{M}}(1/2,S_{par})\\&\leq\frac{t}{2}+\tau_{\widetilde{M}}(1/2,S_{par}).\end{aligned}$$

A union bound yields

$$\begin{aligned}\Pr\left(\sup_{M\in\mathcal{M}}\tau_M(1/2,S_{par})\geq t\right)&\leq\Pr\left(\max_{\widetilde{M}\in\widetilde{\mathcal{M}}}\tau_{\widetilde{M}}(1/2,S_{par})\geq t/2\right)\\&\leq\mathcal{N}_1(t/2^{n+2};\mathcal{M};2^n)\cdot e^{-\frac{t^2}{2\sum c_i^2}}\\&\leq e(\mathsf{Pdim}(\mathcal{M})+1)\left(\frac{2^{n+3}eW(S)}{t}\right)^{\mathsf{Pdim}(\mathcal{M})}\cdot e^{-\frac{t^2}{2\sum c_i^2}}.\end{aligned}$$

The final inequality follows from a well-known L_1-covering-number bound in terms of pseudodimension [Haussler, 1995, Anthony and Bartlett, 1999]. Taking t sufficiently large yields the desired confidence of at least $1-\delta$. □

Combined with Corollary 10.1.3, we have:

Theorem 10.1.9. *Run LWI with parameters N, $p=1/3$, $q=1/2$, where $N\geq N(\varepsilon,\delta,\mathsf{Pdim}(\mathcal{M}))$. Then, with probability $\geq 1-3\delta$,*

$$\mathsf{Rev}_{\widehat{M}}(S_{par})\geq W(S)(L_{\mathcal{M}}(S_{par})-\varepsilon)-2||\tilde{v}||_2\sqrt{2n\mathsf{Pdim}(\mathcal{M})\ln\frac{4e^2\mathsf{Pdim}(\mathcal{M})W(S)}{\delta}}.$$

When $W(S)$ is sufficiently large, we can condense the bound on partition discrepancy to contribute at most an ε loss.

Corollary 10.1.10. *Run LWI with parameters N, $p=1/3$, $q=1/2$, where $N\geq N(\varepsilon,\delta,\mathsf{Pdim}(\mathcal{M}))$. If $W(S)^2-\frac{8n||\tilde{v}||_2^2\mathsf{Pdim}(\mathcal{M})}{\varepsilon^2}\ln(W(S))\geq\frac{8n||\tilde{v}||_2^2\mathsf{Pdim}(\mathcal{M})}{\varepsilon^2}\ln\left(\frac{4e^2\mathsf{Pdim}(\mathcal{M})}{\delta}\right)$,*

$$\mathsf{Rev}_{\widehat{M}}(S_{par})\geq W(S)(L_{\mathcal{M}}(S_{par})-2\varepsilon)$$

with probability $\geq 1-3\delta$.

Remark. We emphasize that small partition discrepancy (for example, stipulating that $\tau_{\mathcal{M}}$ is a fixed constant) should be viewed as a uniformity condition on the set of bidders. Theorem 10.1.8 provides just one way of understanding partition discrepancy by relating it to learning-theoretic quantities.

10.1.3 Population-Size-Independent Auctions

In this section we instantiate our main guarantee for specific mechanism classes $\mathcal{M}$ to obtain more concrete revenue approximations. The following is a naïve lower bound on $L_{\mathcal{M}}(S_{par})$ for auction classes that can run a second-price auction on the grand bundle $\{1, \dots, m\}$ with a reserve price.

Proposition 10.1.11. *Let $v_1 \geq \cdots \geq v_n$ denote the valuations of each bidder in S on the grand bundle. For any $0 < \alpha \leq 1$ such that αn is an integer, any mechanism class $\mathcal{M}$ containing the second-price auction on the grand bundle with reserve price r for every r satisfies $L_{\mathcal{M}}(S_{par}) \geq \frac{v_{\alpha n}}{W(S)}$ with probability $\geq 1 - e^{-\alpha np}$ over the draw of $S_{par} \sim_p S$.*

Proof. Consider running a second-price auction on the grand bundle with reserve price $v_{\alpha n}$. If any bidder who values the grand bundle at least $v_{\alpha n}$ is in S_{par}, the revenue obtained is at least $v_{\alpha n}$. This event occurs with probability $\Pr(\cup_{i \leq \alpha n}\{i \in S_{par}\}) \geq 1 - (1-p)^{\alpha n} \geq 1 - e^{-\alpha np}$. □

However, any bidder's value for the grand bundle can be an arbitrarily bad approximation to $W(S)$. In the remainder of the paper we introduce some new auction classes and prove more fine-tuned approximations for those classes.

We now study a handful of *population-size-independent* auction classes, that is, auction classes that can be parameterized in a way that does not depend on the number of bidders. Traditional variants to the VCG auction including λ-auctions and AMAs specify boosts based on particular allocations and are thus not independent of the population size (and in particular cannot be used with LWI in a natural way since $S_1, \dots, S_N, S_{par}$ can all vary in size). In contrast to these, our auctions specify boosts based on bundles and bundlings.

A bundling is a partition of items $\{1, \dots, m\}$ into bundles. We say that an allocation *respects* a bundling if no two items in the same bundle are allocated to different buyers. For an allocation β, let $\mathsf{blg}(\beta)$ denote the finest bundling respected by β, that is, the bundling with the fewest number of bundles that β respects. For example, if β allocates items 1 and 3 to bidder 1, and the remaining items to bidder 2, $\mathsf{blg}(\beta) = \{\{1, 3\}, \{2, 4, \dots, m\}\}$. Let Φ denote the collection of all bundlings. $|\Phi| < (0.792m/\ln(m+1))^m$ [Berend and Tassa, 2010]. We now introduce two new auction classes that can be viewed as population-size-independent analogues of λ-auctions and VVCAs, respectively.

The class of *bundling-boosted auctions* is the class auctions parameterized by real $|\Phi|$-dimensional vectors $\omega \in \mathbb{R}^{|\Phi|}$ that specify additive boosts $\omega(\phi)$ for each bundling $\phi \in \Phi$. The overall allocation α^* used by a bundling-boosted auction ω is chosen to maximize $W(\alpha) + \omega(\mathsf{blg}(\alpha))$, and bidder i pays $\max_\alpha(W_{-i}(\alpha) + \omega(\mathsf{blg}(\alpha))) - (W_{-i}(\alpha^*) - \omega(\mathsf{blg}(\alpha^*)))$. Equivalently, ω is the λ-auction with $\lambda(\alpha) = \omega(\mathsf{blg}(\alpha))$.

The class of *bundle-boosted auctions* is the class of auctions parameterized by real 2^m-dimensional vectors $\omega \in \mathbb{R}^{2^m}$ that specify additive boosts $\omega(b)$ for each bundle $b \subseteq \{1, \dots, m\}$.

The overall allocation α^* is chosen to maximize $W(\alpha) + \sum_{b \in \mathsf{blg}(\alpha)} \omega(b)$, and bidder i pays $\max_\alpha(W_{-i}(\alpha) + \sum_{b \in \mathsf{blg}(\alpha)} \omega(b)) - (W_{-i}(\alpha^*) + \sum_{b \in \mathsf{blg}(\alpha^*)} \omega(b))$. Equivalently, the class of bundle-boosted auctions is the subclass of VVCAs where the parameters are constant across bidders.

The class of bundling-VCG auctions due to Kroer and Sandholm [2015] consists of all ϕ-VCG auctions, where a ϕ-VCG auction runs VCG while treating each bundle in ϕ as an indivisible item. The class of bundling-VCG auctions is a subclass of the class of bundle-boosted auctions: the ϕ-VCG auction can be represented by the bundle-boosted auction with $\omega(b) = 0$ if b can be represented as a union of bundles from ϕ, and $\omega(b) = -\infty$ otherwise. The class of bundle-boosted auctions is a subclass of the class of bundling-boosted auctions: a bundle-boosted auction is a bundling-boosted auction with the restriction that $\omega(\phi) = \sum_{b \in \phi} \omega(b)$.

Since bundling-boosted and bundle-boosted auctions are subclasses of λ-auctions, they are both delineable with $h(n) = (n+1)^{2m+1}$ due to Balcan et al. [2018d]. The following is a much stronger delineability result that has no dependence on the number of bidders.

Theorem 10.1.12. *The class of bundling-boosted auctions is* $(|\Phi|, |\Phi|^2 + m|\Phi|^3)$*-delineable and the class of bundle-boosted auctions is* $(2^m, |\Phi|^2 + m|\Phi|^3)$*-delineable.*

Proof. We prove the result for bundling-boosted auctions. Fix the input set of bidders. For each bundling ϕ, let $\beta^\phi = \operatorname{argmax}_{\beta : \mathsf{blg}(\beta) = \phi} W(\beta)$, and let $m(\phi)$ denote the number of bundles in ϕ. Note that if β^ϕ is not the ϕ-VCG allocation, then for ϕ' the coarsest bundling respected by the ϕ-VCG allocation, $\beta^{\phi'}$ is the VCG allocation corresponding to both ϕ and ϕ' (ϕ' can be obtained by combining certain bundles in ϕ).

Now, for any bundling-boosted auction parameters ω, the overall allocation chosen must be one of $\{\beta^\phi : \phi \in \Phi\}$ (which contains as a subset the collection of all bundling VCG allocations, as remarked above). This is because for any allocation α, α is given the same boost as $\beta^{\mathsf{blg}(\alpha)}$, which has greater welfare by definition.

We now count the total number allocations that can ever be used by ω when any bidder is absent. For a given $\phi \in \Phi$, exactly $m(\phi)$ bidders are allocated any items by β^ϕ. If bidder i is not allocated any items by β^ϕ, then β^ϕ also maximizes welfare among all allocations α such that $\mathsf{blg}(\alpha) = \phi$ when bidder i is absent. If bidder i is allocated something by β^ϕ, let $\beta^\phi_{-i} = \operatorname{argmax}_{\beta : \mathsf{blg}(\beta) = \phi} W_{-i}(\beta)$. For any setting of the parameters ω, the allocation used when bidder i is absent will be of the form β^ϕ_{-i}. This is because for any allocation α,

$$W_{-i}(\alpha) + \omega(\mathsf{blg}(\alpha)) \le W_{-i}\big(\beta^{\mathsf{blg}(\alpha)}_{-i}\big) + \omega(\mathsf{blg}(\alpha)).$$

There are a total of $m(\phi)$ such unique allocations (not including β^ϕ). Now, if bidder i is not allocated any items by β^ϕ for any $\phi \in \Phi$, the absence of bidder i does not change the allocation used by any $\omega \in \mathbb{R}^{|\Phi|}$. The total number of bidders whose absence can change the allocation used is thus at most

$$\sum_{\phi \in \Phi} m(\phi) \le m|\Phi|.$$

Finally, we count the number of hyperplanes partitioning the parameter space such that the allocations used are constant within any region. There are $\binom{|\Phi|}{2}$ hyperplanes of the form

$$W(\beta^\phi) + \omega(\phi) = W(\beta^{\phi'}) + \omega(\phi')$$

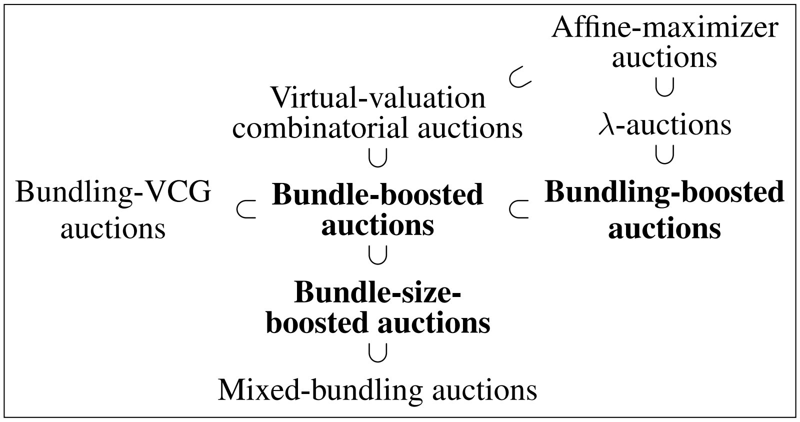


Figure 10.1: Containment relations between auction classes. New auction classes introduced in this section are in boldface.

for every $\phi, \phi' \in \Phi$. For each of the at most $m|\Phi|$ bidders whose absence potentially changes the allocation used, there are at most $\binom{|\Phi|}{2}$ hyperplanes of the form

$$W(\beta_{-i}^{\phi}) + \omega(\phi) = W(\beta_{-i}^{\phi'}) + \omega(\phi')$$

for every $\phi, \phi' \in \Phi$ such that i is allocated a nonempty bundle by β^{ϕ} and $\beta^{\phi'}$. The total number of hyperplanes is thus at most

$$\binom{|\Phi|}{2} + m|\Phi|\binom{|\Phi|}{2} < |\Phi|^2 + m|\Phi|^3.$$

□

Likhodedov and Sandholm [2005], Sandholm and Likhodedov [2015] (implicitly) study properties of the class of auctions parameterized by vectors $\omega \in \mathbb{R}^m$ that specify additive boosts depending on the size of the bundle. We call this class of auctions *bundle-size-boosted* auctions. Bundle-size-boosted auctions are a subclass of bundle-boosted auctions: the equivalent bundle-boosted auction satisfies $\omega(b) = \omega(|b|)$. For the class of bundle-size-boosted auctions, we can prove a stronger delineability result.

Theorem 10.1.13. *The class of bundle-size-boosted auctions is $(m, me^{O(\sqrt{m})})$-delineable.*

Proof. For a given bundle-size-boosted auction, consider the equivalent bundling-boosted auction. Any two bundlings that are indistinguishable with respect to the sizes of their bundles are given the same boost, since $\omega(\phi) = \sum_{b\in\phi}\omega(|b|)$. Hence, we may run the same argument of Theorem 10.1.12 but instead of considering all $|\Phi|$ bundlings we only need to consider the number $p(m)$ of integer partitions of m. It is well known that there is a constant B such that $p(m) < e^{B\sqrt{m}}$. □

Figure 10.1 summarizes the containment relations between the various auction classes.

Guarantees for Bundling-Boosted Auctions

The class of bundling-boosted auctions is a rich class of auctions. If the efficient allocation when bidder i is absent also maximizes welfare when all bidders are present among all allocations respecting the same finest bundling, there is a bundling-boosted auction that extracts revenue equal to the welfare of the efficient allocation. More generally:

Theorem 10.1.14. *Given a set of S bidders, let β denote the efficient allocation and let β_{-i} denote the efficient allocation when bidder i is absent. Let $\Delta_i(S) = \max_{\alpha:\mathsf{blg}(\alpha)=\mathsf{blg}(\beta_{-i})} W(\alpha) - W(\beta_{-i})$. There exists a bundling-boosted auction with revenue $W(S) - \sum_i \Delta_i(S)$.*

Proof. Let $\phi = \mathsf{blg}(\beta)$ and for each i let $\phi_{-i} = \mathsf{blg}(\beta_{-i})$. Consider the bundling-boosted auction with $\omega(\phi) = 0$, $\omega(\phi_{-i}) = W(\beta) - W(\beta_{-i})$, and $\omega(\phi') = -\infty$ for all other bundlings $\phi' \in \Phi \setminus \{\phi, \phi_{-1}, \dots, \phi_{-n}\}$.

We show that when $\Delta_i(S) = 0$ for all i, this auction extracts revenue equal to $W(S)$. The proof of the more general statement in terms of $\Delta_i(S)$ is similar; we describe the necessary modifications at the end. First we show that β is the overall allocation chosen by this auction. For any α such that $\mathsf{blg}(\alpha) \notin \{\phi, \phi_{-1}, \dots, \phi_{-n}\}$, $\lambda(\alpha) = -\infty$, so such allocations are never chosen.

Case 1. $\mathsf{blg}(\alpha) = \phi$. Then as $\lambda(\beta) = \lambda(\alpha) = \omega(\phi) = 0$, $W(\beta) + \lambda(\beta) \geq W(\alpha) + \lambda(\alpha)$, since β is an efficient allocation.

Case 2. $\mathsf{blg}(\alpha) = \phi_{-i}$ for some i. Then $\lambda(\alpha) = \omega(\phi_{-i})$ and $W(\beta_{-i}) \geq W(\alpha)$, so $W(\beta)+\lambda(\beta) = W(\beta) = W(\alpha) + (W(\beta) - W(\alpha)) \geq W(\alpha) + (W(\beta) - W(\beta_{-i})) = W(\alpha) + \lambda(\alpha)$ for all i.

Hence β is the overall allocation used. Next we show that when bidder i is absent, β_{-i} is the allocation used by this auction.

Case 1. $\mathsf{blg}(\alpha) = \phi$. Then $W_{-i}(\beta_{-i}) + \lambda(\beta_{-i}) = W(\beta) \geq W(\alpha) \geq W(\alpha) - v_i(\alpha) = W_{-i}(\alpha) + \lambda(\alpha)$.

Case 2. $\mathsf{blg}(\alpha) = \phi_{-i}$. Then $\lambda(\beta_{-i}) = \lambda(\alpha) = \omega(\phi_{-i})$, so $W_{-i}(\beta_{-i})+\lambda(\beta_{-i}) \geq W_{-i}(\alpha)+\lambda(\alpha)$.

Case 3. $\mathsf{blg}(\alpha) = \phi_{-k}$ for $k \neq i$. Then $W(\beta_{-k}) \geq W(\alpha)$, so $W_{-i}(\alpha) + \lambda(\alpha) = W_{-i}(\alpha) + W(\beta) - W(\beta_{-k}) \leq W_{-i}(\alpha) + W(\beta) - W(\alpha) = W(\beta) - v_i(\alpha) \leq W(\beta) = W(\beta_{-i}) + \lambda(\beta_{-i})$.

Hence β_{-i} is the allocation used when bidder i is not present. We have shown that the allocations used by this bundling-boosted auction are precisely the VCG allocations. The payment of bidder i is therefore $(W_{-i}(\beta_{-i})+\lambda(\beta_{-i}))-(W_{-i}(\beta)+\lambda(\beta)) = W(\beta)-(W(\beta)-v_i(\beta)) = v_i(\beta)$, and so the total revenue is $W(\beta)$. The proof of the general statement is similar. The only difference is that ω might not use the VCG allocations, but is guaranteed to use allocations for which the boosted welfare does not differ much from that of the corresponding VCG allocation by much. □

We give a simple example of bidder valuations that satisfy $\Delta_i(S') = 0$ for every i and every $S' \subseteq S$ involving bidders whose "most desired" bundles intersect.

Example 10.1.15. Let $b_1, \dots, b_n \subseteq \{1, \dots, m\}$ be distinct bundles such that $b_i \cap b_j \neq \emptyset$ for every i, j. Let $c_1 > c_2 > \cdots > c_n > 0$, and let $\varepsilon > 0$ be sufficiently small. The valuation of bidder i satisfies $v_i(b) = c_i$ if $b \supseteq b_i$ and $v_i(b) \leq \varepsilon$ otherwise. Then, for any subset of the bidders $S' \subseteq \{1, \dots, n\}$, the welfare-maximizing allocation gives bundle b_i to $i = \min(S')$ and allocates the remaining items to the other bidders. When i is absent, the welfare-maximizing allocation gives bundle $b_{i'}$ to $i' = \min(S' \setminus \{i\})$ and allocates the remaining items to the other bidders. This is clearly the welfare-maximizing allocation among all allocations respecting the same finest bundling. Now, for each $j \neq i$, if the finest bundlings respected by each of the welfare-maximizing allocations when j is absent are all distinct, then $\Delta_i(S') = 0$ for every i, S'. (since the welfare-maximizing allocation gives b_i to i and uses a distinct bundling on the remaining buyers for each j). Otherwise, $\Delta_i(S')$ is nevertheless small, since the welfare extracted from bidders excluding i when any bidder $j \neq i$ is absent is small.

Concentration inequalities enable us to provide bounds on $L_{\mathcal{M}}(S_{par})$ for the class of bundling-boosted auctions. We have $\mathbb{E}_{S_{par}}[\mathrm{OPT}_{\mathcal{M}}(S_{par})] \geq \mathbb{E}_{S_{par}}[W(S_{par}) - \sum_i \Delta_i(S_{par})] \geq pW(S) - \sum_i \Delta_i(S_{par})$. If we run LWI with parameters p, q, N, we have (assuming for readability that $\Delta_i(S_{par}) = 0$ for all i) $L_{\mathcal{M}}(S_{par}) \geq (1-\eta)p$ with probability $\geq 1 - e^{-2\eta^2 p^2 W(S)^2/||\bar{v}||_2^2}$, where $\bar{v} = (\max_{b \subseteq \{1,\ldots,m\}} v_i(b))_{i \in S} \in \mathbb{R}^n$, by McDiarmid's inequality.

Combined with Theorem 10.1.5, we get our main guarantee for the class of bundling-boosted auctions. For readability, we state our guarantees assuming $\Delta_i(S_{par}) = 0$ for every i.

Theorem 10.1.16. *Let $\mathcal{M}$ be the class of bundling-boosted auctions. Let $N \geq N(\varepsilon, \delta, \mathsf{Pdim}(\mathcal{M}))$ and run LWI with parameters N, p, q. As long as $W(S)^2 \geq ||\bar{v}||_2^2 \ln(1/\delta)/2\eta^2 p^2$, $\mathsf{Rev}_{\widehat{M}}(S_{par}) \geq W(S)((1-\eta)p - \varepsilon) - 2\tau_{\mathcal{M}}(q, S_{par})$ with probability $\geq 1 - 3\delta$ conditional on $\Delta_i(S_{par}) = 0$ for all i.*

Removing the assumption on $\Delta_i(S_{par})$ would replace the $(1-\eta)p$ loss term with $(1-\eta)(p - \sum_i \Delta_i(S_{par})/W(S))$.

10.1.4 Efficient Learning Within an Instance

We now explore two mechanism classes for which LWI can be implemented efficiently by leveraging efficient routines for solving winner determination (a generalization of the problem of computing efficient allocations). Though computing $\widehat{M} = \mathrm{argmax}_{M \in \mathcal{M}} \frac{1}{N} \sum_{t=1}^{N} \mathsf{Rev}_M(S_t)$ is NP-hard since it involves solving winner determination (which is well known to be NP-hard) winner determination can be solved efficiently in practice [Sandholm et al., 2005].

For the class of bundling-VCG auctions, we show that the branch-and-bound technique of Kroer and Sandholm [2015] is compatible with LWI. We did not derive a revenue-guarantee for this class of auctions, however. For the class of *sparse bundling-boosted auctions*, which are bundling-boosted auctions with a constant number of positive boosts, we show that a revenue guarantee similar to (but more sample efficient than) Theorem 10.1.16 holds. We then show how LWI can be efficiently implemented for this class.

Bundling-VCG Auctions

Kroer and Sandholm [2015] give a branch-and-bound algorithm to compute the revenue-maximizing bundling-VCG auction for a given set of bidders. While our setting is different than theirs, their integer-program techniques can be directly used by LWI. Let f denote a function used as an upper bound in branch-and-bound to compute the optimal bundling. For learning groups $S_1, \ldots, S_N$ and x a node in the search tree (corresponding to a partial bundling), let $\hat{f}(x) = \frac{1}{N} \sum_t f(x; S_t)$. Recall that f is *admissible* if its value at any node is an upper bound for the maximum revenue obtainable in the subtree rooted at that node, and f is *monotonic* if it decreases down each path in the search tree. These properties ensure that branch-and-bound finds the revenue-optimal bundling.

Proposition 10.1.17. *If f is admissible for computing the optimal bundling, $\hat{f}$ is admissible for computing the empirically optimal bundling. The same holds for monotonicity.*

Proof. Suppose we are at node x of a branch-and-bound computation (representing a partial bundling). Let T_x denote the subtree rooted at x. Then,

$$\hat{f}(x) = \frac{1}{N}\sum_{t=1}^{N} f(x; S_t) \geq \frac{1}{N}\sum_{t=1}^{N} \max_{\phi \in T_x} \mathsf{Rev}_\phi(S_t) \geq \max_{\phi \in T_x} \frac{1}{N}\sum_{t=1}^{N} \mathsf{Rev}_\phi(S_t).$$

Monotonicity of $\hat{f}$ is also immediate. □

Sparse Bundling-Boosted Auctions

Let $\Phi_0 \subset \Phi$ with $|\Phi_0| = B$, and let m_0 be the number of bundles in the finest bundling in Φ_0. Consider the subclass of bundling-boosted auctions for which $\omega(\phi) > 0$ only if $\phi \in \Phi_0$ (and $\omega(\phi) = 0$ otherwise), which we call Φ_0-bundling-boosted auctions. The same argument used to prove Theorem 10.1.12 shows that the class of Φ_0-bundling-boosted auctions is $(B, B^2 + m_0 B^3)$-delineable. Let $W^{\Phi_0}(S)$ denote the welfare of the welfare-maximizing allocation to bidders in S, subject to the constraint that the finest bundling respected by the allocation is in Φ_0. The same arguments used to obtain Theorem 10.1.14 yield a guarantee with respect to $W^{\Phi_0}(S)$.

Theorem 10.1.18. *Let $\mathcal{M}$ be the class of Φ_0-bundling-boosted auctions. Let $N \geq N(\varepsilon, \delta, \mathsf{Pdim}(\mathcal{M}))$ and run LWI with parameters N, p, q. As long as $W^{\Phi_0}(S)^2 \geq ||\bar{v}||_2^2 \ln(1/\delta)/2\eta^2 p^2$, $\mathsf{Rev}_{\widehat{M}}(S_{par}) \geq W^{\Phi_0}(S)((1-\eta)p - \varepsilon) - 2\tau_{\mathcal{M}}(q, S_{par})$ with probability $\geq 1 - 3\delta$ conditional on $\Delta_i(S_{par}) = 0$ for all i.*

For B a fixed constant, the number of learning groups N required by LWI is $O(B \ln(m_0 B))$ (hiding the dependence on ε and δ). In contrast, optimizing over the entire class of bundling-boosted auctions as in Theorem 10.1.16 would require N to be exponential (in m). For this class of auctions, we describe an algorithm that implements LWI with run-time exponential in B but polynomial in all other parameters (including the run time of the winner determination routine used). A similar algorithm was used in Balcan et al. [2020c], though in a different setting than ours.

Theorem 10.1.19. *Let $B = |\Phi_0|$, and let m_0 be the number of bundles in the finest bundling in Φ_0. Given learning groups $S_1, \ldots, S_N$, the empirical-revenue maximizing Φ_0-bundling-boosted auction can be computed in $(N m_0 B)^{O(B)} + 2w(m_0, n) N m_0 B$ time, where $w(m_0, n)$ is the time required to solve winner determination for n buyers with valuations over m_0 items.*

Proof. For a bundling $\phi \in \Phi$, let $\mathsf{Winner}(\phi, S_t)$ denote the welfare-maximizing allocation among bidders in S_t respecting bundling ϕ. Let $\beta^{\phi,t} = \mathsf{Winner}(\phi, S_t)$ and let $S_t^\phi \subseteq S_t$ be the set of bidders in S_t who get allocated a nonempty bundle by $\beta^{\phi,t}$. For each bidder $i \in S_t^\phi$, let $\beta_{-i}^{\phi,t} = \mathsf{Winner}(\phi, S_t \setminus \{i\})$. The proof of Theorem 10.1.12 shows that any ω must use a subset of these allocations. Winner is called at most $NB + N m_0 B$ times, since $|S_t^\phi| \leq m_0$.

For each t and each pair of bundlings ϕ, ϕ', let $H(t, \phi, \phi')$ denote the hyperplane

$$\sum_{i \in S_t} v_i(\beta^{\phi,t}) + \omega(\phi) = \sum_{i \in S_t} v_i(\beta^{\phi',t}) + \omega(\phi'),$$

and for each $i \in S_t^{\phi}$ let $H_{-i}(t, \phi, \phi')$ denote the hyperplane

$$\sum_{j \in S_t \setminus \{i\}} v_j(\beta_{-i}^{\phi,t}) + \omega(\phi) = \sum_{j \in S_t \setminus \{i\}} v_j(\beta_{-i}^{\phi',t}) + \omega(\phi').$$

Let $\mathcal{H}$ denote the collection of these hyperplanes. The total number of such hyperplanes is at most $NB^2 + Nm_0B^2$ (as argued in Theorem 10.1.12). It is a basic combinatorial fact that $\mathcal{H}$ partitions $\mathbb{R}^B$ into at most $|\mathcal{H}|^B \leq (NB^2 + Nm_0B^2)^B \leq (2Nm_0B^2)^B$ regions (each region is an intersection of at most $|\mathcal{H}|$ halfspaces). The empirical revenue over $S_1, \ldots, S_N$ is linear in ω over each region, since the allocations of the bundling-boosted auction remain constant within each region. Thus, the maximum empirical revenue can be found by solving a linear program within each region. Each linear program involves B variables and at most $|\mathcal{H}|$ constraints and can thus be solved in $poly(|\mathcal{H}|, B)$ time.

A representation of each of the regions as a $0/1$ vector of length $|\mathcal{H}|$ can be computed in $poly(|\mathcal{H}|^B)$ time using the following high-level procedure. Initialize the list of regions as $\mathbb{R}^B$ (represented by the empty set of constraints). Iterate over the set of hyperplanes. For each hyperplane, check whether it intersects any region in the current list of regions. If so, update each region it intersects by adding the two constraints corresponding to the two new halfspaces. Our algorithm solves the corresponding linear program for every such region and picks the solution that yields highest empirical revenue. □

10.1.5 Structural Revenue Maximization

Suppose the mechanism designer can only sample a limited number N of learning groups (due to a run-time constrant, for example). We introduced several new auction classes, but which one should the mechanism designer use in conjunction with LWI? *Structural revenue maximization* (SRM) helps answer this question. SRM suggests maximizing empirical revenue minus a regularization term that penalizes more complex mechanisms to ensure that the chosen auction is indeed likely to generalize well, rather than overfitting to the learning groups. Our generalization guarantee in Theorem 10.1.2 provides the appropriate regularizer $\varepsilon_{\mathcal{M}}(N, \delta) = 240\sqrt{\mathsf{Pdim}(\mathcal{M})/N} + \sqrt{2\ln(1/\delta)/N}$. Say the mechanism designer is deciding between auctions in $\mathcal{M}_1$ and auctions in $\mathcal{M}_2$. Let $\widehat{M_1}$, $\widehat{M_2}$ be the empirical-revenue-maximizing auctions from $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, for one run of LWI. The mechanism designer should use mechanism $\widehat{M_k}$, $k \in \{1, 2\}$, that maximizes $\frac{1}{N}\sum_t \mathsf{Rev}_{\widehat{M_k}}(S_t) - \varepsilon_{\mathcal{M}_k}(N, \delta)$ since empirical revenue minus $\varepsilon_{\mathcal{M}}(N, \delta)$ is a more accurate lower bound on expected revenue than empirical revenue alone. An SRM approach combined with LWI is incentive compatible since the final mechanism only depends on the learning groups of bidders. Our use of SRM is similar to SRM across instances, which was discussed in Balcan et al. [2018d]. SRM for auction design was first proposed by Balcan et al. [2005], also for learning within an instance (but for unlimited supply).

10.2 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 [Aliloupour, 2016, Massad, 2018], a thinning customer base for department stores due to online retailers like Amazon [Goldmanis et al., 2010, Cusumano, 2017], and reduced capacities for restaurants during the COVID-19 pandemic [Song et al., 2021]. 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. 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 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. [2018d] 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 10.2.1. *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 $N = O\left(\frac{d \log(dh)}{\varepsilon^2} \log\left(\frac{1}{\delta}\right)\right)$.

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 10.2.1. 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.

Related work on shrinking markets and uncertainty Shrinking markets have been studied by various researchers in the context of oil companies [Van de Graaf, 2018], cable TV [Aliloupour, 2016, Massad, 2018], labor markets [Jones and Seitani, 2019], telecom markets [Neokosmidis et al., 2018], housing markets [Kawai et al., 2019], and in combinatorial settings including a thinning customer base for department stores due to online retailers like Amazon [Goldmanis et al., 2010, Cusumano, 2017] and reduced capacities for restaurants during the COVID-19 pandemic [Song et al., 2021]. Most of this existing research is extremely domain specific, and provides advisory content based on historical observations, data, and general economic knowledge. We introduce the first formal model of market shrinkage in multi-item settings, and prove the first known guarantees for how much revenue can be preserved in a shrinking market. Our guarantee on the revenue preserved in a shrinking market provides a positive contrast to recent work of Dobzinski and Uziely [2018], who study the effect of market shrinkage on revenue loss. They show that even in the case of selling a single item to n buyers with known valuation distributions, the absence of a single buyer with a fixed "low" value can surprisingly result in a (multiplicative) revenue loss of $\frac{1}{e+1}$ (in expectation). We tackle the significantly more complex multi-item setting. Furthermore, our main results are prior-free (in that they are tailored to the specific set S of bidders and do not require bidders to come from a distribution) and thus provide a strong positive contrast to this negative result.

Our results can also be viewed from the perspective of an uncertain market, since at the point of the mechanism design the subset of bidders that participates in the market is unknown. Mechanism design with uncertainty about bidder valuations has previously studied [Lopomo et al., 2021, Todo, 2020], but to the best of our knowledge the prior-free setting for combinatorial auctions has not been considered.

10.2.1 Problem formulation

We study a model of limited-supply combinatorial auctions that is identical to the one in the previous section.

Market-size uncertainty In our model, the mechanism designer has full knowledge of the entire population of bidders S (described by their valuation functions). An unknown random subset of S participates in the market. We write $S_0 \sim_p S$ to denote a subset S_0 that is sampled from S by including each bidder in S_0 independently with probability p. More generally, for a distribution D over 2^S, we write $S_0 \sim_D S$ to denote a random subset of S chosen according to D. We are interested in what happens to the maximum revenue achievable when only a random fraction of the set S of bidders participates in the auction, that is, $\sup_{M\in\mathcal{M}} \mathbb{E}_{S_0\sim_D S}[\mathsf{Rev}_M(S_0)]$.

Since a variable group of bidders of variable size can participate in the auction mechanisms we run, we require the important assumption that auctions in $\mathcal{M}$ can be run on variable-size sets of bidders in a well-defined manner. Various well-studied classes of auctions satisfy this property: examples include the class of VCG auctions with reserve prices, λ-auctions [Jehiel et al., 2007], affine-maximizer auctions [Roberts, 1979], and various population-size-independent auctions in Balcan et al. [2021c] (covered in the previous section). We assume that the mechanism designer knows the valuations of the bidders in S to begin with. So, each bidder can be thought of having an identity (for example, "the bidder who values apples at x and oranges at y", or "the bidder with valuation function v_4"), and the mechanism designer knows the identities/valuations $v_1, \ldots, v_n$ of all bidders in S. An allocation, formally, is a mapping from items to bidder identities.

The sequence of mechanism design and revelation in our setting is the same as in the standard mechanism design setting. Specifically, the mechanism design (computation of a mechanism from $\mathcal{M}$) takes place before the bidders in the shrunken market are asked to reveal their valuations. This is important for incentive compatibility, that is, for motivating the bidders to reveal their true valuations. If the design/choice of $M \in \mathcal{M}$ is allowed to be based on the revealed valuations, the auction might not be incentive compatible. (The class of second price auctions with reserve prices for a single item serves as an illustrative example. Choosing the reserve price to maximize revenue after the shrunken market is revealed clearly violates incentive compatibility.) Because the designer does not know exactly which bidders are in the shrunken market S_0, the designer has uncertainty about the valuations of the bidders. He only knows that they belong to S.

Assumptions on $\mathcal{M}$ and S For any $S' \subseteq S$, let $W_{\mathcal{M}}(S') = \max_{M \in \mathcal{M}} W_M(S')$ and $\mathsf{Rev}_{\mathcal{M}}(S') = \max_{M \in \mathcal{M}} \mathsf{Rev}_M(S')$. Let $\mathsf{win}_M(S')$ denote the set of bidders in S' that win a nonempty bundle of items per M. Let $\mathsf{win}_{\mathcal{M}}(S')$ denote the set of bidders in S' that win a nonempty bundle of items per the mechanism in M achieving $W_{\mathcal{M}}(S')$. The following two assumptions are the most critical ones.

Welfare submodularity $\forall\, S_1, S_2 \subseteq S, W_{\mathcal{M}}(S_1) + W_{\mathcal{M}}(S_2) \geq W_{\mathcal{M}}(S_1 \cup S_2) + W_{\mathcal{M}}(S_1 \cap S_2)$.

Winner monotonicity $\forall\, S'' \subseteq S' \subseteq S, \forall\, i \in S'', i \in \mathsf{win}_{\mathcal{M}}(S') \implies i \in \mathsf{win}_{\mathcal{M}}(S'')$.

Suppose $W_{\mathcal{M}}(S') = W_{VCG}(S')$ for any $S' \subseteq S$, that is, $\mathcal{M}$ is sufficiently rich to be able to allocate items efficiently (as is the case with all mechanisms in the hierarchies discussed by Balcan et al. [2018d, 2021c]). Then, welfare submodularity implies winner monotonicity [Guo, 2011]. If the valuation functions of bidders in S satisfy the gross-substitutes property, then both welfare submodularity and winner monotonicity hold [Gul and Stacchetti, 1999, Yokoo et al., 2004, Guo, 2011].

The final assumptions stipulate that $\mathcal{M}$ is a sufficiently rich class of mechanisms. We assume that $\mathsf{Rev}_{\mathcal{M}}(S') = W_{\mathcal{M}}(S') = W_{VCG}(S')$ for all $S' \subseteq S$, and further that $\mathcal{M}$ satisfies the following "global VCG-like" property: $\mathsf{Rev}_{\mathcal{M}}(S')$ depends only on $\mathsf{win}_{\mathcal{M}}(S')$ and $\mathsf{win}_{\mathcal{M}}(S' \setminus \{i\})$ for each $i \in \mathsf{win}_{\mathcal{M}}(S')$. In words, these conditions stipulate the following: (1) $\mathcal{M}$ is sufficiently rich such that in a non-truthful full-information setting, $\mathcal{M}$ can always extract the full social surplus $W_{\mathcal{M}}(S') = W_{VCG}(S')$ as revenue and (2) for any S', the payments collected by the revenue-maximizing mechanism M that achieves $\mathsf{Rev}_{\mathcal{M}}(S')$ depend only on $W_{\mathcal{M}}(S)$ and the maximum welfares $W_{\mathcal{M}}(S \setminus \{i\})$ achievable when each bidder drops out.

As a concrete example, if S is a set of bidders with gross-substitutes valuations, then the class of λ-auctions and the class of affine-maximizer auctions satisfy all of the above properties.

10.2.2 Revenue loss can be drastic

At first glance it might appear that the expected revenue preserved by a mechanism M when each bidder participates independently with probability p should simply be $p \cdot \mathsf{Rev}_M(S)$ (or more if one thinks of revenue as having diminishing returns in the number of bidders). This intuition is indeed accurate if Rev_M is a submodular function (which captures the diminishing returns property). However, revenue can shrink by more than this when mechanisms in $\mathcal{M}$ do not have submodular revenue. One reason for greater revenue loss is reduced competition among buyers. For example, suppose there are m items and $2m$ bidders, where bidder i for $1 \leq i \leq m$ has valuation $v_i(b) = c$ if $i \in b$ and $v_i(b) = 0$ otherwise, and bidder $m + i$ for $1 \leq i \leq m$ has valuation $v_{m+i}(b) = c - \varepsilon/m$ if $i \in b$ and $v_{m+i}(b) = 0$ otherwise (bidders have combinatorial valuations in this example, so valuation functions only depend on the bundle of items received). The VCG auction will allocate item $i \in \{1, \ldots, m\}$ to bidder i. The payment collected from bidder i will be $c - \varepsilon/m$, which is the second highest value for item i. The revenue from VCG is thus $mc - \varepsilon = W(S) - \varepsilon$. Now, suppose each bidder participates in the auction independently with probability p. The expected revenue can be computed by breaking it up across items:

$$\mathbb{E}[\mathsf{Rev}_{VCG}(S_0)] = \sum_{i=1}^{m} \mathbb{E}[\text{Revenue from item } i] = \sum_{i=1}^{m} p^2(c - \varepsilon/m) = p^2(W(S) - \varepsilon).$$

The third equality is due to the fact that VCG generates nonzero revenue from item i if and only if both bidders i and $m + i$ participate, since if at most one of them shows up there is no competition for that item. So $\mathbb{E}[\mathsf{Rev}_{VCG}(S_0)] = p^2 \cdot \mathsf{Rev}_{VCG}(S)$.

Furthermore, if bidders' valuations are *allocational*, that is, $v_i(\alpha)$ can depend on what items other bidders receive, revenue loss can be even more dramatic.

Theorem 10.2.1. *For any $\varepsilon > 0$ there exists a set S of bidders with allocational valuations such that for $S_0 \sim_p S$, $\mathbb{E}[\mathsf{Rev}_M(S_0)] \leq p^{m/2} \cdot (\mathsf{Rev}_{VCG}(S) + 2\varepsilon) + \varepsilon$ for any individually rational auction M.*

Proof. For each item $1 \leq i \leq m/2$ we introduce two buyers with valuations $v_{i,1}$, $v_{i,2}$. For each item $m/2 + 1 \leq j \leq m$ we introduce a single buyer with valuation v_j. For $1 \leq i \leq m/2$ valuations $v_{i,1}$ are defined by $v_{i,1}(\alpha) = c$ if bidder $(i,1)$ is allocated item i and bidders $j = m/2+1, \ldots, m$ each receive at least one item, and $v_{i,1}(\alpha) = 0$ otherwise. Valuations $v_{i,2}$ are defined by $v_{i,2}(\alpha) = c - 2\varepsilon/m$ if bidder $(i,2)$ is allocated item i and bidders $j = m/2+1, \ldots, m$ each receive at least one item, and $v_{i,2}(\alpha) = 0$ otherwise. The only requirement on the valuations of bidders $j = m/2+1, \ldots, m$ is that $v_j(\alpha) \leq 2\varepsilon/m$ for all α. The VCG auction would allocate item i to bidder $(i,1)$ for each $i = 1, \ldots, m/2$, and allocate the remaining $m/2$ items to bidders $j = m/2+1, \ldots, m$ such that each bidder j receives exactly one item. The welfare of this (efficient) allocation is at most $cm/2 + \varepsilon$. The revenue obtained by VCG is at least $cm/2 - \varepsilon = W(S) - 2\varepsilon$. Let S^* denote the set of small-valuation bidders $j = m/2+1, \ldots, m$. If each bidder shows up independently with probability p, the expected revenue of any auction M is

$$\begin{aligned}
\mathbb{E}[\mathsf{Rev}_M(S_0)] &= \mathbb{E}[\mathsf{Rev}_M(S_0) \mid S^* \subseteq S_0] \cdot \Pr(S^* \subseteq S_0) + \mathbb{E}[\mathsf{Rev}_M(S_0) \mid S^* \nsubseteq S_0] \cdot \Pr(S^* \nsubseteq S_0) \\
&\leq p^{m/2} \cdot \mathbb{E}[\mathsf{Rev}_M(S_0) \mid S^* \subseteq S_0] + \mathbb{E}[\mathsf{Rev}_M(S_0) \mid S^* \nsubseteq S_0] \\
&\leq p^{m/2} \cdot \mathbb{E}[W(S_0) \mid S^* \subseteq S_0] + W(S^*) \\
&\leq p^{m/2} \cdot W(S) + \varepsilon,
\end{aligned}$$

as desired. □

The exponential revenue decay in the number of items means that even if the shrunken market is large in expectation, the revenue loss can be dramatic. For example, if 50 items are for sale and each bidder shows up independently with 90% probability, our construction shows that any auction can guarantee only at most 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.

10.2.3 Main guarantee on preserved revenue

We now present our main revenue guarantee when each bidder participates in the auction independently with probability p. For a set of bidders $S' \subseteq S$, let $\omega(S') = \mathsf{win}_{\mathcal{M}}(S') \cup (\cup_{i \in S'} \mathsf{win}_{\mathcal{M}}(S' \setminus \{i\}))$ be the set of bidders in S' whose valuations determine $\mathsf{Rev}_{\mathcal{M}}(S')$. Define an equivalence relation $\equiv$ on subsets of S by $S_1 \equiv S_2$ if and only if $\omega(S_1) = \omega(S_2)$. Let $\varphi(S') = \frac{1}{n} \sum_{i=1}^{n} W_{\mathcal{M}}(S' \setminus \{i\})$. φ serves as a potential function in the proof of the following theorem and represents the average max-welfare of S' when a uniformly random bidder in S drops out.

<u>Mechanism $\mathcal{A}$</u>
(1) Let $S_1, \ldots, S_\ell$ be an enumeration of the equivalence classes with $\varphi(S_i) > \frac{p}{4} W_{\mathcal{M}}(S)$.
(2) Let $M_1, \ldots, M_\ell$ denote the mechanisms that achieve $\mathsf{Rev}_{\mathcal{M}}(S_1), \ldots, \mathsf{Rev}_{\mathcal{M}}(S_\ell)$.
(3) Choose M uniformly at random from $\{M_1, \ldots, M_\ell\}$, and run M.

The main challenge in analyzing this mechanism is bounding ℓ. Before we do that, we analyze the revenue guarantee it satisfies in terms of ℓ.

Lemma 10.2.2. *Let* $|S| \geq 2$. *For* $S_0 \sim_p S$, $\mathbb{E}[\mathsf{Rev}_{\mathcal{A}}(S_0)] \geq \Omega(p^2/\ell) W_{\mathcal{M}}(S)$.

Proof. By definition of ω, if $S_1 \equiv S_2$, then $W_{\mathcal{M}}(S_1) = W_{\mathcal{M}}(S_2)$, $\varphi(S_1) = \varphi(S_2)$, and $\mathsf{Rev}_{\mathcal{M}}(S_1) = \mathsf{Rev}_{\mathcal{M}}(S_2)$ (and the maximum revenue is achieved by the same $M \in \mathcal{M}$ for both sets). Call a set of bidders $S' \subseteq S$ *heavy* if $\varphi(S') > \frac{p}{4} W_{\mathcal{M}}(S)$. If S_0 is heavy, there is $M \in \{M_1, \ldots, M_\ell\}$ such that $\mathsf{Rev}_M(S_0) = W_{\mathcal{M}}(S_0)$, so $\mathbb{E}_{\mathcal{A}}[\mathsf{Rev}_{\mathcal{A}}(S_0)] \geq \frac{1}{\ell} W(S_0) \geq \frac{1}{\ell} \varphi(S_0) > \frac{p/4}{\ell} W_{\mathcal{M}}(S)$. Let H denote the event that S_0 is heavy. Then,

$$\mathop{\mathbb{E}}_{\mathcal{A}}[\mathop{\mathbb{E}}_{S_0}[\mathsf{Rev}_{\mathcal{A}}(S_0)]] = \mathop{\mathbb{E}}_{S_0}\left[\mathop{\mathbb{E}}_{\mathcal{A}}[\mathsf{Rev}_{\mathcal{A}}(S_0)]\right] \geq \mathop{\mathbb{E}}_{S_0}\left[\mathop{\mathbb{E}}_{\mathcal{A}}[\mathsf{Rev}_{\mathcal{A}}(S_0) | H]\right] \cdot \Pr(H) \geq \frac{p/4}{\ell} W(S) \cdot \Pr(H).$$

We now derive a lower bound on $\Pr(H)$. We have

$$\mathop{\mathbb{E}}_{S_0}[\varphi(S_0)] = \mathop{\mathbb{E}}_{S_0}\left[\frac{1}{n}\sum_{i=1}^{n} W_{\mathcal{M}}(S_0 \setminus \{i\})\right] = \mathop{\mathbb{E}}_{i \sim S}\left[\mathop{\mathbb{E}}_{S_0}[W_{\mathcal{M}}(S_0 \setminus \{i\})]\right] \geq \frac{p}{2} W_{\mathcal{M}}(S)$$

where in the final inequality we use the fact that $|S| \geq 2$ and that $W_{\mathcal{M}}$ is submodular, and so by Hartline et al. [2008], $\mathbb{E}_{S_0}[W_{\mathcal{M}}(S_0)] \geq p W_{\mathcal{M}}(S)$. By Markov's inequality on the (nonnegative) random variable $W_{\mathcal{M}}(S) - \varphi(S_0)$,

$$\Pr(S_0 \text{ is heavy}) \geq \frac{(p/2) W_{\mathcal{M}}(S) - (p/4) W_{\mathcal{M}}(S)}{W_{\mathcal{M}}(S) - (p/4) W_{\mathcal{M}}(S)} = \frac{p/4}{1 - p/4} \geq \frac{p}{4}.$$

Substituting this into our previous bound yields $\mathbb{E}_{\mathcal{A}}[\mathbb{E}_{S_0}[\mathsf{Rev}_{\mathcal{A}}(S_0)]] \geq \frac{p^2}{16\ell} W_{\mathcal{M}}(S)$, as desired. □

We now bound the number of heavy equivalence classes ℓ. In order to do this, we introduce the notion of a *winner diagram*, which is a subgraph of the Hasse diagram of S. The winner diagram for S is the following directed graph $\mathcal{G}$: each node is labeled $(S', \omega(S'))$ for some subset

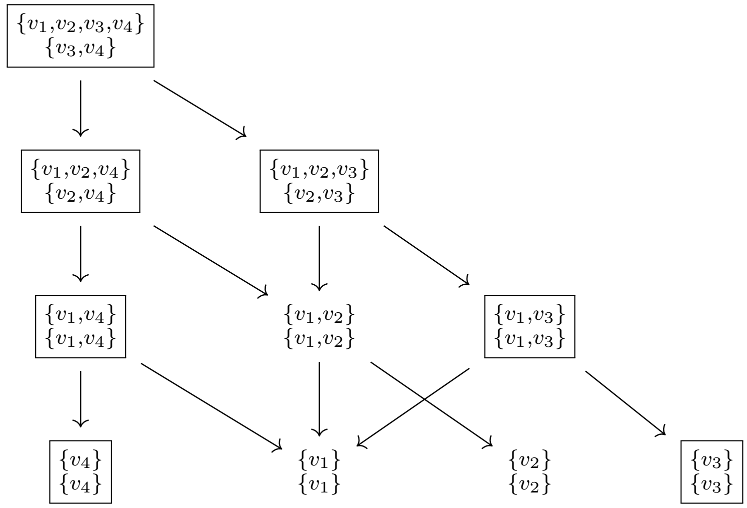


Figure 10.2: A winner diagram representing a second-price auction with a single item and four bidders with valuations $S = \{v_1 = 1, v_2 = 2, v_3 = 4, v_4 = 8\}$. At each node, the top set S' is the set of remaining bidders, and the bottom set is the set of bidders $\omega(S')$ that actually determine revenue. Boxed nodes represent heavy equivalence classes for $p = 8/9$, which is the subgraph of the winner diagram $\mathcal{A}$ randomizes over.

$S' \subset S$. The root node is labeled $(S, \omega(S))$. The children of node $(S', \omega(S'))$ are given by $(S' \setminus \{i\}, \omega(S' \setminus \{i\}))$ for each $i \in \omega(S')$. Figure 10.2 illustrates the winner diagram corresponding to a second-price auction for a single item with four bidders. Winner monotonicity will allow us to show that $\mathcal{G}$ contains a node that represents every equivalence class of $\equiv$.

Lemma 10.2.3. *$\mathcal{G}$ contains all equivalence classes of $\equiv$.*

Proof. Let $S^* \subseteq S$ be a set of bidders that arises as a winner set, that is, $S^* = \omega(S'')$ for some $S'' \supseteq S^*$. The set $S' \supset S'' \supset S^*$ is *maximal* for S^* if $\omega(S') = S^*$ and $\omega(S' \cup \{i\}) \neq S^*$ for every $i \notin S'$. We show that for a given winner set of bidders S^*, there is a unique maximal set of bidders $S' \supseteq S^*$ such that $\omega(S') = S^*$. Initialize $S' = S^*$, and greedily add bidders from S to S' while $\omega(S') = S^*$ does not change. Due to winner monotonicity, if $i \notin \omega(S')$, then $i \notin \omega(S' \cup \{j\})$ for any bidder j. Hence, the order in which bidders are added by the greedy procedure does not matter, and therefore the final set S' is the unique maximal set for S^*. Let the representative element of each equivalence class $[(S', \omega(S'))]$ be the one such that S' is maximal for $\omega(S')$.

We prove the lemma by backwards induction on the size of the representative set S' of any equivalence class. The base case of $|S'| = n$ is immediate since the root $(S, \omega(S))$ is the only node for which the representative set has size n. For the inductive step suppose that $\mathcal{G}$ contains a node for every equivalence class for which the representative set is of size at least n'. Let $(S', \omega(S'))$ be the representative of an equivalence class with $|S'| = n' - 1$. Let $i \notin S'$ be a bidder such that $i \in \omega(S' \cup \{i\})$. Such an i exists due to winner monotonicity: if $i \in \omega(S)$, then $i \in \omega(S' \cup \{i\})$, since $S' \cup \{i\} \subset S$. Let S'' be the maximal set such that $\omega(S'') = \omega(S' \cup \{i\})$. We have $|S''| \geq |S' \cup \{i\}| > |S'|$, so by the induction hypothesis $\mathcal{G}$

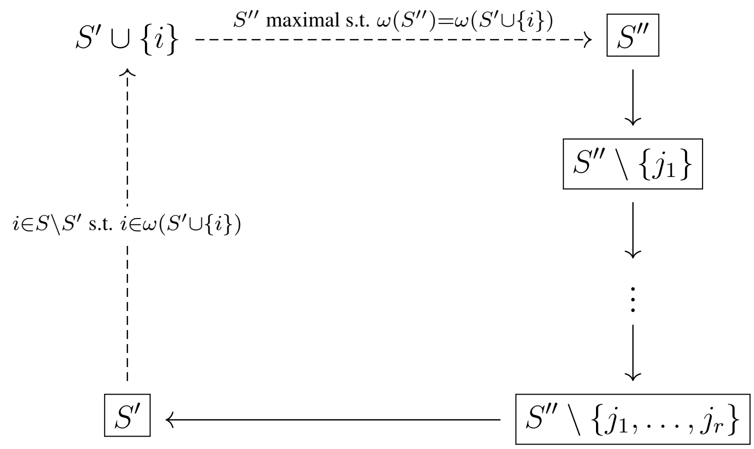


Figure 10.3: Illustration of the inductive step in Lemma 10.2.3. Boxed sets correspond to representative elements of equivalence classes in $\mathcal{G}$. Solid arrows represent directed edges in $\mathcal{G}$ from parent to child.

contains a node labelled $(S'', \omega(S''))$. Now, there must exist a bidder $j_1 \in S'' \setminus S'$ such that $j_1 \in \omega(S'')$. If not, adding all the bidders in $S'' \setminus S'$ to S' would not introduce any new winners, that is, $\omega\left(S' \cup (S'' \setminus S')\right) = \omega\left(S''\right) = \omega\left(S'\right)$, which contradicts the maximality of S' for $\omega(S')$. Therefore, the node $(S'', \omega(S''))$ has a child

$$(S'' \setminus \{j_1\}, \omega\left(S'' \setminus \{j_1\}\right))$$

($S'' \setminus \{j_1\}$ is maximal due to winner monotonicity). We may now find a bidder $j_2 \in S'' \setminus \{j_1\} \setminus S'$ such that $j_2 \in \omega(S'' \setminus \{j_1\} \setminus S')$ for the same reason as before. Continuing in this fashion yields a path from $(S'', \omega(S''))$ to $(S', \omega(S'))$, so $(S', \omega(S')) \in \mathcal{G}$, as desired. □

Combining Lemmas 10.2.2 and 10.2.3 yields our main guarantee. Let $\gamma = \max_{S', i \in \omega(S')} \frac{\varphi(S' \setminus \{i\})}{\varphi(S')}$ and let $k = \max_{S'} |\omega(S')|$. We have $k \leq 2m$. The parameter γ measures the smallest decrease in φ between any two levels of $\mathcal{G}$, which we use to control the depth of nodes considered by our main mechanism $\mathcal{A}$. We stipulate that $\gamma < 1$. The full version of the paper discusses how to remove this assumption and replace γ with an appropriate parameter that is unconditionally strictly less than 1.

Theorem 10.2.4. *Let $|S| \geq 2$ and $S_0 \sim_p S$. We have*

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

If $\mathcal{M}$ consists of revenue-monotonic mechanisms the slightly improved bound $\Omega(\frac{p^2}{k^{\log_{1/\gamma}(4/p)}}) W_{\mathcal{M}}(S)$ holds. In particular, there exist mechanisms in $\mathcal{M}$ achieving the above guarantees in expectation.

Proof. Let $\mathcal{G}'$ denote the restriction of the winner diagram $\mathcal{G}$ to nodes representing heavy equivalence classes. Each node of $\mathcal{G}'$ has out-degree at most $k = \max_{S'} |\omega(S')|$, and the depth of $\mathcal{G}'$ is at most $\log_{1/\gamma}\left(\frac{W_{\mathcal{M}}(S)}{(p/4)W_{\mathcal{M}}(S)}\right) = \log_{1/\gamma}(4/p)$ since φ decreases by a factor of at least γ when passing from a parent node to a child node (and $\mathcal{G}$ is truncated at nodes that are not heavy). Hence the number of nodes in $\mathcal{G}'$ is at most $k^{1+\log_{1/\gamma}(4/p)}$. If mechanisms in $\mathcal{M}$ are revenue monotonic, then we may modify $\mathcal{A}$ to randomize only over mechanisms corresponding to nodes of $\mathcal{G}'$ with out-degree 0. The number of such nodes is at most $k^{\log_{1/\gamma}(4/p)}$. By Lemma 10.2.3, we may substitute this quantity for ℓ in Lemma 10.2.2, which completes the proof. □

We have thus shown, via an application of the probabilistic method, that if $\mathcal{M}$ is a sufficiently rich mechanism class, there always exists $M \in \mathcal{M}$ that is robust to uncertainty in the market. Thus, $\sup_{M\in\mathcal{M}} \mathbb{E}[\mathsf{Rev}_M(S_0)] \geq \Omega(\frac{p^2}{k^{1+\log_{1/\gamma}(4/p)}})W_{\mathcal{M}}(S)$, with a slight improvement under revenue monotonicity.

Applications

The dependence of Theorem 10.2.4 on $\max_{S'} |\omega(S')|$ allows us to derive interesting families of guarantees when the seller places practical constraints on the auction setting. The first is a constraint on the number of winners, and the second is a bundling constraint that favors allocations that sell certain items together. Reasons for limiting the number of winners include: (1) avoiding the logistical hassle of having a large number of winners – this constraint is commonly used in sourcing auctions [Hohner et al., 2003, Sandholm et al., 2006, Sandholm, 2007, 2013] and (2) increasing competition to boost revenue, as is studied in Roughgarden et al. [2020] (though in a different setting than ours). Kroer and Sandholm [2015] show that even the vanilla VCG auction run with bundling constraints can yield significant revenue gains compared to VCG with no bundling constraints.

Limiting the number of winners Suppose $\mathcal{M}$ is a class of mechanisms such that $|\mathsf{win}_M(S)| \leq n_0$ for all $M \in \mathcal{M}$ such that $W_{\mathcal{M}}$ is submodular, winner-monotonic, and satisfies the global-VCG-like property discussed previously. The proofs of all previous theorems go through with this constraint taken into account, with parameters modified correspondingly. Let φ, γ be defined as previously.

Theorem 10.2.5. *Let $|S| \geq 2$ and $S_0 \sim p$, and let $\mathcal{M}$ be a class of mechanisms that sell to at most n_0 bidders. Then, there exists $M \in \mathcal{M}$ such that*

$$\mathbb{E}[\mathsf{Rev}_M(S_0)] \geq \Omega\left(\frac{p^2}{(2n_0)^{1+\log_{1/\gamma}(4/p)}}\right) W_{\mathcal{M}}(S).$$

In practical settings the auction designer might limit the number of bidders that can win a nonempty bundle of items – and in such cases n_0 can potentially be treated as a constant relative to m and n.

Bundling constraints A *bundling* is a partition of the set of items $\{1, \dots, m\}$. An allocation α *respects* a bundling ϕ if no two items in the same bundle according to ϕ are allocated to different buyers. For a set of bundlings Φ, the class of Φ-*boosted* λ-*auctions* consists of all λ-auctions satisfying $\lambda(\alpha) \geq 0$ for all α that respects a bundling in Φ and $\lambda(\alpha) = 0$ otherwise. Let $W^{\Phi}(S)$ denote the maximum welfare of any allocation that respects a bundling in Φ. In the following theorem statement, $\varphi(S') = \frac{1}{n}\sum_{i=1}^{n} W^{\Phi}(S' \setminus \{i\})$.

Theorem 10.2.6. *Let Φ be a set of bundlings. Let S be a set of $n \geq 2$ bidders with valuations such that $W^{\Phi} : 2^S \to \mathbb{R}_{\geq 0}$ is submodular. Let $\gamma = \max_{S', i \in \omega(S')} \frac{\varphi(S' \setminus \{i\})}{\varphi(S')}$. Let m_0 be the greatest number of bundles in any bundling in Φ. Let $\mathcal{M}$ be the class of Φ-boosted λ-auctions. Then,*

$$\sup_{\lambda \in \mathcal{M}} \mathop{\mathbb{E}}_{S_0 \sim_p S}[\mathsf{Rev}_{\lambda}(S_0)] \geq \Omega\left(\frac{p^2}{(2m_0)^{1+\log_{1/\gamma}(4/p)}}\right) W^{\Phi}(S).$$

Proof. At most m_0 bidders can win a nonempty bundle of items, so $\max_{S'} |\omega(S')| \leq 2m_0$ by the same reasoning used to prove Theorem 10.2.5. The arguments used to prove Theorem 10.2.4 yield the desired bound. □

General distribution over submarkets Our proof techniques easily generalize to handle any distribution D over subsets of bidders since the only statistic of the distribution required is the expected welfare of a random subset of bidders $\mathbb{E}_{S_0 \sim_D S}[W_{\mathcal{M}}(S_0)]$. When bidders participated independently with probability p, submodularity of the welfare function was required to ensure that $\mathbb{E}[W_{\mathcal{M}}(S_0)] \geq pW_{\mathcal{M}}(S)$. In the following more general guarantee, which is in terms of $\mathbb{E}[W_{\mathcal{M}}(S_0)]$, we only need the more general condition of winner monotonicity.

Theorem 10.2.7. *Let S be a set of $n \geq 2$ bidders with valuations that satisfy winner monotonicity. Let D be a distribution supported on 2^S with $\mathbb{E}_{S_0 \sim_D S}[W_{\mathcal{M}}(S_0)] = \mu \cdot W_{\mathcal{M}}(S)$. Let $\gamma = \max_{S', i \in \omega(S')} \frac{\varphi(S' \setminus \{i\})}{\varphi(S')}$ and let $k = \max_{S'} |\omega(S')|$. We have*

$$\sup_{M \in \mathcal{M}} \mathop{\mathbb{E}}_{S_0 \sim_D S}[\mathsf{Rev}_M(S_0)] \geq \frac{\eta\mu}{k^{1+\log_{1/\gamma}(1/\eta\mu)}} \left(\frac{\mu - 2\eta\mu}{2(1-\eta\mu)}\right) \cdot W_{\mathcal{M}}(S)$$

for all $0 \leq \eta \leq 1/2$.

Proof. The proof is nearly identical to that of Theorem 10.2.4. The main modification is that $S' \subset S$ is heavy if $\varphi(S') \geq \eta\mu \cdot W_{\mathcal{M}}(S)$, and $\mathcal{A}$ randomizes over mechanisms corresponding to sets S' with this property. Then, $\mathbb{E}_{S_0 \sim_D S}[\varphi(S_0)] \geq \frac{\mu}{2} W_{\mathcal{M}}(S)$ and so Markov's inequality yields $\Pr(S_0 \text{ is heavy}) \geq \frac{\mu/2 - \eta}{1-\eta}$. The remainder of the proof is identical. □

Versions of Theorems 10.2.5 and Theorems 10.2.6 for general distributions can be similarly obtained.

10.2.4 How to choose an auction

Computing the mechanism $M \in \mathcal{M}$ that achieves the revenue guarantee of Theorem 10.2.4 can be accomplished by searching over the set $\{M_1, \ldots, M_\ell\}$ that $\mathcal{A}$ randomizes over, but this would potentially be a highly-inefficient procedure. Moreover, $\mathcal{A}$ itself is not computationally-efficient: determining the heavy sets of bidders, and determining the mechanisms $M_1, \ldots, M_\ell$ that are revenue maximizing for the heavy sets is an exhaustive procedure that would require enumerating over a potentially exponential number of subsets of S.

A more natural way for the mechanism designer to arrive at a mechanism is to learn from samples, which ensures that the mechanism designer uses the auction that (nearly) optimizes the expected preserved revenue, which could be significantly higher than what Theorem 10.2.4 guarantees.

We give a learning algorithm that the mechanism designer can use to learn a mechanism $\widetilde{M} \in \mathcal{M}$ that achieves an expected revenue of nearly $\sup_{M\in\mathcal{M}} \mathbb{E}[\mathsf{Rev}_M(S_0)]$. Our algorithm is similar in spirit to the learning-within-an-instance paradigm of Balcan et al. [2021c]. To describe the algorithm, we require the structural notion of mechanism *delineability* introduced by Balcan et al. [2018d] (discussed in the previous section).

Theorem 10.2.1. *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 $N = O\left(\frac{d\log(dh)}{\varepsilon^2}\log(\frac{1}{\delta})\right)$.

Proof. We design a mechanism $\widetilde{M}$ such that $\mathbb{E}[\mathsf{Rev}_{\widetilde{M}}(S_0)] \geq \sup_{M\in\mathcal{M}} \mathbb{E}[\mathsf{Rev}_M(S_0)] - \varepsilon$ with high probability. The theorem statement then follows from Theorem 10.2.4. Our algorithm is based on the framework of empirical risk minimization from machine learning. The mechanism designer samples $S_1, \dots, S_N \subseteq S$ independently and identically according to distribution D on 2^S. (We assume for simplicity that sampling according to D can be done in a computationally efficient manner. If bidders participate independently with probability p, then the mechanism designer simply needs to flip N coins of bias p for each of the n bidders in S.) The auction used will be the one that maximizes empirical revenue $\widetilde{M} = \operatorname{argmax}_{M\in\mathcal{M}} \frac{1}{N}\sum_{t=1}^{N} \mathsf{Rev}_M(S_t)$. Balcan et al. [2018d] show that $N = O\left(\frac{d\log(dh)}{\varepsilon^2}\log(\frac{1}{\delta})\right)$ samples suffice to guarantee that the expected revenue of $\widetilde{M}$ is ε-close to optimal with probability at least $1-\delta$ over the draw of $S_1, \dots, S_N$.

We now determine the computational complexity of maximizing empirical revenue. Our algorithm exploits similar geometric intuition that was used by Balcan et al. [2018d] to derive the above sample complexity guarantee. A similar approach has been used in other settings as well [Balcan et al., 2020c, 2021c].

For each $S_t \in \{S_1, \dots, S_n\}$, let $\mathcal{H}_t$ denote the set of at most h hyperplanes witnessing (d, h)-delineability of $\mathcal{M}$, and let $\mathcal{H} = \cup_t \mathcal{H}_t$, so $|\mathcal{H}| \leq Nh$. The number of connected components of $\mathbb{R}^d \setminus \mathcal{H}$ is at most $|\mathcal{H}|^d \leq (Nh)^d$. Each connected component is a convex polyhedron that is the intersection of at most $|\mathcal{H}|$ halfspaces. Representations of these regions as $0/1$ constraint-vectors of length $\mathcal{H}$ (a 0 in entry $h \in \mathcal{H}$ corresponds to one side of h, a 1 corresponds to the other side) can be computed in $\mathrm{poly}(|\mathcal{H}|^d)$ time using standard techniques [Tóth et al., 2017]. Empirical revenue is linear as a function of θ in each connected component due to delineability, so the parameter θ that maximizes empirical revenue within a given component can be found by solving a linear program that involves d variables and at most $|\mathcal{H}|$ constraints, which can be done in $\mathrm{poly}(|\mathcal{H}|, d)$ time. □

Our algorithm has a run-time that is exponential in the number of parameters d required to describe mechanisms in $\mathcal{M}$. In the full version of the paper, we study a class of sparse λ-auctions that can be described by a constant number of parameters. By leveraging practically-efficient routines for winner determination [Sandholm, 2002a, Sandholm and Suri, 2003, Sandholm et al., 2005] (a generalization of the problem of computing welfare-maximizing allocations), we show how our empirical revenue maximization is computationally tractable for this setting (in particular, the run-time T of computing the hyperplanes witnessing delineability is in terms of the run-time of winner determination).

10.2.5 Conclusions and future research

Our work in this section is the first to formally study the problem of preserving revenue in a shrinking market via a simple and natural model. We gave a sample-based learning algorithm to design a mechanism that is robust to shrinkage and uncertainty in the market. The crux of our analysis was a new combinatorial construction we introduced called a winner diagram.

There are several open questions and new interesting research directions that stem from this study. The most immediate question is to derive tight bounds on revenue loss. There is a gap between the bound of Theorem 10.2.4 and the $(1-p^2)$-fraction revenue loss of the simple example of a market with competition. Where does the true answer lie? Another interesting, and seemingly more difficult, setting is the one where the mechanism designer does not know the distribution D over 2^S beforehand. Can he still arrive at an auction that is robust to the shrinking market? If the mechanism designer knows that each bidder participates independently with probability p, but does not know p, is it still possible to design a robust auction? Finally, we believe that the combinatorial bidder structure uncovered by our notion of a winner diagram could have interesting applications to other areas in mechanism design. While our analysis required a number of assumptions on the set of bidders, it would be interesting to extend the concept of a winner diagram to prove more general results with weaker assumptions. It would be interesting to extend our techniques to understand market shrinkage in other settings including objectives beyond revenue, other auction classes, and unlimited supply.

Chapter 11

Conclusions and Future Research Directions

Mechanism design and integer programming have been used to great avail to drive the efficient operation of various markets for several decades. In this thesis we pushed the boundary of possibilities at the convergence of mechanism design, integer programming, and machine learning.

There are many remaining open problems in the threads presented so far in this thesis, and all of those are important directions for future research. In this closing chapter, we focus on new directions. These directions will directly use the knowledge and tools involved the research covered in this thesis, spanning the development of new theory to practical implementation.

Side Information and Computation in Mechanism Design

Type space models To further empirically test our proposed approaches, realistic distributions/generators of side information will be needed. Directly building atop the Combinatorial Auction Test Suite (CATS) [Leyton-Brown et al., 2000] that is a mainstay of auction design research (some other lesser-studied generators worth looking into include the Spectrum Auction Test Suite (SATS) [Weiss et al., 2017], the spectrum auction generators in Bichler et al. [2023], and the distributions for TV advertisement markets in Goetzendorff et al. [2015]) is a promising start. Such side-information distributions should have direct ties to information an auctioneer would have access to in real settings: some examples include known appraisal values for items, known relative spending powers of different bidders, known complementarity/substitutability structures on the bids, *etc.* The *interdependent values model* of Milgrom and Weber [1982] might provide some guidance here.

Learning sophisticated and accurate type space models from data is also an interesting future research direction. In our work on improved pricing structures and mechanisms that exploit the weakest type consistent with the type space model (Chapters 6 and 7), it is critical that an agent's true type is contained in the type space. In other words, an agent's true type must be consistent with the mechanism designer's knowledge. There are two challenges to the learning problem that arises. First, the only examples seen by the learner/mechanism designer are positively-labeled examples, so appropriate learning-theoretic techniques that deal with positive-only examples must be ported over [Shvaytser, 1990, Denis, 1998]. Second, the usual notion of classification error is insufficient to judge the merits of a learned type space. That is because a learned type space that contains the true type space can still be used to run an efficient mechanism in an IC and IR way, but a learned type space that misses parts of the type space risks IR violations. Here, the notion of one-sided error introduced by Valiant [1984] (see also Natarajan [1987] and Kalai et al. [2012]) might be amenable to better capture the learning problem.

One can also interpret the constraints defining agents' type spaces simply as *rules of the market*, whether or not agents' true types satisfy those constraints. This interpretation could beget a more practical use of the resulting pricing structures that we developed in this thesis, since ultimately the mechanism designer might not know, with certainty, too much about true private types. The resulting design questions involving incentives, efficiency, equilibrium, and implementation form a compelling research direction.

Side information languages What is the best way to express side information? This analogous question in the context of bid expression spurred a productive line of research on *bidding languages* (starting with Sandholm [2002a] and Nisan [2000], see also, *e.g.,* Boutilier and Hoos [2001]). Can the paradigm of *expressive bidding* [Sandholm, 2007] be used as an analogy for a new paradigm, *expressive side information*, that might enable significant economic improvements? As a first step, a syntactic characterization of side information is needed. In this thesis we have only dealt with expressions of side information compatible with standard optimization paradigms like linear programming. How the expression of information affects the resulting computational tasks is a wide-open research question (owing in part to the fact that the study of weakest-competitor computation has only been initiated in the past two years by the work covered in this thesis [Balcan et al., 2023, 2025c, Prasad et al., 2025a,b]).

Iterative combinatorial auctions and nonlinear pricing Market clearing in combinatorial auctions is a fundamental problem [Bikhchandani and Ostroy, 2002]. 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 [Lahaie and Lubin, 2019] 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 [Ausubel et al., 2006]). This procedure involves cut generation and column generation, and a deeper dive into the integer programming techniques called for here is a fruitful research direction. How can side information guide the design of such procedures?

New Cutting Plane Generation and Selection Techniques

Rank-2 Gomory cuts 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 the procedure just described, 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 applied to the tableau cuts are guaranteed to produce a cut that separates the LP optimum. We thus obtain an infinite family of cuts that can be optimized over (either exactly or with machine learning). Cornuéjols et al. [2003] and Andersen et al. [2005] 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, to obtain rank-2 cuts. Fischetti and Lodi [2007] study optimization of rank-1 CG cut multipliers. Chételat and Lodi [2023] also study a similar idea of running optimization algorithms like gradient descent to tune GMI multipliers.

Dominance relations for lifted cover inequalities In our work on sequence-independent lifting [Prasad et al., 2024] 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 [Balas et al., 1996b].

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 in Chapter 3 [Prasad et al., 2024].

Efficient algorithms and online learning for cut configuration Development of the algorithmic aspects of learning to cut remains an important open direction. Empirical risk minimization (ERM) algorithms enjoy the sample complexity guarantees established in our work, but are too computationally expensive to be relevant in practice. And while there is a growing corpus of attempts to make machine learning work for cutting plane configuration, none of those have been integrated yet into integer programming solvers in a meaningful way. (Solvers like Gurobi offer parameter tuning capabilities, but these only operate on the predefined parameters that the solver allows the user to configure.) Much work remains to truly enhance the decisions throughout branch-and-cut tree search in a fine-grained way using machine learning.

One approach is to shift to an online-learning setting, rather than the distributional setting that has been studied in this thesis, since no IID assumptions about the data-generating process are needed. Many tools for analyzing data-driven algorithm design in the online learning setting have been developed [Balcan et al., 2018b, 2020b, 2021b, Sharma et al., 2020], and application of those to branch-and-cut parameter tuning—with the end goal of practical gains over default solver settings—is a ripe direction for future research.

Large-Scale Integer Programming and Applications to Mechanism Design

Branch-and-price is an algorithm for solving huge integer programs that are too large to be represented explicitly in memory (which is often the case for integer programs arising in modern applications). Branch-and-price solvers have not been commercialized nor have they achieved the same widespread use as branch-and-cut integer programming solvers (despite the many successful developments of custom branch-and-price approaches for kidney exchange [Abraham et al., 2007], vehicle routing, *etc.*) largely due to the optimization expertise required to formulate the important problem-specific components of the algorithm. (How should one formulate and solve the pricing problem? How should one branch? How are cuts generated?) We envision market design as a fertile application area that is ripe for innovations in branch-and-price methods. For example, some of the more difficult design choices of branch-and-price can be guided by what is economically meaningful—and some can even be delegated to market participants (for example, an otherwise intractable pricing subproblem might be solvable by asking an agent to make some decision). Finally, machine learning for branch-and-price configuration is a promising research area that is in its infancy [Morabit et al., 2021, Chi et al., 2022].

Generative AI and Mechanism Design

Large language models (LLMs) present new opportunities for the representation and solving of the large-scale combinatorial allocation problems inherent to most marketplaces. A compelling research agenda here is the principled integration of LLMs to improve the economic performance of mechanisms without compromising any integrity in the form of fairness, incentives, and so on. The research on mechanism design with side information covered in this thesis already takes an important step in this direction, but there are several other challenges in the deployment of market design solutions that LLMs can aid with. A growing body of work has started to explore and pave out this brand new research area [Dütting et al., 2024, Hajiaghayi et al., 2024, Soumalias et al., 2025, Huang et al., 2025].

One of the most important design decisions in any combinatorial market is how to tame communication costs via a tractable language for type/preference expression. LLMs present significant opportunities for innovation along this vein by expanding the possibilities for what kinds of valuations a market participant can express concisely. An LLM that has been fine-tuned based on an agent's preferences can furthermore serve as a bidder on the agent's behalf. Prompting as a bidding language is a promising paradigm here: the agent could describe their high level intent and bidding goal via prompts to the LLM and allow it to choose what to bid on and how to bid.

Critical to these directions is the development of new theory and new models. The Internet boom over two decades ago in-part lead to the development of elegant and far-reaching new concepts in algorithmic game theory and computational mechanism design that in turn directly influenced the design of Internet-based marketplaces that are prevalent today. Generative AI yields such an opportunity as well. Overall, there is massive untapped potential to build and nurture new and improved economic systems that take advantage of the deep interaction between mechanism design and discrete optimization. Rapid advancements in AI will bolster this vision. Building principled pipelines from data to optimization to market design and back will require an interdisciplinary approach drawing from artificial intelligence, economics, and operations research. This thesis represents my first foray towards those goals.

■

Appendix A

Omitted Details About Lifting in Chapter 3

Counterexample to Claim of Gu, Nemhauser, and Savelsbergh

Gu, Nemhauser, and Savelsbergh [2000] remark that if $\mu_1 - \lambda \geq \rho_1$, a large family of superadditive lifting functions can be constructed by considering any nondecreasing function $w(x)$ of $x \in [0, \rho_1]$ with $w(x) + w(\rho_1 - x) = 1$. Consider the class of logistic functions centered at $\rho_1/2$ of the form

$$w_k(x) = \frac{1}{1 + e^{-k(x-\rho_1/2)}}$$

where $k \geq 0$. Each w_k is nondecreasing on $[0, \rho_1]$, and satisfies $w_k(x) + w_k(\rho_1 - x) = 1$ for all x. In the following example, we show that using $w_{0.9}$ for sequence-independent lifting can yield invalid cuts. Moreover, the lifting function $g_{w_{0.9}}$ is not superadditive.

$$\begin{array}{ll} \text{maximize} & 112x_1 + 108x_2 + 107x_3 + 106x_4 + 102x_5 + 84x_6 + 82x_7 \\ \text{subject to} & 112x_1 + 108x_2 + 107x_3 + 106x_4 + 102x_5 + 84x_6 + 82x_7 \leq 268 \\ & \boldsymbol{x} \in \{0,1\}^7 \end{array}$$

An optimal solution is given by $\boldsymbol{x}^* = (0, 0, 0, 0, 1, 1, 1)$, which has an objective value of 268, satisfying the single knapsack constraint with equality. The set $C = \{2, 3, 4\}$ is a minimal cover, and the corresponding minimal cover inequality is $x_2 + x_3 + x_4 \leq 2$. We compute the relevant parameters needed for sequence-independent lifting. We have $\mu_0 = 0, \mu_1 = 108, \mu_2 = 215, \mu_3 = 321, \lambda = 53$, and $\rho_0 = 53, \rho_1 = 52, \rho_2 = 51$. We have that $\mu_1 - \lambda \geq \rho_1$ is satisfied. Thus, the lifting function g_w (truncated to the range $[0, 213]$) is given by

$$g_w(z) = \begin{cases} 0 & 0 < z \leq 55 \\ 1 - w(107 - z) & 55 < z \leq 107 \\ 1 & 107 < z \leq 162 \\ 2 - w(213 - z) & 162 < z \leq 213. \end{cases}$$

Using $g_{w_{0.9}}$ yields the lifted cover inequality

$$x_1 + x_2 + x_3 + x_4 + 0.99x_5 + 0.93x_6 + 0.71x_7 \leq 2.$$

But $0.99 + 0.93 + 0.71 = 2.63 > 2$, so $\boldsymbol{x}^*$ violates this inequality, so the lifted cover inequality is invalid for our problem. Furthermore, $g_{w_{0.9}}$ is not superadditive. We have

$$g_{w_{0.9}}(82) + g_{w_{0.9}}(82) = 0.71 + 0.71 > g_{w_{0.9}}(82 + 82) \approx 1.00.$$

One need not look at logistic functions to derive this counterexample. The step function

$$w(x) = \begin{cases} 0 & x < \rho_1/2 \\ 1/2 & x = \rho_1/2 \\ 1 & x > \rho_1/2 \end{cases}$$

disproves the claim as well (in that g_w is not superadditive, and results in an invalid cut in the above example).

Omitted Proofs

Proof of Theorem 3.1.1

The proof that g_k is superadditive closely follows the proof that g_{1/ρ_1} is superadditive by Gu et al. [2000] (Lemma 1 in their paper), with a couple key modifications. As done in Gu et al. [2000], we will establish superadditivity of a function that is defined slightly more generally.

Given $v_1 > 0$, $u_i \geq 0, u_i \geq u_{i+1}, i = 1, 2, \ldots, v_i \geq 0, v_i \geq v_{i+1}, i = 1, 2, \ldots$ such that $u_i + v_i > 0$ for all i, let $M_0 = 0$ and $M_h = \sum_{i=1}^{h}(u_i + v_i)$ for $h = 1, 2, \ldots, \infty$. Define

$$\tilde{g}_k(z) = \begin{cases} 0 & z = 0 \\ h & M_h < z \leq M_h + u_{h+1}, \qquad h = 0, 1, \ldots \\ h + 1 - w_k(M_{h+1} - z) & M_h + u_{h+1} < z \leq M_{h+1}, \qquad h = 0, 1, \ldots \end{cases}$$

where $w_k(x) = kx + \frac{1-kv_1}{2}$.

The superadditive lifting function g_k is recovered by letting $u_i = a_i - \rho_{i-1}$ for $i \in \{1, \ldots, t\}$, $v_i = \rho_i$ for $i \in \{1, \ldots, t-1\}$. This yields $M_h = \mu_h - \lambda + \rho_h$ and $M_h + u_{h+1} = \mu_{h+1} - \lambda$ (see Gu et al. Gu et al. [2000] for further details).

Lemma A.0.1. *Let $k \in [0, 1/v_1]$. If $u_1 \geq v_1$, the function $\tilde{g}_k$ is superadditive on $[0, \infty)$.*

Proof. We prove that $\max\{\tilde{g}(z_1) + \tilde{g}(z_2) - \tilde{g}(z_1 + z_2) : z_1, z_2 \in [0, \infty)\} \leq 0$, which is equivalent to superadditivity. We break the analysis into cases as done in Gu et al. [2000].

Case 1: $M_{h_1} + u_{h_1+1} < z_1 \leq M_{h_1+1}$ and $M_{h_2} + u_{h_2+1} < z_2 \leq M_{h_2+1}$. Then, as in Gu et al. [2000], $z_1 + z_2 \geq M_{h_1+h_2} + u_{h_1+h_2+1}$. The first subcase of Gu et al. [2000] (Case 1.1) is $z_1 + z_2 \leq M_{h_1+h_2+1}$, that is, $z_1 + z_2$ lies on a "sloped" segment of $\tilde{g}$. We show that the assumption $u_1 \geq v_1$ rules this case out, that is, $z_1 + z_2 > M_{h_1+h_2+1}$.

Proof that $z_1 + z_2 > M_{h_1+h_2+1}$: First, suppose $h_1 = 0$. We have

$$\begin{aligned} z_1 + z_2 &> u_1 + M_{h_2} + u_{h_2+1} \\ &\geq v_{h_2+1} + M_{h_2} + u_{h_2+1} \qquad \text{(since } u_1 \geq v_1 \geq v_{h_2+1}) \end{aligned}$$

$$= M_{h_2+1},$$

as desired. The case where $h_2 = 0$ is symmetric. Thus, suppose $h_1, h_2 \geq 1$, and without loss of generality let $h_1 \leq h_2$ (the case where $h_2 \leq h_1$ is symmetric). We have

$$\begin{aligned}
z_1 + z_2 &> M_{h_1} + u_{h_1+1} + M_{h_2} + u_{h_2+1} \\
&= \sum_{i=1}^{h_1}(u_i + v_i) + u_{h_1+1} + \sum_{i=1}^{h_2}(u_i + v_i) + u_{h_2+1} \\
&= (u_1 + v_1) + \sum_{i=2}^{h_1}(u_i + v_i) + u_{h_1+1} + \sum_{i=1}^{h_2}(u_i + v_i) + u_{h_2+1} \\
&\geq (u_1 + v_1) + \sum_{i=h_2+1}^{h_1+h_2-1}(u_i + v_i) + u_{h_1+1} + \sum_{i=1}^{h_2}(u_i + v_i) + u_{h_2+1} \\
&= M_{h_1+h_2-1} + u_1 + v_1 + u_{h_1+1} + u_{h_2+1}
\end{aligned}$$

Now, $u_{h_1+1} \geq u_{h_1+h_2+1}$ and as $h_1, h_2 \geq 1$, $u_{h_2+1} \geq u_{h_1+h_2}$. Furthermore, $u_1 \geq v_1 \geq v_{h_1+h_2} \geq v_{h_1+h_2+1}$. Therefore, the right-hand-side is at least

$$M_{h_1+h_2-1} + (u_{h_1+h_2} + v_{h_1+h_2}) + (u_{h_1+h_2+1} + v_{h_1+h_2+1}) = M_{h_1+h_2+1},$$

as desired. This completes the proof that under Case 1, $z_1 + z_2 > M_{h_1+h_2+1}$. We now proceed with the casework under Case 1, starting our numbering from Case 1.2 so that the labeling of the cases coincides with Gu et al. [2000].

Case 1.2: $M_{h_1+h_2+1} < z_1 + z_2 \leq M_{h_1+h_2+1} + u_{h_1+h_2+2}$. Then

$$\begin{aligned}
&\tilde{g}(z_1) + \tilde{g}(z_2) - \tilde{g}(z_1 + z_2) \\
&= \left(h_1 + 1 - k(M_{h_1+1} - z_1) - \frac{1 - kv_1}{2}\right) + \left(h_2 + 1 - k(M_{h_2+1} - z_2) - \frac{1 - kv_1}{2}\right) - (h_1 + h_2 + 1) \\
&= 1 - k(M_{h_1+1} + M_{h_2+1} - z_1 - z_2) - (1 - kv_1) \\
&\leq 1 - k(M_{h_1+1} + M_{h_2+1} - M_{h_1+h_2+1} - u_{h_1+h_2+2}) - (1 - kv_1) \\
&= 1 - k\Big((u_1 + v_1) + \sum_{i=2}^{h_1+1}(u_i + v_i) + \sum_{i=1}^{h_2+1}(u_i + v_i) - \sum_{i=1}^{h_1+h_2+1}(u_i + v_i) - u_{h_1+h_2+2}\Big) - (1 - kv_1) \\
&= 1 - k(u_1 + v_1 - u_{h_1+h_2+2}) - k\sum_{i=2}^{h_1+1}((u_i + v_i) - (u_{i+h_2} + v_{i+h_2})) - (1 - kv_1) \\
&= -k\underbrace{(u_1 - u_{h_1+h_2+2})}_{\geq 0} - k\underbrace{\sum_{i=2}^{h_1+1}((u_i + v_i) - (u_{i+h_2} + v_{i+h_2}))}_{\geq 0} \\
&\leq 0.
\end{aligned}$$

Case 1.3: $M_{h_1+h_2+1} + u_{h_1+h_2+2} < z_1 + z_2 \leq M_{h_1+h_2+2}$. Then

$$\tilde{g}(z_1) + \tilde{g}(z_2) - \tilde{g}(z_1 + z_2)$$

$$
\begin{aligned}
&= \left(h_1 + 1 - k(M_{h_1+1} - z_1) - \frac{1-kv_1}{2}\right) + \left(h_2 + 1 - k(M_{h_2+1} - z_2) - \frac{1-kv_1}{2}\right) \\
&\qquad - \left((h_1 + h_2 + 2) - k(M_{h_1+h_2+2} - z_1 - z_2) - \frac{1-kv_1}{2}\right) \\
&= -k(M_{h_1+1} + M_{h_2+1} - M_{h_1+h_2+2}) - \frac{1-kv_1}{2} \\
&\leq -k(M_{h_1+1} + M_{h_2+1} - M_{h_1+h_2+2}) \qquad \text{(since } k \leq 1/v_1\text{)} \\
&\leq 0.
\end{aligned}
$$

The remaining cases (1.4, 2, 3.1 - 3.2 from Gu et al. [2000]) and their proofs follow Gu et al. [2000] verbatim, so we omit them. This completes the proof that $\tilde{g}$ is superadditive. □

Now, the proofs that g is superadditive, maximal, and undominated are identical to those in Gu et al. [2000].

We also show here that the condition $\mu_1 - \lambda \geq \rho_1$ is necessary and sufficient for g_0 (PC lifting) to be superadditive. Theorem 3.1.1 shows that the condition is sufficient. To show necessity, suppose $\mu_1 - \lambda < \rho_1$. Let ε be sufficiently small so that $\mu_1 - \lambda + 2\varepsilon < \rho_1$. We have

$$g_0(\mu_1 - \lambda + \varepsilon) = 1/2$$

and as $2(\mu_1 - \lambda + \varepsilon) \in (\mu_1 - \lambda, \mu_1 - \lambda + \rho_1)$,

$$g_0(2(\mu_1 - \lambda + \varepsilon)) = 1/2,$$

so

$$g_0(2(\mu_1 - \lambda + \varepsilon)) < 2g_0(\mu_1 - \lambda + \varepsilon)$$

violating superadditivity.

Proof of Proposition 3.1.4

We prove Proposition 3.1.4, which states that for any $\varepsilon > 0$ and any $t \in \mathbb{N}$ there exists a knapsack constraint $\boldsymbol{a}^\top \boldsymbol{x} \leq b$ with a minimal cover C of size t such that PC lifting yields $\sum_{j \in C} x_j + \sum_{j \notin C} \frac{1}{2} x_j \leq |C| - 1$ and GNS lifting is dominated by $\sum_{j \in C} x_j + \sum_{j \notin C} \varepsilon x_j \leq |C| - 1$.

Proof. Let $a_1 \geq \cdots \geq a_t$ and let b be such that $a_1 + \cdots + a_t > b$ and $a_1 + \cdots + a_{t-1} \leq b$. Let $\lambda' = a_1 + \cdots + a_t - b$. Furthermore, choose $a_1, \ldots, a_t, b$ so that $a_1 - \lambda' \geq a_2 - a_1 + \lambda' > 0$. Let $M \geq \frac{1}{(a_2 - a_1 + \lambda')\varepsilon}$ and consider the knapsack constraint

$$M(a_1 x_1 + \cdots + a_t x_t) + (1 + M(a_1 - \lambda'))(x_{t+1} + \cdots + x_n) \leq Mb.$$

$C = \{1, \ldots, t\}$ is clearly a minimal cover with $\mu_1 = Ma_1$, $\lambda = M\lambda'$ and $\rho_1 = M(a_2 - a_1 + \lambda')$, and by the choice of $a_1, \ldots, a_t$, $\mu_1 - \lambda = M(a_1 - \lambda') \geq M(a_2 - a_1 + \lambda') = \rho_1$. Hence, PC lifting can be used to yield a valid lifted cut. As $1 + M(a_1 - \lambda') = 1 + (\mu_1 - \lambda)$, we have $g_0(1 + M(a_1 - \lambda')) = \frac{1}{2}$ and

$$g_{1/\rho_1}(1 + M(a_1 - \lambda')) = 1 - \frac{\mu_1 - \lambda + \rho_1 - (1 + \mu_1 - \lambda)}{\rho_1} = \frac{1}{\rho_1} = \frac{1}{M(a_2 - a_1 + \lambda')} \leq \varepsilon.$$

□

Proof of claim in Theorem 3.1.8

In the proof of Theorem 3.1.8 we critically used the claim that given $Q \in \mathcal{Q}(J)$, $|Q| \geq 3$, with h_j such that $a_j \in S_{h_j}$,

$$\sum_{j \in Q} a_j > \mu_{\sum_{j \in Q} h_j - \lfloor |Q|/2 \rfloor} - \lambda.$$

This allowed us to ensure that the constraint induced by every such Q was satisfied by the point $(1/2, \ldots, 1/2)$. We prove this claim here.

Proof. We use the quantities u_h, v_h, M_h defined in Appendix A. We break the proof into two cases. We will use the observation that $u_h \geq v_h$ for any h such that $\rho_h > 0$. This is because for such h, $u_h = u_1 = a_1 - \lambda$, and so $u_h = u_1 \geq v_1 \geq v_h$.
Case 1: $|Q| = 2\ell$ is even. Let (without loss of generality) $Q = \{a_1, \ldots, a_{2\ell}\}$, let h_j be such that $a_j \in S_{h_j}$, and let $H = h_1 + \cdots + h_{2\ell}$. We have

$$\begin{aligned}
\sum_{j=1}^{2\ell} a_j > \sum_{j=1}^{2\ell} \mu_{h_j} - \lambda &= \sum_{j=1}^{2\ell} M_{h_j - 1} + u_{h_j} \\
&\geq M_{H-2\ell} + \sum_{j=1}^{2\ell} u_{h_j} \\
&= \sum_{i=1}^{H-2\ell} (u_i + v_i) + \sum_{j=1}^{2\ell} u_{h_j} \\
&\geq \sum_{i=1}^{H-2\ell} (u_i + v_i) + \sum_{j=\ell+1}^{2\ell} (u_{h_j} + v_{h_j}) \\
&\geq \sum_{i=1}^{H-2\ell} (u_i + v_i) + \sum_{j=1}^{\ell} (u_{H-2\ell+j} + v_{H-2\ell+j}) \\
&= \sum_{i=1}^{H-\ell} (u_i + v_i) \\
&= M_{H-\ell} \\
&= \mu_{H-\ell} - \lambda + \rho_{H-\ell} \\
&\geq \mu_{H-\ell} - \lambda,
\end{aligned}$$

as desired.
Case 2: $|Q| = 2\ell + 1$ is odd. Let $Q = \{a_1, \ldots, a_{2\ell+1}\}$, let h_j be such that $a_j \in S_{h_j}$, and let $H = h_1 + \cdots + h_{2\ell+1}$. We have

$$\sum_{j=1}^{2\ell+1} a_j > \sum_{j=1}^{2\ell+1} \mu_{h_j} - \lambda$$

$$
\begin{aligned}
&= \sum_{j=1}^{2\ell+1} M_{h_j-1} + u_{h_j} \\
&\geq M_{H-(2\ell+1)} + \sum_{j=1}^{2\ell+1} u_{h_j} \\
&= \sum_{i=1}^{H-(2\ell+1)} (u_i + v_i) + \sum_{j=1}^{2\ell+1} u_{h_j} \\
&\geq \sum_{i=1}^{H-(2\ell+1)} (u_i + v_i) + \sum_{j=\ell+1}^{2\ell} (u_{h_j} + v_{h_j}) + u_{h_{2\ell+1}} \\
&\geq \sum_{i=1}^{H-(2\ell+1)} (u_i + v_i) + \sum_{j=1}^{\ell} (u_{H-(2\ell+1)+j} + v_{H-(2\ell+1)+j}) + u_{H-(2\ell+1)+(\ell+1)} \\
&= \sum_{i=1}^{H-\ell-1} (u_i + v_i) + u_{H-\ell} \\
&= M_{H-\ell-1} + u_{H-\ell} \\
&= \mu_{H-\ell} - \lambda,
\end{aligned}
$$

as desired. □

Appendix B

Omitted Details About Plots in Section 4.5

The version of the *facility location* problem we study involves a set of locations J and a set of clients C. Facilities are to be constructed at some subset of the locations, and the clients in C are served by these facilities. Each location $j \in J$ has a cost f_j of being the site of a facility, and a cost $s_{c,j}$ of serving client $c \in C$. Finally, each location j has a capacity κ_j which is a limit on the number of clients j can serve. The goal of the facility location problem is to arrive at a feasible set of locations for facilities and a feasible assignment of clients to these locations that minimizes the overall cost incurred.

The facility location problem can be formulated as the following $0, 1$ IP:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{j \in J} f_j x_j + \sum_{j \in J} \sum_{c \in C} s_{c,j} y_{c,j} \\ \text{subject to} & \displaystyle\sum_{j \in J} y_{c,j} = 1 \qquad\qquad \forall\, c \in C \\ & \displaystyle\sum_{c \in C} y_{c,j} \leq \kappa_j x_j \qquad\quad \forall\, j \in J \\ & y_{c,j} \in \{0,1\} \qquad \forall\, c \in C, j \in J \\ & x_j \in \{0,1\} \qquad\qquad \forall\, j \in J \end{array}$$

We consider the following two distributions over facility location IPs.

First distribution Facility location IPs are generated by perturbing the costs and capacities of a base facility location IP. We generated the base IP with 40 locations and 40 clients by choosing the location costs and client-location costs uniformly at random from $[0, 100]$ and the capacities uniformly at random from $\{0, \ldots, 39\}$. To sample from the distribution, we perturb this base IP by adding independent Gaussian noise with mean 0 and standard deviation 10 to the cost of each location, the cost of each client-location pair, and the capacity of each location.

Second distribution Facility location IPs are generated by placing 80 evenly-spaced locations along the line segment connecting the points $(0, 1/2)$ and $(1, 1/2)$ in the Cartesian plane. The location costs are all uniformly set to 1. Then, 80 clients are placed uniformly at random in the unit square $[0, 1]^2$. The cost $s_{c,j}$ of serving client c from location j is the distance between j and c. Location capacities are chosen uniformly at random from $\{0, \ldots, 43\}$.

In our experiments, we add five cuts at the root of the B&C tree. These five cuts come from the set of Chvátal-Gomory and Gomory mixed integer cuts derived from the optimal simplex tableau of the LP relaxation. The five cuts added are chosen to maximize a weighting of cutting-plane scores:

$$\mu \cdot \texttt{score}_1 + (1 - \mu) \cdot \texttt{score}_2. \tag{B.1}$$

$\texttt{score}_1$ is the *parallelism* of a cut, which intuitively measures the angle formed by the objective vector and the normal vector of the cutting plane—promoting cutting planes that are nearly parallel with the objective direction. $\texttt{score}_2$ is the *efficacy*, or depth, of a cut, which measures the perpendicular distance from the LP optimum to the cut—promoting cutting planes that are "deeper", as measured with respect to the LP optimum. More details about these scoring rules can be found in Balcan et al. [2021d] and references therein. Given an IP, for each $\mu \in [0, 1]$ (discretized at steps of 0.01) we choose the five cuts among the set of Chvátal-Gomory and Gomory mixed integer cuts that maximize (B.1). Figures 4.5a and 4.5b display the average tree size over 1000 samples drawn from the respective distribution for each value of μ used to choose cuts at the root. We ran our experiments in C++ using the IBM ILOG CPLEX C Callable Library, version 20.1.0, with default cut generation disabled.

Bibliography

Vineet Abhishek and Bruce Hajek. Efficiency loss in revenue optimal auctions. In *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010. 5, 5

David Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *ACM Conference on Electronic Commerce (EC)*, pages 295–304, 2007. 11

Tobias Achterberg. Constraint integer programming. 2007. 4.4, 4.4.1

Tobias Achterberg and Roland Wunderling. *Mixed Integer Programming: Analyzing 12 Years of Progress*, pages 449–481. Springer Berlin Heidelberg, 2013. 4.5

Gagan Aggarwal, Gagan Goel, and Aranyak Mehta. Efficiency of (revenue-)optimal mechanisms. In *ACM Conference on Electronic Commerce (EC)*, 2009. 5

Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. Learning-augmented mechanism design: Leveraging predictions for facility location. In *ACM Conference on Economics and Computation (EC)*, 2022. 5, 5

Mohammad Akbarpour, Scott Duke Kominers, Shengwu Li, and Paul R Milgrom. Investment incentives in near-optimal mechanisms. In *ACM Conference on Economics and Computation (EC)*, pages 26–26, 2021. 5.6

Saeed Alaei, Azarakhsh Malekian, and Aravind Srinivasan. On random sampling auctions for digital goods. In *ACM Conference on Electronic Commerce (EC)*, pages 187–196, 2009. 10.1

Nicole P Aliloupour. The impact of technology on the entertainment distribution market: The effects of Netflix and Hulu on cable revenue. 2016. 10.2, 10.2

Alejandro Marcos Alvarez, Quentin Louveaux, and Louis Wehenkel. A machine learning-based approximation of strong branching. *INFORMS Journal on Computing*, 29(1):185–195, 2017. 4, 4.2

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. 11

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. 5, 5

Martin Anthony and Peter Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 1999. 4.1, 4.4, 4.5.3, 10.1.1, 10.1.2

Mark Armstrong and John Vickers. Competitive price discrimination. *RAND Journal of Eco-*

nomics, pages 579–605, 2001. 9.1

Robert J Aumann. Markets with a continuum of traders. *Econometrica*, 1964. 10.1.2

Lawrence Ausubel and Oleg Baranov. A practical guide to the combinatorial clock auction. *Economic Journal*, 127(605), 2017. 5.2, 8.3

Lawrence Ausubel and Oleg Baranov. Core-selecting auctions with incomplete information. *International Journal of Game Theory*, 49(1):251–273, 2020. 6, 6.6

Lawrence Ausubel and Oleg Baranov. The VCG mechanism, the core, and assignment stages in auctions. *Working paper*, 2023. 6, 6.5.1

Lawrence Ausubel and Peter Cramton. Auction design for wind rights. *Report to Bureau of Ocean Energy Management, Regulation and Enforcement*, 1, 2011. 6

Lawrence Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1, 2002. No. 1, Article 1. 6

Lawrence Ausubel and Paul Milgrom. The lovely but lonely Vickrey auction. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 1. MIT Press, 2006. 5, 5.1.1, 6

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

Lawrence Ausubel, Christina Aperjis, and Oleg Baranov. Market design and the FCC incentive auction. *Presentation at the NBER Market Design Meeting*, 2017. 6, 6.5.1

Moshe Babaioff, Yannai A Gonczarowski, and Noam Nisan. The menu-size complexity of revenue approximation. In *ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 869–877, 2017. 9.1

Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S Matthew Weinberg. A simple and approximately optimal mechanism for an additive buyer. *Journal of the ACM (JACM)*, 67(4): 1–40, 2020. 5

Adib Bagh and Hemant K Bhargava. How to price discriminate when tariff size matters. *Marketing Science*, 32(1):111–126, 2013. 9.1

Patrick Bajari and Jungwon Yeo. Auction design and tacit collusion in FCC spectrum auctions. *Information Economics and Policy*, 21(2):90–100, 2009. 7, 7.1.2

Egon Balas. Facets of the knapsack polytope. *Mathematical programming*, 8(1):146–164, 1975. 3, 1

Egon Balas and Eitan Zemel. Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, 34(1):119–148, 1978. 3.1.2, 3.1.7, 3.1.2

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, 1996a. 4.4, 4.4.1

Egon Balas, Sebastian Ceria, Gérard Cornuéjols, and N Natraj. Gomory cuts revisited. *Operations Research Letters*, 19(1):1–9, 1996b. 1, 4.5, 4.5, 11

Maria-Florina Balcan. Data-driven algorithm design. In Tim Roughgarden, editor, *Beyond Worst Case Analysis of Algorithms*. Cambridge University Press, 2020. 4.2, 5, 5.3.1, 5.3.2

Maria-Florina Balcan and Hedyeh Beyhaghi. New guarantees for learning revenue maximizing menus of lotteries and two-part tariffs. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. 9.1

Maria-Florina Balcan and Dravyansh Sharma. Data driven semi-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:14782–14794, 2021. 4.2

Maria Florina Balcan and Dravyansh Sharma. Learning accurate and interpretable decision trees. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2024. 4.2

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. 1, 5, 5, 5, 5.1, 8.3.1, 10.1, 10.1, 10.1.5

Maria-Florina Balcan, Nikhil Devanur, Jason D Hartline, and Kunal Talwar. Random sampling auctions for limited supply. 2007. Technical report, Carnegie Mellon University. 10.1

Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. Item pricing for revenue maximization. In *ACM Conference on Electronic Commerce (EC)*, 2008. 5, 10.1

Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016. 1

Maria-Florina Balcan, Vaishnavh Nagarajan, Ellen Vitercik, and Colin White. Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems. In *Conference on Learning Theory (COLT)*, 2017. 4.2, 4.4.1

Maria-Florina Balcan, Travis Dick, Tuomas Sandholm, and Ellen Vitercik. Learning to branch. In *International Conference on Machine Learning (ICML)*, 2018a. 2, 4.2, 4.4, 4.4.1, 4.4.1, 4.4.3, 4.4.3, 4.4.5

Maria-Florina Balcan, Travis Dick, and Ellen Vitercik. Dispersion for data-driven algorithm design, online learning, and private optimization. In *Symposium on Foundations of Computer Science (FOCS)*, pages 603–614. IEEE, 2018b. 11

Maria-Florina Balcan, Travis Dick, and Colin White. Data-driven clustering via parameterized lloyd's families. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018c. 4.2

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)*, 2018d. 1, 5, 5, 5, 9, 9.1, 9.1, 9.2.2, 9.3, 9.3.1, 10.1, 10.1.1, 10.1.1, 10.1.1, 10.1.3, 10.1.5, 10.2, 10.2.1, 10.2.4, 10.2.4

Maria-Florina Balcan, Travis Dick, and Manuel Lang. Learning to link. In *International Conference on Learning Representations (ICLR)*, 2020a. 4.2

Maria-Florina Balcan, Travis Dick, and Wesley Pegden. Semi-bandit optimization in the dispersed setting. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 909–918. PMLR, 2020b. 11

Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Efficient algorithms for learning revenue-maximizing two-part tariffs. In *International Joint Conference on Artificial Intel-*

ligence (IJCAI), 2020c. 5, 10.1.4, 10.2.4

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 *ACM Symposium on Theory of Computing (STOC)*, 2021a. 4.2, 4.3, 4.3.1, 4.3.1, 4.3.2, 4.3.4, 4.3.4, 4.3.4, 4.4.3, 4.4.3, 4.5.3, 4.5.4

Maria-Florina Balcan, Mikhail Khodak, Dravyansh Sharma, and Ameet Talwalkar. Learning-to-learn non-convex piecewise-lipschitz functions. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:15056–15069, 2021b. 11

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)*, 2021c. 5.1, 8.3.1, 8.3.2, 10.2.1, 10.2.1, 10.2.4, 10.2.4

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)*, 2021d. 3.3, 4.2.1, 4.4, 4.4.1, 4.4.4, 4.4.4, 4.4.5, 4.4.6, 4.5, 4.5.3, B

Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Maximizing revenue under market shrinkage and market uncertainty. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022a. 8.3.2

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)*, 2022b. 3.2, 3.2, 3.3, 4.2.1

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)*, 2022c. 3.3, 4.2.1

Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Bicriteria multidimensional mechanism design with side information. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 5, 5, 6, 6, 6, 6.1, 6.1, 6.2, 6.2, 6.4, 6.4.2, 6.4.2, 6.6, 7, 7, 7.1.3, 8.1, 8.2, 8.2.2, 8.2.1, 8.2.3, 8.4, 11

Maria-Florina Balcan, Christopher Seiler, and Dravyansh Sharma. Accelerating ERM for data-driven algorithm design using output-sensitive techniques. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:72648–72687, 2024. 9.1

Maria-Florina Balcan, Anh Tuan Nguyen, and Dravyansh Sharma. Algorithm configuration for structured Pfaffian settings. *Transactions on Machine Learning Research (TMLR)*, 2025a. ISSN 2835-8856. 4.2.1

Maria-Florina Balcan, Anh Tuan Nguyen, and Dravyansh Sharma. Sample complexity of data-driven tuning of model hyperparameters in neural networks with structured parameter-dependent dual function. *arXiv preprint arXiv:2501.13734*, 2025b. 4.2.1

Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Increasing revenue in efficient combinatorial auctions by learning to generate artificial competition. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pages 13572–13580, 2025c. 7, 11

Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Generalization guarantees for multi-item profit maximization: Pricing, auctions, and randomized mechanisms. *Operations Research*, 73(2):648–663, 2025d. 8.3.1, 8.3.1, 8.3.2

Sandeep Baliga and Rakesh Vohra. Market research and market design. *Advances in theoretical Economics*, 3(1), 2003. 5.1, 8.3.1

Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. Strategyproof scheduling with predictions. In *Innovations in Theoretical Computer Science (ITCS)*, 2023. 5, 5

Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. Mechanism design with predictions: An annotated reading list. *ACM SIGecom Exchanges*, 21(1):54–57, 2024a. 6.6

Eric Balkanski, Vasilis Gkatzelis, Xizhi Tan, and Cherlin Zhu. Online mechanism design with predictions. In *ACM Conference on Economics and Computation (EC)*, pages 1184–1184, 2024b. 5

Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. Online nash social welfare maximization with predictions. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022. 5, 5

Siddharth Barman and Federico Echenique. The Edgeworth conjecture with small coalitions and approximate equilibria in large economies. In *ACM Conference on Economics and Computation (EC)*, pages 765–766, 2020. 10.1.2

Peter Bartlett, Piotr Indyk, and Tal Wagner. Generalization bounds for data-driven numerical linear algebra. In *Conference on Learning Theory (COLT)*, pages 2013–2040. PMLR, 2022. 4.2

Eleni Batziou, Martin Bichler, and Maximilian Fichtl. Core-stability in assignment markets with financially constrained buyers. In *ACM Conference on Economics and Computation (EC)*, pages 473–474, 2022. 6

Berkay Becu, Santanu S Dey, Feng Qiu, and Alinson S Xavier. Approximating the Gomory mixed-integer cut closure using historical data. *arXiv preprint arXiv:2411.15090*, 2024. 4.2.1

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010. 8.3.1

Daniel Berend and Tamir Tassa. Improved bounds on Bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 2010. 10.1.3

Dirk Bergemann and Stephen Morris. Robust mechanism design. *Econometrica*, pages 1771–1813, 2005. 5.6

Martin Bichler and Jacob K Goeree. *Handbook of spectrum auction design*. Cambridge University Press, 2017. 5

Martin Bichler and Stefan Waldherr. Core and pricing equilibria in combinatorial exchanges. *Economics Letters*, 157:145–147, 2017. 6

Martin Bichler and Stefan Waldherr. Core pricing in combinatorial exchanges with financially constrained buyers: Computational hardness and algorithmic solutions. *Operations Research*, 70(1):241–264, 2022. 6

Martin Bichler, Pasha Shabalin, and Jürgen Wolf. Do core-selecting combinatorial clock auctions always lead to high efficiency? An experimental analysis of spectrum auction designs. *Experimental Economics*, 16:511–545, 2013. 6

Martin Bichler, Vladimir Fux, and Jacob K Goeree. Designing combinatorial exchanges for the reallocation of resource rights. *Proceedings of the National Academy of Sciences*, 116(3): 786–791, 2019. 5.6

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. 6.6, 11

Sushil Bikhchandani. Information acquisition and full surplus extraction. *Journal of Economic Theory*, 145(6):2282–2308, 2010. 8

Sushil Bikhchandani and Joseph M. Ostroy. The package assignment model. *Journal of Economic Theory*, 107:377–406, 2002. 6, 6.4.1, 6.4.2, 6.4.2, 6.6, 11

Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh V. Vohra. Linear programming and Vickrey auctions, 2001. Draft. 6.4.1

Robert Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling. MIP: Theory and practice—closing the gap. In *IFIP Conference on System Modeling and Optimization*, pages 19–49. Springer, 1999. 1, 4.5

Tilman Börgers. *An introduction to the theory of mechanism design*. Oxford University Press, USA, 2015. 5.1.1

Vitor Bosshard, Benedikt Bünz, Benjamin Lubin, and Sven Seuken. Computing Bayes-Nash equilibria in combinatorial auctions with continuous value and action spaces. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 119–127, 2017. 6

Craig Boutilier. Solving concisely expressed combinatorial auction problems. In *National Conference on Artificial Intelligence (AAAI)*, pages 359–366, Edmonton, Canada, 2002. 6.6

Craig Boutilier and Holger Hoos. Bidding languages for combinatorial auctions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1211–1217, Seattle, WA, 2001. 6.6, 11

Craig Boutilier, Martin Mladenov, and Guy Tennenholtz. Recommender ecosystems: A mechanism design perspective on holistic modeling and optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 22575–22583, 2024. 7

Eric Budish, Peter Cramton, Albert S Kyle, Jeongmin Lee, and David Malec. Flow trading. Technical report, National Bureau of Economic Research, 2023. 7

Jeremy Bulow and Paul Klemperer. Auctions versus negotiations. *The American Economic Review*, 1996. 5, 8

Benedikt Bünz, Sven Seuken, and Benjamin Lubin. A faster core constraint generation algorithm for combinatorial auctions. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 29, 2015. 6, 6.1

Benedikt Bünz, Benjamin Lubin, and Sven Seuken. Designing core-selecting payment rules: A computational search approach. *Information Systems Research*, 33(4):1157–1173, 2022. 6, 6.1, 6.5.2, 6.6

Ioannis Caragiannis and Georgios Kalantzis. Randomized learning-augmented auctions with revenue guarantees. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2687–2694, 2024. 5

Tanmoy Chakraborty, Zhiyi Huang, and Sanjeev Khanna. Dynamic and nonuniform pricing strategies for revenue maximization. *SIAM Journal on Computing*, 2013. 5, 10.1

Shuchi Chawla and J Benjamin Miller. Mechanism design for subadditive agents via an ex ante relaxation. In *Proc. 2016 ACM Conference on Economics and Computation*, pages 579–596. ACM, 2016. 9.1

Hongyu Cheng and Amitabh Basu. Learning cut generating functions for integer programming. *Advances in Neural Information Processing Systems*, 37:61455–61480, 2024. 4.2.1

Hongyu Cheng and Amitabh Basu. Generalization guarantees for learning branch-and-cut policies in integer programming. *arXiv preprint arXiv:2505.11636*, 2025. 4.2.1

Hongyu Cheng, Sammy Khalife, Barbara Fiedorowicz, and Amitabh Basu. Sample complexity of algorithm selection using neural networks and its applications to branch-and-cut. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 4.2.1

Didier Chételat and Andrea Lodi. Continuous cutting plane algorithms in integer programming. *Operations Research Letters*, 51(4):439–445, 2023. 11

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 (NeurIPS)*, 35:9633–9644, 2022. 11

Alessandro Chiesa, Silvio Micali, and Zeyuan Allen Zhu. Knightian analysis of the Vickrey mechanism. *Econometrica*, 83(5):1727–1754, 2015. 5

Antonia Chmiela, Elias B Khalil, Ambros Gleixner, Andrea Lodi, and Sebastian Pokutta. Learning to schedule heuristics in branch-and-bound. In *Conference on Neural Information Processing Systems (NIPS)*, 2021. 4.2

Vasek Chvátal. Hard knapsack problems. *Operations Research*, 28(6):1402–1411, 1980. 4.4.4

Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete mathematics*, 4(4):305–337, 1973. 4.3, 4.3, 4.5

Ed H. Clarke. Multipart pricing of public goods. *Public Choice*, 1971. 5, 5.1.1, 6.1

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *ACM Symposium on Theory of Computing (STOC)*, pages 243–252, 2014. 5

Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*, volume 271. Springer, 2014. 2, 3, 3.1.2, 3.1.2, 4.3, 4.4.4, 4.5

Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 103–110, Edmonton, Canada, 2002. 5, 5.4.3, 9, 10.1

Gérard Cornuéjols. Revival of the Gomory cuts in the 1990's. *Annals of Operations Research*, 149(1):63–66, 2007. 1, 1

Gérard Cornuéjols. Valid inequalities for mixed integer linear programs. *Mathematical Programming*, 112(1):3–44, 2008. 4.5

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. 11

Peter Cramton. Spectrum auction design. *Review of industrial organization*, 42:161–190, 2013. 5, 6

Peter Cramton and Jesse Schwartz. Collusive bidding: Lessons from the FCC spectrum auctions. *Journal of Regulatory Economics*, 17:229–252, 2000. 7, 7.1.2

Peter Cramton, Robert Gibbons, and Paul Klemperer. Dissolving a partnership efficiently. *Econometrica: Journal of the Econometric Society*, pages 615–632, 1987. 5, 6, 7

Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. MIT Press, 2006. 5

Jacques Crémer and Richard P McLean. Full extraction of the surplus in Bayesian and dominant strategy auctions. *Econometrica: Journal of the Econometric Society*, pages 1247–1257, 1988. 5.1.1, 8

Harlan Crowder, Ellis L Johnson, and Manfred Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 1983. 3

Rachel Cummings, Nikhil R Devanur, Zhiyi Huang, and Xiangning Wang. Algorithmic price discrimination. In *Proc. Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2432–2451. SIAM, 2020. 9.1

Michael Curry, Tuomas Sandholm, and John Dickerson. Differentiable economics for randomized affine maximizer auctions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023. 5, 5, 8.3.2

Michael A Cusumano. Amazon and Whole Foods: follow the strategy (and the money). *Communications of the ACM*, 60(10):24–26, 2017. 10.2, 10.2

Anirban Dasgupta, John Hopcroft, Jon Kleinberg, and Mark Sandler. On learning mixtures of heavy-tailed distributions. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 491–500. IEEE, 2005. 7.4

P. Dasgupta and Eric Maskin. Efficient auctions. *Quarterly Journal of Economics*, 115:341–388, 2000. 8

Constantinos Daskalakis and George Pierrakos. Simple, optimal and efficient auctions. In *International Workshop on Internet and Network Economics (WINE)*. Springer, 2011. 5

Robert Day and Peter Cramton. Quadratic core-selecting payment rules for combinatorial auctions. *Operations Research*, 60(3):588–603, 2012. 1, 6, 6, 6, 6, 6.1, 6.3, 6.3.5, 6.5, 6.5, 6.5.1, 6.5.1, 6.5.1, 6.5.1

Robert Day and Paul Milgrom. Optimal incentives in core-selecting auctions. *Handbook of*

Market Design, 2010. 6, 6, 6.1, 6.3, 6.3, 6.3, 6.5.1, 6.6

Robert Day and S. Raghavan. Fair payments for efficient allocations in public sector combinatorial auctions. *Management Science*, 53(9):1389–1406, 2007. 6, 6, 6, 6.1, 6.3, 6.3, 6.5.1, 6.6, 8

Sven de Vries, James Schummer, and Rakesh V Vohra. On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, 132(1):95–118, 2007. 6.4.1

Gerard Debreu and Herbert Scarf. A limit theorem on the core of an economy. *International Economic Review*, 1963. 10.1.2

François Denis. PAC learning from positive statistical queries. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 112–126. Springer, 1998. 11

Nikhil R Devanur and Jason D Hartline. Limited and online supply and the Bayesian foundations of prior-free mechanism design. In *EC*, 2009. 10.1

Nikhil R Devanur, Jason D Hartline, and Qiqi Yan. Envy freedom and prior-free mechanism design. *Journal of Economic Theory*, 2015. 10.1

Nikhil R Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *ACM Symposium on Theory of Computing (STOC)*, 2016. 5

Arnaud Deza and Elias B Khalil. Machine learning for cutting planes in integer programming: a survey. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6592–6600, 2023. 4.2.1

Arnaud Deza, Elias B Khalil, Zhenan Fan, Zirui Zhou, and Yong Zhang. Learn2Aggregate: Supervised generation of Chvátal-Gomory cuts using graph neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pages 26947–26954, 2025. 4.2.1

Giovanni Di Liberto, Serdar Kadioglu, Kevin Leo, and Yuri Malitsky. Dash: Dynamic approach for switching heuristics. *European Journal of Operational Research*, 248(3):943–953, 2016. 4.2

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. 5, 5

Shahar Dobzinski and Nitzan Uziely. Revenue loss in shrinking markets. In *ACM Conference on Economics and Computation (EC)*, 2018. 10.2

Zhijian Duan, Haoran Sun, Yurong Chen, and Xiaotie Deng. A scalable neural network for DSIC affine maximizer auction design. *Advances in Neural Information Processing Systems*, 36, 2023. 5, 8.3.2

Richard M Dudley. Universal Donsker classes and metric entropy. *The Annals of Probability*, 1987. 10.1.1, 10.1.1, 10.1.1

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. 5, 5, 7.4, 8.3.2

Paul Dütting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. In *ACM Web Conference (WWW)*, pages 144–155, 2024. 7, 11

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American Economic Review*, 97(1):242–259, March 2007. ISSN 0002-8282. 5, 7

Aytek Erdil and Paul Klemperer. A new payment rule for core-selecting package auctions. *Journal of the European Economic Association*, 8(2-3):537–547, 2010. 6, 6, 6, 6.1, 6.5, 6.5.1, 6.5.1

Jeff Erickson. Linear programming. `https://courses.grainger.illinois.edu/cs473/sp2017/notes/H-lp.pdf`, 2017. 7.3.1

Adejuyigbe O Fajemisin, Donato Maragno, and Dick den Hertog. Optimization with constraint learning: A framework and survey. *European Journal of Operational Research*, 314(1):1–14, 2024. 7.4

Martin S Feldstein. Equity and efficiency in public sector pricing: the optimal two-part tariff. *The Quarterly Journal of Economics*, pages 176–187, 1972. 9.1

Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2002. 4.4.4

Matteo Fischetti and Andrea Lodi. Optimizing over the first chvátal closure. *Mathematical Programming*, 110(1):3–20, 2007. 11

Alex S Fukunaga. A branch-and-bound algorithm for hard multiple knapsack problems. *Annals of Operations Research*, 184(1):97–119, 2011. 3.2

Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. Technical report, Optimization Online, March 2020. URL `http://www.optimization-online.org/DB_HTML/2020/03/7705.html`. 4.4, 4.4.1, 4.5.2

Maxime Gasse, Didier Chételat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. Exact combinatorial optimization with graph convolutional neural networks. In *Conference on Neural Information Processing Systems (NIPS)*, pages 15554–15566, 2019. 4.2

Andrew Gilpin and Tuomas Sandholm. Information-theoretic approaches to branching in search. *Discrete Optimization*, 8(2):147–159, 2011. Early version in IJCAI-07. 4.4.4

Vasilis Gkatzelis, Kostas Kollias, Alkmini Sgouritsa, and Xizhi Tan. Improved price of anarchy via predictions. In *ACM Conference on Economics and Computation (EC)*, 2022. 5, 5

Vasilis Gkatzelis, Daniel Schoepflin, and Xizhi Tan. Clock auctions augmented with unreliable advice. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2629–2655. SIAM, 2025. 5

Gagan Goel, Mohammad Reza Khani, and Renato Paes Leme. Core-competitive auctions. In *ACM Conference on Economics and Computation (EC)*, pages 149–166, 2015. 6

Jacob K Goeree and Yuanchuan Lien. On the impossibility of core-selecting auctions. *Theoretical economics*, 11(1):41–52, 2016. 6, 6, 6.2, 6.2, 6.6, 8

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. 5.6, 6, 6, 11

Andrew Goldberg, Jason Hartline, and Andrew Wright. Competitive auctions and digital goods. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Washington, DC, 2001. 5, 5.4.2, 10.1

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. 10.2, 10.2

Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275 – 278, 1958. 1, 4.3, 4.3, 4.4.4, 4.5

J Green and J-J Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977. 7.1.1, 7.1.2

J Green and J-J Laffont. *Incentives in Public Decision Making*. Amsterdam: North-Holland, 1979. 1

M. Grotschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimizations*. Springer-Verlag, 1993. 5.2.1, 8.3.2

Theodore Groves. Incentives in teams. *Econometrica*, 1973. 5, 5.1.1, 5.1.1, 6.1, 7

Zonghao Gu, George L Nemhauser, and Martin WP Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 1998. 3, 3.2

Zonghao Gu, George L Nemhauser, and Martin WP Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Complexity. *INFORMS Journal on Computing*, 1999. 3, 3.1.1

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. 1, 3, 3, 3, 3, 3.1, 3.1, 3.1.2, 3.1.2, A, A, A

Oscar Guaje, Arnaud Deza, Aleksandr M Kazachkov, and Elias B Khalil. Machine learning for optimization-based separation: the case of mixed-integer rounding cuts. *arXiv preprint arXiv:2408.08449*, 2024. 4.2.1

Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:95–124, 1999. 10.2.1

Mingyu Guo. VCG redistribution with gross substitutes. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2011. 10.2.1

Prateek Gupta, Maxime Gasse, Elias B Khalil, M Pawan Kumar, Andrea Lodi, and Yoshua Bengio. Hybrid models for learning to branch. In *Conference on Neural Information Processing Systems (NIPS)*, 2020. 4.2

Rishi Gupta and Tim Roughgarden. A PAC approach to application-specific algorithm selection.

SIAM Journal on Computing, 46(3):992–1017, 2017. 4.2

Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On profit-maximizing envy-free pricing. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1164–1173, 2005. 5

MohammadTaghi Hajiaghayi, Sebastien Lahaie, Keivan Rezaei, and Suho Shin. Ad auctions for LLMs via retrieval augmented generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 7, 11

Sergiu Hart and Noam Nisan. The menu-size complexity of auctions. In *ACM Conference on Electronic Commerce (EC)*, pages 565–566, 2013. 5.4.3

Sergiu Hart and Noam Nisan. Selling multiple correlated goods: Revenue maximization and menu-size complexity. *Journal of Economic Theory*, 183:991–1029, 2019. 9.1

Sergiu Hart and Philip J Reny. Maximal revenue with multiple goods: Nonmonotonicity and other observations. *Theoretical Economics*, 10(3):893–922, 2015. 5.4.3

Jason Hartline, Vahab Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *International Conference on World Wide Web (WWW)*, pages 189–198, 2008. 10.2.3

Jason D Hartline and Tim Roughgarden. Simple versus optimal mechanisms. In *ACM conference on Electronic Commerce (EC)*, pages 225–234, 2009. 5

David Hartvigsen and Eitan Zemel. The complexity of lifted inequalities for the knapsack problem. *Discrete Applied Mathematics*, 39(2):113–123, 1992. 3.1.2

David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnik-Chervonenkis dimension. *J. Comb. Theory, Ser. A*, 1995. 10.1.2

He He, Hal Daume III, and Jason M Eisner. Learning to search in branch and bound algorithms. In *Conference on Neural Information Processing Systems (NIPS)*, 2014. 4.2

Gail Hohner, John Rich, Ed Ng, Grant Reid, Andrew J. Davenport, Jayant R. Kalagnanam, Ho Soo Lee, and Chae An. Combinatorial and quantity-discount procurement auctions benefit Mars, Incorporated and its suppliers. *Interfaces*, 33(1):23–35, 2003. 5, 6, 10.2.3

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. 3, 3.1.2

Bengt Holmström. Groves' scheme on restricted domains. *Econometrica*, 47(5):1137–1144, 1979. 1, 5.1.1, 5.1.1, 7.1.1, 7.1.2, 7.1.1

Ron Holzman, Noa Kfir-Dahav, Dov Monderer, and Moshe Tennenholtz. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*, 47(1):104–123, April 2004. 5

Eric Horvitz, Yongshao Ruan, Carla Gomez, Henry Kautz, Bart Selman, and Max Chickering. A Bayesian approach to tackling hard computational problems. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001. 4, 4.4.2, 5

David Huang, Francisco Marmolejo-Cossío, Edwin Lock, and David Parkes. Accelerated preference elicitation with LLM-based proxies. *arXiv preprint arXiv:2501.14625*, 2025. 11

Zeren Huang, Kerong Wang, Furui Liu, Hui-ling Zhen, Weinan Zhang, Mingxuan Yuan, Jianye Hao, Yong Yu, and Jun Wang. Learning to select cuts for efficient mixed-integer programming. *arXiv preprint arXiv:2105.13645*, 2021. 4.2

Brady Hunsaker and Craig A Tovey. Simple lifted cover inequalities and hard knapsack problems. *Discrete Optimization*, 2(3):219–228, 2005. 3.1.1

Frank Hutter, Holger Hoos, Kevin Leyton-Brown, and Thomas Stützle. Paramils: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009. ISSN 1076-9757. 2, 4, 4.2, 4.4.2

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proc. of LION-5*, pages 507–523, 2011. 4.2

Nathanael Hyafil and Craig Boutilier. Regret minimizing equilibria and mechanisms for games with strict type uncertainty. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004. 5

Raghuram Iyengar, Kamel Jedidi, and Rajeev Kohli. A conjoint approach to multipart pricing. *Journal of Marketing Research*, 45(2):195–210, 2008. 9.1

Philippe Jehiel, Moritz Meyer-ter Vehn, Benny Moldovanu, and William R Zame. The limits of ex post implementation. *Econometrica*, 74(3):585–610, 2006. 8

Philippe Jehiel, Moritz Meyer-Ter-Vehn, and Benny Moldovanu. Mixed bundling auctions. *Journal of Economic Theory*, 2007. 10.1, 10.2.1

Robert G Jeroslow. Trivial integer programs unsolvable by branch-and-bound. *Mathematical Programming*, 6(1):105–109, 1974. 4.3.1, 4.4.4, 4.5.3

Randall S Jones and Haruki Seitani. Labour market reform in Japan to cope with a shrinking and ageing population. *OECD Economic Department Working Papers*, (1568), 2019. 10.2

Serdar Kadioglu, Yuri Malitsky, Meinolf Sellmann, and Kevin Tierney. ISAC—instance-specific algorithm configuration. In *European Conference on Artificial Intelligence (ECAI)*, 2010. 4, 4.2, 4.4.2

Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012. 11

Konstantinos Kaparis and Adam N Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical programming*, 124:69–91, 2010. 3, 3

Orcun Karaca and Maryam Kamgarpour. Core-selecting mechanisms in electricity markets. *IEEE Transactions on Smart Grid*, 11(3):2604–2614, 2019. 6

Miroslav Karamanov and Gérard Cornuéjols. Branching on general disjunctions. *Mathematical Programming*, 128(1-2):403–436, 2011. 4.4.4

Kohei Kawai, Masatomo Suzuki, and Chihiro Shimizu. Shrinkage in Tokyo's central business district: Large-scale redevelopment in the spatially shrinking office market. *Sustainability*, 11 (10):2742, 2019. 10.2

Elias Khalil, Pierre Le Bodic, Le Song, George Nemhauser, and Bistra Dilkina. Learning to branch in mixed integer programming. In *AAAI Conference on Artificial Intelligence (AAAI)*,

2016. 4, 4.2

Elias Khalil, Bistra Dilkina, George Nemhauser, Shabbir Ahmed, and Yufen Shao. Learning to run heuristics in tree search. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. 4.2

Mikhail Khodak, Edmond Chow, Maria Florina Balcan, and Ameet Talwalkar. Learning to relax: Setting solver parameters across a sequence of linear system instances. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 4.2

Misha Khodak, Maria-Florina Balcan, Ameet Talwalkar, and Sergei Vassilvitskii. Learning predictions for algorithms with predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 5, 5.3.2

Diego Klabjan, George L Nemhauser, and Craig Tovey. The complexity of cover inequality separation. *Operations Research Letters*, 23(1-2):35–40, 1998. 3

Robert Kleinberg and Yang Yuan. On the ratio of revenue to welfare in single-parameter mechanism design. In *ACM Conference on Electronic Commerce (EC)*, 2013. 5, 5

Robert Kleinberg, Kevin Leyton-Brown, and Brendan Lucier. Efficiency through procrastination: Approximately optimal algorithm configuration with runtime guarantees. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. 2, 4.2

Robert Kleinberg, Kevin Leyton-Brown, Brendan Lucier, and Devon Graham. Procrastinating with confidence: Near-optimal, anytime, adaptive algorithm configuration. *Conference on Neural Information Processing Systems (NIPS)*, 2019. 2

Sreya Kolay and Greg Shaffer. Bundling and menus of two-part tariffs. *The Journal of Industrial Economics*, 51(3):383–403, 2003. 9.1

Vijay Krishna and Motty Perry. Efficient mechanism design. *Available at SSRN 64934*, 1998. 5, 5, 5.1.1, 5.1.1, 5.6, 6, 6, 6.1, 6.2, 7, 7, 7.1.3, 8, 8.1, 8.2.1, 8.2.5, 8.4

Christian Kroer and Tuomas Sandholm. Computational bundling for auctions. In *AAMAS*, 2015. 10.1.3, 10.1.4, 10.1.4, 10.2.3

Sébastien Lahaie and Benjamin Lubin. Adaptive-price combinatorial auctions. In *ACM Conference on Economics and Computation (EC)*, pages 749–750, 2019. 11

Anja Lambrecht, Katja Seim, and Bernd Skiera. Does uncertainty matter? Consumer behavior under three-part tariffs. *Marketing Science*, 26(5):698–710, 2007. 9.1

Connor Lawless, Yingxi Li, Anders Wikum, Madeleine Udell, and Ellen Vitercik. LLMs for cold-start cutting plane separator configuration. *arXiv preprint arXiv:2412.12038*, 2024. 4.2.1

Hayne E Leland and Robert A Meyer. Monopoly pricing structures with imperfect discrimination. *The Bell Journal of Economics*, pages 449–462, 1976. 9.1

Adam N Letchford and Georgia Souli. On lifted cover inequalities: A new lifting procedure with unusual properties. *Operations Research Letters*, 47(2):83–87, 2019. 3, 3, 3.1.2

Adam N Letchford and Georgia Souli. Lifting the knapsack cover inequalities for the knapsack polytope. *Operations Research Letters*, 48(5):607–611, 2020. 3

W Arthur Lewis. The two-part tariff. *Economica*, 8(31):249–270, 1941. 9

Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *ACM Conference on Electronic Commerce (ACM-EC)*, pages 66–76, Minneapolis, MN, 2000. 3.2, 6.5.1, 11

Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. Empirical hardness models: Methodology and a case study on combinatorial auctions. *Journal of the ACM*, 56(4):1–52, 2009. ISSN 0004-5411. 4, 4.4.2

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. 5, 5.6, 6

Shengwu Li. Obviously strategy-proof mechanisms. *American Economic Review*, 107(11): 3257–3287, 2017. 5.6

Sirui Li, Wenbin Ouyang, Max B. Paulus, and Cathy Wu. Learning to configure separators in branch-and-cut. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 3.3, 4.2.1

Anton Likhodedov and Tuomas Sandholm. Methods for boosting revenue in combinatorial auctions. In *National Conference on Artificial Intelligence (AAAI)*, pages 232–237, San Jose, CA, 2004. 1, 5, 5, 9, 10.1

Anton Likhodedov and Tuomas Sandholm. Approximating revenue-maximizing combinatorial auctions. In *National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, 2005. 5, 5, 9, 10.1, 10.1, 10.1.3

Haotian Ling, Zhihai Wang, and Jie Wang. Learning to stop cut generation for efficient mixed-integer linear programming. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 20759–20767, 2024. 4.2.1

Giuseppe Lopomo, Luca Rigotti, and Chris Shannon. Uncertainty in mechanism design. *arXiv preprint arXiv:2108.12633*, 2021. 10.2

William S. Lovejoy. Optimal mechanisms with finite agent types. *Management Science*, 53(5): 788–803, 2006. 7

Pinyan Lu, Zongqi Wan, and Jialin Zhang. Competitive auctions with imperfect predictions. In *ACM Conference on Economics and Computation (EC)*, pages 1155–1183, 2024. 5, 6

Ashutosh Mahajan and Theodore K Ralphs. Experiments with branching using general disjunctions. In *Operations Research and Cyber-Infrastructure*, pages 101–118. Springer, 2009. 4.4.4

Alexey Malakhov and Rakesh V Vohra. An optimal auction for capacity constrained bidders: a network perspective. *Economic Theory*, 2009. 5

Kyle Mana, Fernando Acero, Stephen Mak, Parisa Zehtabi, Michael Cashmore, Daniele Magazzeni, and Manuela Veloso. Accelerating cutting-plane algorithms via reinforcement learning surrogates. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 20786–20793, 2024. 4.2.1

Hugues Marchand, Alexander Martin, Robert Weismantel, and Laurence Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 2002. 3

Evangelos Markakis and Artem Tsikiridis. On core-selecting and core-competitive mechanisms for binary single-parameter auctions. In *International Conference on Web and Internet Economics (WINE)*, pages 271–285. Springer, 2019. 6

Leslie M Marx and Greg Shaffer. Opportunism and menus of two-part tariffs. *International Journal of Industrial Organization*, 22(10):1399–1414, 2004. 9.1

Eric Maskin and John Riley. Monopoly with incomplete information. *The RAND Journal of Economics*, 15(2):171–196, 1984. 9.1

Victor J Massad. Understanding the cord-cutters: An adoption/self-efficacy approach. *International Journal on Media Management*, 20(3):216–237, 2018. 10.2, 10.2

R Preston McAfee and Philip J Reny. Correlated information and mechanism design. *Econometrica: Journal of the Econometric Society*, pages 395–421, 1992. 5.1.1

Andres Muñoz Medina and Sergei Vassilvitskii. Revenue optimization with approximate bid predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5, 5

Nimrod Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, pages 414–424, 1979. 4.4.3

Cade Metz. Facebook doesn't make as much money as it could—on purpose. *Wired*, 2015. 5.1.1

Paul Milgrom and R Weber. A theory of auctions and competitive bidding. *Econometrica*, 50: 1089–1122, 1982. 8, 11

John Milnor. On the Betti numbers of real varieties. *Proceedings of the American Mathematical Society*, 15(2):275–280, 1964. 4.5

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 2022. 5, 5

Mehryar Mohri and Andres Muñoz Medina. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *ICML*, 2014. 10.1

Paulo Klinger Monteiro. Abstract types and distributions in independent private value auctions. *Economic Theory*, 40(3):497–507, 2009. 7

Dmitry Moor, Sven Seuken, Tobias Grubenmann, and Abraham Bernstein. Core-selecting payment rules for combinatorial auctions with uncertain availability of goods. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 424–432, 2016. 6

Mouad Morabit, Guy Desaulniers, and Andrea Lodi. Machine-learning–based column selection for column generation. *Transportation Science*, 55(4):815–831, 2021. 11

Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In *Conference on Learning Theory (COLT)*, 2016. 1, 5, 5

Jamie H Morgenstern and Tim Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *NIPS*, 2015. 1, 10.1

Ahuva Mu'alem and Michael Schapira. Mechanism design over discrete domains. In *ACM Conference on Electronic Commerce (EC)*, pages 31–37, 2008. 7

Michael M Murphy. Price discrimination, market separation, and the multi-part tariff. *Economic Inquiry*, 15(4):587, 1977. 9.1

Roger Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981. 1, 5, 5, 5, 5.4.3, 7

Roger Myerson and Mark Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983. 5, 6, 7

Sridhar Narayanan, Pradeep K Chintagunta, and Eugenio J Miravete. The role of self selection, usage uncertainty and learning in the demand for local telephone service. *Quantitative Marketing and Economics*, 5(1):1–34, 2007. 9.1

Balaubramaniam Kausik Natarajan. On learning boolean functions. In *ACM Symposium on Theory of Computing (STOC)*, pages 296–304, 1987. 11

Ioannis Neokosmidis, Theodoros Rokkas, Dimitris Xydias, Antonino Albanese, Muhammad Shuaib Siddiqui, Carlos Colman-Meixner, and Dimitra Simeonidou. Are 5g networks and the neutral host model the solution to the shrinking telecom market. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 70–77. Springer, 2018. 10.2

Yew-Kwang Ng and Mendel Weisser. Optimal pricing with a budget constraint–the case of the two-part tariff. *The Review of Economic Studies*, 41(3):337–345, 1974. 9.1

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. 6, 6

Noam Nisan. Bidding and allocation in combinatorial auctions. In *ACM Conference on Electronic Commerce (ACM-EC)*, pages 1–12, Minneapolis, MN, 2000. 6.1, 6.6, 11

Noam Nisan. Introduction to mechanism design (for computer scientists). *Algorithmic Game Theory*, 9:209–242, 2007. 7.1.1, 7.1.1, 7.3.1

Walter Y Oi. A Disneyland dilemma: Two-part tariffs for a Mickey Mouse monopoly. *The Quarterly Journal of Economics*, 85(1):77–96, 1971. 9

Abraham Othman and Tuomas Sandholm. Envy quotes and the iterated core-selecting combinatorial auction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2010. 6, 6, 6.2, 6.2, 8

Marion Ott and Marissa Beck. Incentives for overbidding in minimum-revenue core-selecting auctions. 2013. 6

Jonathan H. Owen and Sanjay Mehrotra. Experimental results on using general disjunctions in branch-and-bound for general-integer linear programs. *Computational Optimization and Applications*, 20(2):159–170, November 2001. 4.4.4

Mallesh M Pai and Rakesh Vohra. Optimal auctions with financially constrained buyers. *Journal of Economic Theory*, 150:383–425, 2014. 5

Ignacio Palacios-Huerta, David C Parkes, and Richard Steinberg. Combinatorial auctions in practice. *Journal of Economic Literature*, 62(2):517–553, 2024. 6

Christos H Papadimitriou and Mihalis Yannakakis. The complexity of facets (and some facets of complexity). In *ACM symposium on Theory of computing*, pages 255–260, 1982. 3.1.2

David Parkes, Jayant Kalagnanam, and Marta Eso. Achieving budget-balance with Vickrey-based payment schemes in exchanges. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1161–1168, Seattle, WA, 2001. 6, 6.5.2

Max B Paulus, Giulia Zarpellon, Andreas Krause, Laurent Charlin, and Chris Maddison. Learning to cut by looking ahead: Cutting plane selection via imitation learning. In *International Conference on Machine Learning (2022)*, pages 17584–17600. PMLR, 2022. 4.2.1

Michael Perregard. New in FICO Xpress solver. `https://plato.asu.edu/ftp/informs_talks_2024/perregard.pdf`, 2024. 3, 3.1.2

Ivan PL Png and Hao Wang. Buyer uncertainty and two-part pricing: theory and applications. *Management Science*, 56(2):334–342, 2010. 9.1

David Pollard. *Convergence of Stochastic Processes*. Springer, 1984. 4.1

Siddharth Prasad, Martin Mladenov, and Craig Boutilier. Content prompting: Modeling content provider dynamics to improve user welfare in recommender ecosystems. *arXiv preprint arXiv:2309.00940*, 2023. 7

Siddharth Prasad, Ellen Vitercik, Maria-Florina Balcan, and Tuomas Sandholm. New sequence-independent lifting techniques for cutting planes and when they induce facets. *arXiv preprint arXiv:2401.13773*, 2024. 4.4.4, 11

Siddharth Prasad, Maria-Florina Balcan, and Tuomas Sandholm. Revenue-optimal efficient mechanism design with general type spaces. *arXiv preprint arXiv:2505.13687*, 2025a. 8.4, 11

Siddharth Prasad, Maria-Florina Balcan, and Tuomas Sandholm. Weakest bidder types and new core-selecting combinatorial auctions. *arXiv preprint arXiv:2505.13680*, 2025b. 8, 11

Pol Puigdemont, Stratis Skoulakis, Grigorios Chrysos, and Volkan Cevher. Learning to remove cuts in integer linear programming. *arXiv preprint arXiv:2406.18781*, 2024. 4.2.1

Kevin Roberts. The characterization of implementable social choice rules. In J-J Laffont, editor, *Aggregation and Revelation of Preferences*. 1979. 5, 5.5, 8.3.2, 10.1, 10.2.1

Marzena Rostek and Nathan Yoder. Core selection in auctions and exchanges. *Working paper*, 2015. 6

Marzena Rostek and Nathan Yoder. Reallocative auctions and core selection. *Available at SSRN 4450987*, 2023. 6

Alvin E Roth. Marketplaces, markets, and market design. *American Economic Review*, 108(7): 1609–1658, 2018. 5

Michael Rothkopf, Aleksandar Pekeč, and Ronald Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998. 5.2.1

Tim Roughgarden, Inbal Talgam-Cohen, and Qiqi Yan. Robust auctions for revenue via enhanced competition. *Operations Research*, 68(4):1074–1094, 2020. 10.2.3

Ashish Sabharwal, Horst Samulowitz, and Chandra Reddy. Guiding combinatorial optimization with UCT. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer, 2012. 4.2

Shinsaku Sakaue and Taihei Oki. Sample complexity of learning heuristic functions for greedy-best-first and a* search. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 2889–2901, 2022. 4.2

Shinsaku Sakaue and Taihei Oki. Generalization bound and learning methods for data-driven projections in linear programming. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:12825–12846, 2024. 4.2.1

Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, January 2002a. 6.1, 6.6, 7.2, 10.2.4, 11

Tuomas Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002b. Earlier versions: Washington U. tech report WU-CS-99-02 Jan. 1999, AAAI-99 Workshop on AI in Ecommerce, AGENTS-00. 6.6

Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *International Conference on Principles and Practice of Constraint Programming*, pages 19–36, Cork, Ireland, 2003. 10.1

Tuomas Sandholm. Optimal winner determination algorithms. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, pages 337–368. MIT Press, 2006. Chapter 14. 8.3.2

Tuomas Sandholm. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine*, 28(3):45–58, 2007. 1, 5, 5.6, 10.2.3, 11

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. 1, 4, 4.2, 4.4.2, 5, 5.1.1, 5.6, 6, 6.6, 8, 10.2.3

Tuomas Sandholm and Anton Likhodedov. Automated design of revenue-maximizing combinatorial auctions. *Operations Research*, 2015. 5, 5, 5, 9, 10.1, 10.1, 10.1.3

Tuomas Sandholm and Subhash Suri. BOB: Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence*, 145:33–58, 2003. 10.2.4

Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *AAMAS*, pages 69–76, Bologna, Italy, July 2002. 3.2

Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for winner determination in combinatorial auctions. *Management Science*, 2005. 5.2.1, 8.3.2, 10.1.4, 10.2.4

Tuomas Sandholm, David Levine, Michael Concordia, Paul Martyn, Rick Hughes, Jim Jacobs, and Dennis Begg. Changing the game in strategic sourcing at Procter & Gamble: Expressive competition enabled by optimization. *Interfaces*, 36(1):55–68, 2006. 1, 6, 10.2.3

Christian Schlereth, Tanja Stepanchuk, and Bernd Skiera. Optimization and analysis of the profitability of tariff structures with two-part tariffs. *European Journal of Operational Research*, 206(3):691–701, 2010. 9.1

Clayton Scott and Robert Nowak. Learning minimum volume sets. *Advances in Neural Information Processing Systems (NeurIPS)*, 18, 2005. 7.4

Dravyansh Sharma, Maria-Florina Balcan, and Travis Dick. Learning piecewise lipschitz functions in changing environments. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3567–3577. PMLR, 2020. 11

Benlin Shi, Yang Song, Changchen Liu, and Yunfeng Luo. Competition and evolution of linear and two-part tariff. *Computers & Mathematics with Applications*, 57(11-12):1895–1900, 2009. 9.1

Haim Shvaytser. A necessary condition for learning from positive examples. *Machine Learning*, 5(1):101–113, 1990. 11

Vasiliki Skreta. Mechanism design for arbitrary type spaces. *Economics Letters*, 91(2):293–299, 2006. 7

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. 10.2, 10.2

Jialin Song, Ravi Lanka, Yisong Yue, and Bistra Dilkina. A general large neighborhood search framework for solving integer programs. In *Conference on Neural Information Processing Systems (NIPS)*, 2020. 4.2

Ermis Soumalias, Yanchen Jiang, Kehang Zhu, Michael Curry, Sven Seuken, and David C Parkes. LLM-powered preference elicitation in combinatorial assignment. *arXiv preprint arXiv:2502.10308*, 2025. 11

Arun Sundararajan. Nonlinear pricing of information goods. *Management Science*, 50(12): 1660–1673, 2004. 9.1

Pingzhong Tang and Tuomas Sandholm. Mixed-bundling auctions with reserve prices. In *AAMAS*, 2012. 10.1

Yunhao Tang, Shipra Agrawal, and Yuri Faenza. Reinforcement learning for integer programming: Learning to cut. *International Conference on Machine Learning (ICML)*, 2020. 4.2

René Thom. Sur l'homologie des varietes algebriques reelles. In *Differential and combinatorial topology*, pages 255–265. Princeton University Press, 1965. 4.5

Taiki Todo. Mechanism design with uncertainty. In *International Joint Conference on Artificial Intelligence, (IJCAI)*, 2020. 10.2

Csaba D Tóth, Joseph O'Rourke, and Jacob E Goodman. *Handbook of Discrete and Computational Geometry*. Chapman and Hall/CRC, 2017. 10.2.4

Mark Turner, Timo Berthold, Mathieu Besançon, and Thorsten Koch. Cutting plane selection with analytic centers and multiregression. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, pages 52–68. Springer, 2023. 3.3

Leslie Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984. 11

Thijs Van de Graaf. Battling for a shrinking market: oil producers, the renewables revolution, and the risk of stranded assets. In *The Geopolitics of Renewables*, pages 97–121. Springer,

2018. 10.2

Vladimir Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. 4.1

Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, pages 1163–1178, 2007. 7

Hal R Varian and Christopher Harris. The VCG auction in theory and practice. *American Economic Review*, 2014. 5.1.1

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 1961. 5, 5.1.1, 6.1

Ellen Vitercik. *Automated Algorithm and Mechanism Configuration.* PhD thesis, Carnegie Mellon University, USA, 2021. 4.4.1, 4.4.5

Rakesh V Vohra. *Mechanism design: a linear programming approach*, volume 47. Cambridge University Press, 2011. 5.1.1, 7.3.1, 8.2.1, 8.2.2

William Walsh, David Parkes, Tuomas Sandholm, and Craig Boutilier. Computing reserve prices and identifying the value distribution in real-world auctions with market disruptions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2008. Short paper. 5

Tonghan Wang, Yanchen Jiang, and David C Parkes. Gemnet: Menu-based, strategy-proof multi-bidder auctions through deep learning. In *ACM Conference on Economics and Computation (EC)*, pages 1100–1100, 2024. 7.4

Hugh E Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968. 4.5

Michael Weiss, Benjamin Lubin, and Sven Seuken. SATS: A universal spectrum auction test suite. In *Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 51–59, 2017. 6.6, 11

Gellért Weisz, András György, and Csaba Szepesvári. LEAPSANDBOUNDS: A method for approximately optimal algorithm configuration. 2018. 4.2

Franz Wesselmann and Uwe Stuhl. Implementing cutting plane management and selection techniques. In *Technical Report*. University of Paderborn, 2012. 4.4

Robert B Wilson. *Nonlinear pricing*. Oxford University Press on Demand, 1993. 9.1

Laurence A Wolsey. Valid inequalities and superadditivity for 0–1 integer programs. *Mathematics of Operations Research*, 2(1):66–77, 1977. 3

Kati Wolter. Implementation of cutting plane separators for mixed integer programs. *Dipolma thesis, Technische Universität Berlin*, 2006. 3, 3.2

Chenyang Xu and Pinyan Lu. Mechanism design with predictions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. 5, 5

Haifeng Xu. On the tractability of public persuasion with no externalities. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2708–2727. SIAM, 2020. 9.2.2

L. Xu, F. Hutter, H.H. Hoos, and K. Leyton-Brown. Satzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32(1):565–606, 2008. 4, 4.4.2

Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming. In *RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion at the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011. 4, 4.4.2

Yu Yang, Natashia Boland, Bistra Dilkina, and Martin Savelsbergh. Learning generalized strong branching for set covering, set packing, and 0-1 knapsack problems. Technical report, Technical Report, 2020., 2020. 4.2, 4.4.4

Yu Yang, Natashia Boland, and Martin Savelsbergh. Multivariable branching: A 0-1 knapsack problem case study. *INFORMS Journal on Computing*, 2021. 4.4.4, 4.4.4

Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior*, 46(1): 174–188, 2004. 10.2.1