ABSTRACT

Title of dissertation: MECHANISM DESIGN WITH GENERAL UTILITIES

Saeed Alaei, Doctor of Philosophy, 2012

Dissertation directed by: Professor Samir Khuller
Department of COMPUTER SCIENCE

This thesis studies mechanism design from an optimization perspective.

Our main contribution is to characterize fundamental structural properties of optimization problems arising in mechanism design and to exploit them to design general frameworks and techniques for efficiently solving the underlying problems. Not only do our characterizations allow for efficient computation, they also reveal qualitative characteristics of optimal mechanisms which are important even from a non-computational standpoint. Furthermore, most of our techniques are widely applicable to optimization problems outside of mechanism design such as online algorithms or stochastic optimization.

Our frameworks can be summarized as follows. When the input to an optimization problem (e.g., a mechanism design problem) comes from independent sources (e.g., independent agents), the complexity of the problem can be exponentially reduced by (i) decomposing the problem into smaller subproblems, each one involving one input source, (ii) simultaneously optimizing the subproblems subject to certain relaxation of coupling constraints, and (iii) combining the solutions of the subproblems in a

certain way to obtain an (approximately) optimal solution for the original problem.

We use our proposed framework to construct optimal or approximately optimal mechanisms for several settings previously considered in the literature and to improve upon the best previously known results. We also present applications of our techniques to non-mechanism design problems such as online stochastic generalized assignment problem which itself captures online and stochastic versions of various other problems such as resource allocation and job scheduling.

MECHANISM DESIGN WITH GENERAL UTILITIES

Saeed Alaei

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advising Committee:
Professor Samir Khuller, Chair
Professor Lawrence Ausubel, Department of Economics
Professor MohammadTaghi HajiAghayi
Professor Jason Hartline, Department of Electrical Engineering and Computer Science, Northwestern University
Professor Neil Spring

Dedication

To my parents

Morteza and Fatemeh

and my sisters

Samaneh and Samineh

for their endless love and support

Acknowledgments

This thesis would not have been possible without the help, encouragement, and support of many people.

First and foremost, I thank my advisor Samir Khuller for his guidance and invaluable advice in all aspects of my academic life and for giving me the freedom to pursue my research in algorithmic game theory. I also thank the other members of my dissertation committee, Lawrence Ausubel, MohammadTaghi HajiAghayi, Jason Hartline and Neil Spring for their time and comments.

During my graduate years, I had the opportunity to collaborate with many people. I thank my friend and co-author Azarakhsh Malekian with whom I have done most of my graduate research and coauthored many papers. I thank Jason Hartline for fruitful collaborations and for his insightful comments for a major part of my research. I thank MohammadTaghi HajiAghayi for his support and advice. I thank my other friends and coauthors Vahid Liaghat, Nima Haghpanah, Hu Fu, Kamal Jain, Barna Saha, and those whom I have not listed here.

Last but not the least, I am forever grateful to my parents to whom I owe everything I have achieved in my life.

Contents

List of Tables

List of Figures

Part I

Introduction and Background

Chapter 1

Introduction

1.1 Mechanism Design

Over the past decades, the growth of computer science has resulted in overlaps with economics, operations research, and related fields. Many problems arising in such fields can be posed as optimization problems, yet each field studies such problems from a different perspective; for example economics focuses on studying qualitative and structural properties of the solution, whereas computer science deals with the computational aspects of finding the solution. In the past decade, the massive growth of internet and computers has led to an increase in market mechanisms which are controlled and run by computers, which has consequently heightened the importance of considering the computational aspects of solving the corresponding optimization problems. Online and dynamic pricing, electronic financial markets, and algorithmic trading are some of the most prominent examples of this trend. While the new technology has allowed for far more complex market mechanisms, it has also accentuated the computational issues pertaining to designing optimal mechanisms, finding optimal strategies, and characterizing equilibria.

In solving optimization problems, we typically assume that the input to problem is not affected by how we compute the output. However this assumption is not true when the input comes from strategic agents that have their own preferences over the possible outcomes of the optimization. For example, consider the problem of matching a set of items to a set of agents. The problem can be cast as a weighted matching in a bipartite graph in which the weight of the edge from an agent to an item represents the valuation of the agent for that item. The input to the problem consists of the edge weights which are reported by the agents. However, a strategic agent would try to manipulate the matching outcome by misreporting her valuations in order to receive a better item (i.e., over reporting her value for the most preferred item or under reporting her value for the less preferred items). It can be shown that the weighted matching algorithm can be accompanied by a payment scheme that would incentivize the agents to report truthfully. In this case the matching algorithm together with the payment scheme comprise a truthful mechanism. A mechanism design problem, in general, is an optimization problem where the input comes from self interested and strategic agents who may misreport their information to manipulate the outcome in their favor. Consequently, the optimality of a mechanism is measured with respect to the true input and not the reported input. To design an optimal mechanism, without loss of generality, one can restrict attention only to truthful mechanisms, i.e., mechanisms that incentivize the agents to report truthful. Unfortunately enforcing truthfulness largely increases the computational complexity of designing optimal mechanisms, except for welfare maximizing mechanism (e.g., mechanisms whose objective is to maximize the total welfare of the agents). This increased computational complexity is due to the requirement of truthfulness. Enforcing truthfulness requires simultaneously optimizing the mechanism's outcome for all possible inputs and not just the reported input, however the number of possible input profiles grows exponentially in the number of agents.

Truthful mechanisms. In mechanism design, we often restrict our attention to *truthful* (a.k.a, *incentive compatible*) mechanisms. A truthful mechanism incentivizes the agents to report truthfully. By *revelation principle*, restriction to truthful mechanisms is without loss of generality as any non-truthful mechanism that has a Nash equilibrium can be converted to an equivalent truthful mechanism as follows. Pick any set of equilibrium strategies for the agents and embed them into the mechanism; in other words, the new mechanism collects the private information of the agents and simulates their equilibrium strategies on behalf of them. The resulting mechanism is obviously truthful and leads to the same equilibrium outcome as the original mechanism.

Mechanism design objectives. Welfare and revenue maximization are the two most common objectives considered in mechanism design. Welfare maximization aims to maximize the social welfare, i.e., the combined welfare of the society; whereas revenue maximization aims to optimize the revenue of the principal running the mechanism (e.g., the auctioneer). Although welfare maximization is the most well studied and well understood objective in mechanism design, it is the objective of interest perhaps only for governments an non-profit organizations (e.g., spectrum auctions, medicare auctions, natural resources, etc). On the other hand, revenue maximization, despite being less well studied, is a more common objective of interest in today's markets (e.g., dynamic pricing, airline tickets, hotels, online retailers). This thesis considers a general and abstract class of objectives that include both welfare and revenue maximization. In particular, most of the results of this thesis apply to any objective that can be linearly separated overt the set of agents; both welfare and revenue objectives satisfy this requirement. There are however other less common objectives which do not satisfy this requirement, such as minimizing make span in job scheduling problems.

Mechanisms with/without money. Mechanisms can be divided in two classes based on their dependence on money. Money can be defined as any common medium for measuring or exchanging utility among agents in a system. Money plays a vital role in mechanism design as it allows a mechanism to measure, to compare, and to aggregate agents' preferences while incentivizing the agents to report their preferences truthfully. Mechanism design without money is quite limited as suggested by Arrow's impossibility theorem. . However, there are scenarios were a monetary transfer is inappropriate or prohibited so a mechanism without money is required; such scenarios include voting/election, resident–hospital matching, roommate assignment, kidney exchange, etc.

Bayesian vs. prior free. The two general approaches to mechanism design, based on prior assumption about agents' preferences, are Bayesian and prior free. A Bayesian mechanism makes use of the available stochastic information about agents' preferences to optimize its average performance (i.e., to maximize the expected value of an objective). On the other hand, a prior free mechanism guarantees a certain level of performance for all possible realizations of agents' preferences. For a large class of welfare maximization problems, the VickreyClarkeGroves mechanism (VCG) obtains the optimal outcome. However almost for any other objective, including revenue maximization, any incentive compatible mechanism has to sacrifice its performance under some realizations of agents' preferences in order to maintain incentive compatibility. In other words there is no incentive compatible mechanism that is optimal under all possible realizations of agents' preferences. Sacrificing optimality is inevitable in order to maintain incentive compatibility. A Bayesian mechanism, having access to the distribution of preferences, optimizes for the most likely realizations of preferences; whereas a prior free mechanism, not making any assumption about the distribution of preferences, has to guarantee a certain level of performance for all possible realizations of agents' preferences. Consequently, Bayesian mechanisms are typically far superior to prior free mechanism in terms of approximate optimality for non-welfare objectives. The first half of this thesis discusses and develops techniques for constructing optimal or approximately optimal Bayesian mechanisms. Most of the techniques presented here can be applied to environments without money as well. The second half of this thesis considers prior free mechanism design and is mostly limited to two sided matching markets.

Theory vs. practice. Many of the results presented in this thesis may appear to be more of theoretical interest and hard to apply literally in practice. For example, optimal mechanisms often (a) randomize over outcomes, (b) discriminate against agents based on their priors, or (c) require complex or unconventional interaction between the mechanism and the agents. However, many of the theoretical results can be translated into practical approaches by indirect means. Here are some examples: (a) Online surveys often solicit participants by offering them to enter a lottery to win a prize; this can be considered as a mechanism with a randomized outcome. (b) Retailers often provide discount coupons, holiday sales, or student pricing; all of which can be considered as discrimination based on the prior (e.g., wealthy people often don't use any of these options due to the extra time and effort they will have to spend). (c) Penny auctions implement an ascending form of all pay auctions. While such mechanisms were hard, if not impossible, to implement in the past, the advent of computers and internet has allowed for mechanisms with complex behavior.

1.2 Preliminaries

Common notation. The following general notations are used throughout this thesis. Capital letter and/or Sans Serif font indicate random variables (i.e., X or x). Bold letters are used when multiple agents are involved whereas non-bold letters indicate a single agent (e.g., $\mathbf{t} = (t_1, \ldots, t_n)$ denotes a type profile of all agents where t_i denotes the type of agent i). For a vector (e.g., $\mathbf{t}$) and an index (e.g., i) we subscript the vector with minus the index (e.g., $\mathbf{t}_{-i}$) to denote the vector resulting from removing the indexed entry. We use $[n]$ to denote the set of integer numbers $\{1, \ldots, n\}$.

Agent's type. An agent's type consist of any part of the agent's information that is relevant to the outcome of a mechanism. For example, in a single item auction, an agent's type may consist of her valuation for the item and possibly her budget. We often represent the type of an agent i with $t_i \in \mathrm{T}_i$ where T_i is her type space, i.e., the space of all possible types for agent i. An agent's type space can be either discrete or continuous. A type profile of n agents is often denoted by $\mathbf{t} = (t_1, \ldots, t_n)$. We typically assume that an agent's type is her private information, although in designing Bayesian mechanisms we assume that the distribution of agent's types is publicly available.

Mechanism. A mechanism can be mathematically defined as a function that maps type profiles to outcomes. For mechanisms with money, an outcome often consists of an allocation and a payment for each agent. For example, in a single item auction, an outcome specifies for each agent (a) whether the agent receives the item or not, and (b) a payment. A *deterministic mechanism* maps each type profile to an outcome; whereas a *randomized mechanism* maps each type profile to a distribution over outcomes. A *Bayesian mechanism* optimizes its objective in expectation where the expectation is taken over the randomness of the agents' types, i.e., optimizing the average performance. Consequently, a Bayesian mechanism needs to make assumption about the distributions of agents' types. On the other hand, a *prior free* mechanism optimizes its objective for all possible profiles of agents' types, i.e., optimizing the worst case performance.

Utility function. An agent's utility function maps the types of the agents and the possible outcomes to real valued utilities; hence, it allows comparing the agent's preferences over different outcomes. We often denote the utility function of an agent i with $u_i(x_i, p_i)$ for allocation x_i and payment p_i. Agent i has *quasilinear utility* if her utility function can be written as $u_i(x_i, p_i) = v_i(x_i) - p_i$ in which $v_i(x_i)$ denotes the valuation of agent i for allocation x_i. In an environment with a set J of items where bundles of items are allocated to the agents, agent i is said to have *submodular valuations* iff

$$v_i(x_i) + v_i(x'_i) \geq v_i(x_i \cap x'_i) + v_i(x_i \cup x'_i), \qquad \forall x_i, x'_i \subseteq J$$

Incentive compatibility (IC). A mechanism is incentive compatible iff truthful reporting is a (weakly) dominant strategy for each agent. The various classes of incentive compatibility, which are often considered in the computer science literature, are listed below in the decreasing order of their restrictiveness.

- **Universally truthful (UT).** Truthful reporting is a (weakly) dominant strategy even if an agent knows the random choices of the mechanism.
- **Truthful in expectation (TIE).** Truthful reporting is a (weakly) dominant strategy in expectation over the random choices of the mechanism.

- **Dominant strategy incentive compatible (DSIC).** Truthful reporting is a (weakly) dominant strategy even if an agent knows the types of the other agents. This is also referred to as *ex post incentive compatible*
- **Bayesian incentive compatible (BIC).** Truthful reporting is a (weakly) dominant strategy in expectation over the types of the other agents. This is also referred to as *interim incentive compatible*

Note that the above definitions are not completely disjoint. In particular, UT often includes DSIC, and BIC often includes TIE, whereas DSIC may or may not include TIE. Throughout this thesis, we assume that both BIC and DSIC indicate TIE.

Individual rationality (IR). A mechanism is individually rational if every agent, regardless of her type, makes a non-negative utility by participating in the mechanism. A mechanism is *interim individually rational* if the utility of each agent is non-negative in expectation over the types of other agents. A mechanism is *ex post individually rational* if the utility of each agent is non-negative under every profile of types.

Combinatorial Auctions. A combinatorial auction is any market mechanism for allocating a set of heterogenous items (often indivisible) to a set of agents. In this case, agents have valuations for bundles of items. Combinatorial auctions are among the hardest mechanism design problems since even computing an optimal allocation (i.e, an allocation that maximizes the total valuations of the agents) is NP-hard.

1.3 Mechanism Design as an Optimization Problem

We present an example of a Bayesian mechanism design problem in the form of an optimization problem (e.g., a linear program). Consider the problem of allocating a single indivisible item to one of n budget constrained agents with the objective of maximizing revenue. For each agent $i \in [n]$, T_i denotes her type space (discrete), $t_i \in \mathrm{T}_i$ denotes her type (private), $f_i(t_i) \in [0,1]$ denotes the probability of her type being t_i, $v_i(t_i) \in \mathbb{R}_+$ denotes her valuation for the item conditioned on her type being t_i, and $B_i \in \mathbb{R}_+$ denotes her budget (public). Agents types are distributed independently.

Also $\mathbf{T} = \mathrm{T}_1 \times \cdots \times \mathrm{T}_n$ denotes the space of type profiles, $\mathbf{t} = (t_1, \ldots, t_n) \in \mathbf{T}$ denotes a type profile, and $\mathbf{f}(\mathbf{t}) = f_1(t_1) \times \cdots \times f_n(t_n)$ denotes a joint probability mass function. An optimal mechanism can be computed by the following linear program in which the variables $x_i(t_i)$ and $p_i(t_i)$ denote respectively the probability of allocation and the payment for agent i conditioned on reporting type t_i and $\mathbf{x}_i(\mathbf{t})$ denotes her probability of allocation conditioned on the reported profile of types being $\mathbf{t}$.

$$
\begin{array}{lll}
\text{maximize} & \sum_i \sum_{t_i \in \mathrm{T}_i} f_i(t_i) p_i(t_i) & \\
\text{subject to} & v_i(t_i)x_i(t_i) - p_i(t_i) \geq v_i(t_i)x_i(t_i') - p_i(t_i'), \forall i \in [n], \forall t_i, t_i' \in \mathrm{T}_i & \text{(IC)} \\
& v_i(t_i)x_i(t_i) - p_i(t_i) \geq 0, \quad \forall i \in [n], \forall t_i \in \mathrm{T}_i & \text{(IR)} \\
& p_i(t_i) \leq B_i, \quad \forall i \in [n], \forall t_i \in \mathrm{T}_i & \text{(Budget)} \\
& x_i(t_i) = \sum_{\mathbf{t}_{-i} \in \mathbf{T}_{-i}} \mathbf{f}_{-i}(\mathbf{t}_{-i}) \mathbf{x}_i(t_i, \mathbf{t}_{-i}), \quad \forall i \in [n], \forall t_i \in \mathrm{T}_i & \\
& \sum_i \mathbf{x}_i(\mathbf{t}) \leq 1, \quad \forall \mathbf{t} \in \mathbf{T} & \\
& \mathbf{x} \in [0,1]^{\mathbf{T} \times n} & \\
& x_i \in [0,1]^{\mathrm{T}_i}, \quad \forall i \in [n] & \\
& p_i \in \mathbb{R}_+^{\mathrm{T}_i}, \quad \forall i \in [n] &
\end{array}
$$

An optimal assignment for the above LP yields an optimal mechanism as follows. Given the reported profile of types $\mathbf{t}$, allocate the item to agent i with probability $x_i(\mathbf{t})$ and charge her $p_i(t_i)$ for each $i \in [n]$. Note that an agent is charged regardless of whether she wins the item or not. In other words, the mechanism collects the payments first and then allocates the item at random. The resulting mechanism is interim incentive compatible (i.e., BIC) and interim individual rational. Observe that the size of the above LP grows exponentially in the number of agents because the size of $\mathbf{x}$ is proportional to the size of $\mathbf{T}$. In Part II of this thesis we present various approaches to avoid this exponential blow up.

1.4 Outline

Part II of this thesis studies structural characteristics of optimization problems arising in Bayesian mechanism design problems and proposes various approaches for solving such problems efficiently. The size of such optimization problems usually grow exponentially in the number of agents. chapter 2 and chapter 4 present two fundamental approaches for solving such optimization problems. chapter 5 abstracts away some of the main technical contributions of chapter 4 and present direct applications of them to problems outside of mechanism design.

Part III of this thesis considers prior free mechanism design. chapter 6 studies competitive equilibrium in matching markets with general (non-quasilinear) utilities.

1.4.1 Bibliographical Notes

Most of the results of chapter 2 have appeared in Alaei et al. (2012a). A preliminary version of the results of chapter 4 have appeared in Alaei (2011). Some of the results of chapter 5 have appeared in Alaei et al. (2012b) and Alaei (2011). A subset of chapter 6 has appeared in Alaei et al. (2011).

Part II

Bayesian Mechanism Design

Chapter 2

Multi to Single Agent Reduction (Interim)

2.1 Introduction

The main challenge of Bayesian mechanism design arises from the fact that the corresponding optimization problem has to consider all joint type profiles of agents simultaneously due to incentive compatibility constraints, and the number of such joint type profiles grows exponentially in the number of agents. We aim to address this challenge by providing a general decomposition technique for mechanism design problems where (i) the objective is linearly separable over the agents (e.g., welfare or revenue), (ii) agents' types are distributed independently, and (iii) inter agent constraints only consist of allocation constraints (e.g., supply constraint).

Our decomposition approach relies on the assumption that the utility of each agent only depends on her own type/outcome, and not the types/outcomes of other agents. Every multi agent mechanism induces a single agent mechanism on each agent which can be obtained by fixing one agent and simulating the other agents as follows: draw the types of the other agents at random from their corresponding distribution; run the multi agent mechanism on the designated agent and the simulated agents; and ignore the outcomes for the simulated agents. The fundamental idea behind the decomposition is to optimize over single agent mechanisms simultaneously, while ensuring that the resulting single agent mechanisms can be combined into a feasible multi agent mechanism.

The main challenge faced by such a decomposition approach is that the joint feasibility constraints over the allocations introduce couplings in the outcome of the optimal solution. The joint feasibility constraints are typically the supply constraints. For example, when agents are independent, a revenue maximizing seller with unlimited supply can decompose the problem over the agents and optimize for each agent independently; however, in the presence of supply constraints, a direct decomposition is not possible. Our decomposition approach is based on characterizing the space of jointly feasibly allocation rules and simultaneously optimizing the single agent mechanisms subject to the feasibility of the joint allocation rule.

Related Work. Myerson (1981) characterized Bayesian optimal auctions in environments with quasi-linear risk-neutral single-dimensional agent preferences. Bulow and Roberts (1989) reinterpreted Myerson's approach as reducing the multi-agent auction problem to a related single-agent problem. Our work generalize this reduction-based approach to multi dimensional auction problems.

An important aspect of our approach is that it can be applied to multi-dimensional agent preferences. Multi-dimensional preferences can arise as distinct values for different goods or services or different configurations of a good or service being auctioned, in specifying a private budget and a private value, or in specifying preferences over risk. We briefly review related work for agent preferences with multiple values, budgets, or risk parameters.

Multi-dimensional valuations are well known to be difficult. For example, Rochet and Chone (1998), showed that, because *bunching*[1] can not be ruled out easily, the optimal auctions for multi-dimensional valuations are dramatically different from those for single dimensional valuations. Because of this, most results are for cases with special structure (e.g., Armstrong, 1996; Wilson, 1994; McAfee and McMillan, 1988) and often, by using such structures, reduce the problems to single-dimensional ones (e.g., Spence, 1980; Roberts, 1979; Mirman and Sibley, 1980). Our framework does not need any such structure.

A number of papers consider optimal auctions for agents with budgets (see, e.g.,

[1]Bunching refers to the situation in which a group of distinct types are treated the same way in by the mechanism.

Pai and Vohra, 2008; Che and Gale, 1995; Maskin, 2000). These papers rely on budgets being public or the agents being symmetric; our technique allows for a non-identical prior distribution and private budgets. Mechanism design with risk averse agents was studied by Maskin and Riley (1984) and Matthews (1983). Both works assume i.i.d. prior distributions and have additional assumptions on risk attitudes; our reduction does not require these assumptions.

Characterization of interim feasibility plays a vital role in this work. For single-item single-unit auctions, necessary and sufficient conditions for interim feasibility were developed through a series of works (Maskin and Riley, 1984; Matthews, 1984; Border, 1991, 2007; Mierendorff, 2011); this characterization has proved useful for deriving properties of mechanisms, Manelli and Vincent (2010) being a recent example. Border (1991) characterized symmetric interim feasible auctions for single-item auctions with identically distributed agent preferences. His characterization is based on the definition of "hierarchical auctions." He observes that the space of interim feasible mechanisms is given by a polytope, where vertices of this polytope corresponding to hierarchical auctions, and interior points corresponding to convex combinations of vertices. Mierendorff (2011) generalize Border's approach and characterization to asymmetric single-item auctions. Our work generalizes the characterization from asymmetric single-unit auctions to asymmetric multi-unit and matroid auctions.

Our main result provides computational foundations to the interim feasibility characterizations discussed above. We show that interim feasibility can be checked, that interim feasible allocation rules can be optimized over, and that corresponding ex post implementations can be found. Independently and contemporaneously Cai et al. (2012) provided similar computational foundations for the single-unit auction problem and multi-item auctions with agents with additive preferences. Their approach to the single-unit auction problem is most comparable to our approach for the multi-unit and matroid auction problems where the optimization problem is written as a convex program which can be solved by the ellipsoid method; while these methods result in strongly polynomial time algorithms they are not considered practical. In contrast, our single-unit approach, when the single-agent problems can be solved by a linear program, gives a single linear program which can be practically solved.

2.2 Preliminaries

We begin by defining the model and some notation.

Model. There are n agents; each agent $i \in [n]$ has a discrete type space T_i and a private type $t_i \in \mathrm{T}_i$. The agents' type profile $\mathbf{t} = (t_1, \ldots, t_n) \in \mathrm{T}_1 \times \ldots \times \mathrm{T}_n = \mathbf{T}$ is distributed according to a publicly known distribution with probability mass function $\mathbf{f} : \mathbf{T} \rightarrow [0, 1]$; WLOG, assume that all types have non-zero probability mass. A multi agent mechanism maps each type profile to a distribution over outcomes; an outcome specifies the allocation as well as extra attributes (e.g., payment, etc) for each agent. For each agent i, let $\mathrm{X}_i \subset \mathbb{R}_+^m$ and W_i denote the space of feasible allocations and the space of feasible attributes respectively. The space of jointly feasible allocations is denoted by $\mathbf{X} \subseteq \mathrm{X}_1 \times \ldots \times \mathrm{X}_n$. Also let $\mathbf{W} = \mathrm{W}_1 \times \ldots \times \mathrm{W}_n$. A multi agent mechanism $M : \mathbf{T} \rightarrow \Delta(\mathbf{X}) \times \Delta(\mathbf{W})$ maps type profiles to distributions over allocations and attributes.Each agent i has a utility function $u_i : \mathrm{T}_i \times \mathrm{X}_i \times \mathrm{W}_i \rightarrow \mathbb{R}$ that maps the agent's type and outcome to a real valued utility. Given a space of feasible mechanisms[2] $\mathbf{M} \subset [\mathbf{T} \rightarrow \Delta(\mathbf{X}) \times \Delta(\mathbf{W})]$, we are interested in computing a mechanism in $\mathbf{M}$ that maximizes[3] the expected value of a given objective function $\textsc{Obj} : \mathbf{T} \times \mathbf{X} \times \mathbf{W} \rightarrow \mathbb{R}$; formally we want to compute $M \in \mathbf{M}$ that maximizes $\mathbf{E}_{\mathbf{t} \sim \mathbf{f}, (\mathbf{x}, \mathbf{w}) \sim M(\mathbf{t})}[\textsc{Obj}(\mathbf{t}, \mathbf{x}, \mathbf{w})]$.

Notation. A function $\widehat{\mathbf{x}} : \mathbf{T} \rightarrow \Delta(\mathbf{X})$ that maps type profiles to distributions over allocations is called an *allocation rule*; the space of feasible allocation rules is denoted by $\widehat{\mathbf{X}} = [\mathbf{T} \rightarrow \Delta(\mathbf{X})]$ and $\widehat{\mathrm{X}}_i = [\mathrm{T}_i \rightarrow \Delta(\mathrm{X}_i)]$ for each agent i. Similarly, a function $\widehat{\mathbf{w}} : \mathbf{T} \rightarrow \Delta(\mathbf{W})$ is called an *attribute rule*; the space of feasible attribute rules is denoted by $\widehat{\mathbf{W}} = [\mathbf{T} \rightarrow \Delta(\mathbf{W})]$, and $\widehat{\mathrm{W}}_i = [\mathrm{T}_i \rightarrow \Delta(\mathrm{W}_i)]$ for each agent i. Any pair of $\widehat{\mathbf{x}} \in \widehat{\mathbf{X}}$ and $\widehat{\mathbf{w}} \in \widehat{\mathbf{W}}$ define a multi agent mechanism which is specified as $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$; similarly any $\widehat{x}_i \in \widehat{\mathrm{X}}_i$ and $\widehat{w}_i \in \widehat{\mathrm{W}}_i$ define a single agent mechanism

[2]Note that additional constraints (e.g., incentive compatibility, budget, etc) can be incorporated in $\mathbf{M}$.

[3]All of our results can be applied to minimization problems by simply maximizing the negation of the objective function.

$M_i = (\widehat{x}_i, \widehat{w}_i)$ for agent i. Note that every mechanism can be uniquely specified by its allocation rule and attribute rule.

For notational convenience, we use $\mathsf{x}^M(\mathbf{t})$ and $\mathsf{w}^M(\mathbf{t})$ to denote the random variables corresponding to the allocations and attributes of a mechanism M for type profile $\mathbf{t}$ (i.e., assuming $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$, random variable $\mathsf{x}^M(\mathbf{t})$ and $\mathsf{w}^M(\mathbf{t})$ are drawn from distributions $\widehat{\mathbf{x}}(\mathbf{t})$ and $\widehat{\mathbf{w}}(\mathbf{t})$ respectively.

The single agent mechanism induced on agent i by a multi agent mechanism $M \in \widehat{\mathbf{X}} \times \widehat{\mathbf{W}}$ is denoted by $[[M]]_i$. Such a single agent mechanism can be obtained by simulating the other agents according to their respective distributions[4]; furthermore $\mathrm{M}_i \subseteq \widehat{\mathrm{X}}_i \times \widehat{\mathrm{W}}_i$ denotes the space of all feasible single agent mechanisms for agent i, i.e., $\mathrm{M}_i = \{[[M]]_i | M \in \mathbf{M}\}$.

Assumptions. We make the following assumptions.

(A1) **Independence.** The agents' types must be distributed independently, i.e., $\mathbf{f} = f_1 \times \ldots \times f_n$ where $f_i : \mathrm{T}_i \to [0, 1]$ is the probability mass function for t_i. Note that if agent i has multidimensional types, f_i itself does not need to be a product distribution.

(A2) **Linear Separability of Objective.** The objective function must be linearly separable over the agents, i.e., $\textsc{Obj}(\mathbf{t}, \mathsf{x}, \mathsf{w}) = \sum_i \textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{w}_i)$ where t_i, x_i, and w_i respectively represent the type, the allocation, and the payment of agent i.

(A3) **Linearity in Allocation.** The utility functions of the agents, the objective function, and the space of feasible mechanisms $\mathbf{M}$ must be linear in allocation, as formally defined next. For any arbitrary mechanism $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}}) \in \mathbf{M}$ and any allocation rule $\widehat{\mathbf{x}}' \in \widehat{\mathbf{X}}$ whose expected allocation is the same as $\widehat{\mathbf{x}}$ for each type of each agent, the mechanism $M' = (\widehat{\mathbf{x}}', \widehat{\mathbf{w}})$ must also be feasible (i.e., must be in $\mathbf{M}$) and must yield the same expected objective value and the same expected utilities as M' for each type of each agent.

[4] The single agent mechanism induced on agent i can be obtained by simulating all agents other than i by drawing a random $\mathbf{t}_{-i}$ from $\mathbf{f}_{-i}$ and running M on agent i and the $n-1$ simulated agents with types $\mathbf{t}_{-i}$; note that this is a single agent mechanism because the simulated agents are just part of the mechanism.

We linearly extend u_i and $\textsc{Obj}_i$ to any allocation in the convex hall of X_i. Throughout the rest of this note, we treat u_i and $\textsc{Obj}_i$ as the linear extensions of the corresponding functions.

(A4) **Convexity.** The space of feasible mechanisms, $\mathbf{M}$, must be convex. In other words, for any two mechanisms $M, M' \in \mathbf{M}$ and any $\beta \in [0, 1]$, the mechanism $M'' = \beta M + (1-\beta)M'$ must also be in $\mathbf{M}$. M'' can be interpreted as a mechanism which runes M with probability β and runs M' with probability $1 - \beta$.

(A5) **Decomposability.** The constraints imposed by the space of feasible mechanisms, $\mathbf{M}$, must be decomposable to allocation constraints which are dictated by $\mathbf{X}$ and single agent constraints (e.g., incentive compatibility, budget, etc). In other words, $\mathbf{M}$ must impose no inter agent constrains except for those implied by $\mathbf{X}$. Formally, the decomposability assumption requires that for any mechanism $M \in \widehat{\mathbf{X}} \times \widehat{\mathbf{W}}$, if $[[M]]_i \in \mathrm{M}_i$ (for all agents i), then it must be $M \in \mathbf{M}$.

Multi agent problem. A multi agent mechanism for n agents induces n single agent mechanisms, one per each agent. Furthermore, the expected objective value of the mechanism and the expected utilities of the agents only depend on the induced single agent mechanisms (this follows from assumption $A2$ and given that the utility of each agent depends only on her own outcome). Consequently, one would hope to obtain an optimal multi agent mechanism by combining optimal single agent mechanisms; however, the resulting multi agent mechanism could yield infeasible allocations due to the joint feasibility constraints imposed by $\mathbf{X}$. To ensure joint feasibility of allocations, one can simultaneously optimize the single agent mechanisms over the space of feasible allocation rules. The multi agent optimization problem is captured by the following program.

$$
\begin{aligned}
\text{maximize} \quad & \sum_i \sum_{t_i \in \mathrm{T}_i} \mathbf{E}_{\mathsf{x}_i \sim \widehat{x}_i(t_i), \mathsf{w}_i \sim \widehat{w}_i(t_i)} \left[\mathrm{O{\small BJ}}_i(t_i, \mathsf{x}_i, \mathsf{w}_i) \right] && (OPT) \\
\text{subject to} \quad & \widehat{x}_i(t_i) = \sum_{\mathbf{t}_{-i} \in \mathbf{T}_{-i}} \mathbf{f}_{-i}(\mathbf{t}_{-i}) \widehat{\mathbf{x}}_i(t_i, \mathbf{t}_{-i}), && \forall i \in [n], \forall t_i \in \mathrm{T}_i \\
& (\widehat{x}_i, \widehat{w}_i) \in \mathrm{M}_i, && \forall i \in [n] \\
& \widehat{\mathbf{x}} \in \widehat{\mathrm{X}}
\end{aligned}
$$

Theorem 1. *Given an optimal assignment of $\widehat{\mathbf{x}}$ and $\widehat{x}_i$ and $\widehat{w}_i$ (for all i) in program (OPT), an optimal multi agent mechanism is given by $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$, in which $\widehat{\mathbf{w}} = \widehat{w}_1 \times \cdots \times \widehat{w}_n$ (i.e., $\widehat{\mathbf{w}}$ is a product distribution). Furthermore, the optimal value of that program is equal to the expected objective value of M.*

Proof. Consider a hypothetical optimal multi agent mechanism and let $\widehat{\mathbf{x}}$ be its allocation rule and let $(\widehat{x}_i, \widehat{w}_i)$ be the single agent mechanism induced on agent i for each i. By $A1$ and $A3$, the contribution of agent i to the expected objective value of the optimal mechanism is exactly the same as its contribution to the expected objective value of the single agent mechanism $(\widehat{x}_i, \widehat{w}_i)$ which is $\sum_{t_i \in \mathrm{T}_i} \mathbf{E}_{\mathsf{x}_i \sim \widehat{x}_i(t_i), \mathsf{w}_i \sim \widehat{w}_i(t_i)}[\mathrm{O{\small BJ}}_i(t_i, \mathsf{x}_i, \mathsf{w}_i)]$. Since $\widehat{\mathbf{x}}$, $\widehat{x}_i$ and $\widehat{w}_i$ form a feasible assignment for convex program (OPT), the optimal value of the convex program must be at least as much as the expected objective value of the optimal mechanism. On the other hand, by $A5$, any feasible assignment of $\widehat{\mathbf{x}}$ and $\widehat{x}_i$ and $\widehat{w}_i$ for the convex program can be turned into a feasible multi agent mechanism $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$, in which $\widehat{\mathbf{w}} = \widehat{w}_1 \times \cdots \times \widehat{w}_n$, therefore the optimal assignment for the convex program must yield an optimal multi agent mechanism. □

Unfortunately, the dimension of the space of feasible allocation rules, $\widehat{\mathbf{X}}$, is proportional to the dimension of $\mathbf{T}$ (i.e., $|\mathrm{T}_1| \times \cdots \times |\mathrm{T}_n|$) which grows exponentially in the number of agents. So the size of program (OPT) is exponential in the size of the input. However, because of assumption $A3$ on linearity in allocations, only the expected allocation of each type is relevant from the perspective of the multi agent optimization problem; in other words, two mechanisms that are identical except for their allocation rules, but have the same expected allocation for each type, are equivalent in terms of feasibility, optimality (i.e., expected objective value), and expected utility for each type. This observation is the key idea to the multi to singe agent decomposition of the next section.

Interim allocation rule. For a mechanism $M \in \widehat{\mathbf{X}} \times \widehat{\mathbf{W}}$, the *interim allocation rule* specifies the expected allocation for every type of every agent normalized by the probability of that type which can be formally defined as follows. WLOG, assume that $\mathrm{T}_1, \ldots, \mathrm{T}_n$ are disjoint[5] and let $\mathrm{T}_N = \bigcup_i \mathrm{T}_i$ be the set of all types. The interim allocation rule $\widetilde{\mathbf{x}} \in \mathbb{R}_+^{\mathrm{T}_N \times m}$ for a mechanism $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$ is defined as follows.

$$\widetilde{\mathbf{x}}(t_i) = f_i(t_i) \quad \mathbf{E}_{\mathbf{t}_{-i} \sim \mathbf{f}_{-i}} \left[\mathsf{x}_i^M(t_i, \mathbf{t}_{-i}) \right], \qquad \forall i \in [n], \forall t_i \in \mathrm{T}_i \qquad \text{(IA)}$$

An interim allocation rule is subscripted by an agent (e.g., $\widetilde{x}_i$) to denote its restriction to the types of that agent.

To avoid confusion we often refer to an allocation rule $\widehat{\mathbf{x}} \in \widehat{\mathbf{X}}$ as an *ex post allocation rule*. An ex post allocation rule $\widehat{\mathbf{x}}$ *implements* an interim allocation rule $\widetilde{\mathbf{x}}$ iff the following equation holds.

$$\widetilde{\mathbf{x}}(t_i) = \sum_{\mathbf{t}_{-i} \in \mathbf{T}_{-i}} f(t_i, \mathbf{t}_{-i}) \widehat{\mathbf{x}}_i(t_i, \mathbf{t}_{-i}), \qquad \forall i \in [n], \forall t_i \in \mathrm{T}_i \qquad \text{(IA-XPA)}$$

An interim allocation rule is feasible iff it can be implemented by a feasible ex post allocation rule. Note that there could be many ex post allocation rules that implement the same interim allocation rule. The space of feasible interim allocation rules is denoted by $\widetilde{\mathbf{X}}$.

2.3 Decomposition via Interim Allocation Rule

We show that an optimal multi agent mechanism can be computed by simultaneously optimizing the single agent mechanisms over the space of feasible interim allocation

[5]If not, label all types of each agent with the name of that agent, i.e., for each $i \in [n]$ replace T_i with $\mathrm{T}'_i = \{(i,t) | t \in \mathrm{T}_i\}$ so that $\mathrm{T}'_1, \cdots, \mathrm{T}'_n$ are disjoint.

rules. Note that the dimension of the space of interim allocation rules is proportional to the size of T_N (i.e., $|\mathrm{T}_1| + \cdots + |\mathrm{T}_n|$) which grows linearly in the number of agents. Note that by assumption $A3$, the objective function, the utilities, and the space of feasible mechanisms are linear in allocations and hence depend only on the expected allocation of each type, i.e., the interim allocation rule. In other words, two ex post allocation rules, which yield the same interim allocation rule, are equivalent in all relevant aspects. Next, we define the single agent problem formally and then discuss the multi to single agent decomposition.

Single agent problem. The single agent problem for agent i is to compute an optimal single agent mechanism subject to a given interim allocation rule [6] $\widetilde{x}_i$, and to compute its expected objective value which is denoted by $R_i(\widetilde{x}_i)$. The single agent problem is captured by the following program whose optimal value is equal to $R_i(\widetilde{x}_i)$ and whose optimal assignment for $(\widehat{x}_i, \widehat{w}_i)$ gives an optimal single agent mechanism subject to $\widetilde{x}_i$.

$$\begin{aligned} \text{maximize} \quad & \sum_{t_i \in \mathrm{T}_i} \mathbf{E}_{\mathsf{x}_i \sim \widehat{x}_i(t_i), \mathsf{w}_i \sim \widehat{w}_i(t_i)} \left[\mathrm{O{\scriptstyle BJ}}_i(t_i, \mathsf{x}_i, \mathsf{w}_i)\right] && (OPT_i) \\ \text{subject to} \quad & \mathbf{E}_{\mathsf{x}_i \sim \widehat{x}_i(t_i)} \left[\mathsf{x}_i\right] = \widetilde{x}_i(t_i), && \forall t_i \in \mathrm{T}_i \\ & (\widehat{x}_i, \widehat{w}_i) \in \mathrm{M}_i \end{aligned}$$

We will refer to R_i as the *optimal benchmark* for agent i. If there are no feasible single agent mechanisms for a given $\widetilde{x}_i$, $R_i(\widetilde{x}_i)$ is defined as $-\infty$.

Theorem 2. *$R_i(\widetilde{x}_i)$, the expected objective value of the optimal single agent mechanism for agent i subject to an interim allocation rule $\widetilde{x}_i$, is concave in $\widetilde{x}_i$ and has a convex domain.*

Proof. Consider any $\widetilde{x}_i$ and $\widetilde{x}'_i$ in the domain of R_i and any $\beta \in [0, 1]$. We shall show that $\widetilde{x}''_i = \beta\widetilde{x}_i + (1-\beta)\widetilde{x}'_i$ is in the domain of R_i and $R_i(\widetilde{x}''_i) \geq \beta R_i(\widetilde{x}_i) + (1-\beta) R_i(\widetilde{x}'_i)$. Let M_i and M'_i denote optimal single agent mechanisms for agent i subject to $\widetilde{x}_i$ and

[6] I.e., the expected allocation of the mechanism to type t_i must be $\widetilde{x}_i(t_i)$.

$\tilde{x}'_i$ respectively. Define $M''_i = \beta M_i + (1-\beta)M'_i$ (i.e., M''_i runs M_i with probability β and runs M'_i with probability $1-\beta$). M''_i is feasible by assumption $A4$ and has interim allocation rule $\tilde{x}''_i$, so $\tilde{x}''_i$ is in the domain of R_i. Furthermore, the expected objective value of the optimal single agent mechanism subject to $\tilde{x}''_i$ is at least as much as the expected objective value of M''_i which is exactly $R_i(\tilde{x}_i) + (1-\beta)R_i(\tilde{x}'_i)$; so R_i is concave. □

Multi to single agent decomposition. The multi agent optimization problem of program (OPT) can be rephrased in terms of the interim allocation rule as the following program.

$$\begin{aligned} \text{maximize} \quad & \sum_i R_i(\tilde{x}_i) \qquad (OPT_{\text{interim}}) \\ \text{subject to} \quad & \tilde{\mathbf{x}} \in \widetilde{\mathbf{X}} \end{aligned}$$

Note that the above program is a convex program because R_i are concave (Theorem 2) and $\widetilde{\mathbf{X}}$ is a convex space. Therefore an optimal assignment can be computed efficiently (e.g., in polynomial time) assuming that R_i can be computed efficiently for each i.

Theorem 3. *Given an optimal assignment of $\tilde{\mathbf{x}}$ in convex program (OPT_{interim}) and an optimal single agent mechanism $M_i = (\widehat{x}_i, \widehat{w}_i)$ subject to $\tilde{x}_i$ for each agent i, an optimal multi agent mechanism is given by $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$ in which $\widehat{\mathbf{x}}$ is any ex post allocation rule implementing $\tilde{\mathbf{x}}$ and $\widehat{\mathbf{w}} = \widehat{w}_1 \times \cdots \times \widehat{w}_n$ (i.e., a product distribution).*

Proof. Let $\widehat{\mathbf{x}}$ be an ex post allocation rule corresponding to $\tilde{\mathbf{x}}$. Let $\widehat{x}'_i$ denote the ex post allocation rule induced by $\widehat{\mathbf{x}}$ on agent i. Note that $\widehat{x}_i$ and $\widehat{x}'_i$ may not necessarily be the same for each agent i, but they both produce the same interim allocation rule $\tilde{\mathbf{x}}$. By $A3$, $(\widehat{x}'_i, \widehat{w}_i)$ is feasible mechanism for each agent i which is equivalent to $(\widehat{x}'_i, \widehat{w}_i)$ and is also an optimal assignment for the convex program. Observe that $\widehat{\mathbf{x}}$ and $\widehat{x}'_i$ and $\widehat{w}_i$ (for all i) form an optimal assignment for convex program (OPT); consequently $M = (\widehat{\mathbf{x}}, \widehat{w}_1 \times \cdots \times \widehat{w}_n)$ in an optimal multi agent mechanism by Theorem 1. □

Convex program (OPT_{interim}) together with Theorem 3 provide a generic approach for computing an optimal multi agent mechanism; however, in order for this approach to be computationally efficient, each of the following steps must be computationally efficient.

1. Computing $R_i(\tilde{x}_i)$.
2. Optimizing over $\widetilde{\mathbf{X}}$; typically, this can be done using the Ellipsoid method together with a separation oracle or an explicit representation of $\widetilde{\mathbf{X}}$ (in the form of a collection of linear constraints).
3. Computing an optimal single agent mechanism $M_i = (\hat{x}_i, \widehat{w}_i) \in \mathrm{M}_i$ subject to $\tilde{x}_i$, for each agent i.
4. Computing an ex post allocation rule $\hat{\mathbf{x}}$ that implements $\tilde{\mathbf{x}}$.

The second and fourth steps are discussed in §2.4. The first and the third steps together form the single agent problem which can be computed for each agent i typically in time polynomial in the size of T_i, as shown in the next example.

Example. There is one indivisible item which can be allocated to at most one of n agents with the objective of maximizing revenue. The item can be painted with one of c available colors. $v_i : \mathrm{T}_i \to \mathbb{R}_+^c$ denotes the valuations of each type of agent i for each color. An outcome for agent i specifies an allocation (either 0 or 1) and an attribute which specifies the color and payment; therefore $\mathrm{X}_i = \{0,1\}$ and $\mathrm{W}_i = [c] \times \mathbb{R}$. The space of jointly feasible allocations is given by

$$\mathbf{X} = \left\{ \mathsf{x} \in \{0,1\}^n \middle| \sum_i \mathsf{x}_i \leq 1 \right\}$$

Let $\tilde{x} \in [0,1]^{\mathrm{T}_N \times 1}$ be an interim allocation rule. Pick an arbitrary agent i. Recall that $\tilde{x}_i \in [0,1]^{\mathrm{T}_i \times 1}$ denotes the restriction of $\tilde{x}$ to the types of agent i. An optimal single agent mechanism for agent i, subject to $\tilde{x}_i$, is computed by the following linear program and $R_i(\tilde{x}_i)$ is given by the optimal value of this program as a function of $\tilde{x}_i$.

$$\begin{aligned}
\text{maximize} \quad & \sum_{t_i \in \mathrm{T}_i} f_i(t_i) p_i(t_i) \\
\text{subject to} \quad & v_i(t_i) \cdot y_i(t_i) - p_i(t_i) \geq v_i(t_i) \cdot y_i(t_i') - p_i(t_i'), && \forall t_i, t_i' \in \mathrm{T}_i && \text{(IC)} \\
& v_i(t_i) \cdot y_i(t_i) - p_i(t_i) \geq 0, && \forall t_i \in \mathrm{T}_i && \text{(IR)} \\
& \sum_{\ell=1}^{c} y_i(t_i, \ell) = \widetilde{x}_i(t_i) / f_i(t_i), && \forall t_i \in \mathrm{T}_i \\
& y_i \in [0,1]^{\mathrm{T}_i \times c} \\
& p_i \in \mathbb{R}_+^{\mathrm{T}_i}
\end{aligned}$$

For agent i, conditioned on having type t_i, $y_i(t_i, \ell)$ denotes the probability of receiving the item painted with color ℓ, and $p(t_i)$ denotes the corresponding payment. Recall that a single agent mechanism maps each type of the agent to distributions over allocations and attributes. In the above example, each type $t_i \in \mathrm{T}_i$ is mapped to the following distributions over allocations and attributes.

- x_i is 1 with probability $\frac{\widetilde{x}_i(t_i)}{f_i(t_i)}$, and is 0 otherwise.
- w_i is $(\ell, p_i(t_i))$ with probability $\frac{y_i(t_i,\ell)}{\widetilde{x}_i(t_i)} f_i(t_i)$, for each $\ell \in [c]$.

2.4 Optimization and Implementation of Interim Allocation Rules

We address the computational issues pertaining to (i) optimization over the space of interim allocation rules, and (ii) ex post implementation of interim allocation rules. We study fundamental characteristic of the space of interim allocation rules which allow computationally tractable methods for both optimization and implementation over this space; such characteristics are important even from a non-computational stand point as they relate to qualitative characteristics of optimal multi agent mechanisms.

Optimization over $\widetilde{\mathbf{X}}$ requires either a separation oracle or an explicit representation of $\widetilde{\mathrm{X}}$ as a collection of linear constraints; however the latter approach is often not computationally efficient since it requires exponentially many linear constraints. In the rest of this section, we develop solutions based on either of those approaches.

An ex post implementation of a feasible interim allocation rule can be obtained without an explicit construction of a corresponding ex post allocation rule. Recall that an ex post allocation rule $\hat{\mathbf{x}}$ maps every type profile $\mathbf{t}$ to a distribution $\hat{x}(\mathbf{t})$ over feasible allocations. Furthermore, to run a mechanism it is enough to sample from $\hat{x}(\mathbf{t})$. We show that the interim allocation rules corresponding to the vertices of $\widetilde{\mathbf{X}}$ have simple deterministic implementations, i.e., the corresponding $\hat{x}(\mathbf{t})$ is deterministic and efficiently computable for all $\mathbf{t}$. Consequently, any $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ can be implemented by sampling a vertex $\tilde{x}'$ of $\widetilde{\mathbf{X}}$ at random such that $\mathbf{E}[\tilde{\mathbf{x}}'] = \tilde{x}$ and then choosing the determinist allocation corresponding to an ex post implementation of $\tilde{x}'$.

2.4.1 Single Unit Allocation Constraints

We consider the space of feasible allocations defined by single unit allocation constraints (i.e., at most one agent can be allocated to). For such space of feasible allocations, we characterize feasibility of interim allocation rules as implementability via a particular, simple *stochastic sequential allocation rule* whose dynamics can be captured by $O(|\mathrm{T}_N|^2)$ linear equations. Consequently, we obtain a compact formulation of the space of feasible interim allocation rules.

The space of feasible allocations defined by single unit allocation constrains can be formulated as follows.

$$\mathbf{X} = \left\{ \mathsf{x} \in \{0,1\}^n \middle| \sum_i \mathsf{x}_i \leq 1 \right\}$$

Observe that the space of feasible allocations for each agent i is exactly $\mathrm{X}_i = \{0,1\}$.

A stochastic sequential allocation algorithm is parameterized by a stochastic transition table. Such a table specifies the probability by which an agent with a given type can steal a token from a preceding agent with a given type. For simplicity in describing the process we will assume the token starts under the possession of a "dummy agent" indexed by 0; the agents are then considered in the arbitrary order from 1 to n; and the agent with the token at the end of the process is the one that is allocated (or none are allocated if the dummy agent retains the token).

Definition 1 (stochastic sequential allocation algorithm)**.** *Parameterized by a stochastic transition table π, the* stochastic sequential allocation algorithm (SSA) *computes the allocations for a type profile $\mathbf{t} \in \mathbf{T}$ as follows:*

1. *Give the token to the dummy agent 0 with dummy type t_0.*

2. *For each agent $i \in \{1, \ldots, n\}$ (in order):*

   If agent i' has the token, transfer the token to agent i with probability $\pi(t_{i'}, t_i)$.

3. *Allocate to the agent who has the token (or none if the dummy agent has it).*

First, we present a dynamic program, in the form of a collection of linear equations, for calculating the interim allocation rule implemented by SSA for a given π. Let $y(t_{i'}, i)$ denote the ex-ante probability of the event that agent i' has type $t_{i'}$ and is holding the token at the end of iteration i. Let $z(t_{i'}, t_i)$ denote the ex-ante probability in iteration i of SSA that agent i has type t_i and takes the token from agent i' who has type $t_{i'}$.

The following additional notation will be useful in this section. For any subset of agents $A \subseteq N = \{1, \ldots, n\}$, we define $\mathrm{T}_A = \bigcup_{i \in A} \mathrm{T}_i$ (Recall that without loss of generality agent type spaces are assumed to be disjoint.). The shorthand notation $t_i \in S$ for $S \subseteq \mathrm{T}_N$ will be used to quantify over all types in S and their corresponding agents (i.e., $\forall t_i \in S$ is equivalent to $\forall i \in N, \forall t_i \in S \cap \mathrm{T}_i$).

The interim allocation rule $\tilde{\mathbf{x}}$ resulting from the SSA is exactly given by the dynamic program specified by the following linear equations.

$$y(t_0, 0) = 1, \tag{S.1}$$

$$y(t_i, i) = \sum_{t_{i'} \in \mathrm{T}_{\{0,\dots,i-1\}}} z(t_{i'}, t_i), \qquad \forall t_i \in \mathrm{T}_{\{1,\dots,n\}} \tag{S.2}$$

$$y(t_{i'}, i) = y(t_{i'}, i-1) - \sum_{t_i \in \mathrm{T}_i} z(t_{i'}, t_i), \quad \forall i \in \{1, \dots, n\}, \forall t_{i'} \in \mathrm{T}_{\{0,\dots,i-1\}} \tag{S.3}$$

$$z(t_{i'}, t_i) = y(t_{i'}, i-1)\pi(t_{i'}, t_i) f_i(t_i), \qquad \forall t_i \in \mathrm{T}_{\{1,\dots,n\}}, \forall t_{i'} \in \mathrm{T}_{\{0,\dots,i-1\}} \tag{π}$$

$$y(t_i, n) = \tilde{\mathbf{x}}(t_i), \qquad \forall t_i \in \mathrm{T}_{\{1,\dots,n\}}$$

Note that π is the only adjustable parameter in the SSA algorithm, so by relaxing the equation (π) and replacing it with the following inequality we can specify all possible dynamics of the SSA algorithm.

$$0 \leq z(t_{i'}, t_i) \leq y(t_{i'}, i-1) f_i(t_i), \qquad \forall t_i \in \mathrm{T}_{\{1,\dots,n\}}, \forall t_{i'} \in \mathrm{T}_{\{0,\dots,i-1\}} \tag{S.4}$$

Let S denote the convex polytope captured by the 4 sets of linear constraints (S.1) through (S.4) above, i.e., $(y, z) \in \mathrm{S}$ iff y and z satisfy the aforementioned constraints. Note that every $(y, z) \in \mathrm{S}$ corresponds to some stochastic transition table π by solving equation (π) for $\pi(t_i, t_{i'})$. We show that S captures all feasible interim allocation rules; in other words, the space of feasible interim allocation rules, $\widetilde{\mathbf{X}}$, is the projection of S through equation $\tilde{\mathbf{x}}(\cdot) = y(\cdot, n)$.

Theorem 4. *An interim allocation rule $\tilde{\mathbf{x}}$ is feasible if and only if it can be implemented by the SSA algorithm for some choice of stochastic transition table π. In other words, $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ iff there exists $(y, z) \in \mathrm{S}$ such that $\tilde{\mathbf{x}}(t_i) = y(t_i, n)$ for all $t_i \in \mathrm{T}_N$.*

Corollary 1. *$\widetilde{\mathbf{X}}$ can be compactly formulated by $O(|\mathrm{T}_N|^2)$ variables and linear constraints.*

Corollary 2. *The convex program* (OPT_{interim}) *can be rephrased as follows for computing an optimal interim allocation rule* $\tilde{\mathbf{x}}$.

$$\begin{aligned} &\textit{maximize} && \sum_i R_i(\tilde{x}_i) \\ &\textit{subject to} && y(t_i, n) = \tilde{\mathbf{x}}(t_i), \qquad \forall t_i \in \mathrm{T}_N \\ &&& (y, z) \in \mathrm{S}. \end{aligned}$$

Furthermore, the resulting interim allocation rule can be implemented ex post by SSA using the the stochastic transition table defined by:[7]

$$\pi(t_{i'}, t_i) = \frac{z(t_{i'}, t_i)}{y(t_{i'}, i-1) f_i(t_i)}, \qquad \forall t_i \in \mathrm{T}_{\{1,\ldots,n\}}, \forall t_{i'} \in \mathrm{T}_{\{0,\ldots,i-1\}}.$$

Next, we present a few definitions and lemmas that are used in the proof of Theorem 4. Two transition tables π and π' are considered *equivalent* if their induced interim allocation rules for SSA are equal. Type t_i is called *degenerate* for π if in the execution of SSA the token is sometimes passed to type t_i but it is always taken away from t_i later, i.e., if $y(t_i, i) > 0$ but $y(t_i, n) = 0$. The stochastic transition table π is degenerate if there is a degenerate type. For π, type t_i is *augmentable* if there exists a π' (with a corresponding y') which is *equivalent* to π for all types expect t_i and has $y(t_i, n) > y'(t_i, n)$.[8]

Lemma 1. *For any stochastic transition table* π *there exists an equivalent* π' *that is non-degenerate.*

Lemma 2. *For any non-degenerate stochastic transition table* π*, any non-augmentable type* t_i *always wins against any augmentable type* $t_{i'}$*. I.e.,*

- *if* $i' < i$ *and* $t_{i'}$ *has non-zero probability of holding the token then* $\pi(t_{i'}, t_i) = 1$*, i.e.,* t_i *always takes the token away from* $t_{i'}$*, and*

[7] If the denominator is zero, i.e., $y(t_i, i'-1) = 0$, we can set $\pi(t_i, t_{i'})$ to an arbitrary value in $[0, 1]$.

[8] We define t_0 to be augmentable unless the dummy agent never retains the token in which case all agents are non-augmentable (and for technical reasons we declare the dummy agent to be non-augmentable as well).

- *if $i < i'$ and t_i has non-zero probability of holding the token then $\pi(t_i, t_{i'}) = 0$, i.e., $t_{i'}$ never takes the token away from t_i.*

It is possible to view the token passing in stochastic sequential allocation as a network flow. From this perspective, the augmentable and non-augmentable types form a minimum-cut and Lemma 2 states that the token must eventually flow from the augmentable to non-augmentable types. We defer the proof of this lemma to §2.5 where the main difficulty in its proof is that the edges in the relevant flow problem have dynamic(non-constant) capacities.

Proof of Theorem 4. Any interim allocation rule that can be implemented by the SSA algorithm is obviously feasible, so we only need to prove the opposite direction. The proof is by contradiction, i.e., given an interim allocation rule $\tilde{\mathbf{x}}$ we show that if there is no $(y, z) \in \mathrm{S}$ such that $\tilde{\mathbf{x}}(\cdot) = y(\cdot, n)$, then $\tilde{\mathbf{x}}$ must be infeasible. Consider the following linear program for a given $\tilde{\mathbf{x}}$ (i.e., $\tilde{\mathbf{x}}$ is constant).

$$
\begin{aligned}
\text{maximize} \quad & \sum_{t_i \in \mathrm{T}_{\{1,\ldots,n\}}} y(t_i, n) \\
\text{subject to} \quad & y(t_i, n) \leq \tilde{\mathbf{x}}(t_i), \qquad \forall t_i \in \mathrm{T}_{\{1,\ldots,n\}} \\
& (y, z) \in \mathrm{S}.
\end{aligned}
$$

Let (y, z) be an optimal assignment of this LP. If the first set of inequalities are all tight (i.e., $\tilde{\mathbf{x}}(\cdot) = y(\cdot, n)$) then $\tilde{\mathbf{x}}$ can be implemented by the SSA, so by contradiction there must exists a type $\tau^* \in \mathrm{T}_N$ for which the inequality is not tight. Note that τ^* cannot be augmentable; otherwise, by the definition of augmentability, the objective of the LP could be improved. Partition T_N to augmentable types T_N^+ and non-augmentable types T_N^-. Note that T_N^- is non-empty because $\tau^* \in \mathrm{T}_N^-$. Without loss of generality, by Lemma 1 we may assume that (y, z) is non-degenerate.[9]

An agent wins if she holds the token at the end of the SSA algorithm. The ex ante probability that some agent with non-augmentable type wins is $\sum_{t_i \in \mathrm{T}_N^-} y(t_i, n)$. On the other hand, Lemma 2 implies that the first (in the order agents are considered by

[9] By Lemma 1, there exits an non-degenerate assignment with the same objective value.

SSA) agent with non-augmentable type will take the token from her predecessors and, while she may lose the token to another non-augmentable type, the token will not be relinquished to any augmentable type. Therefore, the probability that an agent with a non-augmentable type is the winner is exactly equal to the probability that at least one such agent exists, therefore

$$\mathbf{Pr}_{\mathbf{t}\sim\mathbf{f}}\left[\exists i : t_i \in \mathrm{T}_N^-\right] = \sum_{t_i \in \mathrm{T}_N^-} y(t_i, n) < \sum_{t_i \in \mathrm{T}_N^-} \widetilde{\mathbf{x}}(t_i).$$

The second inequality follows from the assumption above that τ^* satisfies $y(\tau^*, n) < \widetilde{\mathbf{x}}(\tau^*)$. We conclude that $\widetilde{\mathbf{x}}$ requires an agent with non-augmentable type to win more frequently than such an agent exists, which is a contradiction to interim feasibility of $\widetilde{\mathbf{x}}$. □

The contradiction that we derived in the proof of Theorem 4 yields a necessary and sufficient condition, as formally stated in the following theorem for feasibility of any given interim allocation rule.

Theorem 5. *If the space of feasible allocations, $\widehat{\mathbf{x}}$, is defined by single unit allocation constraints, an interim allocation rule $\widetilde{\mathbf{x}}$ is feasible iff*

$$\sum_{\tau \in S} \widetilde{\mathbf{x}}(\tau) \leq \mathbf{Pr}_{\mathbf{t}\sim\mathbf{f}}\left[\exists i : t_i \in S\right], \qquad \forall S \subseteq \mathrm{T}_N \tag{MRMB}$$

The necessity of condition (MRMB) was first discovered both by Maskin and Riley (1984) and independently by Matthews (1984) and its sufficiency was first proved by Border (1991, 2007). This condition implies that the space of feasible interim allocation rules, $\widetilde{\mathbf{X}}$, can be specified by $O(2^{|\mathrm{T}_N|})$ linear constraints on a $|\mathrm{T}_N|$-dimensional space. An important consequence of Theorem 4 is that $\widetilde{\mathbf{X}}$ can equivalently be formulated by only $O(|\mathrm{T}_N|^2)$ variables and $O(|\mathrm{T}_N|^2)$ linear constraints as a projection of S, therefore any optimization problem over $\widetilde{\mathbf{X}}$ can equivalently be solved over S.

2.4.2 General Allocation Constraints

In this section we consider environments where the set of feasible allocations $\mathbf{X}$ is arbitrary, however we assume blackbox access to an algorithm for optimizing linear objectives over $\mathbf{X}$. We present computationally tractable algorithms both for optimization over the space of interim allocation rules, and for ex post implementation of interim allocation rules; however most of the details (including the actual algorithms and the proofs) are deferred to chapter 3.

Flat allocation vector. A *"flat allocation vector"* $\mathbf{x} \in \mathbb{R}^{\mathrm{T}_N \times [m]}$ specifies the allocation for each type of each agent, i.e., $\mathbf{x}(\tau)$ is the allocation of type $\tau \in \mathrm{T}_N$. Conditioned on type profile $\mathbf{t} \in \mathbf{T}$ being reported, the space of feasible flat allocation vectors is denoted by $\mathbf{X}^{\mathbf{t}}$ which is defined as

$$\mathbf{X}^{\mathbf{t}} = \left\{ \mathbf{x} \in \mathbb{R}^{\mathrm{T}_N \times [m]} \middle| (\mathbf{x}(t_1), \ldots, \mathbf{x}(t_n)) \in \mathbf{X} \text{ and } \mathbf{x}(\tau) = 0, \forall \tau \in \mathrm{T}_N \setminus \mathbf{t} \right\}$$

where $\mathrm{T}_N \setminus \mathbf{t}$ denotes the set of all types minus $t_1, \ldots, t_n$. Observe that $\mathbf{x}(\tau)$ is forced to be 0 for any type τ that is not among $\{t_1, \ldots, t_n\}$.

For the rest of this section we assume all allocation vectors are flat in order to simplify the exposition of our algorithms and proofs. For instance, $\mathbf{x} \sim \widehat{\mathbf{x}}(\mathbf{t})$ will denote an allocation vector drawn from $\widehat{\mathbf{x}}(\mathbf{t})$ but represented as a flat allocation vector. Observe that $\mathbf{X}^{\mathbf{t}}$ has the same dimension as $\widetilde{\mathbf{X}}$; furthermore, for any mechanism $M = (\widehat{\mathbf{x}}, \widehat{\mathbf{w}})$ the interim allocation rule is now given by

$$\widetilde{\mathbf{x}} = \mathbf{E}_{\mathbf{t} \sim \mathbf{f}} \left[\mathbf{E}_{\mathbf{x} \sim \widehat{\mathbf{x}}(\mathbf{t})} \left[\mathbf{x} \right] \right].$$

Observe that the implementation problem for an interim allocation rule $\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ is to select an allocation vector from each $\mathbf{X}^{\mathbf{t}}$ for each $\mathbf{t} \in \mathbf{T}$ such that in expectation $\widetilde{\mathbf{x}}$ is obtained.

chapter 3 studies the optimization and the implementation problem for $\widetilde{\mathbf{X}}$ in detail. All of the theorems presented throughout the rest of this section are directly implied by similar theorems in chapter 3 so the proofs are omitted here.

Theorem 6. *For any $\epsilon, \delta > 0$, the problems of computing and implementing an optimal interim allocation rule can be solved with an error of less than δ with probability at least $1-\epsilon$ in time polynomial in $\delta, \log\frac{1}{\epsilon}$ and the size of the problem, assuming black box access to an algorithm for optimizing linear objective over $\mathbf{X}$.*

We show that the interim allocation rules corresponding to the vertices of $\widetilde{\mathbf{X}}$ are easy to implement.

Theorem 7. *Consider any vertex $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ and let $\mathbf{w}$ be the normal vector to a strictly supporting hyperplane for $\tilde{\mathbf{x}}$ (i.e., $\mathbf{w} \cdot \tilde{\mathbf{x}}' - \mathbf{w} \cdot \tilde{\mathbf{x}} < 0$, for all $\tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}} \setminus \{\tilde{\mathbf{x}}\}$). $\tilde{\mathbf{x}}$ has a deterministic implementation as follows: under each type profile $\mathbf{t} \in \mathbf{T}$, select the allocation $\mathbf{x} = \arg\max_{\mathbf{x} \in \mathbf{X}^{\mathbf{t}}} \mathbf{w} \cdot \mathbf{x}$.*

In the above theorem, if we interpret $\mathbf{w}$ as a vector of virtual valuations, then $\tilde{\mathbf{x}}$ can be implemented by choosing for each $\mathbf{t} \in \mathbf{T}$ the allocation $\mathbf{x} \in \mathbf{X}^{\mathbf{t}}$ that maximizes the virtual surplus. Note that that by linearity of expectation every interim allocation rule which is not a vertex of $\widetilde{\mathbf{X}}$ can be implemented by randomizing over the implementations of the vertices of $\widetilde{\mathbf{X}}$.

2.4.3 Polymatroidal Allocation Constraints

For environments where for every $\mathbf{t} \in \mathbf{T}$ the convex hall of $\mathbf{X}^{\mathbf{t}}$ is a polymatroid, we present a polymatroidal characterization of $\widetilde{\mathbf{X}}$. Since our implementation algorithms only choose allocations that are vertices of $\mathbf{X}^{\mathbf{t}}$, we can replace each $\mathbf{X}^{\mathbf{t}}$ with its convex hall without loss of generality. We assume that a non-decreasing submodular function $\mathcal{F}^{\mathbf{t}}$ is given for each $\mathbf{t} \in \mathbf{T}$ such that $\mathbf{X}^{\mathbf{t}}$ is the polymatroid associated with $\mathcal{F}^{\mathbf{t}}$.

Definition 2. *Given a non-decreasing submodular function $\mathcal{F} : [d] \to \mathbb{R}_+$, the polymatroid associated with $\mathcal{F}$ is the convex polytope defined by*

$$\mathbf{P}_{\mathcal{F}} = \left\{\mathbf{x} \in \mathbb{R}_+^d \middle| \mathbf{x}(S) \le \mathcal{F}(S), \forall S \subseteq [d]\right\}$$

where $\mathbf{x}(S)$ is a shorthand notation for $\sum_{j \in S} \mathbf{x}(j)$.

Example. Suppose there is one indivisible item to be allocated to at most one of n agents; Let $\mathbf{t} \in \mathbf{T}$ be the type profile reported by the agents. Then it is easy to see that an allocation vector $\mathbf{x} \in \mathbb{R}^{\mathrm{T}_N \times [m]}$ is feasible if and only if it satisfies the following set of inequalities.

$$\mathbf{x}(S) \leq \min(|S \cap \mathbf{t}|, 1), \qquad \forall S \subseteq \mathrm{T}_N$$

The above set of inequalities ensures that the item can be allocated to at most one of the reported types and the allocations for all other types should be zero. Therefore $\mathbf{X}^{\mathbf{t}}$ is the polymatroid associated with the submodular function $\mathcal{F}^{\mathbf{t}}(S) = \min(|S \cap \mathbf{t}|, 1)$.

Polymatroidal characterization of $\tilde{\mathbf{x}}$. Recall that an interim allocation rule $\tilde{\mathbf{x}}$ is feasible (i.e., $\tilde{\mathbf{x}} \in \widetilde{\mathrm{X}}$) iff there exists an ex post allocation rule $\hat{\mathbf{x}}$ that implements it (i.e., $\tilde{\mathbf{x}} = \mathbf{E}_{\mathbf{t}\sim\mathbf{f}}[\mathbf{E}_{\mathsf{x}\sim\hat{\mathbf{x}}(\mathbf{t})}[\mathsf{x}]]$). Note that every allocation vector in the support of $\hat{\mathbf{x}}(\mathbf{t})$ must be in $\mathbf{X}^{\mathbf{t}}$ therefore

$$\mathsf{x}(S) \leq \mathcal{F}^{\mathbf{t}}(S), \qquad \forall \mathbf{t} \in \mathrm{T}, \forall \mathsf{x} \in \hat{\mathbf{x}}(\mathbf{t}), \forall S \subseteq \mathrm{T}_N \times [m] \,.$$

Taking the expectation over $\mathsf{x} \sim \hat{\mathbf{x}}(\mathbf{t})$ and then over $\mathbf{t} \sim \mathbf{f}$ we get

$$\mathbf{E}_{\mathbf{t}\sim\mathbf{f}}\left[\mathbf{E}_{\mathsf{x}\sim\hat{\mathbf{x}}(\mathbf{t})}\left[\mathsf{x}(S)\right]\right] \leq \mathbf{E}_{\mathbf{t}\sim\mathbf{f}}\left[\mathcal{F}^{\mathbf{t}}(S)\right], \qquad \forall S \subseteq \mathrm{T}_N \times [m] \,.$$

Observe that the left hand side of the above inequality is exactly $\tilde{\mathbf{x}}$ and the right hand side is itself a non-decreasing submodular function of S.[10]. Consequently, the above inequality implies that $\tilde{\mathbf{x}} \in \mathbf{P}_{\widetilde{\mathcal{F}}}$ where $\widetilde{\mathcal{F}}$ is defined as follows.

Definition 3. *$\widetilde{\mathcal{F}} : \mathrm{T}_N \times [m] \to \mathbb{R}_+$ is a non-decreasing submodular function given by*

$$\widetilde{\mathcal{F}}(S) = \mathbf{E}_{\mathbf{t}\sim\mathbf{f}}\left[\mathcal{F}^{\mathbf{t}}(S)\right], \qquad S \subseteq \mathrm{T}_N \times [m] \,.$$

The above discussion shows that if $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ then $\tilde{\mathbf{x}} \in \mathbf{P}_{\widetilde{\mathcal{F}}}$ which implies $\widetilde{\mathbf{X}} \subseteq \mathbf{P}_{\widetilde{\mathcal{F}}}$; we will show that the two are in fact equal which implies that $\widetilde{\mathbf{X}}$ is itself a polymatroid.

[10]Because it is the expectation of (i.e., weighted sum of) non-decreasing submodular functions

Theorem 8. *$\widetilde{\mathbf{X}}$ is the polymatroid associated with $\widetilde{\mathcal{F}}$ (i.e., $\widetilde{\mathbf{X}} = \mathbf{P}_{\widetilde{\mathcal{F}}}$).*

The polymatroidal characterization of $\tilde{\mathbf{x}}$ is important both because it relates to qualitative characteristics of optimal mechanisms and because it leads to more efficient algorithms for optimization and implementation of interim allocation rules.

Theorem 9. *The problems of computing and implementing an optimal interim allocation rule can be reduced in polynomial time to the problem of computing $\widetilde{\mathcal{F}}$.*

We also show that the interim allocation rules corresponding to the vertices of $\widetilde{\mathbf{X}}$ have particularly easy implementations.

Proposition 1 (Polymatroid Vertices)**.** *Consider an arbitrary non-decreasing submodular function $\mathcal{F} : [d] \to \mathbb{R}_+$ and the associated polymatroid $\mathbf{P}_{\mathcal{F}}$. Every ordered subset $\pi = (\pi_1, \pi_2, \ldots) \subseteq [d]$ identifies a vertex $\mathbf{x} \in \mathbf{P}_{\mathcal{F}}$ whose value at each coordinate $j \in [d]$ is given by*

$$\mathbf{x}(j) = \begin{cases} \mathcal{F}(\{\pi_1, \ldots, \pi_r\}) - \mathcal{F}(\{\pi_1, \ldots, \pi_{r-1}\}) & \text{if } j = \pi_r \text{ for some } r \in [|\pi|] \\ 0 & \text{if } j \notin \pi \end{cases}.$$

Furthermore, every vertex of $\mathbf{P}_{\mathcal{F}}$ is identified by one or more such ordered subsets.

Theorem 10. *Consider an arbitrary vertex $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ and a corresponding ordered subset $\pi \subseteq \mathrm{T}_N \times [m]$. Then $\tilde{\mathbf{x}}$ can be implemented by choosing, for each $\mathbf{t} \in \mathbf{T}$, the allocation corresponding to the vertex of $\mathbf{X}^{\mathbf{t}}$ associated with π.*

Suppose there are m indivisible items so that each coordinate of the allocation vector corresponds to a pair of type and item. Then the previous theorem implies that for every interim allocation rule $\tilde{\mathbf{x}}$ which is a vertex of $\widetilde{\mathbf{X}}$ there exists an ordering π over a subset of pairs of types and items, such that $\tilde{\mathbf{x}}$ can be implemented by greedily allocating items to agents (types) according to the ordering specified by π subject to the constraint imposed by $\mathbf{X}^{\mathbf{t}}$.

k-unit allocation constraint. Suppose there are k units of an indivisible item and agents are unit demand. Therefore an allocation vector is feasible iff the the total number of allocated items is no more than k, each one of the reported types get at most one unit, and types that are not reported have a zero allocation. It is easy to see that an allocation vector $\mathbf{x}$ is feasible, conditioned on type profile $\mathbf{t}$ being reported, iff

$$\mathbf{x}(S) \leq \min(|S \cap \mathbf{t}|, k), \qquad \forall S \subseteq \mathrm{T}_N.$$

Therefore $\mathbf{X}^{\mathbf{t}} = \mathbf{P}_{\mathcal{F}^{\mathbf{t}}}$ where $\mathcal{F}^{\mathbf{t}}(S) = \min(|S \cap \mathbf{t}|, k)$. Then by Theorem 8 we can argue that $\tilde{\mathbf{x}}$ is a feasible interim allocation rule iff

$$\tilde{\mathbf{x}}(S) \leq \mathbf{E}_{\mathbf{t} \sim \mathbf{f}}\left[\min(|S \cap \mathbf{t}|, k)\right], \qquad \forall S \subseteq \mathrm{T}_N.$$

Observe that the result of Border (1991, 2007) can be obtained from the above inequalities by setting $k = 1$.

Lemma 3. *$\widetilde{\mathcal{F}}(S) = \mathbf{E}_{\mathbf{t} \sim \mathbf{f}}[\min(|S \cap \mathbf{t}|, k)]$ can be exactly computed in time $O((n + |S|) \cdot k)$ for any $S \in \mathrm{T}_N$ using dynamic programming.*

Proof. It can be computed using the following dynamic program in which G^i_j denotes the probability of the event that $\min(|\mathbf{t} \cap S \cap \mathrm{T}_{\{1,\ldots,i\}}|, k) = j$.

$$\mathbf{E}_{\mathbf{t} \sim \mathbf{f}}\left[\min(|S \cap \mathbf{t}|, k)\right] = \sum_{j=1}^{k} j \cdot G^n_j$$

$$G^i_j = \begin{cases} G^{i-1}_k + (\sum_{t_i \in S \cap \mathrm{T}_i} f_i(t_i)) \cdot G^{i-1}_{k-1} & 1 \leq i \leq n, j = k \\ G^{i-1}_j + (\sum_{t_i \in S \cap \mathrm{T}_i} f_i(t_i)) \cdot (G^{i-1}_{j-1} - G^{i-1}_j) & 1 \leq i \leq n, 0 \leq j < k \\ 1 & i = 0, j = 0 \\ 0 & \text{otherwise} \end{cases}$$

□

The above lemma can be combined with Theorem 9 to yields the following theorem.

Theorem 11. *For an environment with k-unit allocation constraint, the problems of computing and implementing an optimal interim allocation rule can be solved exactly, in polynomial time.*

Matroid allocation constraint. Suppose a subset of agents are to be selected to receive some service. Suppose the set of all feasible subsets of agents is given by a matroid $\mathcal{M} = (N, \mathcal{I})$, i.e., a subset $A \subseteq N$ can be simultaneously allocated to (e.g., served) iff $A \in \mathcal{I}$. Let $r_{\mathcal{M}}$ denote the rank function of $\mathcal{M}$. It is easy to see that an allocation vector $\mathbf{x}$ is feasible, conditioned on type profile $\mathbf{t}$ being reported, iff

$$\mathbf{x}(S) \leq r_{\mathcal{M}}(\{i \in N | t_i \in S\}), \qquad \forall S \subseteq \mathrm{T}_N.$$

Therefore $\mathbf{X}^{\mathbf{t}} = \mathbf{P}_{\mathcal{F}^{\mathbf{t}}}$ where $\mathcal{F}^{\mathbf{t}}(S) = r_{\mathcal{M}}(\{i \in N | t_i \in S\})$. Then by Theorem 8 we can argue that $\tilde{\mathbf{x}}$ is a feasible interim allocation rule iff

$$\tilde{\mathbf{x}}(S) \leq \mathbf{E}_{\mathbf{t} \sim \mathbf{f}}\left[r_{\mathcal{M}}(\{i \in N | t_i \in S\})\right], \qquad \forall S \subseteq \mathrm{T}_N.$$

Observe that k-unit allocation constraint is a special case of matroid allocation constraint in which the matroid is k-uniform. In particular the characterization of interim feasibility for k-unit allocation constraint can be obtained from the above inequalities by setting $r_{\mathcal{M}}(A) = \min(|A|, k)$.

2.5 Omitted Proofs

We first describe a network flow formulation of S, which is used to prove Lemma 1 and Lemma 2.

A network flow formulation of S. We construct a network in which every feasible flow corresponds to some $(y, z) \in$ S. The network (see Figure 2.1) has a source node $\langle$Src$\rangle$, a sink node $\langle$Snk$\rangle$, and $n - i + 1$ nodes for every $t_i \in \mathrm{T}_N$ labeled as $\langle t_i, i\rangle, \ldots, \langle t_i, n\rangle$ where each node $\langle t_{i'}, i\rangle$ corresponds to the type $t_{i'}$ at the time SSA algorithm is visiting agent i. For each $t_{i'} \in \mathrm{T}_N$ and for each $i \in \{i', \ldots, n-1\}$ there

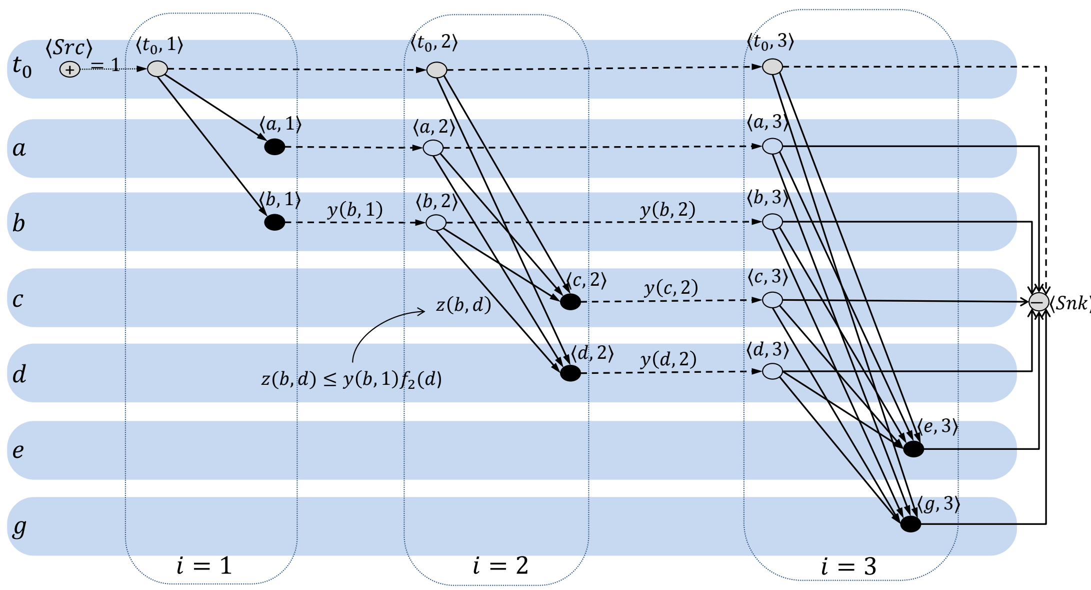


Figure 2.1: The flow network corresponding to the SSA algorithm 1.
. In this instance, there are three agents with type spaces $\mathrm{T}_1 = \{a, b\}$, $\mathrm{T}_2 = \{c, d\}$, and $\mathrm{T}_3 = \{e, g\}$. All nodes in the same row correspond to the same type. The diagonal edges have dynamic capacity constraints while all other edges have no capacity constraints. The flow going from $\langle t_{i'}, i\rangle$ to $\langle t_i, i\rangle$ corresponds to the ex-ante probability of t_i taking the token away from $t_{i'}$. The flow going from $\langle t_{i'}, i\rangle$ to $\langle t_{i'}, i+1\rangle$ corresponds to the ex-ante probability of $t_{i'}$ still holding the token after agent i is visited.

is an edge $(\langle t_{i'}, i\rangle, \langle t_{i'}, i+1\rangle)$ with infinite capacity whose flow is equal to $y(t_{i'}, i)$; we refer to these edges as *"horizontal edges"*. For every $t_{i'}$ and every t_i where $i' < i$ there is an edge $(\langle t_{i'}, i\rangle, \langle t_i, i\rangle)$ whose flow is equal to $z(t_{i'}, t_i)$ and whose capacity is equal to the total amount of flow that enters $\langle t_{i'}, i\rangle$ multiplied by $f_i(t_i)$, i.e., it has a dynamic capacity which is equal to $y(t_{i'}, i-1)f_i(t_i)$; we refer to these edges as *"diagonal edges"*. There is an edge $(\langle \text{SRC}\rangle, t_0)$ through which the source node pushes exactly one unit of flow. Finally, for every $t_i \in \mathrm{T}_N$, there is an edge $(\langle t_i, n\rangle, \langle \text{SNK}\rangle)$ with unlimited capacity whose flow is equal to $y(t_i, n)$. To simplify the proofs we sometimes use $\langle t_0, 0\rangle$ as an alias for the source node $\langle \text{SRC}\rangle$ and $\langle t_i, n+1\rangle$ for $i \in [n]$ as aliases for the sink node $\langle \text{SNK}\rangle$. The network always has a feasible flow because all the flow can be routed along the path $\langle \text{SRC}\rangle, \langle t_0, 1\rangle, \ldots, \langle t_0, n\rangle, \langle \text{SNK}\rangle$.

We define the *residual capacity* between two types $t_{i'}, t_i \in \mathrm{T}_N$ with respect to a given $(y, z) \in \mathrm{S}$ as follows.

$$\text{RESCAP}_{y,z}(t_{i'}, t_i) = \begin{cases} y(t_{i'}, i-1)f_i(t_i) - z(t_{i'}, t_i) & i > i' \\ z(t_i, t_{i'}) & i < i' \\ 0 & \text{otherwise} \end{cases} \qquad (\text{RESCAP})$$

Due to dynamic capacity constraints, it is not possible to augment a flow along a path with positive residual capacity by simply changing the amount of the flow along the edges of the path, because reducing the total flow entering a node also decreases the capacity of the diagonal edges leaving that node, which could potentially violate their capacity constraints. Therefore, we introduce an operator $\text{REROUTE}(t_{i'}, t_i, \rho)$ (algorithm 1 and Figure 2.2) which modifies an existing $(y, z) \in \mathrm{S}$, while maintaining its feasibility, to transfer a ρ-fraction of $y(t_i, n)$ to $y(t_{i'}, n)$ by changing the flow along the cycle

$$\langle \text{SNK}\rangle, \langle t_{i'}, n\rangle, \langle t_{i'}, n-1\rangle, \ldots, \langle t_{i'}, \max(i', i)\rangle, \langle t_i, \max(i', i)\rangle, \ldots, \langle t_i, n-1\rangle, \langle t_i, n\rangle, \langle \text{SNK}\rangle$$

and adjusting the flow of the the diagonal edges which leave this cycle. More precisely, $\text{REROUTE}(t_{i'}, t_i, \rho)$ takes out a ρ-fraction of the flow going through the subtree rooted at $\langle t_{i'}, \max(i', i)\rangle$ [11] and reassigns it to the subtree rooted at $\langle t_i, \max(i', i)\rangle$ (see Figure 2.2).

Algorithm 1 $\textsc{Reroute}(t_{i'}, t_i, \rho)$.

Input: An existing $(y, z) \in \mathrm{S}$ given implicitly, a source type $t_{i'} \in \mathrm{T}_N$, a destination type $t_i \in \mathrm{T}_N$ where $i' \neq i$, and a fraction $\rho \in [0, 1]$.
Output: Modify (y, z) to transfer a ρ-fraction of $y(t_{i'}, n)$ to $y(t_i, n)$ while ensuring that the modified assignment is still in S.

if $i' < i$ **then**
  Increase $z(t_{i'}, t_i)$ by $\rho \cdot y(t_{i'}, i)$.
else
  Decrease $z(t_i, t_{i'})$ by $\rho \cdot y(t_{i'}, i')$.
end if
for $i'' = \max(i', i)$ to n **do**
  Increase $y(t_i, i'')$ by $\rho \cdot y(t_{i'}, i'')$.
  Decrease $y(t_{i'}, i'')$ by $\rho \cdot y(t_{i'}, i'')$.
end for
for $t_{i''} \in \mathrm{T}_{\{\max(i',i)+1,\ldots,n\}}$ **do**
  Increase $z(t_i, t_{i''})$ by $\rho \cdot z(t_{i'}, t_{i''})$.
  Decrease $z(t_{i'}, t_{i''})$ by $\rho \cdot z(t_{i'}, t_{i''})$.
end for

Proof of Lemma 1. For any given $(y, z) \in \mathrm{S}$ we show that it is always possible to modify y and z to obtain a non-degenerate feasible assignment with the same induced interim allocation probabilities (i.e., the same $y(\cdot, n)$). Let d denote the number of degenerate types with respect to (y, z), i.e., define

$$d = \#\left\{t_i \in \mathrm{T}_{\{1,\ldots,n\}} \middle| y(t_i, n) = 0, y(t_i, i) > 0\right\}$$

The proof is by induction on d. The base case is $d = 0$ which is trivial. We prove the claim for $d > 0$ by modifying y and z, reducing the number of degenerate types to $d - 1$, and then applying the induction hypothesis. Let t_i be a degenerate type. For each $t_{i'} \in \mathrm{T}_{\{0,\ldots,i-1\}}$, we apply the operator $\textsc{Reroute}(t_i, t_{i'}, \frac{z(t_{i'},t_i)}{y(t_i,i)})$ unless $y(t_i, i)$ has

[11] This subtree consists of the path $\langle t_{i'}, \max(i', i)\rangle, \ldots, \langle t_{i'}, n\rangle, \langle \textsc{Snk}\rangle$ and all the diagonal edges leaving this path.

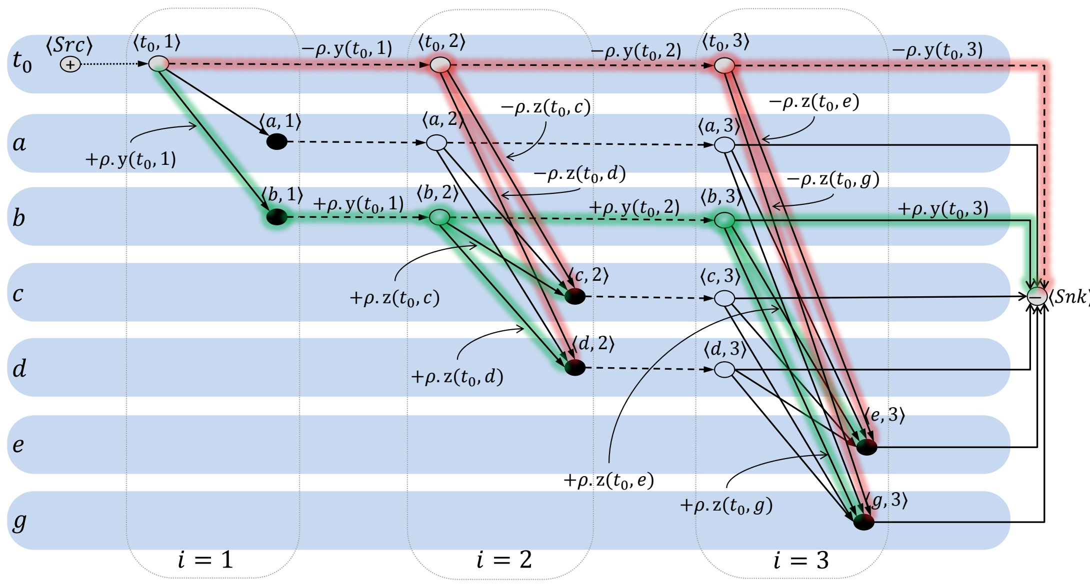


Figure 2.2: Changes made by applying $\textsc{Reroute}(t_0, b, \rho)$.
. A ρ-fraction of the red subtree rooted at t_0 is take out and reassigned to the green subtree rooted at b. The exact amount of change is indicated for each green and each red edge. The flow along all other edges stay intact. The operator has the effect of reassigning ρ-fraction of ex-ante probability of allocation for type t_0 to type b.

already reached 0. Applying this operator to each type $t_{i'}$ eliminates the flow from $\langle t_{i'}, i\rangle$ to $\langle t_i, i\rangle$, so eventually $y(t_i, i)$ reaches 0 and t_i is no longer degenerate and also no new degenerate type is introduced, so the number of degenerate types is reduced to $d-1$. It is also easy to see that $y(t_{i'}, n)$ is not modified because $y(t_i, n) = 0$. That completes the proof. □

Proof of Lemma 2. To prove the lemma it is enough to show that for any augmentable type $t_{i'}$ and any non-augmentable type t_i, $\textsc{ResCap}_{y,z}(t_{i'}, t_i) = 0$ which is equivalent to the statement of the lemma (the equivalence follows from the definition of $\textsc{ResCap}$ and equation (π)). The proof is by contradiction. Suppose $t_{i'}$ is augmentable and $\textsc{ResCap}_{y,z}(t_{i'}, t_i) = \delta$ for some positive δ; we show that t_i is also augmentable. Since $t_{i'}$ is augmentable, there exists a $(y', z') \in \mathrm{S}$ such that $y'(\tau, n) = y(\tau, n)$ for all

$\tau \in \mathrm{T}_N \setminus \{t_0, t_{i'}\}$ and $y'(t_{i'}, n) - y(t_{i'}, n) = \epsilon > 0$. Define

$$(y'', z'') = (1 - \beta) \cdot (y, z) + \beta \cdot (y', z')$$

where $\beta \in [0, 1]$ is a parameter that we specify later. Note that in (y'', z''), $t_{i'}$ is augmented by $\beta\epsilon$, and $\textsc{ResCap}_{y'',z''}(t_{i'}, t_i) \geq (1 - \beta)\delta$, and $(y'', z'') \in \mathrm{S}$ because it is a convex combination of (y, z) and (y', z'). Consider applying $\textsc{Reroute}(t_{i'}, t_i, \rho)$ to (y'', z'') for some parameter $\rho \in [0, 1]$. The idea is to choose β and ρ such that the exact amount, by which $t_{i'}$ was augmented, gets reassigned to t_i, by applying $\textsc{Reroute}(t_{i'}, t_i, \rho)$; so that eventually t_i is augmented while every other type (except t_0) has the same allocation probabilities as they originally had in (y, z). It is easy to verify that by setting

$$\beta = \frac{y(t_{i'}, n)\delta}{2} \qquad\qquad \rho = \frac{\epsilon\delta}{2 + \epsilon\delta}$$

we get a feasible assignment in which the allocation probability of t_i is augmented by $\beta\epsilon$ while every other type (except t_0) has the same allocation probabilities as in (y, z). We still need to show that $\beta > 0$. The proof is again by contradiction. Suppose $\beta = 0$, so it must be $y(t_{i'}, n) = 0$, which would imply that $t_{i'}$ is a degenerate type because $y(t_{i'}, i') > 0$ (because $\textsc{ResCap}_{y,z}(t_{i'}, t_i) > 0$), however (y, z) is a non-degenerate assignment by the hypothesis of the lemma, which is a contradiction. That completes the proof. $\square$

Chapter 3

Optimizing Over A Stochastic Polytope

We consider the following abstract stochastic optimization problem: we would like to select (possibly at random) a point $\mathsf{x} \in \mathbb{R}^d$ so as to maximize the the value of a concave objective function $\textsc{Obj}(\mathbf{E}[\mathsf{x}])$, i.e., the objective function depends only on the expected value of x; the set of feasible choices for x depends on a random variable $\mathbf{t}$ with known distribution $\mathbf{D}$; for each realization of $\mathbf{t}$, the set of feasible choices for x, denoted by $\mathbf{X}^{\mathbf{t}}$, is a bounded subset of $\mathbb{R}^d$. Formally, the problem is to identify a *"selection policy"*, say $\widehat{\mathbf{x}}$, which maps each realization of $\mathbf{t}$ to a distribution over $\mathbf{X}^{\mathbf{t}}$, with the objective of maximizing $\textsc{Obj}(\mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathbf{E}_{\mathsf{x}\sim\widehat{\mathbf{x}}(\mathbf{t})}[\mathsf{x}]])$. Note that it is not required to explicitly compute the optimal $\widehat{\mathbf{x}}$; it is enough that a point x can be efficiently sampled from $\widehat{\mathbf{x}}(\mathbf{t})$, for any given $\mathbf{t}$. We present computationally efficient algorithms (i.e., polynomial running time) for this problem, assuming we can efficiently sample $\mathbf{t}$ from $\mathbf{D}$ and efficiently optimize any linear objective over $\mathbf{X}^{\mathbf{t}}$. In particular, the running times of our proposed algorithms do not depend on the size of the support of $\mathbf{D}$.

Definition 4 ($\widetilde{\mathbf{X}}$)**.** *A point* $\widetilde{\mathbf{x}} \in \mathbb{R}^d$ *is* "implementable" *iff there exists a feasible selection policy* $\widehat{\mathbf{x}}$ *such that* $\widetilde{\mathbf{x}} = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathbf{E}_{\mathsf{x}\sim\widehat{\mathbf{x}}(\mathbf{t})}[\mathsf{x}]]$*; subsequently, we say* $\widehat{\mathbf{x}}$ "implements" $\widetilde{\mathbf{x}}$*. The set of all implementable points is denoted by . Conceptually, one can define* $\widetilde{\mathbf{X}} = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathbf{X}^{\mathbf{t}}]$.

Proposition 2. $\widetilde{\mathbf{X}}$ *is convex.*

Proof. Consider any $\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}$ and $\beta \in [0,1]$. We show that $\widetilde{\mathbf{x}}'' = \beta\widetilde{\mathbf{x}} + (1-\beta)\widetilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}$

which proves that $\widetilde{\mathbf{X}}$ is convex. Let $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{x}}'$ be selection policies implementing $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{x}}'$ respectively. Let $\widehat{\mathbf{x}}''$ be the selection policy which with probability β selects according to $\widehat{\mathbf{x}}$ and otherwise selects according to $\widehat{\mathbf{x}}'$. It is easy to see that $\widehat{\mathbf{x}}''$ implements $\widetilde{\mathbf{x}}''$. □

Consider the following convex program.

$$\begin{aligned} \text{maximize} \quad & \textsc{Obj}(\widetilde{\mathbf{x}}) \\ \text{subject to} \quad & \widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}} \end{aligned} \tag{$\widetilde{OPT}$}$$

The problem can be broken down to two parts:

- **Optimization problem.** Compute an optimal assignment for program ($\widetilde{OPT}$).
- **Implementation problem.** Given a point $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ and given $\mathbf{X}^{\mathbf{t}}$ (specified by $\mathbf{t}$), identify a selection polity $\widehat{\mathbf{x}}$ that implements $\tilde{\mathbf{x}}$, and draw at random $\mathbf{x} \sim \widehat{\mathbf{x}}(\mathbf{t})$.

Main result. The main result of this chapter can be summarized in the following informal theorem.

Theorem 12. *For any $\epsilon, \delta > 0$, both the optimization problem and the implementation problem can be solved within an absolute error less than δ (in terms of Euclidian distance) with probability at least $1 - \epsilon$ in time polynomial in $\delta, \log \frac{1}{\epsilon}$ and the size of the problem, assuming black box access to polynomial time algorithms for sampling $\mathbf{t} \sim \mathbf{D}$ and optimizing linear objectives over $\mathbf{X}^{\mathbf{t}}$.*

Road map. Section §3.1 presents a characterization of the vertices of $\widetilde{\mathbf{X}}$ which leads to a simple algorithm (based on sampling) for optimizing a linear objective over $\widetilde{\mathbf{X}}$; this characterization also yields a simple deterministic implementation for the vertices of $\widetilde{\mathbf{X}}$. Furthermore, any point in $\widetilde{\mathbf{X}}$ can be decomposed as a convex combination of vertices of $\widetilde{\mathbf{X}}$ and therefore can be implemented by randomizing over the implementations of those vertices. Section §3.2 presents a separation oracle for $\widetilde{\mathbf{X}}$, which can be used with the Ellipsoid method to optimize a concave objective over $\widetilde{\mathbf{X}}$.

The separation oracle itself makes use of the Ellipsoid method to reduce the separation problem to the problem of optimizing a linear objective over $\widetilde{\mathbf{X}}$. Section §3.3 presents a polymatroidal characterization of $\widetilde{\mathbf{X}}$ assuming each $\mathbf{X^t}$ is itself a polymatroid. This polymatroidal characterization yields more efficient algorithms for optimization and implementation over $\widetilde{\mathbf{X}}$.

3.1 Preliminaries

This section presents basic separating/supporting hyperplane theorems from convex optimization and provides a simple characterization of the vertices of $\widetilde{\mathbf{X}}$ which yields a simple algorithm for optimizing linear functions over $\widetilde{\mathbf{X}}$ and also yields a simple determinist implementation for the vertices of $\widetilde{\mathbf{X}}$.

Sampling and optimization over $\mathbf{X^t}$. We assume blackbox access to algorithms with polynomial running time for:

- Sampling $\mathbf{t} \sim \mathbf{D}$.
- Computing $\arg^\dagger\max_{\mathbf{x}\in\mathbf{X^t}} \mathbf{w}\cdot\mathbf{x}$ for any given $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{X^t} \subset \mathbb{R}^d$ (specified by $\mathbf{t}$), where $\arg^\dagger\max$ breaks ties in lexicographical order [1].

Separating/Supporting hyperplanes. $\widetilde{\mathbf{X}}$ is a closed convex polytope, therefore for any point outside of $\widetilde{\mathbf{X}}$ there exists a separating hyperplane (i.e., a hyperplane that separates that point from the polytope); also for any vertex of $\widetilde{\mathbf{X}}$ there exists a strict supporting hyperplane (i.e., a hyperplane through that vertex such that the rest of $\widetilde{\mathbf{X}}$ lies strictly on one side of the hyperplane).

Proposition 3 (Separating Hyperplane)**.** *Consider any $\tilde{\mathbf{x}} \in \mathbb{R}^d$; $\tilde{\mathbf{x}}$ is not in $\widetilde{\mathbf{X}}$ iff there exists $\mathbf{w} \in \mathbb{R}^d$ such that*

$$\mathbf{w}\cdot\tilde{\mathbf{x}}' - \mathbf{w}\cdot\tilde{\mathbf{x}} < 0, \qquad \textit{for all} \qquad \tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}.$$

[1] The lexicographical tie breaking is not strictly necessary, but simplifies the exposition.

Proposition 4 (Strict Supporting Hyperplane)**.** *Consider any* $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$*;* $\tilde{\mathbf{x}}$ *is a vertex of* $\widetilde{\mathbf{X}}$ *iff there exists* $\mathbf{w} \in \mathbb{R}^d$ *such that*

$$\mathbf{w} \cdot \tilde{\mathbf{x}}' - \mathbf{w} \cdot \tilde{\mathbf{x}} < 0, \qquad\qquad \textit{for all} \qquad \tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}} \setminus \{\tilde{\mathbf{x}}\}$$

Optimization of a linear objective over $\widetilde{\mathbf{X}}$. Suppose the objective function $\textsc{Obj}(\tilde{\mathbf{x}})$ is a linear function of $\tilde{\mathbf{x}}$, i.e., $\textsc{Obj}(\tilde{\mathbf{x}}) = \mathbf{w} \cdot \tilde{\mathbf{x}}$ for some $\mathbf{w} \in \mathbb{R}^d$; then an optimal assignment for program $(\widetilde{OPT})$ is given by $\arg^{\dagger}\max_{\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \tilde{\mathbf{x}}$.

Lemma 4. *For any* $\mathbf{w} \in \mathbb{R}^d$*,*

$$\arg^{\dagger}\max_{\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \tilde{\mathbf{x}} = \mathbf{E}_{\mathbf{t} \sim \mathbf{D}} \left[\arg^{\dagger}\max_{\mathbf{x} \in \mathbf{X}^{\mathbf{t}}} \mathbf{w} \cdot \mathbf{x}\right].$$

Proof. Let $\tilde{\mathbf{x}} = \arg^{\dagger}\max_{\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \tilde{\mathbf{x}}$ and let $\hat{\mathbf{x}}$ be a selection policy that implements $\tilde{\mathbf{x}}$. The proof is by contradiction. Suppose the equation does not hold, so there must be some $\mathbf{t}^* \in \mathbf{D}$ for which $\hat{\mathbf{x}}(\mathbf{t}^*) \neq \mathbf{x}^* = \arg^{\dagger}\max_{\mathbf{x} \in \mathbf{X}^{\mathbf{t}^*}} \mathbf{x} \cdot \mathbf{x}$ which implies that either $\mathbf{w} \cdot \hat{\mathbf{x}}(\mathbf{t}^*) < \mathbf{w} \cdot \mathbf{x}^*$ or $\mathbf{w} \cdot \hat{\mathbf{x}}(\mathbf{t}^*) = \mathbf{w} \cdot \mathbf{x}^*$ and $\mathbf{x}^*$ comes before $\hat{\mathbf{x}}(\mathbf{t}^*)$ in lexicographical order; in either case, we can obtain a new selection policy $\hat{\mathbf{x}}'$ which is the same as $\hat{\mathbf{x}}$ everywhere except that $\hat{\mathbf{x}}'(\mathbf{t}^*) = \mathbf{x}^*$; let $\tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}$ be the point implemented by $\hat{\mathbf{x}}'$; it is easy to see that either $\mathbf{w} \cdot \tilde{\mathbf{x}} < \mathbf{w} \cdot \tilde{\mathbf{x}}'$ or $\mathbf{w} \cdot \tilde{\mathbf{x}} = \mathbf{w} \cdot \tilde{\mathbf{x}}'$ and $\tilde{\mathbf{x}}'$ comes before $\tilde{\mathbf{x}}$ in lexicographical order; in either case we have a contradiction which completes the proof [2]. □

The previous lemma implies that the optimal along the direction of $\mathbf{w}$ can be achieved in expectation by optimizing along the direction of $\mathbf{w}$ over every realization of $\mathbf{X}^{\mathbf{t}}$; consequently, $\arg^{\dagger}\max_{\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \tilde{\mathbf{x}}$ can be approximated to any degree of accuracy

[2]The proof implicitly assumes that every $\mathbf{t} \in \mathbf{D}$ has a probability mass of non-zero measure; this assumption is without loss of generality if $\mathbf{t}$ is a discrete random variable. Also, it is easy to extend the proof to the case of $\mathbf{t}$ being a continuous random variable.

with high probability by sampling and taking the average.

Definition 5. ApxSolve($\mathbf{w}$). *Given* $\mathbf{w} \in \mathbb{R}^d$ *and an implicit parameter* $\sigma \in \mathbb{N}_0$*, compute and return an approximate solution for* $\arg^\dagger\max_{\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \widetilde{\mathbf{x}}$*:*

1. *For each* $\ell \in [\sigma]$*: sample* $\mathbf{t}^\ell \sim \mathbf{D}$ *and compute* $\mathbf{x}^\ell = \arg^\dagger\max_{\mathbf{x} \in \mathbf{X}^{\mathbf{t}^\ell}} \mathbf{w} \cdot \mathbf{x}$.
2. *Compute* $\widetilde{\mathbf{x}}^{APX} \leftarrow \left(\sum_{\ell=1}^{\sigma} \mathbf{x}^\ell\right)/\sigma$ *and return* $\widetilde{\mathbf{x}}^{APX}$.

Definition 6. η *is the length of the smallest hypercube which contains* $\mathbf{X}^{\mathbf{t}}$ *for all* $\mathbf{t} \in \mathbf{D}$.

Theorem 13. *For any* $\epsilon, \delta > 0$ *there exists* $\sigma \in \mathbb{N}_0$ *where* $\sigma = O(\frac{1}{\delta} d\eta\sqrt{\log\frac{d}{\epsilon}})$ *such that for any* $\mathbf{w} \in \mathbb{R}^d$*, if* $\widetilde{\mathbf{x}}^{OPT} = \arg^\dagger\max_{\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \widetilde{\mathbf{x}}$ *and* $\widetilde{\mathbf{x}}^{APX} = $ ApxSolve($\mathbf{w}$)*, then* $\|\widetilde{\mathbf{x}}^{APX} - \widetilde{\mathbf{x}}^{OPT}\| < \delta$ *with probability at least* $1 - \epsilon$.

Proof. Recall that $\widetilde{\mathbf{x}}^{\text{OPT}} = \mathbf{E}_{\mathbf{t} \sim \mathbf{D}}[\arg^\dagger\max_{\mathbf{x} \in \mathbf{X}^{\mathbf{t}}} \mathbf{w} \cdot \mathbf{x}]$ by Lemma 4, therefore

$$\begin{aligned}
\mathbf{Pr}\left[\|\widetilde{\mathbf{x}}^{\text{APX}} - \widetilde{\mathbf{x}}^{\text{OPT}}\| < \delta\right] &\geq \mathbf{Pr}\left[|\widetilde{\mathbf{x}}^{\text{APX}}(j) - \widetilde{\mathbf{x}}^{\text{OPT}}(j)| < \delta/\sqrt{d}, \forall j \in [d]\right] \\
&\geq 1 - \sum_{j=1}^{d} \mathbf{Pr}\left[|\widetilde{\mathbf{x}}^{\text{APX}}(j) - \widetilde{\mathbf{x}}^{\text{OPT}}(j)| \geq \delta/\sqrt{d}\right] \\
&\geq 1 - 2d\exp\left(-\frac{2\delta^2\sigma^2}{d^2\eta^2}\right)
\end{aligned}$$

where the last inequality follows from Hoeffding's inequality (recall that $\widetilde{\mathbf{x}}^1(j), \ldots, \widetilde{\mathbf{x}}^\sigma(j)$ are *i.i.d* random variables with a support of length at most η). By choosing $\sigma = \frac{1}{\delta\sqrt{2}} d\eta\sqrt{\ln\frac{2d}{\epsilon}}$ we get $\mathbf{Pr}[\|\widetilde{\mathbf{x}}^{\text{APX}} - \widetilde{\mathbf{x}}^{\text{OPT}}\| < \delta] \geq 1 - \epsilon$ which completes the proof. □

Corollary 3. *Consider any algorithm that makes up to* $c \in \mathbb{N}$ *calls to* ApxSolve*. For any* $\epsilon, \delta > 0$*, there exists* $\sigma \in \mathbb{N}_0$ *such that* σ *is polynomial in* $c, \log\frac{1}{\epsilon}, \delta$ *and such that with probability* $1 - \epsilon$ *all calls to* ApxSolve *have errors less than* δ.

Proof. By Theorem 13, there exists $\sigma = O(\frac{1}{\delta} d\eta\sqrt{\log\frac{dc}{\epsilon}})$ such that a single call to ApxSolve returns a point with an error of at most δ with probability at least $1 - \epsilon/\sigma$ and consequently with probability $1 - \epsilon$ all calls have errors of at most δ. □

Implementation of vertices of $\widetilde{\mathbf{X}}$. Another consequence of Lemma 4 is that every vertex of $\widetilde{\mathbf{X}}$ has a simple implementation.

Proposition 5. *Consider any vertex $\tilde{\mathbf{x}}$ of $\widetilde{\mathbf{X}}$ and let $\mathbf{w}$ be the normal vector to a strictly supporting hyperplane for $\tilde{\mathbf{x}}$ (i.e., $\mathbf{w} \cdot \tilde{\mathbf{x}}' - \mathbf{w} \cdot \tilde{\mathbf{x}} < 0$, for all $\tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}} \setminus \{\tilde{\mathbf{x}}\}$). $\tilde{\mathbf{x}}$ has a deterministic implementation as follows: under each realization of $\mathbf{t}$, select the point $\mathsf{x} = \arg^{\dagger}\max_{\mathsf{x} \in \mathbf{X}^{\mathsf{t}}} \mathbf{w} \cdot \mathsf{x}$.*

Furthermore, every point in $\widetilde{\mathbf{X}}$ can be implemented by decomposing it as a convex combination of vertices of $\widetilde{\mathbf{X}}$ and randomizing over the implementations of those vertices. This observation is the basis of our algorithms for implementing arbitrary points in $\widetilde{\mathbf{X}}$.

Proposition 6. *Every $\tilde{\mathbf{x}}$ of $\widetilde{\mathbf{X}}$ has a randomized implementation as follows: round $\tilde{\mathbf{x}}$ to a vertex $\tilde{\mathbf{x}}^*$ at random such that $\mathbf{E}[\tilde{\mathbf{x}}^*] = \tilde{\mathbf{x}}$; then implement $\tilde{\mathbf{x}}^*$ (as in Proposition 5).*

Proof. Since $\widetilde{\mathbf{X}}$ is a convex polytope, every $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ can be written as $\tilde{\mathbf{x}} = \sum_{\ell} \lambda_{\ell} \tilde{\mathbf{x}}^*_{\ell}$ where each $\tilde{\mathbf{x}}^*_{\ell}$ is a vertex of $\widetilde{\mathbf{X}}$ and $\sum_{\ell} \lambda_{\ell} = 1$. Consequently, $\tilde{\mathbf{x}}$ can be implemented by rounding it to $\tilde{\mathbf{x}}^*_{\ell}$ with probability λ_{ℓ} for each ℓ, and then using the corresponding deterministic implementation. □

3.2 General Polytopes

For both the optimization problem and the implementation problem over $\widetilde{\mathbf{X}}$, we present polynomial time reductions to the linear optimization problem over $\widetilde{\mathbf{X}}$; recall that the linear optimization problem over $\widetilde{\mathbf{X}}$ can be approximated to any degree of accuracy and with high probability via sampling in polynomial time. To avoid complicating the exposition we omit the details related to preserving the sampling error through the reduction.

Optimization over $\widetilde{\mathbf{X}}$. Any concave optimization problem over $\widetilde{\mathbf{X}}$ can be solved by the Ellipsoid method as long as a separation oracle for $\widetilde{\mathbf{X}}$ is available. Grötschel et al. (1981) showed that, for any convex polytope, the separation problem and the linear optimization problem are *equivalent*, i.e., there exists a polynomial time reduction between the two problems. Our approach is to reduce the separation problem for $\widetilde{\mathbf{X}}$ to the linear optimization problem over $\widetilde{\mathbf{X}}$ which can then be solved via sampling as explained in the previous section.

The separation problem for $\widetilde{\mathbf{X}}$ can be reduced to the linear optimization problem over $\widetilde{\mathbf{X}}$ by using the Ellipsoid method itself. Recall that for any point $\tilde{\mathbf{x}} \notin \widetilde{\mathbf{X}}$ there can be several hyperplanes separating $\tilde{\mathbf{x}}$ from $\widetilde{\mathbf{X}}$; each such separating hyperplane is determined by its normal vector.

Definition 7 (Separating Witness)**.** *Consider any* $\tilde{\mathbf{x}}, \mathbf{w} \in \mathbb{R}^d$*.* $\mathbf{w}$ *is a* separating witness *for* $\tilde{\mathbf{x}}$ *iff*

$$\mathbf{w} \cdot \tilde{\mathbf{x}}' - \mathbf{w} \cdot \tilde{\mathbf{x}} < 0, \qquad \textit{for all } \tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}.$$

The set of all separating witnesses for $\tilde{\mathbf{x}}$ *is*

$$\mathbf{W}(\tilde{\mathbf{x}}) = \left\{ \mathbf{w} \in \mathbb{R}^d \middle| \mathbf{w} \cdot \tilde{\mathbf{x}}' - \mathbf{w} \cdot \tilde{\mathbf{x}} < 0, \forall \tilde{\mathbf{x}}' \in \widetilde{\mathbf{X}} \right\}.$$

$\mathbf{W}(\tilde{\mathbf{x}})$ is a convex cone consisting of all vectors that are normal to a separating hyperplane for $\tilde{\mathbf{x}}$; in particular, $\mathbf{W}(\tilde{\mathbf{x}})$ is empty for every $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$.

The separation problem for $\widetilde{\mathbf{X}}$ can be reduced to linear optimization problem over $\widetilde{\mathbf{X}}$ as follows: given a query $\tilde{\mathbf{x}} \in \mathbb{R}^d$, use the Ellipsoid method to verify whether $\mathbf{W}(\tilde{\mathbf{x}})$ is empty (hence $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$), and if not, to find a $\mathbf{w} \in \mathbf{W}(\tilde{\mathbf{x}})$; for every $\mathbf{w} \in \mathbb{R}^d$ queried by the Ellipsoid method, optimizing over $\widetilde{\mathbf{X}}$ along the direction of $\mathbf{w}$ either yields a separating hyperplane for $\tilde{\mathbf{x}}$, or yields a separating hyperplane for $\mathbf{w}$ itself. The reduction is formally described in Definition 8 and Definition 9.

Definition 8. DUALSEP$(\widetilde{\mathbf{x}}, \mathbf{w})$. *Given a query* $\mathbf{w} \in \mathbb{R}^d$ *and* $\widetilde{\mathbf{x}} \in \mathbb{R}^d$*, either confirm that* $\mathbf{w} \in \mathbf{W}(\widetilde{\mathbf{x}})$*, or find a hyperplane that separates* $\mathbf{w}$ *from* $\mathbf{W}(\widetilde{\mathbf{x}})$*:*

1. *Compute* $\widetilde{\mathbf{x}}^* = \arg^\dagger\max_{\widetilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}} \mathbf{w} \cdot \widetilde{\mathbf{x}}'$.
2. *If* $\mathbf{w} \cdot (\widetilde{\mathbf{x}} - \widetilde{\mathbf{x}}^*) > 0$*, assert* $\mathbf{w} \in \mathbf{W}(\widetilde{\mathbf{x}})$.
3. *Otherwise,* $\mathbf{w}$ *can be separated from* $\mathbf{W}(\widetilde{\mathbf{x}})$ *by*

$$\mathbf{w}' \cdot (\widetilde{\mathbf{x}} - \widetilde{\mathbf{x}}^*) > 0, \qquad\qquad \textit{for all} \quad \mathbf{w}' \in \mathbf{W}(\widetilde{\mathbf{x}}).$$

Definition 9. SEP$(\widetilde{\mathbf{x}})$. *Given a query* $\widetilde{\mathbf{x}} \in \mathbb{R}^d$*, either confirm that* $\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$*, or find a hyperplane that separates* $\widetilde{\mathbf{x}}$ *from* $\widetilde{\mathbf{X}}$*:*

1. *Find a* $\mathbf{w} \in \mathbf{W}(\widetilde{\mathbf{x}})$ *using the Ellipsoid method with separation oracle* DUALSEP$(\widetilde{\mathbf{x}}, \cdot)$.
2. *If the Ellipsoid method concludes that* $\mathbf{W}(\widetilde{\mathbf{x}})$ *is empty, assert* $\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$.
3. *Otherwise,* $\widetilde{\mathbf{x}}$ *can be separated from* $\widetilde{\mathbf{X}}$ *by*

$$\mathbf{w} \cdot (\widetilde{\mathbf{x}}' - \widetilde{\mathbf{x}}) < 0, \qquad\qquad \textit{for all} \quad \widetilde{\mathbf{x}}' \in \widetilde{\mathbf{X}}.$$

Theorem 14. *The separation problem over* $\widetilde{\mathbf{X}}$ *can be reduced to the linear optimization problem over* $\widetilde{\mathbf{X}}$ *in polynomial time.*

Proof. The claim follows from the algorithms of Definition 8 and Definition 9. □

The linear optimization problems over $\widetilde{\mathbf{X}}$ can be approximated to any degree of accuracy with high probability by sampling (Definition 5). By modifying the above reduction to take the sampling error into account and then combining it with Corollary 3 the following theorem can be proved.

Theorem 15. *For any* $\epsilon, \delta > 0$*, there exists an algorithm for the separation problem over* $\widetilde{\mathbf{X}}$*, and consequently an algorithm for the concave optimization problem over* $\widetilde{\mathbf{X}}$*, with an error of less than* δ *with probability at least* $1 - \epsilon$*, and with running time polynomial in* $d, \eta, \delta, \log \frac{1}{\epsilon}$.

Implementation of points in $\widetilde{\mathbf{X}}$. Recall that any $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ can be implemented by randomizing over the implementations of the vertices of $\widetilde{\mathbf{X}}$. This can be done by (a) finding a subset of vertices (of polynomial size) such that $\tilde{\mathbf{x}}$ falls in their convex hall, (b) decomposing $\tilde{\mathbf{x}}$ as a convex combination of those vertices (i.e., by solving a linear program), and (c) picking a vertex at random according to that convex combination and then implementing that vertex. The only non-trivial step is (a). However, it turns out the separation oracle SEP in fact does the step (a) implicitly! The algorithm is sketched in Definition 11.

Definition 10. IMPLEMENTVERTEX$(\mathbf{w}, \mathbf{X^t})$. *Given* $\mathbf{w} \in \mathbb{R}^d$ *and* $\mathbf{X^t} \subset \mathbb{R}^d$ *(specified by* $\mathbf{t}$*), select a vertex* $\mathsf{x} \in \mathbf{X^t}$ *so as to implement the vertex* $\tilde{\mathbf{x}} = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\arg^\dagger\max_{\mathsf{x}\in\mathbf{X^t}} \mathbf{w} \cdot \mathsf{x}] \in \widetilde{\mathbf{X}}$ *in expectation:*

1. *Return* $\arg^\dagger\max_{\mathsf{x}\in\mathbf{X^t}} \mathbf{w} \cdot \mathsf{x}$.

Definition 11. IMPLEMENT$(\tilde{\mathbf{x}}, \mathbf{X^t})$. *Given* $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ *and* $\mathbf{X^t} \subset \mathbb{R}^d$ *(specified by* $\mathbf{t}$*), select a vertex* $\mathsf{x} \in \mathbf{X^t}$ *at random so as to implement* $\tilde{\mathbf{x}}$ *in expectation:*

1. *Run the Ellipsoid method with separation oracle* DUALSEP$(\tilde{\mathbf{x}}, \cdot)$.

2. *Let* $\mathbf{w}_1, \ldots, \mathbf{w}_c \in \mathbb{R}^d$ *denote the queries by the Ellipsoid method to* DUALSEP*; also let* $\tilde{\mathbf{x}}_1^*, \ldots, \tilde{\mathbf{x}}_c^* \in \widetilde{\mathbf{X}}$ *denote the corresponding points computed by* DUALSEP *where, for each* $\ell \in [c]$, $\mathbf{w}_\ell$ *is separated from* $\mathbf{W}(\tilde{\mathbf{x}})$ *by*

$$\mathbf{w}' \cdot (\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_\ell^*) > 0, \qquad\qquad \textit{for all} \quad \mathbf{w}' \in \mathbf{W}(\tilde{\mathbf{x}}).$$

3. *Compute* $\lambda_1, \ldots, \lambda_c \in [0, 1]$ *such that* $\sum_{\ell=1}^c \lambda_\ell \tilde{\mathbf{x}}_\ell^* = \tilde{\mathbf{x}}$ *and* $\sum_{\ell=1}^c \lambda_\ell = 1$ *(i.e., using linear programming).*

4. *Pick* $\ell \in [c]$ *at random with probability* λ_ℓ.

5. *Return* IMPLEMENTVERTEX$(\mathbf{w}_\ell, \mathbf{X^t})$.

Observe that the first step of the algorithm IMPLEMENT (Definition 11) is practically the same as invoking the separation oracle SEP on point $\tilde{\mathbf{x}}$. Given that $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$, the Ellipsoid method will confirm $\mathbf{W}(\tilde{\mathbf{x}})$ is empty after making polynomially many calls to DUALSEP($\tilde{\mathbf{x}}, \cdot$), so c is polynomially bounded. For each $\mathbf{w}_\ell$ queried by the Ellipsoid method, DUALSEP computes a vertex $\widetilde{\mathbf{x}}_\ell^*$ which is optimal along the direction of $\mathbf{w}_\ell$. It is easy to see that the Ellipsoid method concludes the emptiness of $\mathbf{W}(\tilde{\mathbf{x}})$ if and only if $\tilde{\mathbf{x}}$ falls in the convex hall of $\widetilde{\mathbf{x}}_1^*, \ldots, \widetilde{\mathbf{x}}_c^*$. Therefore, the first step of the algorithm indeed finds polynomially many vertices of $\widetilde{\mathbf{X}}$ whose convex hall contain $\tilde{\mathbf{x}}$. The algorithm picks $\ell \in [c]$ at random such that $\mathbf{E}_\ell[\widetilde{\mathbf{x}}_\ell^*] = \tilde{\mathbf{x}}$ and then implements $\widetilde{\mathbf{x}}_\ell^*$.

Theorem 16. *For any point in $\widetilde{\mathbf{X}}$, the implementation problem can be reduced to the linear optimization problem over $\widetilde{\mathbf{X}}$ in polynomial time.*

Proof. The claim follows from the algorithms of Definition 11. □

Recall that the linear optimization problems over $\widetilde{\mathbf{X}}$ can be approximated to any degree of accuracy with high probability by sampling (Definition 5). Consequently, from Definition 11 and Corollary 3 the following theorem can be proved.

Theorem 17. *For any $\epsilon, \delta > 0$, there exists an algorithm for the implementation problem for any point in $\widetilde{\mathbf{X}}$ which has an error of less than δ with probability at least $1 - \epsilon$, and with running time polynomial in $d, \eta, \delta, \log \frac{1}{\epsilon}$.*

3.3 Polymatroids

In this section we consider problem instances where the convex hall of each $\mathbf{X}^{\mathbf{t}}$ is a polymatroid. We present a polymatroidal characterization of $\widetilde{\mathbf{X}}$ which also leads to specialized algorithms for both optimization and implementation over $\widetilde{\mathbf{X}}$. Since our implementation algorithms only select points that are vertices of $\mathbf{X}^{\mathbf{t}}$, we can replace each $\mathbf{X}^{\mathbf{t}}$ with its convex hall without loss of generality. We assume that a non-decreasing submodular function $\mathcal{F}^{\mathbf{t}}$ is given for each $\mathbf{t} \in \mathbf{D}$ such that $\mathbf{X}^{\mathbf{t}}$ is the polymatroid associated with $\mathcal{F}^{\mathbf{t}}$.

Definition 12. *Given a non-decreasing submodular function $\mathcal{F} : [d] \to \mathbb{R}_+$, the polymatroid associated with $\mathcal{F}$ is the convex polytope defined by*

$$\mathbf{P}_{\mathcal{F}} = \left\{\mathbf{x} \in \mathbb{R}_+^d \middle| \mathbf{x}(S) \leq \mathcal{F}(S), \forall S \subseteq [d]\right\}$$

where $\mathbf{x}(S)$ is a shorthand notation for $\sum_{j \in S} \mathbf{x}(j)$.

Consider an arbitrary $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ and a corresponding selection policy $\hat{\mathbf{x}}$ that implements $\tilde{\mathbf{x}}$, i.e., $\mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathbf{E}_{\mathsf{x}\sim\hat{\mathbf{x}}(\mathbf{t})}[\mathsf{x}]] = \tilde{\mathbf{x}}$. Recall that $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ iff such a selection policy exists. Observe that x must be selected form $\mathbf{X}^{\mathbf{t}}$ therefore

$$\mathsf{x}(S) \leq \mathcal{F}^{\mathbf{t}}(S), \qquad \forall \mathbf{t} \in \mathbf{D}, \forall \mathsf{x} \in \hat{\mathbf{x}}(\mathbf{t}), \forall S \subseteq [d]\,.$$

Taking the expectation over $\mathsf{x} \sim \hat{\mathbf{x}}(\mathbf{t})$ and over $\mathbf{t} \sim \mathbf{D}$ we get

$$\mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\mathbf{E}_{\mathsf{x}\sim\hat{\mathbf{x}}(\mathbf{t})}\left[\mathsf{x}(S)\right]\right] \leq \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\mathcal{F}^{\mathbf{t}}(S)\right], \qquad \forall S \subseteq [d]\,.$$

Observe that the left hand side of the above inequality is exactly $\tilde{\mathbf{x}}$ and the right hand side is a non-decreasing submodular function of S.[3]. Consequently, the above

[3]Because it is the expectation of (i.e., weighted sum of) non-decreasing submodular functions

inequality implies that $\tilde{\mathbf{x}} \in \mathbf{P}_{\widetilde{\mathcal{F}}}$ where $\widetilde{\mathcal{F}}$ is defined as follows.

Definition 13. $\widetilde{\mathcal{F}} : [d] \to \mathbb{R}_+$ *is a non-decreasing submodular function defined as*

$$\widetilde{\mathcal{F}}(S) = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\mathcal{F}^{\mathbf{t}}(S)\right], \qquad\qquad S \subseteq [d]$$

So far we have shown that if $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ then $\tilde{\mathbf{x}} \in \mathbf{P}_{\widetilde{\mathcal{F}}}$ which implies that $\widetilde{\mathbf{X}}$ is a subset of $\mathbf{P}_{\widetilde{\mathcal{F}}}$; we will show that the two are in fact equal which implies that $\widetilde{\mathbf{X}}$ is itself a polymatroid.

Theorem 18. $\widetilde{\mathbf{X}}$ *is the polymatroid associated with* $\widetilde{\mathcal{F}}$ *(i.e.,* $\widetilde{\mathbf{X}} = \mathbf{P}_{\widetilde{\mathcal{F}}}$*).*

The proof of the above theorem heavily relies on the following characterization of the vertices of a polymatroid.

Proposition 7 (Polymatroid Vertices)**.** *Consider an arbitrary non-decreasing submodular function* $\mathcal{F} : [d] \to \mathbb{R}_+$ *and the associated polymatroid* $\mathbf{P}_{\mathcal{F}}$*. Every ordered subset* $\pi = (\pi_1, \pi_2, \ldots) \subseteq [d]$ *identifies a vertex* $\mathbf{x} \in \mathbf{P}_{\mathcal{F}}$ *whose value at each coordinate* $j \in [d]$ *is given by*

$$\mathbf{x}(j) = \begin{cases} \mathcal{F}(\{\pi_1, \ldots, \pi_r\}) - \mathcal{F}(\{\pi_1, \ldots, \pi_{r-1}\}) & \text{if } j = \pi_r \text{ for some } r \in [|\pi|] \\ 0 & \text{if } j \notin \pi \end{cases}.$$

Furthermore, every vertex of $\mathbf{P}_{\mathcal{F}}$ *is identified by one or more such ordered subsets.*

We now proceed to prove Theorem 18. Note that we have already shown $\widetilde{\mathbf{X}} \subseteq \mathbf{P}_{\widetilde{\mathcal{F}}}$, so we only need to show $\mathbf{P}_{\widetilde{\mathcal{F}}} \subseteq \widetilde{\mathbf{X}}$. Since both $\mathbf{P}_{\widetilde{\mathcal{F}}}$ and $\widetilde{\mathbf{X}}$ are convex polytopes it is enough to show that every vertex of $\mathbf{P}_{\widetilde{\mathcal{F}}}$ is in $\widetilde{\mathbf{X}}$. Consider an arbitrary vertex $\tilde{\mathbf{x}} \in \mathbf{P}_{\mathcal{F}}$ and a corresponding ordered subset $\pi = (\pi_1, \pi_2, \ldots) \subseteq [d]$; such a π exists by Proposition 7. We explicitly present a selection policy that implements $\tilde{\mathbf{x}}$ which implies that $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$. For each $\mathbf{t} \in \mathbf{D}$, we select the vertex $\mathbf{x}^{\mathbf{t}} \in \mathbf{X}^{\mathbf{t}} = \mathbf{P}_{\mathcal{F}^{\mathbf{t}}}$ associated with π. We show that $\mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathbf{x}^{\mathbf{t}}] = \tilde{\mathbf{x}}$. Observe that for each $j \in [d]$:

- If $j = \pi_r$ for some $r \in [|\pi|]$, then

$$\begin{aligned}\widetilde{\mathbf{x}}(j) &= \widetilde{\mathcal{F}}(\{\pi_1, \ldots, \pi_r\}) - \widetilde{\mathcal{F}}(\{\pi_1, \ldots, \pi_{r-1}\}) \\ &= \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\mathcal{F}^{\mathbf{t}}(\{\pi_1, \ldots, \pi_r\}) - \mathcal{F}^{\mathbf{t}}(\{\pi_1, \ldots, \pi_{r-1}\})\right] \\ &= \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\mathbf{x}^{\mathbf{t}}(j)\right].\end{aligned}$$

- Otherwise $j \notin \pi$ which implies that

$$\widetilde{\mathbf{x}}(j) = 0 = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\mathbf{x}^{\mathbf{t}}(j)\right].$$

Therefore $\mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathbf{x}^{\mathbf{t}}] = \widetilde{\mathbf{x}}$ which completes the proof. The above proof also implies the following theorem.

Theorem 19. *Consider an arbitrary vertex $\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ and a corresponding ordered subset $\pi \subseteq [d]$. Then $\widetilde{\mathbf{x}}$ can be implemented by selecting, for each $\mathbf{t} \in \mathbf{D}$, the vertex of $\mathbf{X}^{\mathbf{t}}$ associated with π.*

Optimization over $\widetilde{\mathbf{X}}$. Recall that any concave optimization problem over $\widetilde{\mathbf{X}}$ can be solved in polynomial time by the Ellipsoid method assuming a polynomial time separation oracle is available. $\widetilde{\mathbf{X}}$ is a polymatroid which is defined by an exponential number of linear inequalities, however the separation problem for any given $\widetilde{\mathbf{x}} \in \mathbb{R}^d$ can be solved in polynomial time as follows: find $S^* = \arg\min_{S\subseteq[d]} \widetilde{\mathcal{F}}(S) - \widetilde{\mathbf{x}}(S)$; if $\widetilde{\mathbf{x}} \notin \widetilde{\mathbf{X}}$, the inequality $\widetilde{\mathbf{x}}(S^*) \leq \widetilde{\mathcal{F}}(S^*)$ must be violated, and that yields a separating hyperplane for $\widetilde{\mathbf{x}}$. Note that $\widetilde{\mathcal{F}}(S) - \widetilde{\mathbf{x}}(S)$ is itself submodular in S, so it can be minimized in strong polynomial time. Consequently, optimization problems can be solved over polymatroids in polynomial time. In some cases $\widetilde{\mathcal{F}}$ can be computed exactly in strong polynomial time (e.g., using dynamic programming); otherwise it can be approximated to any degree of accuracy with high probability via sampling in polynomial time. In either case, the separation problem for $\widetilde{\mathbf{X}}$ can be solved without relying on the the Ellipsoid method; that makes concave optimization over $\widetilde{\mathbf{X}}$ more efficient than the general case.

Implementation of points in $\widetilde{\mathbf{X}}$. Recall that any $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ can be implemented by randomizing over the implementations of the vertices of $\widetilde{\mathbf{X}}$. This can be done by (a) iteratively making small changes to $\tilde{\mathbf{x}}$ to arrive at a vertex while ensuring that the expected change at each iteration is 0, and (b) implementing the vertex obtained in the previous step. Recall that by Theorem 19 any vertex of $\widetilde{\mathbf{X}}$ can be implemented as in Definition 15. Next we describe our approach for (a).

Definition 14 (Tight Sets). *Consider an arbitrary non-decreasing submodular function $\mathcal{F} : [d] \to \mathbb{R}_+$ where $\mathcal{F}(\emptyset) = 0$ and consider the associated polymatroid $\mathbf{P}_{\mathcal{F}}$. A subset $S \subseteq [d]$ is* tight *with respect to a given $\mathbf{x} \in \mathbf{P}_{\mathcal{F}}$ iff $\mathbf{x}(S) = \mathcal{F}(S)$. A set $\Psi = \{S_0, S_1, \ldots\} \subset 2^{[d]}$ is called a* nested family of tight sets *with respect to $\mathbf{x} \in \mathbf{P}_{\mathcal{F}}$, if and only the elements of Ψ can be ordered/relabeled such that $\emptyset = S_0 \subset \cdots \subset S_{|S|-1} \subseteq [d]$, and such that S_r is tight with respect to $\mathbf{x}$ (for every $r \in \{0, \ldots, |S|\}$).*

Definition 16 sketches the algorithm ImplementVertexP for implementing any $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}} = \mathbf{P}_{\widetilde{\mathcal{F}}}$. The algorithm makes small changes to $\tilde{\mathbf{x}}$ iteratively until a vertex is reached. At each iteration ℓ, it computes $\tilde{\mathbf{x}}^{\ell} \in \mathbf{P}_{\widetilde{\mathcal{F}}}$, and a nested family of tight sets Ψ^{ℓ} (with respect to $\tilde{\mathbf{x}}^{\ell}$) such that

- $\mathbf{E}[\tilde{\mathbf{x}}^{\ell} | \tilde{\mathbf{x}}^{\ell-1}] = \tilde{\mathbf{x}}^{\ell-1}$, and
- $\tilde{\mathbf{x}}^{\ell}$ is closer to a vertex in the sense that either the number of non-zero coordinate values increases by one or the number of tight sets decreases by one.

Observe that the above process must stop after at most $2d$ iterations[4]. At the ℓ-th iteration of the rounding process, a vector $\mathbf{\Delta} \in \mathbb{R}^d$ and $\delta, \delta' \in \mathbb{R}_+$ are computed such that both $\tilde{\mathbf{x}}^{\ell-1} + \delta\mathbf{\Delta}$ and $\tilde{\mathbf{x}}^{\ell-1} - \delta'\mathbf{\Delta}$ are still in $\mathbf{P}_{\widetilde{\mathcal{F}}}$, but closer to a vertex. The algorithm then chooses at random $\delta'' \in \{\delta, -\delta'\}$ such that $\mathbf{E}[\delta''] = 0$, and sets $\tilde{\mathbf{x}}^{\ell} \leftarrow \tilde{\mathbf{x}}^{\ell-1} + \delta''\mathbf{\Delta}$.

Definition 15. ImplementVertexP$(\pi, \mathbf{X}^{\mathbf{t}})$. *Given an ordered subset $\pi \subseteq [d]$ and $\mathbf{X}^{\mathbf{t}} \subset \mathbb{R}^d$ (specified by $\mathbf{t}$), select a vertex $\mathsf{x} \in \mathbf{X}^{\mathbf{t}}$ so as to implement the vertex of $\widetilde{\mathbf{X}}$ associated with π in expectation:*

1. *Return the vertex of $\mathsf{x} \in \mathbf{X}^{\mathbf{t}}$ associated with π (see Proposition 7).*

[4] In fact we will show that it stops after at most d iterations.

Definition 16 (ImplementP($\tilde{\mathbf{x}}, \mathbf{X^t}$)). *Given $\tilde{\mathbf{x}} \in \widetilde{\mathbf{X}}$ and $\mathbf{X^t} \subset \mathbb{R}^d$ (specified by $\mathbf{t}$), select at random a vertex $\mathbf{x} \in \mathbf{X^t}$ so as to implement $\tilde{\mathbf{x}}$ in expectation.*
In the following, let $\mathbf{1}_j \in [0,1]^d$ be the vector whose value is 1 at coordinate j and 0 everywhere else.

1. *Initialize $\Psi \leftarrow \{\emptyset\}$. Ψ will maintain a nested family of tight sets with respect to $\tilde{\mathbf{x}}$. Whenever Ψ is updated, order/relabel its element as $\Psi = \{S_0, S_1, \ldots\}$ so that $S_0 \subset S_1 \subseteq \cdots$.*

2. *Repeat each of the following steps until no longer applicable:*

   - *If there exist distinct $j, j' \in S_r \setminus S_{r-1}$ for some $r \in [|\Psi| - 1]$:*

     (a) *Let $\mathbf{\Delta} \leftarrow \mathbf{1}_j - \mathbf{1}_{j'}$ Compute $\delta, \delta' \in \mathbb{R}_+$ such that $\tilde{\mathbf{x}} + \delta\mathbf{\Delta}$ has a new tight set S and $\tilde{\mathbf{x}} - \delta'\mathbf{\Delta}$ has a new tight set S', i.e:*
       - *Let $S \leftarrow \arg\min_{S^{r-1}+j \subseteq S \subseteq S_r - j'} \widetilde{\mathcal{F}}(S) - \tilde{\mathbf{x}}(S)$, and $\delta \leftarrow \widetilde{\mathcal{F}}(S) - \tilde{\mathbf{x}}(S)$.*
       - *Let $S' \leftarrow \arg\min_{S_{r-1}+j' \subseteq S' \subseteq S_r - j} \widetilde{\mathcal{F}}(S') - \tilde{\mathbf{x}}(S')$, and $\delta' \leftarrow \widetilde{\mathcal{F}}(S') - \tilde{\mathbf{x}}(S')$.*

     (b) $\begin{cases} \textit{with prob. } \frac{\delta}{\delta+\delta'}: & \textit{set } \tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} + \delta\mathbf{\Delta}, \textit{ and add } S \textit{ to } \Psi. \\ \textit{with prob. } \frac{\delta'}{\delta+\delta'}: & \textit{set } \tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \delta'\mathbf{\Delta}, \textit{ and add } S' \textit{ to } \Psi. \end{cases}$

   - *If there exists $j \in [d] \setminus S_{|S|-1}$ for which $\tilde{\mathbf{x}}(j) > 0$:*

     (a) *Let $\mathbf{\Delta} \leftarrow \mathbf{1}_j$, and compute $\delta, \delta' \in \mathbb{R}_+$ such that $\tilde{\mathbf{x}} + \delta\mathbf{\Delta}$ has a new tight set S and $\tilde{\mathbf{x}} - \delta'\mathbf{\Delta}$ has a zero at coordinate j, i.e:*
       - *Let $S \leftarrow \arg\min_{S \supseteq S_{|S|-1}+j} \widetilde{\mathcal{F}}(S) - \tilde{\mathbf{x}}(S)$, and $\delta \leftarrow \widetilde{\mathcal{F}}(S) - \tilde{\mathbf{x}}(S)$.*
       - *Let $\delta' \leftarrow \tilde{\mathbf{x}}(j)$.*

     (b) $\begin{cases} \textit{with prob. } \frac{\delta}{\delta+\delta'}: & \textit{set } \tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} + \delta\mathbf{\Delta}, \textit{ and add } S \textit{ to } \Psi. \\ \textit{with prob. } \frac{\delta'}{\delta+\delta'}: & \textit{set } \tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \delta'\mathbf{\Delta} \end{cases}$

3. *Let $\pi = (\pi_1, \ldots, \pi_{|\Psi|-1}) \subseteq [d]$ be the ordered subset associated with Ψ where $\pi_r \in S_r \setminus S_{r-1}$ for each $r \in [|\pi|]$.*

4. *Return* ImplementVertexP($\pi, \mathbf{X^t}$).

Theorem 20. *The implementation problem for any point* $\widetilde{\mathbf{x}} \in \widetilde{\mathbf{X}} = \mathbf{P}_{\widetilde{\mathcal{F}}}$ *can be reduced to computing* $\widetilde{\mathcal{F}}$ *in polynomial time.*

Proof. It follows from the algorithm of Definition 16. □

Chapter 4

Multi to Single Agent Reduction (Ex ante)

4.1 Introduction

In this chapter, we present an alternative multi to single agent decomposition approach that leads to approximately optimal mechanisms, but is far more practical compared to the approach presented in §2 in terms of both computation and applicability. In §2, we presented a general decomposition technique to reduce a multi agent mechanism design problem to single agent subproblems. The decomposition technique allowed us avoid the exponential blow up (as a function of the number of agents) in the size of the optimization problem. On the other hand, the proposed techniques are more theoretical than practical as they heavily rely on the use of Ellipsoid method where each query to the separation oracle involves solving a second optimization problem.

The decomposition technique of the current chapter can be roughly described as the following: (i) Construct a mechanism that satisfies the supply constraints only in expectation (ex-ante); the optimization problem for constructing such a mechanism can be fully decomposed over the set of agents. (ii) Convert the mechanism from the previous step to another mechanism that satisfies the supply constraint at every instance.

We restrict our discussion to Bayesian combinatorial auctions. We are interested in mechanisms that allocate a set of heterogenous items with limited supply to a set of agents in order to maximize the expected value of a certain objective function which is linearly separable over the agents (e.g., welfare, revenue, etc). The agents' types are assumed to be distributed independently according to publicly known priors. We defer the formal statement of our assumptions to §4.2.

Our framework can be summarized as follows. We start by relaxing the supply constraints, i.e., we consider the mechanisms for which only the ex-ante expected number of allocated units of each item is no more than the supply of that item. Note that *"ex-ante"* means that the expectation taken over all possible inputs (i.e., all possible types of the agents). We show that the optimal mechanism for the relaxed problem can be constructed by independently running n single agent mechanisms, where each single agent mechanism is subject to an ex-ante probabilistic supply constraint. In particular, we show that if one can construct an α-approximate mechanism for each single agent problem, then running these mechanisms simultaneously and independently yields an α-approximate mechanism for the relaxed multiple agent problem. We then present two methods for converting the mechanism for the relaxed problem to a mechanism for the original problem while losing a small constant factor in the approximation. We present two generic multi agent mechanisms that use the single agent mechanisms from the previous step as blackboxes [1]. In the first mechanism, we serve agents sequentially by running, for each agent, the corresponding single agent mechanism from the previous step. However, we sometimes randomly preclude some of the items from the early agents in order to ensure that late agents get the same chance of being offered with those items; we ensure that the ex-ante expected probability of preclusion is equalized over all agents, regardless of the order in which they are served (i.e., we simultaneously minimize the preclusion probability for all agents). In the second mechanism, we run all of the single agent mechanisms simultaneously and then modify the outcomes by deallocating some units of the over-allocated items at random while adjusting the payments respectively; we ensure that the ex-ante probability of deallocation is equalized among all units of each item and therefore simultaneously minimized for all agents.

We also introduce a toy problem, *the magician's problem*, in §4.4, along with a

[1] Note that the single agent mechanisms can be different for different agents, e.g., to accommodate different classes of agents.

near optimal solution for it, which is used as the main ingredient of our multi agent mechanisms. A more general variant of this toy problem is presented in §5 along with other applications.

As applications of our framework, in §4.6, we present mechanisms with improved approximation factor for several settings from the literature. For each setting we present a single agent mechanism that satisfies the requirements of our framework, and can be plugged in one of our generic multi agent mechanisms.

4.1.1 Related Work

In single dimensional settings, the related works form the CS literature are mostly focused on approximating the VCG mechanism for welfare maximization and/or approximating the Myerson's mechanism Myerson (1981) for revenue maximization (e.g., Bulow and Roberts (1989); Babaioff et al. (2006); Blumrosen and Holenstein (2008); Hartline and Roughgarden (2009); Dhangwatnotai et al. (2010); Chakraborty et al. (2010); Yan (2011)). Most of them consider mechanisms that have simple implementation and are computationally efficient. For welfare maximization in single dimensional settings, Hartline and Lucier (2010) gives a blackbox reduction from mechanism design to algorithmic design.

In multidimensional setting, for welfare maximization, Hartline et al. (2011) presents a blackbox reduction from mechanism design to algorithm design which subsumes the earlier work of Hartline and Lucier (2010).

Our work is also related to a line of work on approximating the Bayesian optimal mechanism. These works tend to look for simple mechanisms that give constant (e.g., two) approximations to the optimal mechanism. Chawla et al. (2007), Briest et al. (2010), and Cai and Daskalakis (2011) consider item pricing and lottery pricing for a single agent; the first two give constant approximations the last gives a $(1 + \epsilon)$-approximation for any ϵ. These problems are related to the single-agent problems we consider. Chawla et al. (2010) and Bhattacharya et al. (2010) extend these approaches to multi-agent auction problems. For revenue maximization, Chawla et al. (2010) presents several sequential posted pricing mechanisms for various settings and various types of matroid feasibility constraints. These mechanisms have simple implementation and approximate the revenue of the optimal mechanism. For unit demand agents whose valuations' for the items are distributed according to product distributions, Chawla et al. (2010) present a sequential posted pricing mechanism that obtains in expectation at least $\frac{1}{6.75}$-fraction of the revenue of the optimal posted pricing mechanism. In §4.6.2, we present an improved sequential posted pricing mechanism for this setting with an approximation factor of $\frac{1}{2}\gamma_k$ in which k is the number of units available of each item, and γ_k is a constant which is at least $1 - \frac{1}{\sqrt{k+3}}$. For combinatorial auctions with additive/correlated valuations with budget and demand constraints, Bhattacharya et al. (2010) presents all-pay $\frac{1}{4}$-approximate BIC mechanisms for revenue maximization and a similar mechanism for welfare maximization. In subsection 4.6.4, we present an improved mechanism for this setting with an approximation factor of γ_k. Note that γ_k is at least $\frac{1}{2}$ and approaches 1 as $k \to \infty$. Bhattacharya et al. (2010) also presents sequential posted pricing mechanisms for the same setting, obtaining $O(1)$ approximation factors. For a similar setting, in §4.6.3, we present an improved sequential posted pricing mechanism with an approximation factor of $(1 - \frac{1}{e})\gamma_k$. Finally, Chawla et al. (2011) also considers various settings with hard budget constraints.

4.2 Preliminaries

We present our framework for combinatorial auctions, but it can be readily applied to Bayesian mechanism design in other contexts. We begin by defining the model and some notation.

Model. We consider the problem of allocating m indivisible heterogenous items to n agents where there are k_j units of each item $j \in [m]$. All the relevant private information of each agent $i \in [n]$ is represented by her type $t_i \in \mathrm{T}_i$ where T_i is the type space of agent i. Let $\mathbf{T} = \mathrm{T}_1 \times \cdots \times \mathrm{T}_n$ be the space of all type profiles. The agents' type profile $\mathbf{t} \in \mathbf{T}$ is distributed according to a publicly known prior $\mathbf{D}$. We use $\mathsf{x}_{ij}(\mathbf{t})$ and $\mathsf{p}_i(\mathbf{t})$ to denote the random variables[2] respectively for the allocation of item j to agent i and the payment of agent i, for type profile $\mathbf{t}$. For a mechanism M, the random variables for allocations and payments are denoted respectively by $\mathsf{x}_{ij}^M(\mathbf{t})$ and $\mathsf{p}_i^M(\mathbf{t})$. We are interested in computing a mechanism that (approximately) maximizes[3] the expected value of a given objective function $\textsc{Obj}(\mathbf{t}, \mathbf{x}, \mathbf{p})$ where $\mathbf{t}$, $\mathbf{x}$, and $\mathbf{p}$ respectively represent the types, the allocations, and the payments of all agents. We are only interested in mechanisms which are within a given space M of feasible mechanisms. Formally, we aim to compute a mechanism $M \in \mathrm{M}$ that (approximately) maximizes $\mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\textsc{Obj}(\mathbf{t}, \mathbf{x}^M(\mathbf{t}), \mathbf{p}^M(\mathbf{t}))]$.

Assumptions. We make the following assumptions.

(A1) **Independence.** The agents' types must be distributed independently, i.e., $\mathbf{D} = \mathcal{D}_1 \times \ldots \times \mathcal{D}_n$ where $\mathcal{D}_i$ is the distribution of types for agent i. Note that if agent i has multidimensional types, $\mathcal{D}_i$ itself does not need to be a product distribution.

(A2) **Linear Separability of Objective.** The objective function must be linearly separable over the agents, i.e., $\textsc{Obj}(\mathbf{t}, \mathbf{x}, \mathbf{p}) = \sum_i \textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{p}_i)$ where t_i, x_i, and p_i respectively represent the type, the allocations, and the payment of agent i.

(A3) **Single–Unit Demands.** No agent should ever need more than *one unit* of each item, i.e., $\mathsf{x}_{ij}(\mathbf{t}) \in \{0, 1\}$ for all $\mathbf{t} \in \mathbf{T}$. This assumption is not necessary and is only to simplify the exposition; it can be removed as explained in §4.7.

(A4) **Incentive Compatibility.** M must be restricted to (Bayesian) incentive compatible mechanisms. By direct revelation principle this assumption is without loss of generality[4],

[2] Note that these random variables are often correlated. Furthermore, for a deterministic mechanism, these variables take deterministic values as a function of $\mathbf{t}$.

[3] All of the results can be applied to minimization problems by simply maximizing the negation of the objective function.

[4] It is WLOG, given that we are only interested in mechanisms that have Bayes-Nash equilibria.

(A5) **Convexity.** M must be a convex space. In other words, every convex combination of every two mechanisms from M must itself be a mechanism in M. A convex combination of two mechanisms $M, M' \in$ M is another mechanism M'' which simply runs M with probability β and runs M' with probability $1-\beta$, for some $\beta \in [0, 1]$. In particular, if M is restricted to deterministic mechanisms, it is not convex; however if M also includes mechanisms that randomize over deterministic mechanisms, then it is convex [5].

(A6) **Decomposability.** The set of constraints specifying M must be decomposable to supply constraints (i.e., $\sum_i x_{ij}(\mathbf{t}) \leq k_j$, for all $\mathbf{t}$ and all items j) and single agent constraints(e.g., incentive compatibility, budget, etc). We define this assumption formally as follows. For any mechanism M, let $[[M]]_i$ be the single agent mechanism perceived by agent i, by simulating[6] the other agents according to their respective distributions $\mathbf{D}_{-i}$. Define $\mathrm{M}_i = \{[[M]]_i | M \in \mathrm{M}\}$ to be the space of all feasible single agent mechanisms for agent i. The decomposability assumption requires that for any arbitrary mechanism M the following holds: if M satisfies the supply constraints and also $[[M]]_i \in \mathrm{M}_i$ (for all agents i), then it must be that $M \in$ M.

We shall clarify the last assumption by giving an example. Suppose M is the space of all agent specific item pricing mechanisms, then M satisfies the last assumption. On the other hand, if M is the space of mechanisms that offer the same set of prices to every agent, it does not satisfy the decomposability assumption, because there is an implicit inter agent constraint that the same prices should be offered to different agents.

Throughout the rest of this chapter, we often omit the range of the sums whenever the range is clear from the context (e.g., $\sum_i$ means $\sum_{i\in[n]}$, and $\sum_j$ means $\sum_{j\in[m]}$).

[5]For an example of a randomized non-convex space of mechanisms, consider the space of mechanisms where the expected payment of every type must be either less than \$2 or more than \$4.

[6]The single agent mechanism induced on agent i can be obtained by simulating all agents other than i by drawing a random $\mathbf{t}_{-i}$ from $\mathbf{D}_{-i}$ and running M on agent i and the $n-1$ simulated agents with types $\mathbf{t}_{-i}$; note that this is a single agent mechanism because the simulated agents are just part of the mechanism.

Multi agent problem. Formally, the multi agent problem is to find a mechanism M which is a solution to the following program.

$$\begin{aligned} \text{maximize} \quad & \sum_i \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}\left[\textsc{Obj}_i(t_i, \mathsf{x}_i^M(\mathbf{t}), \mathsf{p}_i^M(\mathbf{t}))\right] & & (OPT) \\ \text{subject to} \quad & \sum_i \mathsf{x}_{ij}^M(\mathbf{t}) \leq k_j, & \forall \mathbf{t} \in \mathbf{T}, \forall j \in [m] & \\ & [[M]]_i \in \mathrm{M}_i, & \forall i \in [n] & \end{aligned}$$

Observe that, in the absence of the first set of constraints, we could optimize the mechanism for each agent independently. This observation is the key to our multi to single agent decomposition, which allows us to approximately decompose/reduce the multi agent problem to single agent problems. A mechanism M is an α*-approximation* of the optimal mechanism if it is a feasible mechanism for the above program and obtains at least α-fraction of the optimal objective value of the program.

Ex ante allocation rule. For a multi agent mechanism M, the *ex ante allocation rule* is a vector $x \in [0,1]^{n\times m}$ in which $x_{ij} = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathsf{x}_{ij}^M(\mathbf{t})]$ is the expected probability of allocating a unit of item j to agent i, where the expectation is taken over all possible type profiles. Note that for any feasible mechanism M, by linearity of expectation, the ex ante allocation rule satisfies $\sum_i x_{ij} \leq k_j$, for every item j.

Single agent problem. The *single agent problem*, for agent i, is to compute an optimal single agent mechanism and its expected objective value, subject to a given upper bound $\overline{x}_i \in [0,1]^m$ on the ex ante allocation rule; in other words, the single agent mechanism may not allocate a unit of item j to agent i with an expected probability of more than $\overline{x}_i$, where the expectation is taken over $t_i \sim \mathcal{D}_i$. Formally, the single agent problem is to compute the optimal value of the following program along with a corresponding solution (i.e., the optimal M_i), for a given $\overline{x}_i$.

$$\begin{aligned} \text{maximize} \quad & \mathbf{E}_{t_i \sim \mathcal{D}_i}\left[\textsc{Obj}_i(t_i, \mathsf{x}_i^{M_i}(t_i), \mathsf{p}_i^{M_i}(t_i))\right] && (OPT_i) \\ \text{subject to} \quad & \mathbf{E}_{t_i \sim \mathcal{D}_i}\left[\mathsf{x}_{ij}^{M_i}(t_i)\right] \leq \overline{x}_{ij}, && \forall j \in [m] \\ & M_i \in \mathrm{M}_i, \end{aligned}$$

We typically denote an optimal single agent mechanism for agent i, subject to a given $\overline{x}_i$, by $M_i\langle\overline{x}_i\rangle$, and denote its expected objective value (i.e., the optimal value of the above program as a function of $\overline{x}_i$) by $R_i(\overline{x}_i)$. Later, we prove that $R_i(\overline{x}_i)$, which we refer to as the *optimal benchmark* for agent i, is a concave function of $\overline{x}_i$. In the case of approximation, we say that a single agent mechanism M_i together with a concave benchmark R_i provide an α-approximation of the optimal single agent mechanism/optimal benchmark, if the expected objective value of $M_i\langle\overline{x}_i\rangle$ is at least $\alpha R_i(\overline{x}_i)$ and if $R_i(\overline{x}_i)$ is an upper bound on the optimal benchmark, for every $\overline{x}_i$.

To make the exposition more concrete, consider the following single agent problem as an example. Suppose there is only one type of item (i.e., $m = 1$) and the objective is to maximize the expected revenue[7]. Suppose agent i's valuation is drawn from a regular distribution with CDF, F_i. The optimal single agent mechanism for i, subject to $\overline{x}_i \in [0, 1]$, is a deterministic mechanism which offers the item at some fixed price, while ensuring that the probability of sale (i.e., the probability of agent i's valuation being above the offered price) is no more than $\overline{x}_i$. In particular, the optimal benchmark $R_i(\overline{x}_i)$ is the optimal value of the following convex program as a function of $\overline{x}_i$.

$$\begin{aligned} \text{maximize} \quad & x_i F_i^{-1}(1 - x_i) \\ \text{subject to} \quad & x_i \leq \overline{x}_i \\ & x_i \in [0, 1] \end{aligned}$$

Furthermore, the optimal single agent mechanism offers the item at the price $F_i^{-1}(1-$

[7] The optimal multi agent mechanism for this setting is given by Myerson (1981); yet we consider this setting to keep the example simple and intuitive.

x_i) where x_i is the optimal assignment for the above convex program. Note that, for a regular distribution, $x_i F_i^{-1}(1-x_i)$ is concave in x_i, so the above program is a convex program.

4.3 Decomposition via Ex ante Allocation Rule

In this section we present general methods for approximately decomposing/reducing the multi agent problem to single agent problems. Recall that a single agent problem is to compute the optimal single agent mechanism $M_i\langle\overline{x}_i\rangle$ and its expected objective value $R_i(\overline{x}_i)$ (i.e., the optimal benchmark), subject to an upper bound $\overline{x}_i$ on the ex ante allocation rule. We present two methods for constructing an approximately optimal multi agent mechanism, using M_i and R_i as black box. Furthermore, we show that if we can only compute an α-approximation of the optimal single agent mechanism/optimal benchmark for each agent i, then the factor α simply carries over to the approximation factor of the final multi agent mechanism.

Multi agent benchmark. We start by showing that the optimal value of the following convex program gives an upper bound on the expected objective value of the optimal multi agent mechanism.

$$\begin{aligned} \text{maximize} \quad & \sum_i R_i(\overline{x}_i) && (\overline{OPT}) \\ \text{subject to} \quad & \sum_i \overline{x}_{ij} \le k_j, \qquad \forall j \in [m] \\ & \overline{x}_{ij} \in [0,1]\,, \end{aligned}$$

We first show that the above program is indeed a convex program.

Theorem 21. *The optimal benchmarks R_i are always concave.*

Proof. We prove this for an arbitrary agent i. Let M_i and R_i denote the optimal single agent mechanism and the optimal benchmark for agent i. To show that R_i

is concave, it is enough to show that for any $\overline{x}_i, \overline{x}'_i \in [0,1]^m$ and any $\beta \in [0,1]$, the following inequality holds.

$$R_i(\beta\overline{x}_i + (1-\beta)\overline{x}'_i) \geq \beta R_i(\overline{x}_i) + (1-\beta)R_i(\overline{x}'_i)$$

Consider the single agent mechanism M'' that works as follows: M'' runs $M_i(\overline{x}_i)$ with probability β and runs $M_i(\overline{x}'_i)$ with probability $1-\beta$. Note that M_i is a convex space (this follows from $A5$ and $A6$), therefore $M'' \in \mathrm{M}_i$. Observe that by linearly of expectation, the ex ante allocation rule of M'' is no more than $\beta\overline{x}_i + (1-\beta)\overline{x}'_i$ and the expected objective value of M'' is exactly $\beta R_i(\overline{x}_i) + (1-\beta)R_i(\overline{x}'_i)$. So the expected objective value of the optimal single agent mechanism, subject to $\beta\overline{x}_i + (1-\beta)\overline{x}'_i$, may only be higher. That implies $R_i(\beta\overline{x}_i + (1-\beta)\overline{x}'_i) \geq \beta R_i(\overline{x}_i) + (1-\beta)R_i(\overline{x}'_i)$ which proves the claim. □

Theorem 22. *The optimal value of the convex program ($\overline{OPT}$) is an upper bound on the expected objective value of the optimal multi agent mechanism.*

Proof. Let M^* be an optimal multi agent mechanism. Let x^* denote the ex ante allocation rule corresponding to M^*, i.e., $x^*_{ij} = \mathbf{E}_{\mathbf{t}\sim\mathbf{D}}[\mathrm{x}^{M^*}_{ij}]$. Observe that x^* is a feasible assignment for the convex program and yields an objective value of $\sum_i R_i(x^*_i)$ which is upper bounded by the optimal value of the convex program. So to prove the theorem it is enough to show that the contribution of each agent i to the expected objective value of M^* is upper bounded by $R_i(x^*_i)$. Consider $M^*_i = [[M^*]]_i$, i.e, the single agent mechanism induced by M^* on agent i. M^*_i can be obtained by simply running M^* on agent i and simulating the other $n-1$ agents with random types $\mathbf{t}_{-i} \sim \mathbf{D}_{-i}$; Observe that M^*_i is a feasible single agent mechanism subject to x^*_i and obtains the same expected objective value as M^* from agent i, so the expected objective value of the optimal single agent mechanism subject to x^*_i could only be higher. □

Constructing multi agent mechanisms. Theorem 22 suggests that by computing an optimal assignment of $\overline{x}$ for the convex program ($\overline{OPT}$) and running the single agent mechanism $M_i\langle\overline{x}_i\rangle$ for each agent i, one might obtain a reasonable multi agent mechanism; however such a multi agent mechanism would only satisfy the supply constraints in expectation; in other words, there is a good chance that some items are over allocated with a non-zero probability. We present two generic multi agent mechanisms for combining the single agent mechanisms and resolving the conflicts in the allocations in such a way that would ensure the supply constraints are met at every instance and not just in expectation. In both approaches we first solve the convex program ($\overline{OPT}$) to compute the optimal $\overline{x}$. The high level idea of each mechanism is explained below.

1. **Pre-Rounding.** This mechanism serves the agents sequentially (arbitrary order); for each agent i, it selects a subset S_i of available items and runs the single agent mechanism $M_i\langle\overline{x}_i[S_i]\rangle$, where $\overline{x}_i[S_i]$ denotes the vector resulting from $\overline{x}_i$ by zeroing the entries corresponding to items not in S_i. In particular, this mechanism sometimes precludes some of the available items from early agents to make them available to late agents. We show that if there are at least k units of each item, then S_i includes item j with probability at least $1-\frac{1}{\sqrt{k+3}}$, for each agent i and each item j.

2. **Post-Rounding.** This mechanism runs $M_i\langle\overline{x}_i\rangle$ for all agents i simultaneously and independently. It then modifies the outcomes by deallocating the over allocated items at random in such a way that the probability of deallocation observed by all agents are equal, and therefore minimized over all agents. The payments are adjusted respectively. We show that if there are at least k units of each item, every allocation is preserved with probability $1-\frac{1}{\sqrt{k+3}}$ from the perspective of the corresponding agent.

We will explain the above mechanisms in more detail in §4.5 and present some technical assumptions that are sufficient to ensure that they retain at least $1-\frac{1}{\sqrt{k+3}}$ fraction of the expected objective value of each $M_i\langle\overline{x}_i\rangle$.

Approximately optimal single agent mechanisms. Throughout the above discussion, we assumed that we can compute the optimal single agent mechanisms and the corresponding optimal benchmarks. However, it is likely that we can only compute an approximation of them. Suppose for each agent i, M_i and R_i, instead of being optimal, only provide an α-approximation of the optimal single agent mechanism/optimal benchmark, and suppose R_i is concave; then we can still use M_i and R_i in the above construction, but the final approximation factor will be multiplied by α.

Main result. The following informal theorem summarizes the main result of this chapter. The formal statement of this result can be found in Theorem 26 and Theorem 27.

Theorem 23 (Market Expansion)**.** *If for each agent $i \in [n]$, an α-approximate single agent mechanism M_i and a corresponding concave benchmark R_i can be constructed in polynomial time, then, with some further assumptions (explained later), a multi agent mechanism $M \in \mathrm{M}$ can be constructed in polynomial time by using M_i as building blocks, such that M is $\gamma_k\alpha$-approximation of the the optimal multi agent mechanism in M, where $k = \min_j k_j$ and γ_k is a constant which is at least $1 - \frac{1}{\sqrt{k+3}}$.*

In order to explain the construction of the multi agent mechanism, we shall first describe the magician's problem and its solution, which is used in both pre-rounding and post-rounding for equalizing the expected probabilities of preclusion/deallocation over all agents.

4.4 The Magician's Problem

In this section, we present an abstract online stochastic toy problem and a near-optimal solution for it which provides the main ingredient for combining single agent mechanisms to form multi agent mechanisms. A generalization of this problem and its solution is presented in §5.2.

Definition 17 (The Magician's Problem). *A magician is presented with a series of boxes one by one, in an online fashion. There is a prize hidden in one of the boxes. The magician has k magic wands that can be used to open the boxes. If a wand is used on box i, it opens, but with a probability of at most x_i, which written on the box, the wand breaks. The magician wishes to maximize the probability of obtaining the prize, but unfortunately the sequence of boxes, the written probabilities, and the box in which the prize is hidden are arranged by a villain, and the magician has no prior information about them (not even the number of the boxes). However, it is guaranteed that $\sum_i x_i \leq k$, and that the villain has to prepare the sequence of boxes in advance (i.e., cannot make any changes once the process has started).*

The magician could fail to open a box either because (a) he might choose to skip the box, or (b) he might run out of wands before getting to the box. Note that once the magician fixes his strategy, the best strategy for the villain is to put the prize in the box that has the lowest ex ante probability of being opened, based on the magician's strategy. Therefore, in order for the magician to obtain the prize with a probability of at least γ, he has to devise a strategy that guarantees an ex ante probability of at least γ for opening each box. Notice that allowing the prize to be split among multiple boxes does not affect the problem. It is easy to show the following strategy ensures an ex ante probability of at least $\frac{1}{4}$ for opening each box: for each box randomize and use a wand with probability $\frac{1}{2}$. But can we do better? We present an algorithm parameterized by a probability $\gamma \in [0, 1]$ which guarantees a minimum ex-ante probability of γ for opening each box while trying to minimize the number of wands broken. In Theorem 24, we show that for $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$ this algorithm never requires more than k wands.

Definition 18 (γ-Conservative Magician)**.** *The magician adaptively computes a sequence of thresholds $\theta_1, \theta_2, \ldots \in \mathbb{N}_0$ and makes a decision about each box as follows: let W_i denote the number of wands broken prior to seeing the i^{th} box; the magician makes a decision about box i by comparing W_i against θ_i; if $W_i < \theta_i$, it opens the box; if $W_i > \theta_i$, it does not open the box; and if $W_i = \theta_i$, it randomizes and opens the box with some probability (to be defined). The magician chooses the smallest threshold θ_i for which $\mathbf{Pr}[W_i \leq \theta_i] \geq \gamma$ where the probability is computed ex ante (i.e., not conditioned on past broken wands). Note that γ is a parameter that is given. Let $F_{W_i}(\ell) = \mathbf{Pr}[W_i \leq \ell]$ denote the ex ante CDF of random variable W_i, and let Y_i be the indicator random variable which is 1 iff the magician opens the box i. Formally, the probability with which the magician should open box i condition on W_i is computed as follows.*

$$\mathbf{Pr}\left[Y_i = 1 | W_i\right] = \begin{cases} 1 & W_i < \theta_i \\ (\gamma - F_{W_i}(\theta_i - 1))/(F_{W_i}(\theta_i) - F_{W_i}(\theta_i - 1)) & W_i = \theta_i \\ 0 & W_i > \theta_i \end{cases} \qquad (y)$$

$$\theta_i = \min\{\ell | F_{W_i}(\ell) \geq \gamma\} \qquad (\theta)$$

Observe that θ_i is in fact computed before seeing box i itself.

Define $y_i^\ell = \mathbf{Pr}[Y_i = 1 | W_i = \ell]$; the CDF of W_{i+1} can be computed from the CDF of W_i and x_i as follows (assume x_i is the exact probability of breaking a wand for box i).

$$F_{W_{i+1}}(\ell) = \begin{cases} y_i^\ell x_i F_{W_i}(\ell - 1) + (1 - y_i^\ell x_i) F_{W_i}(\ell) & i \geq 1, \ell \geq 0 \\ 1 & i = 0, \ell \geq 0 \\ 0 & \text{otherwise.} \end{cases} \qquad (F_W)$$

Furthermore, if each x_i is just an upper bound on the probability of breaking a wand on box i, then the above definition of $F_{W_i}(\cdot)$ stochastically dominates the actual CDF of W_i, and the magician opens each box with a probability of at least γ.

In order to prove that a γ-conservative magician does not fail for a given choice of γ, we must show that the thresholds θ_i are no more than $k-1$. The following theorem states a condition on γ that is sufficient to guarantee that $\theta_i \le k-1$ for all i.

Theorem 24 (γ-Conservative Magician)**.** *For any $\gamma \le 1 - \frac{1}{\sqrt{k+3}}$, a γ-conservative magician with k wands opens each box with an ex ante probability of at least γ. Furthermore, if x_i is the exact probability (not just an upper bound) of breaking a wand on box i for each i, then each box is opened with an ex ante probability exactly γ*[8]

Proof. We defer the proof of this theorem to §5.2 where we present a more general variant of the magician's problem. □

Definition 19 (γ_k)**.** *We define γ_k to be the largest probability such that for any $k' \ge k$ and any instance of the magician's problem with k' wands, the thresholds computed by a γ_k-conservative magician are less than k'. In other words, γ_k is the optimal choice of γ which works for all instances with $k' \ge k$ wands. By Theorem 24, γ_k must be*[9] *at least $1 - \frac{1}{\sqrt{k+3}}$.*

Observe that γ_k is a non-decreasing function in k which is at least $\frac{1}{2}$ (when $k = 1$) and approaches 1 as $k \to \infty$. The next theorem shows that the lower bound of $1 - \frac{1}{\sqrt{k+3}}$ on γ_k cannot be far from the optimal.

Theorem 25 (Hardness of Magician's Problem)**.** *For any $\epsilon > 0$, it is not possible to guarantee an ex ante probability of $1 - \frac{k^k}{e^k k!} + \epsilon$ for opening each box (i.e., no magician can guarantee it). Note that $1 - \frac{k^k}{e^k k!} \approx 1 - \frac{1}{\sqrt{2\pi k}}$ by Stirling's approximation.*

Proof. See §4.8. □

[8] In particular the fact that the probability of the event of breaking a wand for the i^{th} box is exactly x_i, conditioned on any sequence of prior events, implies that these events are independent for different boxes.

[9] Because for any $k' \ge k$ obviously $1 - \frac{1}{\sqrt{k+3}} \le 1 - \frac{1}{\sqrt{k'+3}}$.

4.5 Generic Multi Agent Mechanisms

In this section, we present a formal description of the two generic multi agent mechanisms outlined toward the end of §4.3. Throughout the rest of this section we assume that for each agent $i \in [n]$ we can compute a single agent mechanism M_i and a corresponding concave benchmark R_i, which together provide α-approximation of the optimal single agent mechanism/optimal benchmark for agent i. We show that the resulting multi agent mechanism will be $\gamma_k\alpha$-approximation of the the optimal multi agent mechanism in M, where $k = \min_j k_j$ and γ_k is the optimal magician parameter which is at least $1 - \frac{1}{\sqrt{k+3}}$ (Definition 19) .

4.5.1 Pre-Rounding

This mechanism serves the agents sequentially (arbitrary order); for each agent i, it selects a subset S_i of available items and runs the single agent mechanism $M_i\langle\overline{x}_i[S_i]\rangle$, where $\overline{x}_i$ is an optimal assignment for the benchmark convex program ($\overline{OPT}$), and $\overline{x}_i[S_i]$ denotes the vector resulting from $\overline{x}_i$ by zeroing the entries corresponding to items not in S_i. In particular, this mechanism sometimes precludes some items from early agents to make them available to late agents. For each item, the mechanism tries to minimize the probability of preclusion for each agent by equalizing it for all agents. Note that, for any given pair of agent and item, we only care about the probability of preclusion in expectation, where the expectation is taken over the types of other agents and the random choices of the mechanism. The mechanism is explained in detail in Definition 20.

Definition 20 (γ-Pre-Rounding)**.**

(I) Solve the convex program ($\overline{OPT}$) and let $\overline{x}$ be an optimal assignment.

(II) For each item $j \in [m]$, create an instance of γ-conservative magician (Definition 18) with k_j wands (this will be referred to as the j^{th} magician). We will use these magicians through the rest of the mechanism. Note that γ is a parameter that is given.

(III) For each agent $i \in [n]$:

- *(a) For each item $j \in [m]$, write $\overline{x}_{ij}$ on a box and present it to the j^{th} magician. Let S_i be the set of items where the corresponding magicians opened the box.*
- *(b) Run $M_i\langle\overline{x}_i[S_i]\rangle$ on agent i and use its outcome as the final outcome for agent i.*
- *(c) For each item $j \in [m]$, if a unit of item j was allocated to agent i in the previous step, break the wand of the j^{th} magician.*

Note that since $\overline{x}$ is a feasible assignment for convex program ($\overline{OPT}$), it must satisfy $\sum_i \overline{x}_{ij} \leq k_j$, so by setting $\gamma \leftarrow \gamma_k$ and by Theorem 24 and Definition 19 we can argue that each S_i includes each item j with probability at least γ_k where γ_k is at least $1 - \frac{1}{\sqrt{k+3}}$.

In order for the above mechanism to retain at least a γ-fraction of the the expected objective value of each $M_i\langle\overline{x}_i\rangle$, further technical assumptions are needed in addition to $\gamma \leq \gamma_k$. We show that it is enough to assume each R_i has a budget-balanced and cross monotonic cost sharing scheme.

Definition 21 (Budget Balanced Cross Monotonic Cost Sharing Scheme)**.** *A function $R : [0, 1]^m \to \mathbb{R}_+$ has a budget balanced cross monotonic cost sharing scheme iff there exists a cost share function $\xi : [m] \times [0, 1]^m \to \mathbb{R}_+$ with the following two properties:*

(i) ξ must be budget balanced which means for all $x \in [0, 1]^m$ and $S \subseteq [m]$, $\sum_{j \in S} \xi(j, x[S]) = R(x[S])$.

(ii) ξ must be cross monotonic which means for all $x \in [0, 1]^m$, $j \in [m]$ and $S, S' \subseteq [m]$, $\xi(j, x[S]) \geq \xi(j, x[S \cup S'])$.

Intuitively, a cost share function associates a fraction of the expected objective value returned by the benchmark function R to each item; and ensures that the fraction associated with each item does not decrease when other items are excluded. In particular, the above assumption holds if $R(x[S])$ is a submodular function of S (e.g., for welfare maximization, assuming that agents' valuations are submodular[10]). Note that it is enough to show that such a cost sharing function exists; however it is never used in the mechanism and its computation is not required.

Theorem 26 (γ-Pre-Rounding)**.** *Suppose for each agent i, M_i is an α-approximate incentive compatible single agent mechanism, and R_i is the corresponding concave benchmark. Also suppose R_i has a budget balanced cross monotonic cost sharing scheme. Then, for any $\gamma \in [0, \gamma_k]$, the γ-pre-rounding mechanism (Definition 20) is dominant strategy incentive compatible (DSIC) mechanism which is in* M *and is a $\gamma\alpha$-approximation of the optimal mechanism in* M*.*

Proof. See §4.8. □

Remark 1. *The γ-pre-rounding mechanism assumes no control and no prior information about the order in which agents are visited. The order specified in the mechanism is arbitrary and could be replaced by any other ordering which may be unknown in advance. In particular, this mechanism can be adopted to online settings in which agents are served in an unknown order.*

[10]We conjecture that it holds in general for revenue maximization, when agents' valuations are submodular and M is restricted to mechanisms which use agent specific item pricing.

Corollary 4. *In any setting where Theorem 26 is applicable and when* M *includes all feasible BIC mechanisms, the gap between the optimal DSIC mechanism and the optimal BIC mechanism is at most* $1/\gamma_k$*. This gap is at most* 2 *(for* $k = 1$*) and vanishes as* $k \to \infty$*. That is because Definition 20 is always DSIC, yet it approximates the optimal mechanism in* M*.*

4.5.2 Post-Rounding

This mechanism runs $M_i\langle\overline{x}_i\rangle$ simultaneously and independently for all agents i to compute a tentative allocation/payment for each agent; it then deallocates some of the items at random to ensure that the supply constraints are met at every instance; it ensures that the probability of deallocation perceived by each agent (i.e., in expectation over the types of other agents and random choices of the mechanism) is equalized and therefore simultaneously minimized for all agents. The payments are also adjusted respectively. The mechanism is explained in detail in Definition 22.

Definition 22 (γ-Post-Rounding)**.**

(I) Solve the convex program ($\overline{OPT}$) and let $\overline{x}$ denote an optimal assignment.

(II) Run $M_i\langle\overline{x}_i\rangle$ simultaneously and independently for all agents $i \in [n]$, and let $\mathsf{x}'_i \subseteq [m]$ and $\mathsf{p}'_i \in \mathbb{R}_+$ denote respectively the allocation (subset of items) and payment computed by $M_i\langle\overline{x}_i\rangle$ for agent i.

(III) For each item $j \in [m]$, create an instance of γ-conservative magician (Definition 18) with k_j wands (this will be referred to as the j^{th} magician). We will use these magicians through the rest of the mechanism. Note that γ is a parameter that is given.

(IV) For each agent $i \in [n]$:

(a) For each item $j \in [m]$, write $\hat{x}_{ij}$ on a box and present it to the j^{th} magician, where $\hat{x}_{ij}$ is the exact probability[11] of $M_i\langle\overline{x}_i\rangle$ allocating a unit of item j to agent i; let S_i be the set of items where the corresponding magicians opened the box.

(b) Let $\mathsf{x}_i \leftarrow S_i \cap \mathsf{x}'_i$ and $\mathsf{p}_i \leftarrow \gamma\mathsf{p}'_i$. The final allocation and payment of agent i is given by x_i and p_i respectively.

(c) For each item $j \in \mathsf{x}_i$, break the wand of the j^{th} magician.

Note that $\sum_i \hat{x}_{ij} \leq \sum_i \overline{x}_{ij} \leq k_j$; so by setting $\gamma \leftarrow \gamma_k$ and by Theorem 24 and Definition 19 we can argue that each S_i includes each item j with probability at least γ_k where γ_k is at least $1 - \frac{1}{\sqrt{k+3}}$. Consequently, any item that is in x'_i will also be in x_i with probability exactly γ.

In order for γ-post-rounding to retain at least a γ-fraction of the the expected objective value of each $M_i\langle\overline{x}_i\rangle$ and preserve incentive compatibility, further technical assumptions are needed in addition to $\gamma \leq \gamma_k$; next, we present a set of assumptions which is sufficient for this purpose[12].

[11] Note that $\overline{x}_{ij}$ is only an upper bound on the probability of allocation, so $\hat{x}_{ij} \leq \overline{x}_{ij}$

[12] I.e., one might come up with other sets of assumptions that are also sufficient.

($A'1$) The exact ex ante allocation rule for each $M_i\langle \overline{x}_i \rangle$ (i.e., $\hat{x}$) must be available (i.e., efficiently computable). Note that $\overline{x}$ is only an upper bound on the ex ante allocation rule.

($A'2$) The objective functions must be of the form $\textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{p}_i) = \textsc{Obj}_i(t_i, \mathsf{x}_i, 0) + c_i \mathsf{p}_i$ in which $c_i \in \mathbb{R}_+$ is an arbitrary fixed constant. Also, each $\textsc{Obj}_i(t_i, \mathsf{x}_i, 0)$ must have cost sharing scheme in x_i which is cross monotonic and budget balanced.

($A'3$) The resulting mechanism must be in M. In particular, that implies M may not be restricted to any from of incentive compatibility stronger than Bayesian incentive compatibility (BIC), because the γ-post-rounding is only BIC.

($A'4$) The valuations of each agent must be in the form of a weighted rank function of some matroid.

Observe that $A'2$ obviously holds for revenue maximization (because $\textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{p}_i) = \mathsf{p}_i$), and also for welfare maximization with quasilinear utilities and submodular valuations (because $\textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{p}_i) = v_i(t_i, \mathsf{x}_i)$ where $v_i(t_i, \mathsf{x}_i)$ is the valuation of agent i for allocation x_i[13]). Next, we formally define $A'4$.

Definition 23 (Matroid Weighted Rank Valuation)**.** *A valuation function $v : 2^m \to \mathbb{R}_+$ is a matroid weighted rank valuation iff there exists a matroid $\mathcal{M} = ([m], \mathcal{I})$, and a weight function $w : [m] \to \mathbb{R}_+$ such that $v(S)$ is equal to the weight of a maximum weight independent subset of S, i.e,*

$$v(S) = \max_{I \in \mathcal{I} \cap 2^S} \sum_{j \in I} w(j), \qquad \forall S \subseteq [m]$$

Matroid weighted rank valuations include additive valuations with demand constraints, unit demand valuations, etc.

[13]Note that the payment terms cancel out because the utility of the auctioneer is counted toward the social welfare of the mechanism

Theorem 27 (γ-Post-Rounding)**.** *Suppose for each agent i, M_i is an α-approximate incentive compatible single agent mechanism, and R_i is a corresponding concave benchmark. Also suppose the assumptions $A'1$ through $A'4$ hold. Then, for any $\gamma \in [0, \gamma_k]$, the γ-post-rounding mechanism (Definition 22) is a Bayesian incentive compatible (BIC) mechanism which is in* M *and is a $\gamma\alpha$-approximation of the optimal mechanism in* M*.*

Proof. See §4.8. □

4.6 Single Agent Mechanisms

In this section, we present approximately optimal single agent mechanisms for several common settings. Each one of the single agent mechanisms presented in this section satisfies the requirements of one of the generic multi agent mechanisms of §4.5, so they can be readily converted to a multi agent mechanisms. Except for §4.6.4, we restrict the space of mechanisms to item pricing mechanisms with budget randomization as defined next.

Definition 24 (Item Pricing with Budget Randomization (IPBR))**.** *An item pricing mechanism is a possibly randomized mechanism that offers a menu of prices to each agent and allows each agent to choose their favorite bundle. The payment of an agent is equal to the total price of the items in her purchased bundle. Note that the prices offered to different agents do not need to be identical and agents can be served sequentially. In the presence of budget constraints, an agent is allowed to pay a fraction of the price of an item and receive the item with a probability equal to the paid fraction*[14]*. A mechanism is considered an item pricing mechanism if its outcome can be interpreted as such*[15]*.*

Item pricing mechanisms are simple and practical as opposed to optimal BIC

[14] A utility maximizing agent, with submodular valuations and budget constraint, always pays the full price for any item she purchases, except potentially for the last item purchased, for which she must have run out of budget.

[15] I.e., an item pricing mechanism may collect all the reports and compute the final outcome along with agent specific prices, such that the outcome of each agent would be the same as if each agent purchased their favorite bundle according to her observed prices, and the prices observed by each agent should be independent of her report.

mechanisms which often involve lotteries. Also budget randomization allows us to get around the hardness of the knapsack problem faced by the budgeted agents; in particular, assuming that budgets are large compared to prices, budget randomization can be safely ignored since the optimal integral solution of the knapsack problem approaches its optimal fractional solution.

Table 4.1 lists several settings for which we obtain a multi agent mechanism with an improved approximation factor compared to previous best known approximations. For each setting, we present a single agent mechanism that satisfies the requirements of one of the generic multi agent mechanisms of §4.5. The corresponding single agent mechanisms are presented in detail throughout the rest of this section. Note that the final approximation factor for each multi agent mechanism is equal to the approximation factor of the corresponding single agent mechanism multiplied by γ_k; recall that $\gamma_k \geq 1 - \frac{1}{\sqrt{k+3}}$ which approaches 1 as $k \to \infty$.

Setting	Approx	Space of Mechanisms	Ref
single item(multi unit), unit demand, budget constraint, revenue maximization	γ_k	item pricing with budget randomization	§4.6.1
multi item(heterogenous), unit demand, product distribution, revenue maximization	$\frac{1}{2}\gamma_k$	deterministic	§4.6.2
multi item(heterogenous), additive valuations, product distribution, budget constraint, revenue maximization	$(1 - \frac{1}{e})\gamma_k$	item pricing with budget randomization	§4.6.3
multi item(heterogenous), additive valuations, correlated distribution with polynomial number of types, budget constraint, matroid constrains, revenue or welfare maximization	γ_k	randomized (BIC)	§4.6.4

Table 4.1: Summary of mechanisms obtained using the current framework.

For each single agent mechanism presented in this chapter, the single agent benchmark function $R(x)$ is defined as the optimal value of some convex program of the following general form, in which u is some concave function, $g_j(\cdot)$ are some convex functions, and $\mathbb{Y}$ is some convex polytope (in the rest of this section we only consider a single agent, so we will omit the subscript i).

$$\begin{aligned} \text{maximize} \quad & u(y) && (\overline{OPT}_1) \\ \text{subject to} \quad & g_j(y) \le \overline{x}_j, \qquad \forall j \in [m] \\ & y \in \mathbb{Y} \end{aligned}$$

Lemma 5. *$R(\overline{x})$ is concave, i.e., the optimal value of a convex program of the form $(\overline{OPT}_1)$ is always concave in $\overline{x}$.*

Proof. See section 4.8. □

Note that we can substitute each $R_i(\cdot)$ in the multi agent benchmark convex program $(\overline{OPT})$ with the corresponding single agent benchmark convex program to obtain a combined convex program which can be solved efficiently. If each R_i is captured by a linear program, the combined multi agent program will also be a linear program.

4.6.1 Single Item, Unit Demand, Budget Constraint

In this section, we consider a unit-demand agent with a publicly known budget B and one type of item (i.e., $m = 1$). The only private information of the agent is her valuation for the item, which is drawn from a publicly known distribution with CDF $F(\cdot)$. To avoid complicating the proofs, we assume that $F(\cdot)$ is continuous and strictly increasing in its domain[16]. We present a single agent mechanism which is optimal among item pricing mechanisms with budget randomization (IPBR). We start by defining the modified CDF function $F^B(\cdot)$ as follows.

$$F^B(v) = \begin{cases} F(v) & v \le B \\ 1 - (1 - F(v))\frac{B}{v} & v \ge B \end{cases} \qquad (F^B)$$

Intuitively, $1 - F^B(p)$ is the probability of allocating the item to the agent if we offer the item at price p. Note that the agent only buys if her valuation is more than p

[16]The proofs can be modified to work without this assumption.

which happens with probability $1-F(p)$; if $p > B$, she will pay her whole budget and only get the item with probability $\frac{B}{p}$, otherwise she pays the full price and receives the item with probability 1. Observe that if we want to allocate the item with probability x we can offer a price of $F^{B^{(-1)}}(1-x)$ which yields a revenue of $xF^{B^{(-1)}}(1-x)$ in expectation. Define $\mathfrak{R}(x) = xF^{B^{(-1)}}(1-x)$ and let $\widehat{\mathfrak{R}}(x)$ denote its concave closure (i.e., the smallest concave function that is an upper bound on $\mathfrak{R}(x)$ for every x). We will address the problem of efficiently computing $\widehat{\mathfrak{R}}(x)$ later in Lemma 6. Next, we show that the optimal value of the following convex program is equal to the expected revenue of the optimal single agent IPBR mechanism subject to $\overline{x}$; therefore we will define the single agent benchmark function $R(\overline{x})$ to be equal to the optimal value of this program as a function of $\overline{x}$.

$$\begin{aligned} \text{maximize} \quad & \widehat{\mathfrak{R}}(x) \\ \text{subject to} \quad & x \leq \overline{x} \\ & x \geq 0 \end{aligned} \tag{$\textsc{Rev}_{single}$}$$

Theorem 28. *The revenue of the optimal single agent IPBR mechanism, subject to an upper bound of $\overline{x}$ on the ex ante allocation rule, is equal to the optimal value of the convex program* ($\textsc{Rev}_{single}$)*. Furthermore, assuming that x^* is the optimal assignment for the convex program, if $\widehat{\mathfrak{R}}(x^*) = \mathfrak{R}(x^*)$, then the optimal mechanism uses a single price $p = F^{B^{(-1)}}(1-x^*)$ otherwise, it randomized between two prices p^-, p^+ with probabilities θ and $1-\theta$ for some $\theta \in [0,1]$ and p^-, p^+.*

Proof. First, we prove that the expected revenue of the optimal single agent IPBR mechanism, subject to $\overline{x}$, is upper bounded by $\widehat{\mathfrak{R}}(x^*)$. We then construct a price distribution that obtains this revenue. Note that any single agent IPBR mechanism can be specified as a distribution over prices. Let $\mathcal{P}$ be the optimal price distribution. So the optimal revenue is $\mathbf{E}_{p\sim\mathcal{P}}[p(1-F^B(p))]$. Note that every price p corresponds to an allocation probability $x = 1 - F^B(p)$. So any probability distribution over p can be specified as a probability distribution over x. Let $\mathcal{Q}$ denote the probability distribution over x that corresponds to price distribution $\mathcal{P}$, so we can write

$$\begin{aligned}
\text{optimal revenue} &= \mathbf{E}_{x\sim\mathcal{Q}}\left[xF^{B^{(-1)}}(1-x)\right] = \mathbf{E}_{x\sim\mathcal{Q}}\left[\mathfrak{R}(x)\right] \\
&\leq \mathbf{E}_{x\sim\mathcal{Q}}\left[\widehat{\mathfrak{R}}(x)\right] \leq \widehat{\mathfrak{R}}(\mathbf{E}_{x\sim\mathcal{Q}}\left[x\right]) \qquad \text{By Jensen's inequality}
\end{aligned}$$

which means the optimal revenue is upper bounded by the value of the convex program for $x = \mathbf{E}_{x\sim\mathcal{Q}}[x]$ [17]; so the optimal revenue is upper bounded by the optimal value of the convex program. That completes the first part of the proof.

Next, we construct an optimal price distribution. If $\widehat{\mathfrak{R}}(x^*) = \mathfrak{R}(x^*)$, the optimal price distribution is just a single price $p = F^{B^{(-1)}}(1-x^*)$; otherwise, by definition of concave closure, there are two points x^- and x^+ and $\theta \in [0,1]$ such that $x^* = \theta x^- + (1-\theta)x^+$ and $\widehat{\mathfrak{R}}(x^*) = \theta\mathfrak{R}(x^-) + (1-\theta)\mathfrak{R}(x^+)$. In the latter case, the optimal price distribution offers price $p^- = F^{B^{(-1)}}(1-x^-)$ with probability θ and offers price $p^+ = F^{B^{(-1)}}(1-x^+)$ with probability $1-\theta$. □

Formally, an optimal single agent IPBR mechanism can be constructed as follows.

Definition 25 (Mechanism)**.**

- *Define the single agent benchmark $R(\overline{x})$ to be the optimal value of the convex program* (Rev$_{single}$) *as a function of $\overline{x}$.*
- *Given $\overline{x}$, solve* (Rev$_{single}$) *and let x be an optimal assignment.*
- *If $\widehat{\mathfrak{R}}(x) = \mathfrak{R}(x)$, offer the single price $p = F^{B^{(-1)}}(1-x)$, otherwise randomize between two prices p^- and p^+ as explained in the proof of Theorem 28.*

Theorem 29. *The mechanism of Definition 25 is the optimal revenue maximizing single agent IPBR mechanism. Furthermore, this mechanism satisfies the requirements of γ-pre-rounding.*

Proof. The proof of the optimality follows from Theorem 28. Furthermore, the benchmark function, $R(x)$, is concave (this follows from Lemma 5) and it has a trivial budget

[17] Note that $\mathbf{E}_{x\sim\mathcal{Q}}[x]$ is exactly the probability of allocating the item by the price distribution $\mathcal{P}$, so it must be no more than $\overline{x}$

balanced cost sharing scheme (because there is only one item), therefore it meets the requirements of γ-pre-rounding. □

Next, we address the problem of efficiently computing $\widehat{\mathfrak{R}}(\cdot)$.

Lemma 6. *A* $(1+\epsilon)$*-approximation of* $\widehat{\mathfrak{R}}(\cdot)$*, which we denote by* $\widehat{\mathfrak{R}}_{1+\epsilon}(\cdot)$*, can be constructed using a piece-wise linear function with* $\ell = \frac{\log L}{\log(1+\epsilon)}$ *pieces and in time* $O(\ell \log \ell)$ *in which* L *is the ratio of the maximum valuation to minimum non-zero valuation. Note that we need at least* $\log_2 L$ *bits just to represent such valuations so this construction is polynomial in the input size for any constant* ϵ*.*

Proof. WLOG, assume that all possible non-zero valuations of the agent are in the range of $[1, L]$. Let $\ell = \lfloor \frac{\log L}{\log(1+\epsilon)} \rfloor$. For $r = 0 \cdots \ell$, consider the prices $p_r = (1+\epsilon)^{\ell - r}$ and compute the corresponding $x_r = 1 - F^B(p_r)$. Construct $\widehat{\mathfrak{R}}_{1+\epsilon}(\cdot)$ by constructing the convex hall of the points:
$(0,0), (x_1, p_1 x_1), (x_2, p_2 x_2), \ldots, (x_\ell, p_\ell x_\ell), (1, 0)$. This can be done in time $O(\ell \log \ell)$. Note that $F^{B(-1)}(1-x)$ is a decreasing function of x so at every $x \in [x_r, x_{r+1}]$, the corresponding price is $F^{B(-1)}(x) \in [p_{r+1}, p_r]$ but $p_r = (1+\epsilon) p_{r+1}$ therefore at every x, $\mathfrak{R}_{1+\epsilon}(x) \leq \widehat{\mathfrak{R}}(x) \leq (1+\epsilon)\mathfrak{R}_{1+\epsilon}(x)$ which completes the proof. □

Remark 2. *In order to use* $\mathfrak{R}_{1+\epsilon}(\cdot)$ *in the single agent mechanism of Definition 25, we need to substitute* $(1+\epsilon)\widehat{\mathfrak{R}}_{1+\epsilon}(\cdot)$ *in the objective function of the convex program* ($\textsc{Rev}_{single}$) *instead of* $\widehat{\mathfrak{R}}(\cdot)$ *for computing the benchmark. Furthermore, the mechanism will be a* $(1-\epsilon)$*-approximation of the optimal single agent IPBR mechanism. Also notice that finding* p^- *and* p^+ *from* $\mathfrak{R}_{1+\epsilon}(\cdot)$ *is trivial.*

4.6.2 Multi Item (Independent), Unit Demand

In this section, we consider a unit demand agent with private independent valuations for m items. We assume that for each item j, the agent's valuation is distributed independently according to a publicly known distribution with CDF $F_j(\cdot)$. We present a single agent mechanism which is a $\frac{1}{2}$-approximation of the optimal deterministic revenue maximizing mechanism. To avoid complicating the proofs, we assume that each $F_j(\cdot)$ is continuous and strictly increasing in its domain. Furthermore, we require the distributions to be regular. This mechanism can be used with γ-pre-rounding (Definition 20) to yield a $\frac{1}{2}\gamma_k$-approximate sequential posted pricing multi agent mechanism. The previous best approximation mechanism for this setting was a $\frac{1}{6.75}$-approximate sequential posted pricing mechanism by Chawla et al. (2010)[18].

We start by defining $\mathfrak{R}_j(x) = xF_j^{-1}(1-x)$ for each item j. Because $F_j(\cdot)$ is corresponds to a regular distribution, $\mathfrak{R}_j(\cdot)$ is concave as shown in the following lemma.

Lemma 7. *If $F(\cdot)$ is the CDF of a regular distribution, the function $\mathfrak{R}(x) = xF^{-1}(1-x)$ is concave.*

Proof. It is enough to show that $\frac{\partial}{\partial x}\mathfrak{R}(x)$ is non-increasing in x. Observe that $\frac{\partial}{\partial x}\mathfrak{R}(x) = F^{-1}(1-x) - \frac{x}{f(F^{-1}(1-x))}$ in which $f(\cdot)$ is the derivative of $F(\cdot)$. By substituting $x = 1 - F(p)$, it is enough to show that the resulting function is non-decreasing in p because x is itself non-increasing in p. However, by this substitution we get $\frac{\partial}{\partial x}\mathfrak{R}(x) = p - \frac{1-F(p)}{f(p)}$ which is non-decreasing in p by definition of regularity. $\square$

Note that any deterministic mechanism for a unit demand agent can be interpreted as item pricing. Consequently, $\mathfrak{R}_j(x_j)$ is the maximum revenue that such a mechanism can obtain if item j is allocated with probability x_j. Next, we show that the following convex program gives an upper bound the on the expected optimal revenue.

$$
\begin{aligned}
\text{maximize} \quad & \sum_j \mathfrak{R}_j(x_j) && && (\textsc{Rev}_{unit}) \\
\text{subject to} \quad & x_j \le \overline{x}_j, && \forall j \in [m] && (\lambda_j) \\
& \sum_j x_j \le 1 && && (\tau) \\
& x_j \ge 0, && \forall j \in [m] && (\mu_j)
\end{aligned}
$$

Theorem 30. *The revenue of the optimal deterministic single agent mechanism, subject to an upper bound of $\overline{x}$ on the ex ante allocation rule, is no more than the*

[18]Note that the mechanism of Chawla et al. (2010) does not work for non-regular distributions despite the authors' claim.

optimal value of the convex program (REV$_{unit}$).

Proof. Let x^* be the ex ante allocation rule of the optimal single agent deterministic mechanism. So the expected revenue obtained from each item j is upper bounded by $\mathfrak{R}_j(x_j^*)$ (proof of this claim is essentially the same as the proof of Theorem 28). Consequently, the expected optimal revenue cannot be more that $\sum_j \mathfrak{R}_j(x_j^*)$. Furthermore, the optimal mechanism never allocates more than one item, so $\sum_j x_j^* \leq 1$, and also $x_j^* \leq \overline{x}_j$; therefore x^* is a feasible solution for the convex program; so the expected optimal revenue is upper bounded by the optimal value of the convex program. □

Next, we present the single agent mechanism.

Definition 26 (Mechanism)**.**

- *Define the benchmark* $R(\overline{x})$ *to be the optimal value of* (REV$_{unit}$) *as a function of* $\overline{x}$.
- *Given* $\overline{x}$, *solve* (REV$_{unit}$) *and let* x *denote an optimal assignment.*
- *For each item* j, *assign the price* $p_j = F_j^{-1}(1 - x_j)$. *WLOG, assume that items are indexed in non-decreasing order of prices, i.e.,* $p_1 \leq \ldots \leq p_m$.
- *For each item* j, *define* $r_j = \max(x_j p_j + (1 - x_j) r_{j+1}, r_{j+1})$ *and let* $r_{m+1} = 0$. *Let* S^* *be the subset of items defined as* $S^* = \{j | p_j \geq r_{j+1}\}$.
- *Only offer the items in* S^* *at prices computed in the previous step (i.e., set the price of other items to infinity).*

Theorem 31. *The mechanism of Definition 26 obtains at least* $\frac{1}{2}$ *of the revenue of the optimal deterministic single agent mechanism in expectation. Furthermore, it satisfies the requirements of* γ*-pre-rounding.*

Proof. First, we show that this mechanism obtains in expectation at least $\frac{1}{2}$ of its benchmark $R(\overline{x})$, which by Theorem 30 is an upper bound on the optimal revenue. Observe that $R(\overline{x}) = \sum_j x_j p_j$ where x_j is exactly the probability that the valuation of the agent for item j is at least p_j. Now consider an "adversary replica" who has the exact same valuations as the original agent, but always buys the item that has the lowest price among all the items priced below her valuation. For any assignment of prices, the revenue obtained from the adversary replica is a lower bound on the revenue obtained from the original agent. So it is enough to show that the mechanism obtains a revenue of at least $\frac{1}{2}\sum_j x_j p_j$ from the adversary replica. Observe that r_j is exactly the expected revenue obtained from the adversary replica when offered the items in $S^* \cap \{j, \ldots, m\}$. In particular, item j is included in S^* if $p_j \geq R_{j+1}$, which implies that the revenue obtained from the purchase of item j, conditioned on purchase, is more than the lower bound on the expected revenue obtained from items $\{j, \ldots, m\}$. Finally, observe that the expected revenue obtained from the adversary replica is exactly r_1. By Lemma 8 we can conclude that $r_1 \geq \frac{1}{2}\sum_j x_j p_j$ which completes the proof of the first claim.

Next, we show that this mechanism satisfies the requirements of γ-pre-rounding. Observe that by Lemma 5, the optimal value of (Rev$_{unit}$) is a concave function of $\overline{x}$; so $R(\overline{x})$ is concave. It only remains to show that $R(\cdot)$ has a budget balanced cross monotonic cost sharing scheme. Let $x_j(\overline{x})$ denote the optimal assignment of variable x_j, in the convex program (Rev$_{unit}$), as a function of $\overline{x}$. Define the cost share function

$$\xi(j, \overline{x}) = \mathfrak{R}_j(x_j(\overline{x})).$$

We shall show that ξ is budget balanced and cross monotonic (see Definition 21).

- **Budget balance.** We shall show that for any $\overline{x} \in [0,1]^m$ and any $S \subseteq [m]$, $R(\overline{x}[S]) = \sum_{j \in S} \xi(j, \overline{x}[S])$. Note that $R(\overline{x}[S]) = \sum_j \mathfrak{R}_j(x_j(\overline{x}[S])) = \sum_{j \in S} \xi(j, x_j(\overline{x}[S]))$ which proved that ξ is budget balanced. Note that $\mathfrak{R}_j(x_j(\overline{x}[S])) = 0$, for any $j \notin S$, because $x_j(\overline{x}[S])$ is forced to be 0.

- **Cross monotonicity.** We shall show that $\xi(j, \overline{x}[S]) \geq \xi(j, \overline{x}[S \cup S'])$, for any $\overline{x} \in [0,1]^m$ and any $S, S' \subseteq [m]$. Let the Lagrangian of (Rev$_{unit}$) be defined as follows.

$$L(x, \lambda, \tau, \mu) = -\sum_j \mathfrak{R}_j(x_j) + \sum_j \lambda_j \ (x_j - \overline{x}_j) + \tau \ (\sum_j x_j - 1) - \sum_j \mu_j x_j$$

The high level idea of the proof is as follows. We show that there is more pressure on the constraint associated with τ when the set of available items is $S \cup S'$ instead of S (i.e., τ is larger for $S \cup S'$); we then show that the optimal x_j can be determined from τ; in particular, we show that, as the optimal τ increases, the optimal x_j decreases, and consequently $\xi(j, x)$ (which is equal to $\mathfrak{R}_j(x_j)$) decreases as well, which proves ξ is cross monotonic. Next we present the proof in detail.

By KKT stationarity conditions, at the optimal assignment the following holds.

$$\frac{\partial}{\partial x_j} L(x, \lambda, \tau, \mu) = -\frac{\partial}{\partial x_j} \mathfrak{R}_j(x_j) + \lambda_j + \tau - \mu_j = 0$$

First we show that the optimal x_j, and consequently $\xi(j, \overline{x})$, can be determined from the optimal τ; and they are both non-increasing in τ. Observe that (a) all dual variables must be non-negative, (b) by complementary slackness λ_j may be non-zero only if $x_j = \overline{x}_j$, and (c) complementary slackness implies that μ_j may be non-zero only if $x_j = 0$; therefore, if the optimal τ is given, the optimal assignment for x_j is uniquely[19] determined by the above equation and the aforementioned complementarity slackness conditions. Let $x_j(\tau)$ denote the optimal assignment of x_j as a function of τ. Due to the concavity of $\mathfrak{R}_j(\cdot)$, and the above KKT condition, we can argue that $x_j(\tau)$ is non-increasing in τ, which also implies that $\xi(j, \overline{x})$ is non-increasing in τ.

Next, we prove by contradiction that ξ is cross monotonic. Let $\tau(\overline{x})$ denote the optimal assignment of τ as a function of $\overline{x}$. By contradiction, suppose ξ is not cross monotonic, i.e. $\xi(j^*, \overline{x}[S \cup S']) > \xi(j^*, \overline{x}[S])$ for some item j^*; therefore $\tau(\overline{x}[S]) > \tau(\overline{x}[S \cup S']) \geq 0$. Since $\tau(\overline{x}[S]) > 0$, the inequality associated with τ must be tight (by complementary slackness), so $\sum_j x_j(\tau(\overline{x}[S])) = 1$. On the other hand, for all j, $x_j(\tau(\overline{x}[S \cup S'])) \geq x_j(\tau(\overline{x}[S]))$, with the inequality being strict for $j = j^*$, which means $\sum_j x_j(\tau(\overline{x}[S \cup S'])) > 1$, which is a contradiction.

□

[19] To avoid complicating the proof, we assume that the functions $\mathfrak{R}_j(\cdot)$ are strictly concave, however this assumption is not necessary.

Lemma 8. *Let $p_1, \ldots, p_m$ and $x_1, \ldots, x_m$ be two sequences of non-negative real numbers and suppose $\sum_j x_j \leq 1$. For each $j \in [m]$, define $r_j = \max(x_j p_j + (1 - x_j) r_{j+1}, r_{j+1})$ and let $r_{m+1} = 0$. Then $r_1 \geq \frac{1}{2} \sum_j x_j p_j$.*

Proof. See section 4.8. □

4.6.3 Multi Item (Independent), Additive, Budget Constraint

In this section, we consider an agent with publicly known budget B who has private independent and additive valuations for m items (i.e., her valuation for a bundle of items is the sum of her valuations for individual items in the bundle). We assume the agent's valuation for each item j is distributed independently according to a publicly known distribution with CDF $F_j(\cdot)$. To avoid complicating the proofs, we assume that each $F_j(\cdot)$ is continuous and strictly increasing in its domain[20]. We present a single agent mechanism which is a $(1 - \frac{1}{e})$-approximation of the optimal revenue maximizing item pricing mechanism with budget randomization (IPBR). This mechanism can be used with γ-pre-rounding (Definition 20) to yield a $(1 - \frac{1}{e})\gamma_k$-approximate sequential posted pricing multi agent mechanism. The previous best approximation mechanism for this setting was an $O(1)$-approximate[21] sequential posted pricing mechanism by Bhattacharya et al. (2010). We should note that the mechanism in Bhattacharya et al. (2010) is more general as it allows the agents to have demand constraints as well, and it does not allow for budget randomization.

As in §4.6.1, we start by defining the modified CDF function $F_j^B(\cdot)$ for each item j as follows.

$$F_j^B(v) = \begin{cases} F_j(v) & v \leq B \\ 1 - (1 - F_j(v))\frac{B}{v} & v \geq B \end{cases} \tag{F_j^B}$$

Furthermore, for each item j, let $\mathfrak{R}_j(x) = x {F_j^B}^{-1}(1 - x)$ and let $\widehat{\mathfrak{R}}_j(\cdot)$ be its concave closure as define in §4.6.1. Also, for each j, define $R_j(\overline{x}_j)$ to be the optimal

[20] The proofs can be modified to work without this assumption.

[21] $\frac{1}{96}$

value of the following convex program as a function of $\overline{x}_j$.

$$\begin{aligned} \text{maximize} \quad & \mathfrak{R}_j(x_j) \\ \text{subject to} \quad & x_j \leq \overline{x}_j \\ & x_j \geq 0 \end{aligned} \qquad (\text{REV}_{add})$$

The next theorem provides an upper bound on the revenue of the optimal single agent IPBR mechanism.

Theorem 32. *The revenue of the optimal single agent item pricing mechanism with budget randomization (IPBR), subject to an upper bound of $\overline{x}$ on the ex ante allocation rule, is no more than* $\min(\sum_j R_j(\overline{x}_j), B)$, .

Proof. For any j, if we were only to sell the item j, by Theorem 28, the maximum revenue we could obtain using an IPBR mechanism would be no more than $R_j(\overline{x}_j)$. Observe that if we compute the optimal price distribution for each item separately, we might only get less revenue because the budget is shared among all items and the agent might not be able to buy some of the items that she would otherwise buy if there were no other items. That means the actually probability of allocating each item j could be less than the optimal assignment of x_j for the convex program (REV_{add}); so the optimal joint price distribution might sell at lower prices; but the extra revenue may only come from lower types which were originally excluded by the optimal single item mechanism. Consequently, the overall revenue from each item j cannot be more than $R_j(\overline{x}_j)$. Finally, observe that the expected revenue of the mechanism cannot be more that B, so it can be no more than $\min(\sum_j R_j(x_j), B)$. □

Next, we present $(1 - \frac{1}{e})$-approximate revenue maximizing single agent IPBR mechanism.

Definition 27 (Mechanism)**.**

- *Define the benchmark* $R(\overline{x}) = \min(\sum_j R_j(\overline{x}_j), B)$.
- *Given* $\overline{x}$*, solve the convex program of* (REV$_{add}$) *for each item* j*, and let* x_j *denote an optimal assignment.*
- *For each item* j*, if* $\widehat{\mathfrak{R}}_j(x_j) = \mathfrak{R}_j(x_j)$*, offer the single price* $p_j = F_j^{B(-1)}(1 - x_j)$*, otherwise randomize between two prices* p_j^- *and* p_j^+ *with probabilities* θ_j *and* $1 - \theta_j$*, as explained in Theorem 28. Note that the randomization must be done for each item independently.*

Theorem 33. *The mechanism of Definition 27 obtains at least* $1 - \frac{1}{e}$ *of the revenue of the optimal single agent IPBR mechanism. Furthermore, this mechanism satisfies the requirements of* γ*-pre-rounding.*

Proof. First, we show that the mechanism obtains at least $1 - \frac{1}{e}$ of its benchmark $R(\overline{x})$, which by Theorem 32 is an upper bound on the optimal revenue. Consider an imaginary replica of the agent who has exactly the same valuations as the original agent, but has a separate budget B for each item. We call this imaginary agent the "super replica". Furthermore, suppose that any payment received from the super replica beyond B is lost (i.e., if the super replica pays Z, the mechanism receives only $\min(Z, B)$). Observe that for any assignment of prices, the payment received from the original agent and the payment received from the super replica are exactly the same because if the original agent has't hit his budget limit then both the original agent and the super replica will buy the same items and pay the exact same amount. Otherwise, if the original agent hits his budget limit, the mechanism receives exactly B from both the original agent and the super replica; therefore we only need to show that the revenue obtained by the mechanism from the super replica is at least $(1 - \frac{1}{e})R(\overline{x})$. Observe that from the view point of the super replica there is no connection between different items, so he makes a decision for each item independently. Let Z_j be the random variable corresponding to the amount paid by the super replica for item

j. By Theorem 28, we know that $\mathbf{E}[Z_j] = R_j(\overline{x}_j)$ and the total revenue received by the mechanism is $Z = \min(\sum_j Z_j, B)$. Notice that $Z_1, \ldots, Z_m$ are independent random variables in the range of $[0, B]$. By applying Lemma 9, we can argue that $\mathbf{E}[\min(\sum_j Z_j, B)] \geq (1 - \frac{1}{e}) \min(\sum_j \mathbf{E}[Z_j], B) = (1 - \frac{1}{e}) R(\overline{x})$ which proves our claim.

Next, we show that the mechanism satisfies the requirements of γ-pre-rounding. Observe that all $R_j(\cdot)$ are concave, and so is $R(\overline{x})$. Furthermore, $R(\overline{x}[S]) = \min(\sum_{j \in S} R_j(\overline{x}_j), B)$ is submodular in S for any $S \subseteq [m]$, and therefore it has a cross monotonic budget balanced cost share scheme (see Definition 21), which completes the proof. □

Lemma 9. *Let B be an arbitrary positive number and let $Z_1, \ldots, Z_m$ be independent random variables such that $Z_j \in [0, B]$, for all j. Then the following inequality holds.*

$$\mathbf{E}\left[\min(\sum_j Z_j, B)\right] \geq (1 - \frac{1}{e^{(\sum_j \mathbf{E}[Z_j])/B}})B \geq (1 - \frac{1}{e}) \min(\sum_j \mathbf{E}\left[Z_j\right], B)$$

Proof. See section 4.8. □

4.6.4 Multi Item (Correlated), Additive, Budget and Matroid Constraints

In this section, we consider an agent with publicly known budget B who has private correlated additive valuations for m items; furthermore, a bundle of items can be allocated to the agent only if it is an independent set of a matroid $\mathcal{M} = ([m], \mathcal{I})$, where $\mathcal{M}$ is publicly known; equivalently, instead of treating $\mathcal{M}$ as a constraint on the allocation, we may assume that the agent has matroid valuations, as defined in Definition 23. We assume that the agent has a discrete type space T. Let $v_t \in \mathbb{R}_+^m$ denote the agent's valuation vector corresponding to type $t \in \mathrm{T}$, and let $f(t)$ denote its probability. We assume that $f(\cdot)$ is represented explicitly as a part of the input, i.e., by enumerating all types along with their respective probabilities. The only private information of the agent is her type. We present an optimal single agent randomized mechanism. This mechanism can be used with γ-post-rounding (Definition 22) to yield a γ_k-approximate multi agent BIC mechanism. Recall that γ_k is at least $\frac{1}{2}$, and approaches 1 as $k \to \infty$, which means the resulting multi agent mechanism approaches the optimal multi agent mechanism as $k \to \infty$. Prior to the preliminary version of this work, the best approximation for this setting was a $\frac{1}{4}$-approximate BIC mechanism by Bhattacharya et al. (2010)[22]. At the time of writing the current version, Henzinger and Vidali (2011) has also presented a $\frac{1}{2}$-approximate BIC mechanism for the same setting. Note that all of the aforementioned mechanisms (including the one presented here) have running times polynomial only in $|\mathrm{T}|$, which means their running time may not be polynomial in the input size if $|\mathrm{T}|$ is of exponential size and $f(\cdot)$ has a compact representation.

Consider the following linear program in which $x_t \in [0,1]^m$ represents the marginal allocation probabilities for type $t \in \mathrm{T}$, and p_t represents the corresponding payment. Also let $r_{\mathcal{M}} : 2^m \to \{0, \ldots, m\}$ denote the rank function of $\mathcal{M}$. The optimal value of this LP is obviously an upper bound on the optimal revenue.

$$
\begin{array}{lll}
\text{maximize} & \sum_{t \in \mathrm{T}} f(t) p_t & (\textsc{Rev}_{corr}) \\
\text{subject to} & \sum_{t \in \mathrm{T}} f(t) x_{tj} \le \overline{x}_j, & \forall j \in [m] \\
& \sum_{j \in S} x_{tj} \le r_{\mathcal{M}}(S), & \forall t \in \mathrm{T}, \forall S \subseteq [m] \\
& v_t \cdot x_t - p_t \ge v_t \cdot x_{t'} - p_{t'}, & \forall t, t' \in \mathrm{T} \\
& x_t \in [0,1]^m, & \forall t \in \mathrm{T} \\
& p_t \in [0, B], & \forall t \in \mathrm{T}
\end{array}
$$

Even though the above LP has exponentially many constraints, it can be solved in polynomial time using the ellipsoid method[23]. Next, we present a mechanism whose expected revenue is equal to the optimal value of the above LP, which also implies

[22] The mechanism in Bhattacharya et al. (2010) considers demand constraint, which is a special case of matroid constraints.

[23] See Schrijver (2003) for optimization over matroid polytope.

that it is optimal.

Definition 28 (Mechanism)**.**

- *Define the optimal benchmark* $R(\overline{x})$ *to be the optimal value of* ($\textsc{Rev}_{corr}$) *as a function of* $\overline{x}$*.*
- *Given* $\overline{x}$*, solve the LP of* ($\textsc{Rev}_{corr}$) *and let* x *an* p *be an optimal assignment.*
- *Let* t *be the agent's reported type. Allocate a random subset* $\mathsf{x} \subseteq [m]$ *of items such that* x *is an independent set of* $\mathcal{M}$ *and each item* $j \in [m]$ *is included in* x *with a marginal probability of exactly* x_{tj}*. This can be archived by rounding* x_t *to a vertex of the matroid polytope using dependent randomized rounding (see Chekuri et al. (2010) and the references therein). Also charge a payment of* p_t*.*

Theorem 34. *The mechanism of Definition 28 is an optimal truthful in expectation revenue maximizing single agent mechanism, subject to an upper bound of* $\overline{x}$ *on the ex ante allocation rule. Furthermore, it satisfies all the requirements of the* γ*-post-rounding.*

Proof. The proof of truthfulness and optimality trivially follows from the linear program of ($\textsc{Rev}_{corr}$). So, we only focus on proving that this mechanism satisfies the requirements of Theorem 27. First, observe that the benchmark function, $R(\overline{x})$, is concave (this follows from Lemma 5). Second, observe that the matroid constrains can be interpreted as matroid valuations for the agent. Third, notice that the exact ex ante allocation rule can be readily computed from the LP solution, i.e., $\hat{x}_j = \sum_t f(t)x_{tj}$ is the exact probability of allocating item j. Therefore, the mechanism satisfies the requirements of γ-post-rounding. □

Remark 3. *Observe that by replacing the objective function of* ($\textsc{Rev}_{corr}$) *with* $\sum_{t\in\mathrm{T}} f(t)v_t \cdot x_t$*, we get a truthful in expectation welfare maximizing single agent mechanism, which can also be used with* γ*-post-rounding to obtain a* γ_k*-approximate welfare maximizing BIC multiple agent mechanism.*

4.7 Multi Unit Demands

In this section, we show that the more general model, in which each agent may need more than one unit but no more than $\frac{1}{k}$ of all units of each item, can be reduced to the simpler model in which there are at least k units of every item and no agent demands more than 1 unit of each item.

Definition 29 (Multi Unit Demand Market Transformation). *Let k_j denote the number of units of item j. Define $c_j = \lfloor \frac{k_j}{k} \rfloor$ and divide the units of item j almost equally into c_j bins (i.e., each bin will contain either c_j or $c_j + 1$ units). Create a new item type for each bin (i.e., units from the same bin has the same type, but units from different bins are treated as different types of item).*

Theorem 35. *Let M be the space of feasible mechanisms, in the original (multi unit demand) market, which do not allocate more than $\frac{1}{k}$ of all units of each item to any single agent. Similarly, let $\mathrm{M}^{(1)}$ be the space of feasible mechanisms, in the transformed market, which do not allocate more than one unit of each item to any single agent. Any mechanism in M can be interpreted as a mechanism in $\mathrm{M}^{(1)}$ and vice-versa with the same allocations/payments. Therefore, in order to find the optimal mechanism in the original market, it is enough to find the optimal mechanism in the transformed market.*

Proof. First, we show that any mechanism in $M \in \mathrm{M}^{(1)}$ can be interpreted as a mechanism in M. That is trivially true because M allocates to each agent at most one unit from each bin, which is at most c_j units of each item j of the original market, which is no more than $\frac{1}{k}$ of all units of item j.

Next, we show that any mechanism $M \in \mathrm{M}$ can be interpreted as a mechanism in $\mathrm{M}^{(1)}$. For every j, we create a list L_j of all the bins of item j. L_j is initially sorted in decreasing order of the size of the bins. Let x_{ij}^{M} be the number of units of item j allocated to agent i by M. We specify the allocations in the transformed market as follows. For each agent i we repeat the following, x_{ij}^{M} times: Allocate one unit from the bin that is first in the list L_j and then move the bin back to the end of the list.

It is easy to see that no two units from the same bin are allocated to the same agent, which completes the proof. □

Note that by Theorem 35, any mechanism in the original market is equivalent to a mechanism in the transformed market with the exact same allocations/payments from the perspective of agents. Therefore, WLOG, we can work with the transformed market and only consider mechanisms in this market. However, to use our generic multi agent mechanisms in the transformed market, the underlying single agent mechanisms should be capable of handling correlated valuations, because units of the same item, even when labeled with different types, are perfect substitutes from the view point of an agent. Among the single agent mechanisms presented in this chapter, only the mechanism explained in §4.6.4 can handle correlated valuations.

4.8 Omitted Proofs

Proof of Theorem 25. Suppose we create n boxes and in each box, independently, we put \$1 with probability $\frac{k}{n}$. If the magician opens a box containing a \$1, then he gets the \$1 but we break his wand (i.e., $x_i = \frac{k}{n}$). Observe that the expected total prize is k dollars, but because we put a dollar in each box independently, there are some instances in which there are more than k non-empty boxes but the magician cannot win more than k dollars at any instance. Let X_i be the indicator random variable which is 1 iff there is a dollar in box i. The expected total prize is $\mathbf{E}[\sum_i X_i] = k$, but the expected prize that the magician can win is at most $\mathbf{E}[\min(\sum_i X_i, k)]$. It can be verified that $\mathbf{E}[\min(\sum_i X_i, k)] \approx (1-\frac{k^k}{e^k k!})k$ asymptotically as $n \to \infty$. In fact, for any positive ϵ, there is a large enough n such that $\mathbf{E}[\min(\sum_i X_i, k)] < (1-\frac{k^k}{e^k k!}+\epsilon)k$. On the other hand, if a magician can guarantee that every box is opened with probability at least $\gamma = 1-\frac{k^k}{e^k k!}+\epsilon$, then he will be able to obtain a prize of at least $\sum_i \gamma\, \mathbf{E}[X_i] = (1-\frac{k^k}{e^k k!}+\epsilon)k$ in expectation which is a contradiction; therefore it is not possible to make such a guarantee. □

Proof of Theorem 26. First, we show that each S_i includes each item j with probability at least γ. Observe that for each item j, a sequence of n boxes are presented to the j^{th} magician with probabilities $\overline{x}_{1j}, \dots, \overline{x}_{nj}$ written on them. Since $\sum_i \overline{x}_{ij} \leq k_j$ and $\gamma \in [0, \gamma_k]$, we can argue that each box is opened with probability at least γ (see Theorem 24 and Definition 19); therefore S_i includes each item j with probability at least γ.

Next, we show that the expected objective value of γ-pre-rounding is at least $\gamma\alpha$-fraction of the expected objective value of the optimal mechanism in M. Note that by Theorem 22, the expected objective value of the optimal mechanism in M is upper bounded by the optimal value of $(\overline{OPT})$ which is $\sum_i R_i(\overline{x}_i)$; therefore, it is enough to show that $\mathbf{E}_{S_i}[R_i(\overline{x}_i[S_i])] \geq \gamma\alpha R_i(\overline{x}_i)$, i.e., the expected objective value that $M_i\langle\overline{x}_i[S_i]\rangle$ obtains from agent i is at least $\gamma\alpha R_i(\overline{x}_i)$. Let ξ_i be a budget balanced cross monotonic cost share function for $R_i(\cdot)$; then

$$\begin{aligned}
\mathbf{E}_{S_i}\left[R_i(\overline{x}_i[S_i])\right] &= \mathbf{E}_{S_i}\left[\sum_{j \in S_i} \xi_i(j, \overline{x}_i[S_i])\right] && \text{because } \xi_i \text{ is budget balanced} \\
&\geq \mathbf{E}_{S_i}\left[\sum_{j \in S_i} \xi_i(j, \overline{x}_i[\{1, \dots, m\}])\right] && \text{because } \xi_i \text{ is cross monotonic} \\
&= \sum_{j \in [m]} \mathbf{Pr}\left[j \in S_i\right] \xi_i(j, \overline{x}_i) \\
&\geq \sum_{j \in [m]} \gamma \xi_i(j, \overline{x}_i) \\
&= \gamma R_i(\overline{x}_i) && \text{because } \xi_i \text{ is budget balanced}
\end{aligned}$$

Next, we show that the multi agent mechanism based on γ-pre-rounding is in M and it is dominant strategy incentive compatible (DSIC). The fact that this mechanism is in M follows from assumption $A6$ and the fact that for each item j, the corresponding magician breaks no more than k_j wands, which means no more than k_j units are allocated at any instance. To show that it is DSIC, observe that the only way the reports of other agents could affect the outcome of agent i is by affecting S_i, yet $M_i\langle\overline{x}_i[S_i]\rangle$ is a mechanism in M_i, so it is incentive compatible mechanism for any choice of S_i; therefore the resulting mechanism is DSIC. Observe that this mechanism also preserves all of the ex post properties of each M_i (e.g., individual rationality). □

Proof of Theorem 27. First, we show that each S_i includes each item j with probability exactly γ. Observe that for each item j, a sequence of n boxes are presented to the j^{th} magician with probabilities $\hat{x}_{1j}, \ldots, \hat{x}_{nj}$ written on them. Since $\gamma \in [0, \gamma_k]$ and $\sum_i \hat{x}_{ij} \leq k_j$ and because each $M_i\langle \overline{x}_i \rangle$ allocates each item j with probability exactly $\hat{x}_{ij}$, we can argue that each box is opened with probability exactly γ (see Theorem 24 and Definition 19); therefore S_i includes each item j with probability exactly γ.

Next, we show that γ-post-rounding obtains in expectation at least $\gamma\alpha$-fraction of the expected objective value of the optimal mechanism in M. Note that by Theorem 22 the expected objective value of the optimal mechanism in M is upper bounded by the optimal value of ($\overline{OPT}$) which is $\sum_i R_i(\overline{x}_i)$; therefore, it is enough to show that $\mathbf{E}_{t_i, \mathsf{x}_i, \mathsf{p}_i}[\textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{p}_i)] \geq \gamma\alpha R_i(\overline{x}_i)$, i.e., the expected objective value that γ-post-rounding obtains from agent i is at least $\gamma\alpha R_i(\overline{x}_i)$. Let ξ_i be a budget balanced cross monotonic cost share function for $\textsc{Obj}_i$ as required by $A'2$; then

$$
\begin{aligned}
\mathbf{E}_{t_i, \mathsf{x}_i, \mathsf{p}_i}\left[\textsc{Obj}_i(t_i, \mathsf{x}_i, \mathsf{p}_i)\right] &= \mathbf{E}_{t_i, \mathsf{x}_i, \mathsf{p}_i}\left[\textsc{Obj}_i(t_i, \mathsf{x}_i, 0) + c_i \mathsf{p}_i\right] && \text{By } A'2 \\
&= \mathbf{E}_{t_i, \mathsf{x}_i, \mathsf{p}_i}\left[\sum_{j \in \mathsf{x}_i} \xi_i(j, t_i, \mathsf{x}_i) + c_i \mathsf{p}_i\right] && \text{because } \xi_i \text{ is budget balanced} \\
&\geq \mathbf{E}_{t_i, \mathsf{x}_i, \mathsf{p}_i}\left[\sum_{j \in \mathsf{x}_i} \xi_i(j, t_i, \mathsf{x}'_i) + c_i \mathsf{p}_i\right] && \text{because } \xi_i \text{ is cross monotonic} \\
&= \mathbf{E}_{t_i, \mathsf{x}'_i, \mathsf{p}'_i, S_i}\left[\sum_{j \in \mathsf{x}'_i} \mathbf{Pr}\left[j \in S_i\right] \xi_i(j, t_i, \mathsf{x}'_i) + c_i \gamma \mathsf{p}'_i\right] \\
&= \mathbf{E}_{t_i, \mathsf{x}'_i, \mathsf{p}'_i}\left[\sum_{j \in \mathsf{x}'_i} \gamma \xi_i(j, t_i, \mathsf{x}'_i) + c_i \gamma \mathsf{p}'_i\right] \\
&= \gamma\, \mathbf{E}_{t_i, \mathsf{x}'_i, \mathsf{p}'_i}\left[\textsc{Obj}_i(t_i, \mathsf{x}'_i, \mathsf{p}'_i)\right] \\
&\geq \gamma\alpha R_i(\overline{x}_i)
\end{aligned}
$$

Note that the last step follows because $\mathbf{E}_{t_i, \mathsf{x}'_i, \mathsf{p}'_i}[\textsc{Obj}_i(t_i, \mathsf{x}'_i, \mathsf{p}'_i)]$ is exactly the expected objective value of $M_i\langle \overline{x}_i \rangle$ which is at least $\alpha R_i(\overline{x}_i)$.

Next, we show that γ-post-rounding is Bayesian incentive compatible (BIC) and does not over allocate any item. Consider any arbitrary agent i. Observe that each item $j \in \mathsf{x}'_i$ is included in x_i with a probability of exactly γ; furthermore, by $A'4$, valuations of agent i can be interpreted as a weighted rank function of some matroid; WLOG, we may assume that x'_i is always an independent set of this matroid[24]; therefore, the valuation of the agent for the items in x'_i is additive; consequently, her expected valuation for x_i is exactly γ times her valuation for x'_i. Observe that both the expected valuation and the expected payment of agent i are scaled by γ for any outcome of $M_i\langle\overline{x}_i\rangle$ and $M_i\langle\overline{x}_i\rangle$ itself was incentive compatible; therefore, the resulting mechanism is incentive also incentive compatible. However, the final mechanism is only Bayesian incentive compatible because S_i depends on the typers/reports of agents other that i [25]. Also note that the mechanism does not over allocate any item, because for each unit of item j being allocated one of the k_j wands of the j^{th} magician breaks. □

Proof of Lemma 5. The proof is very similar to the proof of Theorem 21. To show that $R(\overline{x})$ is concave, it is enough to show that for any $\overline{x}$ and $\overline{x}'$ and any $\beta \in [0,1]$, $R(\beta\overline{x} + (1-\beta)\overline{x}') \geq \beta R(\overline{x}) + (1-\beta)R(\overline{x}')$. Let y and y' be the optimal assignments for the convex program subject to $\overline{x}$ and $\overline{x}'$ respectively; then $y'' = \beta y + (1-\beta)y'$ is also a feasible assignment for the convex program subject to $\beta\overline{x} + (1-\beta)\overline{x}'$; therefore, $R(\beta\overline{x}+(1-\beta)\overline{x}')$ must be at least $u(\beta y + (1-\beta)y')$; on the other hand $u(\cdot)$ is concave, so $u(\beta y + (1-\beta)y') \geq \beta u(y) + (1-\beta)u(y') = \beta R(\overline{x}) + (1-\beta)R(\overline{x}')$. That proves the claim. □

Proof of Lemma 9. Let $\mu = \sum_j \mathbf{E}[Z_j]$. Define the random variables $Y_j = \max(Y_{j-1} - Z_j, 0)$ and $Y_0 = B$. Observe that for each j, $Y_j = \max(B - \sum_{r=1}^{j} Z_r, 0)$, so $\min(\sum_{r=1}^{j} Z_r, B) + Y_j = B$. Therefore $\mathbf{E}[\min(\sum_{r=1}^{j} Z_r, B)] + \mathbf{E}[Y_j] = B$ and to prove the theorem it is enough to show that $\mathbf{E}[Y_m] \leq \frac{1}{e^{\mu/B}} B$. We first show that

$$Y_j \leq (1 - \frac{\mathbf{E}[Z_j]}{B})\, Y_{j-1}. \qquad (Y_j)$$

[24]Otherwise, we could replace x'_i by a maximum weight independent subset of x'_i.

[25]I.e., $\mathbf{Pr}[j \in S_i]$ is equal to γ only in expectation over other agents' reports

Consequently

$$Y_m \leq B\prod_{j=1}^{m}(1-\frac{\mathbf{E}[Z_j]}{B}) \tag{4.1}$$
$$\leq B\frac{1}{e^{\mu/B}} \tag{4.2}$$

The last inequality follows because $\prod_{j=1}^{m}(1-\frac{\mathbf{E}[Z_j]}{B})$ takes its maximum when $\frac{\mathbf{E}[Z_j]}{B} = \frac{\mu}{mB}$ (for all j) and $m \to \infty$.

To prove the second inequality in the statement of the lemma we can use the fact that $(1-x^a) \geq (1-x)a$ for any $a \leq 1$, and conclude that $(1-\frac{1}{e^{\mu/B}})B \geq (1-\frac{1}{e^{\min(\mu,B)/B}})B \geq (1-\frac{1}{e})\frac{\min(\mu,B)}{B}B = (1-\frac{1}{e})\min(\mu,B)$.

To complete the proof, we prove inequality (Y_j) as follows.

$$\begin{aligned}
\mathbf{E}[Y_j] &= \mathbf{E}[\max(Y_{j-1}-Z_j,0)] \\
&\leq \mathbf{E}\left[\max(Y_{j-1}-Z_j\frac{Y_{j-1}}{B},0)\right] && \text{because } \frac{Y_{j-1}}{B} \leq 1 \\
&= \mathbf{E}\left[Y_{j-1}-Z_j\frac{Y_{j-1}}{B}\right] && \text{because } \frac{Z_j}{B} \leq 1 \\
&= \mathbf{E}[Y_{j-1}] - \frac{1}{B}\mathbf{E}[Z_j Y_{j-1}] \\
&\leq \mathbf{E}[Y_{j-1}] - \frac{1}{B}\mathbf{E}[Z_j]\,\mathbf{E}[Y_{j-1}] && \text{because } Z_j \text{ and } Y_{j-1} \text{ are independent.} \\
&= (1-\frac{\mathbf{E}[Z_j]}{B})\,\mathbf{E}[Y_{j-1}]
\end{aligned}$$

□

Proof of Lemma 8. To prove the claim, it is enough to show that $\frac{r_1}{\sum_j x_j p_j} \geq \frac{1}{2}$. WLOG, we may assume that $\sum_j p_j x_j = 1$ since we can scale $p_1,\ldots,p_m$ by a constant $c = \frac{1}{\sum_j x_j p_j}$ and this will also scale $r_1,\ldots,r_m$ by the same constant c, so their ratio is not be affected. Consider the following LP and observe that x_j, p_j, and r_j, as defined in the statement of the lemma, form a feasible assignment for this LP. If we show that the optimal objective value of the LP is bounded below by $\frac{1}{2}$, any feasible assignment yields an objective value of at least $\frac{1}{2}$, and therefore $\frac{r_1}{\sum_j x_j p_j} \geq \frac{1}{2}$ which proves the lemma. In the following LP, p_j and r_j are variables and everything else is constant.

$$
\begin{aligned}
\text{minimize} \quad & r_1 \\
\text{subject to} \quad & r_j \geq x_j p_j + (1 - x_j) r_{j+1}, && \forall j \in [m] && (\alpha_j) \\
& r_j \geq r_{j+1}, && \forall j \in [m] && (\beta_j) \\
& \sum_{j=1}^{m} x_j p_j \geq 1 && && (\gamma) \\
& p_j \geq 0, && \forall j \in [m] \\
& r_j \geq 0, && \forall j \in [m+1]
\end{aligned}
$$

To prove that the optimal value of the above LP is bounded below by $\frac{1}{2}$, we construct a feasible assignment for its dual LP, obtaining a value of $\frac{1}{2}$. The dual LP is as follows.

$$
\begin{aligned}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & \gamma \leq \alpha_j, && \forall j \in [m] && (p_j) \\
& \alpha_1 + \beta_1 \leq 1 && && (r_1) \\
& \alpha_j + \beta_j \leq (1 - x_{j-1}) \alpha_{j-1} + \beta_{j-1}, && \forall j \in \{2, \ldots, m\} && (r_j) \\
& 0 \leq (1 - x_m) \alpha_m + \beta_m && && (r_{m+1}) \\
& \alpha_j \geq 0, \beta_j \geq 0, \gamma \geq 0, && \forall j \in [m]
\end{aligned}
$$

We construct an assignment for the dual LP as follows. Set $\alpha_j = \gamma$ and set $\beta_j = \beta_{j-1} - x_{j-1}\gamma$ for all j, except that for $j = 1$ we set $\beta_1 = 1 - \gamma$. From this assignment we get $\beta_j = 1 - \gamma - \gamma \sum_{\ell=1}^{j-1} x_\ell$. Observe that we get a feasible assignment as long as all β_j resulting from this assignment are non-negative. Furthermore, it is easy to see that $\beta_j \geq 1 - \gamma - \gamma \sum_{\ell=1}^{m} x_\ell \geq 1 - 2\gamma$ because $\sum_j x_j \leq 1$. Therefore, by setting $\gamma = \frac{1}{2}$, all β_j are non-negative and we always get a feasible assignment for the dual LP with an objective value of $\frac{1}{2}$, which completes the proof. □

Chapter 5

The Generalized Magician's Problem and Applications

5.1 Introduction

In this chapter we present a generalization of the magician's problem from §4.4 along with several applications. As the first application, we present an improved algorithm/lower bound for a generalization of prophet inequalities. As the second application, we present an online algorithm for the stochastic generalized assignment problem.

5.2 The Generalized Magician's Problem

We present a generalization of the magician's problem along with a near-optimal solution.

Definition 30 (The Generalized Magician's Problem)**.** *A magician is presented with a series of boxes one by one, in an online fashion. There is a prize hidden in one of the boxes. The magician has a magic wand that can be used to open the boxes. The wand has k units of mana*[1]*. If the wand is used on box i and has at least 1 unit of mana, the box opens, but the wand looses a random amount of mana $X_i \in [0, 1]$ drawn from a distribution specified on the box by its cumulative distribution function F_{X_i} (i.e., the magician learns F_{X_i} upon seeing box i). The magician wishes to maximize the probability of obtaining the prize, but unfortunately the sequence of boxes, the distributions written on the boxes, and the box containing the prize have been arranged by a villain; the magician has no prior information (not even the number of the boxes); however, it is guaranteed that $\sum_i \mathbf{E}[X_i] \leq k$, and that the villain has to prepare the sequence of boxes in advance (i.e., cannot make any changes once the process has started).*

The magician could fail to open a box either because (a) he might choose to skip the box, or (b) his wand might run out of mana before getting to the box. Note that once the magician fixes his strategy, the best strategy for the villain is to put the prize in the box which, based on the magician's strategy, has the lowest ex ante probability of being opened. Therefore, in order for the magician to obtain the prize with a probability of at least γ, he has to devise a strategy that guarantees an ex ante probability of at least γ for opening each box. Notice that allowing the prize to be split among multiple boxes does not affect the problem. We present an algorithm parameterized by a probability $\gamma \in [0, 1]$ which guarantees a minimum ex-ante probability of γ for opening each box while trying to minimize the mana used.

[1] *"Mana is an indigenous Pacific islander concept of an impersonal force or quality that resides in people, animals, and inanimate objects. .... Modern fantasy fiction, computer and role-playing games have adopted mana as a term for magic points, an expendable (and most often rechargeable) resource out of which magic users form their magical spells."* Wikipedia (2012)

We show that for $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ this algorithm never requires more than k units of mana.

Definition 31 (γ-Conservative Magician)**.** *The magician adaptively computes a sequence of thresholds $\theta_1, \theta_2, \ldots \in \mathbb{R}_+$ and makes a decision about each box as follows: let W_i denote the amount of mana lost prior to seeing the i^{th} box; the magician makes a decision about box i by comparing W_i against θ_i; if $W_i < \theta_i$, it opens the box; if $W_i > \theta_i$, it does not open the box; and if $W_i = \theta_i$, it randomizes and opens the box with some probability (to be defined). The magician chooses the smallest threshold θ_i for which $\mathbf{Pr}[W_i \leq \theta_i] \geq \gamma$ where the probability is computed ex ante (i.e., not conditioned on $X_1, \ldots, X_{i-1}$). Note that γ is a parameter that is given. Let $F_{W_i}(w) = \mathbf{Pr}[W_i \leq w]$ denote the ex ante CDF of random variable W_i, and let Y_i be the indicator random variable which is 1 iff the magician opens the box i. Formally, the probability with which the magician should open box i condition on W_i is computed as follows[2].*

$$\mathbf{Pr}\left[Y_i = 1 | W_i\right] = \begin{cases} 1 & W_i < \theta_i \\ (\gamma - F^-_{W_i}(\theta_i))/(F_{W_i}(\theta_i) - F^-_{W_i}(\theta_i)) & W_i = \theta_i \\ 0 & W_i > \theta_i \end{cases} \tag{Y}$$

$$\theta_i = \inf\{w | F_{W_i}(w) \geq \gamma\} \tag{θ}$$

In the above definition, $F^-_{W_i}$ is the left limit of F_{W_i}, i.e., $F^-_{W_i}(w) = \mathbf{Pr}[W_i < w]$.

Note that $F_{W_{i+1}}$ and $F^-_{W_{i+1}}$ are fully determined by F_{W_i} and F_{X_i} and the choice of γ (see Theorem 38). Observe that θ_i is in fact computed before seeing box i itself.

A γ-conservative magician may fail for a choice of γ unless all thresholds θ_i are less than or equal to $k - 1$. The following theorem states a condition on γ that is sufficient to guarantee that $\theta_i \leq k - 1$ for all i.

Theorem 36 (γ-Conservative Magician)**.** *For any $\gamma \leq 1 - \frac{1}{\sqrt{k}}$, a γ-conservative magician with k units of mana opens each box with an ex ante probability of γ exactly.*

Proof. See §5.5. □

[2]Assume $W_0 = 0$

Theorem 37 (γ-Conservative Magician (0–1))**.** *If $X_i \in \{0,1\}$ (i.e., Bernoulli random variable) for all i, then for any $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$, a γ-conservative magician with k units of mana opens each box with an ex ante probability of γ exactly; furthermore, if each F_{X_i} is not the exact CDF of X_i but stochastically dominates it, then the magician opens each box with an ex ante probability of* at least *γ.*

Proof. See §5.5. □

Definition 32 (γ_k^* and γ_k)**.** *We define γ_k^* to be the largest probability such that for any $k' \geq k$ and any instance of the magician's problem with k' units of mana, the thresholds computed by a γ_k^*-conservative magician are no more than $k' - 1$. In other words, γ_k^* is the optimal choice of γ which works for all instances with $k' \geq k$ units of mana. By Theorem 36, γ_k^* must be*[3] *at least $1 - \frac{1}{\sqrt{k}}$. We define γ_k similar to γ_k^* but with the extra assumption that $X_i \in \{0,1\}$ for all i. By Theorem 37, γ_k must be least $1 - \frac{1}{\sqrt{k+3}}$.*

Observe that γ_k^* and γ_k are non-decreasing functions in k and they both approach 1 as $k \to \infty$. However, for $k = 1$, $\gamma_1^* = 0$ whereas $\gamma_1 = \frac{1}{2}$. We can show that both of these bounds are tight for $k = 1$.

Proposition 8. *For the generalized magician's problem for $k = 1$, no algorithm for the magician (online or offline) can guarantee a constant non-zero probability for opening each box.*

Proof. Suppose there is an algorithm for the magician that is guaranteed to open each box with a probability of at least $\gamma \in (0, 1]$. We construct an instance in which the algorithm fails. Let $n = \lceil \frac{1}{\gamma} \rceil + 1$. Suppose all X_i are (independently) drawn from the distribution specified below.

$$X_i = \begin{cases} \frac{1}{2n} & \text{with prob. } 1 - \frac{1}{2n} \\ 1 & \text{with prob. } \frac{1}{2n} \end{cases}, \qquad \forall i \in [n]$$

[3] Because for any $k' \geq k$ obviously $1 - \frac{1}{\sqrt{k}} \leq 1 - \frac{1}{\sqrt{k'}}$.

As soon as the magician opens a box, the remaining mana will be less than 1, so he will not be able to open any other box, i.e., the magician can open only one box at every instance. Let Y_i denote the indicator random variable which is 1 iff the magician opens box i. Since $\sum_i Y_i \leq 1$, it must be $\sum_i \mathbf{E}[Y_i] \leq 1$. On the other hand, $\mathbf{E}[Y_i] \geq \gamma$ because the magician has guaranteed to open each box with a probability of at least γ. However $\sum_i \mathbf{E}[Y_i] \geq n\gamma > 1$ which is a contradiction. Note that $\sum_i \mathbf{E}[X_i] < 1$ so it satisfies the requirement of Definition 30. □

Proposition 9. *For the magician's problem for $k = 1$, assuming $X_i \in \{0, 1\}$ for all i, no algorithm for the magician (online or offline) can guarantee a probability of more than $\frac{1}{2}$ for opening each box.*

Proof. Suppose there is an algorithm for the magician that is guaranteed to open each box with a probability of at least $\gamma \in (0.5, 1]$. We construct an instance in which the algorithm fails. Pick any $\delta \in (\frac{1}{2\gamma}, 1]$. Suppose there are two boxes with distributions specified below.

$$X_1 = \begin{cases} 1 & \text{with prob. } \delta \\ 0 & \text{otherwise} \end{cases}$$

$$X_2 = \begin{cases} 1 & \text{with prob. } 1 - \delta \\ 0 & \text{otherwise} \end{cases}$$

Observe that the algorithm must open the first box with probability at least γ; so the probability that there is enough mana left for the second box is at most $1 - \gamma\delta < \frac{1}{2}$; therefore the algorithm will not be able to open the second box with a probability of $\frac{1}{2}$ or more. Note that $\sum_i \mathbf{E}[X_i] = 1$ so it satisfies the requirement of Definition 30. □

Computation of $F_{W_i}(\cdot)$. For every $i \in [n]$, the equation $W_{i+1} = W_i + Y_i X_i$ relates the distribution of W_{i+1} to those of W_i and X_i [4]. The following lemma shows that the distribution of W_{i+1} is fully determined by the information available to the magician before seeing box $i + 1$.

[4] Note that the distribution of Y_i is dependent on/determined by W_i.

Theorem 38. *In the algorithm of γ-conservative magician (Definition 31), the choice of γ and the distributions of $X_1, \ldots, X_i$ fully determine the distribution of W_{i+1}, for every $i \in [n]$. In particular, $F_{W_{i+1}}$ can be recursively defined as follows.*

$$F_{W_{i+1}}(w) = F_{W_i}(w) - G_i(w) + \mathbf{E}_{X_i \sim F_{X_i}}\left[G_i(w - X_i)\right] \qquad \forall i \in [n], \forall w \in \mathbb{R}_+ \qquad (F_W)$$

$$G_i(w) = \min(F_{W_i}(w), \gamma) \qquad \forall i \in [n], \forall w \in \mathbb{R}_+ \qquad (G)$$

Proof. See §5.5. □

As a corollary of Theorem 38, we show how F_{W_i} can be computed using dynamic programming, assuming X_i can only take discrete values that are proper multiples of some minimum value.

Corollary 5. *If all X_i are proper multiple of $\frac{1}{D}$ for some $D \in \mathbb{N}$, then $F_{W_i}(\cdot)$ can be computed using the following dynamic program.*

$$F_{W_{i+1}}(w) = \begin{cases} F_{W_i}(w) - G_i(w) + \sum_\ell \mathbf{Pr}[X_i = \frac{\ell}{D}] G_i(w - \frac{\ell}{D}) & i \geq 1, w \geq 0 \\ 1 & i = 0, w \geq 0 \\ 0 & \textit{otherwise.} \end{cases}, \quad \forall i \in [n], \forall w \in \mathbb{R}_+$$

In particular, the γ-conservative magician makes a decision for each box in time $O(D)$.

Note that it is enough to compute F_{W_i} only for proper multiples of $\frac{1}{D}$ because $F_{W_i}(w) = F_{W_i}(\frac{\lfloor Dw \rfloor}{D})$ for any $w \in \mathbb{R}_+$.

5.3 Prophet Inequalities

We prove a generalization of prophet inequalities by a direct reduction to the magician's problem. Prophet inequalities have been extensively studied in the past (e.g. Hill and Kertz (1992)). Prior to this work, the best known bound for the generalization to sum of k choices was $1 - O(\frac{\sqrt{\ln k}}{\sqrt{k}})$ by Hajiaghayi et al. (2007). We improve this to $1 - \frac{1}{\sqrt{k+3}}$. Note that the current bound is tight for $k = 1$, and is useful even for small values of k. We start by defining the problem formally.

Definition 33 (k-Choice Sum)**.** *A sequence of n non-negative random numbers $V_1, \dots, V_n$ are drawn from arbitrary distributions $F_1, \dots, F_n$ one by one in an arbitrary order. A gambler observes the process and may select k of the random numbers, with the goal of maximizing the sum of the selected ones; a random number may only be selected at the time it is drawn, and it cannot be unselected later. The gambler knows all the distributions in advance, and observes from which distribution the current number is drawn, but not the order in which the future numbers are drawn. On the other hand, a prophet knows all the actual draws in advance, so he chooses the k highest draws. We assume that the order in which the random numbers are drawn is fixed in advance (i.e., may not change based on the decisions of the gambler).*

Hajiaghayi et al. (2007) proved that there is a strategy for the gambler that guarantees in expectation at least $1 - O(\frac{\sqrt{\ln k}}{\sqrt{k}})$ fraction of the payoff of the prophet, using a non-decreasing sequence of k stopping rules (thresholds) [5]. Next, we construct a gambler that obtains in expectation at least γ_k fraction of the prophet's payoff, using a γ_k-conservative magician as a black box. Note that $\gamma_k \geq 1 - \frac{1}{\sqrt{k+3}}$. This gambler uses only a single threshold. However, he may skip some of the random draws at random.

Theorem 39 (Prophet Inequalities – k-Choice Sum)**.** *The following strategy ensures that the gambler obtains at least γ_k fraction of payoff of the prophet in expectation.* [6]

- *Find a threshold τ such that $\sum_i \mathbf{Pr}[V_i > \tau] = k$ (e.g., by doing a binary search on τ).*
- *Use a γ_k-conservative magician with k units of mana. Upon seeing each V_i, create a box and write $x_i = \mathbf{Pr}[V_i > \tau]$ on it and present it to the magician. If the magician chooses to open the box and also $V_i > \tau$, then select V_i and decrease the magician's mana by 1, otherwise skip V_i.*

[5] A gambler with stopping rules $\tau_1, \dots, \tau_k$ works as follows. Upon seeing V_i, he selects it iff $V_i \geq \tau_{j+1}$ where j is the number of random draws selected so far.

[6] To simplify the exposition we assume that the distributions do not have point masses. The result holds with slight modifications if we allow point masses.

Proof. First, we compute an upper bound on the expected payoff of the prophet. Let x_i be the ex ante probability (i.e., before any random number is drawn) that the prophet chooses V_i (i.e. the probability that V_i is among the k highest draws). Let $u_i(x_i)$ denote the maximum possible contribution of the random variable V_i to the expected payoff of the prophet if V_i is selected with an ex ante probability x_i. Note that $u_i(x_i)$ is equal to the expected value of V_i conditioned on being above the $1 - x_i$ quantile, multiplied by the probability of V_i being above that quantile. Assuming $F_i(\cdot)$ and $f_i(\cdot)$ denote the CDF and PDF of V_i, we can write $u_i(x_i) = \int_{F_i^{-1}(1-x_i)}^{\infty} v f_i(v) dv$. By changing the integration variable and applying the chain rule we get $u_i(x_i) = \int_0^{x_i} F_i^{-1}(1-x)dx$. Observe that $\frac{d}{dx_i} u_i(x_i) = F_i^{-1}(1-x_i)$ is a non-increasing function, so $u_i(x_i)$ is a concave function. Furthermore, $\sum_i x_i \leq k$ because the prophet cannot choose more than k random draws. So the optimal value of the following convex program is an upper bound on the payoff of the prophet.

$$\begin{aligned} \text{maximize} \quad & \sum_i u_i(x_i) & \\ \text{subject to} \quad & \sum_i x_i \leq k & (\tau) \\ & x_i \geq 0, \quad \forall i \in [n] & (\mu_i) \end{aligned}$$

Define the Lagrangian for the above convex program as

$$L(x, \tau, \mu) = -\sum_i u_i(x_i) + \tau \left(\sum_i x_i - k \right) - \sum_i \mu_i x_i.$$

By KKT stationarity condition, at the optimal assignment, it must be $\frac{\partial}{\partial x_i} L(q, \tau, \mu) = 0$. On the other hand, $\frac{\partial}{\partial x_i} L(q, \tau, \mu) = -F_i^{-1}(1-x_i) + \tau - \mu_i$. Assuming that $x_i > 0$, by complementary slackness $\mu_i = 0$, which then implies that $x_i = 1 - F_i(\tau)$, so $x_i = \mathbf{Pr}[V_i > \tau]$. Furthermore, it is easy to show that the first constraint must be tight, which implies that $\sum_i \mathbf{Pr}[V_i > \tau] = k$. Observe that the contribution of each V_i to the objective value of the convex program is exactly $\mathbf{E}[V_i | V_i > \tau] \mathbf{Pr}[V_i > \tau]$. By using a γ_k-conservative magician we can ensure that each box is opened with probability at least γ_k which implies the contribution of each V_i to the expected payoff of the gambler is $\mathbf{E}[V_i | V_i > \tau]\, \mathbf{Pr}[V_i > \tau]\gamma_k$ which proves that the expected payoff of the gambler is at least γ_k fraction of optimal objective value of the convex program, which was itself and upper bound on the expected payoff of the prophet. □

5.4 Online Stochastic Generalized Assignment Problem

5.4.1 Introduction

The generalized assignment problem (GAP) and its special cases, multi knapsack problem and bin packing capture a class of optimization problems with various applications in computer science, operations research, and related disciplines. The (offline) GAP is defined as follows:

Definition 34 (Generalized Assignment Problem)**.** *There is a set of items that can be assigned to a set of bins. Each item has a known size and a known value for each bin. the objective is to maximize the total value of the assignment subject to the total size of the items assigned to each bin not exceeding the capacity of that bin. The size and value of each item may depend on the bin it is assigned to (if assigned).*

For example GAP can be viewed as a scheduling problem on parallel machines, where each machine has a capacity (or a maximum load) and each job has a size (or a processing time) and a profit each possibly dependent on the machine to which it is assigned, and the objective is to find a feasible scheduling which maximizes the total profit. Though multiple knapsack and bin packing have a fully polynomial-time approximation scheme (asymptotic for bin packing) **?** in the offline setting, GAP is APX-hard and the best known approximation ratio is $1 - 1/e + \epsilon$ where $\epsilon \approx 10^{-180}$ **?**, which improves on a previous $(1 - 1/e)$-approximation **?**.

In this section we consider the following online stochastic variant of the problem.

Definition 35 (Online Stochastic Generalized Assignment Problem)**.** *A sequence of items arrive online and each item can be either assigned to a bin from a fixed set of bins, or discarded. Items arrive in an arbitrary unknown order; each item has a size and a value; stochastic information is known about the size/value of each item; the objective is to maximize the total value of the assignment subject to the total size of the items assigned to each bin not exceeding the capacity of that bin. The size and value of each item may depend on the bin it is assigned to (if assigned). The actual size of an item becomes known only after it is placed in a bin. Furthermore, it is given that an item does not take up more than $\frac{1}{k}$ fraction of the capacity of any relevant bin.*

We present a $1 - \frac{1}{\sqrt{k}}$-approximate online algorithm for the online stochastic assignment problem under the assumption that no item takes up more than $\frac{1}{k}$ fraction of the capacity of any bin. Items arrive online; each item has a value and a size; upon arrival, an item can be placed in a bin or discarded; the objective is to maximize the total value of the placement. Both value and size of an item may depend on the bin in which the item is placed; the size of an item is revealed only after it has been placed in a bin; distribution information is available about the value and size of each item in advance (not necessarily i.i.d), however items arrive in adversarial order (non-adaptive adversary).

5.4.2 Related Work

To the best of our knowledge Feldman et al. **?** were the first to consider the generalized assignment problem in an online setting. In the adversarial model where the items and the order of arrivals are chosen by an adversary, there is no competitive algorithm. Consider the simple case of one bin with capacity one and two arriving items each with size one. The value of the first item is 1. The value of the second item would be either $\frac{1}{\epsilon}$ or 0 based on whether we assign the first item to the bin. Thus the online profit cannot be more than ϵ factor of the offline profit. Indeed one can show a much stronger hardness result for the adversarial model: two special cases of GAP, namely the Adword problem[7] and the Display Ad problem[8] are shown to be not competitive even under the large-capacity assumption **??**.

Since no algorithm is competitive for online GAP in the adversarial model, Feldman et al. consider this model with *free disposal*. In free disposal model, the total size of items assigned to a bin may exceed its capacity, however, the profit of the bin is the maximum-valued subset of the assigned items which does not violate the capacity. Feldman et al. give a $(1-\frac{1}{e}-\epsilon)$-competitive primal-dual algorithm for GAP under the free disposal assumption and the additional large-capacity assumption by which the capacity of each bin is at least $O(\frac{1}{\epsilon})$ times larger than the maximum size of a single item. Although the free disposal assumption might be counter-intuitive in time-sensitive applications such as job scheduling, where the machine may start doing a job right after the job assignment, it is a very natural assumption in many applications including applications in economics like Ad allocation – a buyer does not mind receiving more items.

Dean, Goemans, and Vondrak **?** consider the closely related problem of (offline) stochastic knapsack problem. In their model, there is only one bin and the value of each item is known. However, the size of each item is drawn from a known distribution only after it is placed in the knapsack. We note that this is an offline setting in the sense that we may choose any order of items for allocation. This model is motivated by job scheduling on a single machine where the actual processing time required for a job is learned only after the completion of the job. Dean et al. give various adaptive and non-adaptive algorithms for their model where the best one has a competitive ratio $\frac{1}{3}-\epsilon$. This ratio was improved to $\frac{3}{8}-\epsilon$ by Bhalgat et al. **?**. Recently Bhalgat improved the competitive ratio to $\frac{1}{2}-\epsilon$ **?**. Other variations, such as soft capacity constraints, have also been considered for which we refer the reader to **???**. Dean et al. **?** also introduce an *ordered model* where items must be considered in a specific order, which can be seen as a version of the the online model where the order is known. **?** present a $\frac{1}{9.5}$-competitive algorithm. In general, the online model can be considered as a more challenging variation of the models proposed by Dean et al,

[7]The Adword problem is a special case of GAP where the size and the value of assigning an item to a bin is the same, i.e., $s_{ij} = v_{ij}$.

[8]The Display Ad problem is a special case of GAP where all sizes are uniform, i.e., $s_{ij} = 1$.

however we show that the assumption of bound on the maximum size to capacity ratio is enough to overcome this challenge.

Even with stochastic information about the arriving queries, no online algorithm can achieve a competitive ratio better than $\frac{1}{2}$ Hajiaghayi et al. (2007); **?**); **?**); Alaei (2011). Consider the simple example where the value of the first item is 1 with probability one and the value of the second item is $\frac{1}{\epsilon}$ with probability ϵ, and 0 with probability $1-\epsilon$. No online (randomized) algorithm can achieve a profit more than $\max\{1, \epsilon(\frac{1}{\epsilon})\} = 1$ in expectation. However, the expected profit of the optimum offline assignment is $(1-\epsilon)1 + \epsilon(\frac{1}{\epsilon}) = 2-\epsilon$. Therefore without any additional assumption one cannot get a competitive ratio better than 1/2. We overcome this difficulty by considering the natural large-capacity assumption which arises in many applications such as online advertising.

Our techniques can be used to design asymptotically optimum algorithms for other resource allocation settings. For example another application in ad allocation is the banner advertisement problem. Feige et al. **?** propose a new automated system for selling banner advertisements. In this system, each advertiser specifies a collection of webpages which are relevant to his product, a desired total quantity of impressions on these pages, and a maximum per-impression price. The problem of selecting a feasible subset of advertisers with maximum total value does not have any non-trivial approximation. This can be shown by a reduction from the Independent Set problem on a graph; advertisers represent the vertices of the graph and webpages represent the edges of the graph. Advertisers desire all the impressions of the relevant webpages. Thus any feasible subset of advertisers would denote an independent set in the graph. This shows that maximizing the total value does not have a non-trivial approximation. Feige et al. present two greedy heuristics and discuss new techniques to measure their performances. By considering some variants of the banner advertisement problem, they show that their algorithms can achieve a competitive ratio between 0.1 to 0.3 which depends on the structural properties of the optimum solution. We show that one can get near optimum solutions if the number of available impressions on each website is at least k times the required impressions of each relevant advertiser.

5.4.3 Preliminaries

Model. We consider the problem of assigning m items to n bins; items arrive on-line in an arbitrary but unknown order; stochastic information is known about the size/value of each item; the objective is to maximize the total value of the assignment. Each item $i \in [m]$ has r_i possible types with each type $t \in [r_i]$ having a probability of p_{it}, a value of $v_{itj} \in \mathbb{R}_+$, and a size of $S_{itj} \in [0,1]$ if placed in bin j (for each $j \in [n]$); S_{itj} is a random variable which is drawn from a distribution with a CDF of F_{itj} if the item is placed in bin j. Each bin $j \in [m]$ has a capacity of $c_j \in \mathbb{N}_0$ which limits the total size of the items placed in that bin[9]. The type of each item is revealed upon arrival and the item must be either placed in a bin or discarded; this decision cannot be changed later. The size of an item is revealed only after it has been place in a bin, furthermore an item can be placed in a bin only if the bin has at least one unit of capacity left. We assume that m, n, c_j, v_{itj} and F_{itj} are known in advance.

Note that the assumption that all item sizes being in $[0,1]$ is WLOG because all item sizes and the capacity of each bin can be scaled.

Benchmark. Consider the following linear program in which $\tilde{s}_{itj} = \mathbf{E}_{S_{itj} \sim F_{itj}}[S_{itj}]$; the optimal value of this linear program, which corresponds to the expected instance, is an upper bound on the expected value of the optimal offline assignment.

$$
\begin{aligned}
\text{maximize} \quad & \sum_i \sum_t \sum_j v_{itj} x_{itj} && (\overline{OPT}_{GAP}) \\
\text{subject to} \quad & \sum_i \sum_t \tilde{s}_{itj} x_{itj} \le c_j, && \forall j \in [n] \\
& \sum_j x_{itj} \le p_{it}, && \forall i \in [m], \forall t \in [r_i] \\
& x_{itj} \in [0,1],
\end{aligned}
$$

Theorem 40. *The optimal value of the linear program ($\overline{OPT}_{GAP}$) is an upper bound the the expected value of the offline optimal assignment.*

[9]Our results hold for non-integer capacitates, however we assume integer capacities to simplify the exposition.

Proof. Let x^*_{itj} denote the ex ante probability that item i is of type t and is assigned to bin j in the optimal offline assignment. It is easy to see that x^*_{itj} is a feasible assignment for the linear program. Furthermore, the expected value of the optimal offline assignment is exactly $\sum_i \sum_t \sum_j v_{itj} x^*_{itj}$ which is equal to the value of the linear program for x^*_{itj} which is itself no more than the optimal value of the linear program. Note that the optimal value of the linear program may be strictly higher since a feasible assignment of the linear program does not necessarily correspond to a feasible offline assignment policy. □

Section §5.4.4 presents an online algorithm which obtains at least $1 - \frac{1}{\sqrt{k}}$-fraction of the optimal value of the above linear program, where $k = \min_j c_j$. Next section presents a stochastic toy problem and its solution which is used as a black box in the online algorithm of §5.4.4.

5.4.4 The Online Algorithm

We present an online algorithm which obtains at least $1 - \frac{1}{\sqrt{k}}$-fraction of the optimal value of the linear program ($\overline{OPT}_{GAP}$). The algorithm uses, as a black box, the solution of the generalized magician's problem.

Definition 36 (Online Stochastic GAP Algorithm)**.**

1. *Solve the linear program* ($\overline{OPT}_{GAP}$) *and let x be an optimal assignment.*

2. *For each $j \in [n]$, create a γ-conservative magician (Definition 31) with c_j units of mana for bin j. γ is a parameter that is given.*

3. *Upon arrival of each item $i \in [m]$, do the following:*

   (a) *Let t denote the type of item i.*

   (b) *Choose a bin at random such that each bin $j \in [n]$ is chosen with probability $\frac{x_{itj}}{p_{it}}$. Let j^* denote the chosen bin.*

(c) *For each $j \in [n]$, define the random variable X_{ij} as $X_{ij} \leftarrow S_{itj}$ if $j^* = j$, and $X_{ij} \leftarrow 0$ otherwise*[10]. *For each $j \in [n]$, write the CDF of X_{ij} on a box and present it to the magician of bin j. The CDF of X_{ij} is $F_{x_{ij}}(s) = (1 - \sum_{t'} x_{it'j}) + \sum_{t'} x_{it'j} F_{it'j}(s)$.*

(d) *If the magician for bin j^* opened his box in step 3c, then assign item i to bin j^*, otherwise discard the item. For each $j \in [n]$, if the magician of bin j opened his box in step 3c, decrease the mana of that magician by X_{ij}. In particular, $X_{ij} = 0$ for all $j \neq j^*$, and $X_{ij^*} = S_{itj^*}$.*

Theorem 41. *For any $\gamma \leq \gamma^*_k$, the online algorithm of Definition 36 obtains in expectation at least a γ-fraction of the expected value of the optimal offline assignment (recall that $\gamma^*_k \geq 1 - \frac{1}{\sqrt{k}}$).*

Proof. By Theorem 40, it is enough to show that the online algorithm obtains in expectation at least a γ-fraction of the optimal value of the linear program ($\overline{OPT}_{GAP}$). Let x be an optimal assignment for the LP. The contribution of each item $i \in [m]$ to the value of bin $j \in [n]$ in the LP is exactly $\sum_t v_{itj} x_{itj}$. We show that the online algorithm obtains in expectation $\gamma \sum_t v_{itj} x_{itj}$ from each item i and each bin j.

Consider an arbitrary item $i \in [m]$ and an arbitrary bin $j \in [n]$. WLOG, suppose the items are indexed in the order in which they arrive. Observe that

$$\mathbf{E}\left[X_{ij}\right] = \sum_t p_{it} \frac{x_{itj}}{p_{it}} \mathbf{E}\left[S_{itj}\right] = \sum_t x_{itj} \widetilde{s}_{itj}.$$

Consequently,

$$\sum_i \mathbf{E}\left[X_{ij}\right] = \sum_i \sum_t x_{itj} \widetilde{s}_{itj} \leq c_j.$$

The last inequality follows from the first set of constraints in the LP of ($\overline{OPT}_{GAP}$). Given that $\sum_i \mathbf{E}[X_{ij}] \leq c_j$ and $\gamma \leq \gamma^*_k \leq \gamma^*_{c_j}$, Theorem 36 implies that the magician of bin j opens each box with a probability of γ. Therefore, the expected contribution

[10] Note that S_{itj} is learned only after item i is placed in bin j which implies that X_{ij} may not be known at this point, however the algorithm does not use X_{ij} until after it is learned.

of item i to bin j is exactly $\sum_t \gamma p_{it} \frac{x_{itj}}{p_{it}} v_{itj} = \gamma \sum_t x_{itj} v_{itj}$. Consequently, the online algorithm obtains $\gamma \sum_i \sum_j \sum_t x_{itj} v_{itj}$ in expectation which is at least a γ-fraction of the expected value of the optimal offline assignment. Furthermore, each magician guarantees that the total size of the items assigned to each bin does not exceed the capacity of that bin. □

5.5 Analysis of Generalized γ-Conservative Magician

We present the proof of Theorem 36 and Theorem 37. We prove the theorems in two parts. In the first part, we show that the thresholds computed by the γ-conservative magician indeed guarantee that each box is opened with an ex-ante probability of γ, assuming there is enough mana. In the second part, we show that for any $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ (or $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ and assuming $X_i \in \{0, 1\}$), the thresholds θ_i are less than or equal to $k - 1$, for all i, which implies that the magician never requires more than k units of mana. It can be shown that a non-adaptive algorithm cannot guarantee a probability of more than $1 - O(\frac{\sqrt{\ln k}}{\sqrt{k}})$ for opening each box.

Below, we repeat the formulation of the threshold based strategy of the magician.

$$\mathbf{Pr}\left[Y_i = 1 | W_i\right] = \begin{cases} 1 & W_i < \theta_i \\ (\gamma - F^{-}_{W_i}(\theta_i))/(F_{W_i}(\theta_i) - F^{-}_{W_i}(\theta_i)) & W_i = \theta_i \\ 0 & W_i > \theta_i \end{cases} \qquad (Y)$$

$$\theta_i = \inf\{w | F_{W_i}(w) \geq \gamma\} \qquad (\theta)$$

Part 1. We show that the thresholds computed by a γ-conservative magician guarantee that each box is opened with an ex ante probability of γ (i.e., $\mathbf{Pr}[Y_i = 1] = \gamma$), assuming there is enough mana.

$$\begin{aligned}\mathbf{Pr}\left[Y_i \le w\right] &= \mathbf{Pr}\left[Y_i = 1 \cap W_i < \theta_i\right] + \mathbf{Pr}\left[Y_i = 1 \cap W_i = \theta_i\right] + \mathbf{Pr}\left[Y_i = 1 \cap W_i > \theta_i\right] \\ &= \mathbf{Pr}\left[W_i < \theta_i\right] + \frac{\gamma - F^{-}_{W_i}(\theta_i)}{F_{W_i}(\theta_i) - F^{-}_{W_i}(\theta_i)} \mathbf{Pr}\left[W_i = \theta_i\right] \\ &= \gamma\end{aligned}$$

For Theorem 37, we must show that the thresholds computed by a γ-conservative magician guarantee that each box is opened with an ex ante probability at least γ when F_{X_i} stochastically dominates the actual CDF of X_i for all i and assuming there is enough mana. Let $x_i = \mathbf{E}_{X_i \sim F_{X_i}}[X_i]$, i.e., x_i is an upper bound on $\Pr X_i = 1$ (recall that $x_i \in \{0, 1\}$). The proof is as follows.

(a) First we prove that $\mathbf{Pr}[W_i \le \ell] \ge F_{W_i}(\ell)$ by induction on i. The base case is trivial. Suppose the inequality holds for $i \ge 1$, we prove it for $i+1$ as follows.

$$\begin{aligned}\mathbf{Pr}\left[W_{i+1} \le \ell\right] &\ge \mathbf{Pr}\left[W_i \le \ell - 1\right] + \mathbf{Pr}\left[W_i = \ell\right](1 - y_i^\ell x_i) \\ &= \mathbf{Pr}\left[W_i \le \ell - 1\right] y_i^\ell x_i + \mathbf{Pr}\left[W_i \le \ell\right](1 - y_i^\ell x_i) \\ &\ge F_{W_i}(\ell - 1) y_i^\ell x_i + F_{W_i}(\ell)(1 - y_i^\ell x_i) && \text{by induction hypothesis} \\ &= F_{W_{i+1}}(\ell) && \text{by (\textbf{??})}\end{aligned} \tag{1.a}$$

Observe that all of the above inequalities are met with equality if every x_i is the exact probability of breaking wand for the corresponding box instead of just an upper bound.

(b) Next, we show that each box is opened with probability at least γ. We shall show that $\mathbf{Pr}[Y_i = 1] \geq \gamma$.

$$\begin{aligned}
\mathbf{Pr}\left[Y_i = 1\right] &= \sum_{\ell} \mathbf{Pr}\left[Y_i = 1 | W_i = \ell\right] \mathbf{Pr}\left[W_i = \ell\right] \\
&= \sum_{\ell=0}^{\theta_i} y_i^{\ell}\, \mathbf{Pr}\left[W_i = \ell\right] \\
&= \mathbf{Pr}\left[W_i < \theta_i\right] + y_i^{\theta_i}\, \mathbf{Pr}\left[W_i = \theta_i\right] && \text{because } y_i^{\ell} = 1 \text{ for } \ell < \theta_i \\
&= (1 - y_i^{\theta_i})\, \mathbf{Pr}\left[W_i < \theta_i\right] + y_i^{\theta_i}\, \mathbf{Pr}\left[W_i \leq \theta_i\right] \\
&\geq (1 - y_i^{\theta_i}) F_{W_i}(\theta_i - 1) + y_i^{\theta_i} F_{W_i}(\theta_i) && \text{by (1.a)} \\
&= F_{W_i}(\theta_i - 1) + y_i^{\theta_i}(F_{W_i}(\theta_i) - F_{W_i}(\theta_i - 1)) \\
&= \gamma && \text{by substituting } y_i^{\theta_i} \text{ from (\textbf{??})}
\end{aligned}$$

Observe that all of the above inequalities are met with equality if each x_i is the exact probability of breaking a wand for the corresponding box instead of being just an upper bound.

Part 2. Assuming $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ (or $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ and $X_i \in \{0, 1\}$), we show that the thresholds computed by a γ-conservative magician are no more than $k - 1$ (i.e., $\theta_i \leq k - 1$ for all i). First, we present an interpretation of how $F_{W_i}(\cdot)$ evolves in i in terms of a sand displacement process.

Definition 37 (Sand Displacement Process)**.** *Consider one unit of infinitely divisible sand which is initially at position* 0 *on the real line. The sand is gradually moved to the right and distributed over the real line in n rounds. Let $F_{W_i}(w)$ denote the total amount of sand in the interval $[0, w]$ at the beginning of round $i \in [n]$. At each round i the following happens.*

(I) The leftmost γ-fraction of the sand is selected by first identifying the smallest threshold $\theta_i \in \mathbb{R}_+$ such that $F_{W_i}(\theta_i) \geq \gamma$ and then selecting all the sand in the interval $[0, \theta_i)$ and selecting a fraction of the sand at position θ_i itself such that the total amount of selected sand is equal to γ. Formally, if $G_i(w)$ denotes the total amount of sand selected from $[0, w]$, the selection of sand is such that $G_i(w) = \min(F_{W_i}(w), \gamma)$, for every $w \in \mathbb{R}_+$. In particular, this implies that only a fraction of the sand at position θ_i itself might be selected, however all the sand to the left of position θ_i is selected.

(II) The selected sand is moved to the right as follows. Consider the given random variable $X_i \in [0, 1]$ and let $F_{X_i}(\cdot)$ denote its CDF. For every point $w \in [0, \theta_i]$ and every distance $\delta \in [0, 1]$, take a fraction proportional to $\mathbf{Pr}[X_i = \delta]$ out of the sand which was selected from position w and move it to position $w + \delta$.

It is easy to see that θ_i and $F_{W_i}(w)$ resulting from the above process are exactly the same as those computed by the γ-conservative magician.

Lemma 10. *At the end of the i^{th} round of the sand displacement process, the total amount of sand in the interval $[0, w]$ is given by the following equation.*

$$F_{W_{i+1}}(w) = F_{W_i}(w) - G_i(w) + \mathbf{E}_{X_i \sim F_{X_i}}\left[G_i(w - X_i)\right] \qquad \forall i \in [n], \forall w \in \mathbb{R}_+ \qquad (F_W)$$

Proof. According to definition of the sand displacement process, $F_{W_{i+1}}(w)$ can be defined as follows.

$$\begin{aligned} F_{W_{i+1}}(w) &= (F_{W_i}(w) - G_i(w)) + \iint_{\omega+\delta\leq w} dG_i(\omega)\, dF_{X_i}(\delta) \\ &= F_{W_i}(w) - G_i(w) + \int G_i(\omega-\delta)\, dF_{X_i}(\delta) \\ &= F_{W_i}(w) - G_i(w) + \mathbf{E}_{X_i\sim F_{X_i}}\left[G_i(w - X_i)\right] \end{aligned}$$

□

Proof of Theorem 38. The claim follows directly from Lemma 10 □

Consider a conceptual barrier which is at position θ_i+1 at the beginning of round i and is moved to position $\theta_{i+1}+1$ for the next round, for each $i \in [n]$. It is easy to verify (i.e., by induction) that the sand never crosses to the right side of the barrier (i.e., $F_{W_{i+1}}(\theta_i+1) = 1$). The following theorem implies that the sand remains concentrated near the barrier throughout the process.

Theorem 42 (Sand)**.** *Throughout the sand displacement process (Definition 37), at the beginning of round $i \in [n]$, the following inequality holds.*

$$F_{W_i}(w) < \gamma F_{W_i}(w+1), \qquad \forall i \in [n], \forall w \in [0, \theta_i) \qquad (F_W\text{-ineq})$$

Furthermore, at the beginning of round $i \in [n]$, the average distance of the sand from the barrier, denoted by d_i, is upper bounded by the following inequalities[11] *in which the first inequality is strict except for $i = 1$.*

$$d_i \leq (1-\{\theta_i\})\frac{1-\gamma^{\lfloor\theta_i\rfloor+1}}{1-\gamma} + \{\theta_i\}\,\frac{1-\gamma^{\lceil\theta_i\rceil+1}}{1-\gamma} \leq \frac{1-\gamma^{\lceil\theta_i\rceil+1}}{1-\gamma} < \frac{1}{1-\gamma}, \quad \forall i \in [n] \quad (d)$$

Proof. We start by proving the inequality (F_W-ineq). The proof is by induction on i. The case of $i = 1$ is trivial because all the sand is at position 0 and so $\theta_1 = 0$. Suppose the inequality holds at the beginning of round i for all $w \in [0, \theta_i)$; we show that it holds at the beginning of round $i+1$ for all $w \in [0, \theta_{i+1})$. Note that $\theta_i \leq \theta_{i+1} \leq \theta_i+1$, so there are two possible cases:

[11]Note that $\{z\} = z - \lfloor z\rfloor$, for any z.

Case 1. $w \in [0, \theta_i)$. Observe that $G_i(w) = F_{W_i}(w)$ in this interval, so:

$$\begin{aligned}
F_{W_{i+1}}(w) &= F_{W_i}(w) - G_i(w) + \mathbf{E}_{X_i}\left[G_i(w - X_i)\right] && \text{by } (F_W).\\
&= \mathbf{E}_{X_i}\left[F_{W_i}(w - X_i)\right] && \text{by } G_i(w) = F_{W_i}(w) \text{, for } w \in [0, \theta_i)\\
&< \mathbf{E}_{X_i}\left[\gamma F_{W_i}(w - X_i + 1)\right] && \text{by induction hypothesis.}\\
&= \gamma\, \mathbf{E}_{X_i}\left[F_{W_i}(w - X_i + 1) - G_i(w - X_i + 1) + G_i(w - X_i + 1)\right] &&\\
&\leq \gamma\ \left(F_{W_i}(w + 1) - G_i(w + 1) + \mathbf{E}_{X_i}\left[G_i(w - X_i + 1)\right]\right) && \text{by monotonicity of } F_{W_i}(\cdot) - G_i(\cdot).\\
&= \gamma F_{W_{i+1}}(w + 1) && \text{by } (F_W).
\end{aligned}$$

Case 2. $w \in [\theta_i, \theta_{i+1}]$. We prove the claim by showing that $F_{W_{i+1}}(w) < \gamma$ and $F_{W_{i+1}}(w + 1) = 1$. Observe that $F_{W_{i+1}}(w) < \gamma$ because $w < \theta_{i+1}$ and because of the definition of θ_{i+1} in (θ). Furthermore, observe that $F_{W_{i+1}}(w + 1) \geq F_{W_{i+1}}(\theta_i + 1) = 1$ both before and after round i all the sand is still contained in the interval $[0, \theta_i + 1]$.

Next, we prove inequality (d) which upper bounds the average distance of the sand from the barrier at the beginning of round $i \in [n]$.

$$
\begin{aligned}
d_i &= \int_0^{\theta_i+1} (\theta_i + 1 - w)\, dF_{W_i}(w) \\
&= \int_0^{\theta_i+1} F_{W_i}(w)\, dw && \text{by integration by part.} \\
&= \sum_{\ell=0}^{\lceil \theta_i \rceil} \int_{\theta_i-\ell}^{\theta_i+1-\ell} F_{W_i}(w)\, dw \\
&\leq \sum_{\ell=0}^{\lfloor \theta_i \rfloor} \int_{\theta_i}^{\theta_i+1} \gamma^\ell F_{W_i}(w)\, dw \; + \int_{\lfloor \theta_i \rfloor+1}^{\theta_i+1} \gamma^{\lceil \theta_i \rceil} F_{W_i}(w)\, dw && \text{by } (F_W\text{-ineq}). \\
&\leq \sum_{\ell=0}^{\lfloor \theta_i \rfloor} \gamma^\ell \; + \{\theta_i\}\, \gamma^{\lceil \theta_i \rceil} && \text{by } F_{W_i}(w) \leq 1. \\
&= (1 - \{\theta_i\}) \sum_{\ell=0}^{\lfloor \theta_i \rfloor} \gamma^\ell \; + \{\theta_i\} \sum_{\ell=0}^{\lceil \theta_i \rceil} \gamma^\ell \\
&= (1 - \{\theta_i\}) \frac{1 - \gamma^{\lfloor \theta_i \rfloor + 1}}{1 - \gamma} + \{\theta_i\}\, \frac{1 - \gamma^{\lceil \theta_i \rceil + 1}}{1 - \gamma} \\
&\leq \frac{1 - \gamma^{\lceil \theta_i \rceil + 1}}{1 - \gamma}
\end{aligned}
$$

The last inequality follows because $(1 - \beta)L + \beta H \leq H$ for any $\beta \in [0, 1]$ and any L, H with $L \leq H$. Note that at least one of the first two inequalities is strict except for $i = 1$ which proves the claim. □

Theorem 43 (Barrier)**.** *If* $\sum_{i=1}^n \mathbf{E}_{X_i \sim F_{X_i}}[X_i] \leq k$ *for some* $k \in \mathbb{N}$*, and* $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ *(or* $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$ *and also* $X_i \in \{0, 1\}$ *for all* i*), then the distance of the barrier from the origin is no more than* k *throughout the process, i.e.,* $\theta_i \leq k - 1$ *for all* $i \in [n]$*.*

Proof. At the beginning of round i, let d_i and d'_i denote the average distance of the sand from the barrier and from the origin respectively. Recall that the barrier is defined to be at position $\theta_i{+}1$ at the beginning of round i. Observe that $d_i{+}d'_i = \theta_i{+}1$. Furthermore, $d'_{i+1} = d'_i + \gamma\, \mathbf{E}[X_i]$, i.e., the average distance of the sand from the origin is increased exactly by $\gamma\, \mathbf{E}[X_i]$ during round i (because the amount of selected sand is exactly γ and the sand selected from every position $w \in [0, \theta_i]$ is moved to the right by an expected distance of $\mathbf{E}[X_i]$). By applying Theorem 42 we get the following inequality.

$$\begin{aligned}\theta_i + 1 &= d_i' + d_i \\ &< \gamma \sum_{r=1}^{i-1} \mathbf{E}\left[X_i\right] + d_i \\ &\leq \gamma k + (1 - \{\theta_i\}) \frac{1 - \gamma^{\lfloor \theta_i \rfloor + 1}}{1 - \gamma} + \{\theta_i\} \frac{1 - \gamma^{\lceil \theta_i \rceil + 1}}{1 - \gamma}, \qquad \forall i \in [n] \qquad (\Gamma)\end{aligned}$$

In order to show that the distance of the barrier from the origin is no more than k throughout the process, it is enough to show that the above inequality cannot hold for $\theta_i > k - 1$. In fact it is just enough to show that it cannot hold for $\theta_i = k - 1$; alternatively, it is enough to show that the complement of the above inequality holds for $\theta_i = k - 1$.

$$k \geq \gamma k + \frac{1 - \gamma^k}{1 - \gamma}$$

Consider the stronger inequality $k \geq \gamma k + \frac{1}{1-\gamma}$; this inequality is quadratic in γ and can be solved to get a bound of $\gamma \leq 1 - \frac{1}{\sqrt{k}}$.

Next, consider the case in which $X_i \in \{0, 1\}$ for all i. Observe that the barrier can only take integral values; therefore, to show that the distance of the barrier from the origin is no more than k, it is enough to show that inequality (Γ) cannot hold for $\theta_i = k$; alternatively, it is enough to show that the complement of that inequality holds for $\theta_i = k$.

$$k + 1 \geq \gamma k + \frac{1 - \gamma^{k+1}}{1 - \gamma}$$

Consider the the stronger inequality $k+1 \geq \gamma k + \frac{1}{1-\gamma}$ which is quadratic in γ and yields a bound of $\gamma \leq 1 - \frac{1}{1/2 + \sqrt{k + 1/4}}$; this bound in fact imposes a looser constraint than $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$ when $k \geq 7$. Furthermore it can be verified (by direct calculation) that the inequality holds for $k < 7$ and $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$. That completes the proof. $\square$

Theorem 43 implies that a γ-conservative magician requires no more than k units of mana, assuming that $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ (or assuming $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$ and also $X_i \in \{0, 1\}$ for all i). That completes the proof of Theorem 36 and Theorem 37.

Part III

Non-Bayesian Mechanism Design

Chapter 6

Competitive Equilibrium in Two Sided Matching Markets

In this chapter, we study the class of competitive equilibria in two sided matching markets with general (non-quasilinear) utility functions. Mechanism design in general non-quasilinear setting is one of the biggest challenges in mechanism design. General non-quasilinear utilities can for example model smooth budget constraints as a special case. Due to the difficulty of dealing with arbitrary non-quasilinear utilities, a large fraction of the existing work have considered the simpler case of quasilinear utilities with hard budget constraints and they all rely on some form of ascending auction. For general non-quasilinear utilities, we show that such ascending auctions may not even converge in finite time. As such, almost all of the existing work on general non-quasilinear utility function (Demange and Gale (1985); Gale (1984); Quinzii (1984)) have resorted to non-constructive proofs based on fixed point theorems or discretization. In this chapter, we give the first direct characterization of competitive equilibria in such markets. Our approach is constructive and solely based on induction. Our characterization reveals striking similarities between the payments at the lowest competitive equilibrium for general utilities and VCG payments for quasilinear utilities. We also show that the mechanism that outputs the lowest competitive equilibrium is group strategyproof. We also present a class of price discriminating truthful mechanisms for selling heterogeneous goods to unit-demand buyers with general utility functions and from that we derive a natural welfare maximizing mechanism for ad-auctions that combines pay per click and pay per impression advertisers with general utility functions. Our mechanism is group strategyproof even if the search engine and advertisers have different estimates of clickthrough rates.

6.1 Introduction

In this chapter, we study the class of competitive equilibria in two sided matching markets with general utility functions. In these markets, agents form a one-to-one matching and monetary transfers are made between matched agents. Utility of each agent is a function of whom she is matched to and the amount of the monetary transfer to/from her partner. For the most of this chapter, we work with simpler markets consisting of a set of buyers and a set of heterogeneous good. In section 6.5, we show that the more general model can be reduced to this simpler buyer/good model. We assume that the utility of each buyer depends on the choice of good she receives and the price she pays but it is not necessarily a quasi-linear function of the payment. Non-quasilinear utilities can be used for example to model smooth budget constraints.

A competitive equilibrium, in these markets, is essentially an assignment of prices to goods together with a feasible allocation of goods to buyers such that every buyer receives her most preferred good at the announced prices and every unallocated good has a price of 0. This is also referred to as an envy-free equilibrium for the buyer/good model. In the case of unit-demand buyers, each buyer would be allocated at most a single good. With quasi-linear utilities, buyer i's utility for good j as a function of payment can be written as $u_i^j(x) = v_i^j - x$ where v_i^j is the valuation of buyer i for good j and x is the payment. In this case, social welfare is well-defined and VCG is applicable. The efficient allocation can be computed using a maximum weight matching on the bipartite graph consisting of buyers/goods with the edge between buyer i and good j having a weight of v_i^j. The VCG payoffs/payments would then correspond to a minimum weighted cover on this graph Leonard (1983). For general utilities, the functions $u_i^j(x)$ could be any continuous decreasing function of x. In this case, social welfare is not well defined and VCG is not applicable.

As a motivating example of a unit-demand market with non-quasi-linear utilities, consider a housing market in which each seller owns a house and each buyer wants to buy a house. Typically, a buyer will have a smooth budget constraint. For example, they may need to get a loan/mortgage to pay for the house and so the actual cost will include interests, fees, etc. in addition to the actual payment made to the seller. This cost may depend on the choice of the house as well (e.g., the interest rate may depend on the condition of the house). With non-quasilinear utilities, this can be modeled as $u_i^j(x) = v_i^j - c_i^j(x)$ in which $c_i^j(x)$ is the cost as a function of the price of good j.

6.2 Related Work

In the abstract mathematical form, the problem we are looking at is a one-to-one matching with monetary transfers and general utilities as described by Demange and GaleDemange and Gale (1985). In this model, the set of competitive equilibria corresponds exactly to the outcomes that are in the core. Demange and Gale also proved the lattice structure on the set of competitive equilibria although the lattice structure was already discovered by Shapley and Shubik Shapley and Shubik (1971) for the case of quasilinear utilities. Demange, Gale and Sotomayor Demange et al. (1986) proposed an ascending auction for the quasilinear setting to compute a competitive equilibrium. The existence of competitive equilibria for general utilities was proved by Quinzii Quinzii (1984). Quinzii showed that the game defined by this model is a *"Balanced Game"* and for general n-person balanced games it was already shown by Scarf Scarf (1967) that the core is non-empty. Using a different method, Gale Gale (1984) showed that for a more general class of preferences (i.e., preferences are not even required to be monotone in payment, yet they should still satisfy some other milder conditions) a competitive equilibrium always exists. Gale's proof is based on a generalization of the KKM lemma Knaster et al. (1929) which is the continuous variant of the Sperner's lemma. Both of these proofs only show the existence of an equilibrium and are non-constructive. As such, they don't help much in understanding the properties of the equilibria. [1]

Using a completely different approach, Kelso and Crawford Kelso and Crawford (1982) studied the equilibria of the more general case of many-to-one matching with monetary transfers using discretization, i.e. the prices are chosen from a discrete set rather than from a continuum). Their approach can be considered an extension of the deferred acceptance algorithm of Gale and Shapley for college admission and stable marriageGale and Shapley (1962). Kelso and Crawford state their problem in the context of matching workers to firms. They introduce the notion of *"Gross Substitutes"*(GS) and show that if firms' preferences satisfy GS then the core is non-empty. Later on, Hatfield and MilgromHatfield and Milgrom (2005) presented a unified framework of many-to-one matching with contracts which subsumes the Kelso-Crawford model. Their approach is also based on discretization. They replace the finite set of discrete prices with a finite set of contracts where a contract could include any general term which may include a monetary transfer amount as well. They describe their model in the context of hospitals and doctors and show that if hospitals preferences over the set of possible contracts satisfy GS and also if doctors have strict preferences over the set of contracts then the core is non-empty. They show that the set of core outcomes form a lattice and that the infimum of the lattice correspond to the doctor optimal outcome while the supremum of the lattice correspond to the optimal outcome for the hospitals. They provide an iterative procedure for finding the core outcomes based on the discrete version of Tarski's fixed point theorem. They also characterize another condition which they call the *"Law of Aggregate Demand"* under which the doctor optimal outcome is also group strategyproof for the doctors. Recently, Hatfiled and Kominers Hatfield and Kominers (2010) generalized this to many-to-many matchings with contracts.

Leonard Leonard (1983) first showed that in one-to-one markets with quasilinear

[1] Scarf's proof actually provides an algorithm based on the pivoting algorithm of Lemke and Howson Lemke and Howson (1964). When combined with Quinzii's construction, that would lead to a construction that requires $2^{O(n!)}$ operations which runs on a matrix with $O(n!)$ columns. Nevertheless, the resulting algorithm is more of an exhaustive search algorithm and does not provide any insight into the equilibrium structure.

utilities, prices at the lowest competitive equilibrium equal VCG payments. Gul and StacchettiGul and Stacchetti (1999) studied many-to-one matchings in the context of allocation of indivisible goods to consumers with quasi-linear utilities. Their model differs from the model of Kelso and Crawford in that they do not require discrete prices but instead require the utilities to be quasi-linear in money. They show the existence of competitive equilibria given that consumers' preferences satisfy GS. They also show that not only is GS sufficient but it is also necessary. Similarly, Bikhchandani and Mamer Mamer (1997) showed the existence of competitive equilibria for the same model but without indivisibility using a different approach. Their proofs crucially needs the quasilinearity of utilities and their approach cannot be extended to general utilities.

Ausubel and Milgrom Ausubel and Milgrom (2002) also studied the many-to-one matching in the the context of allocation of indivisible goods to consumers with quasi-linear utilities. They propose an ascending package auction to compute the outcome. They assume that all the goods are initially owned by one seller and as such the set of competitive equilibria is only a strict subset of the core. They consider the core outcomes and not just the competitive equilibria. They present an ascending package auction that always results in a core outcome even in the absence of GS preferences. They show that if consumers' preferences satisfy GS then the outcome of their auction coincides with the VCG outcome. More specifically, they show that their auction precisely computes the VCG outcome whenever the VCG outcome is in the core. They also show that GS is the the necessary and sufficient condition for the VCG outcome to be in the core. Their auction, however, requires payments to be chosen from a finite discrete set. Their setting can be modeled as a special case of the Milgrom and Hatfield matching with contracts Hatfield and Milgrom (2005). Their proofs also crucially depend on quasilinearity of utilities.

There are also related work that consider one-to-one matching markets with quasi-linear utilities and hard budget constraint. Aggarwal et al.Aggarwal et al. (2009), consider this problem in the context of Ad-Auctions with advertisers having slot specific hard budget constraints. They prove the existence of a budget-feasible competitive equilibrium and present a truthful auction mechanism based on that. They present an extension of the Hungarian methodKuhn (1956) for computing the equilibrium and their proofs are based on this construction. Ashlagi et al. Ashlagi et al. (2010) also consider a similar problem but they assume a single hard budget constraint and a single value per click for each advertiser and require separable click through rates. In subsection 6.4.2, we discuss the major difficulty of dealing with soft budget constraints as opposed to hard budget constraints.

6.3 Our Contribution

In this chapter we study the class of competitive equilibria in unit demand markets with general utility function in continuous setting. We must emphasis that all of the earlier works except for Demange and Gale (1985); Quinzii (1984); Gale (1984) either crucially require quasilinear utilities or work in a discrete setting. Our main contributions are the following:

- In Theorem 45, we present a construction using an inductive characterization of prices/payoffs at the competitive equilibria that reveals interesting similarities between VCG payments for quasilinear utilities and the prices at the lowest competitive equilibrium for general utilities. In Theorem 46 we give a simple proof for group strategyproofness based on a critical property of the lowset/highest competitive equilibria. All of the earlier works only proved the existence of a competitive equilibrium using either fixed point theorems or discretization without providing an exact characterization of the equilibria. We present a simple characterization that has a natural interpretation. Our characterization provides a deeper insight into the structure of the equilibria.

- In section 6.6, we suggest a mechanism for ad-auctions that can naturally combine both pay per click and pay per impression advertisers in a general setting in which advertisers could submit a separate utility function for each slot as a price of that slot. These utility functions could be any arbitrary function[2] of the price of that slot. Furthermore, our mechanism is group strategyproof even

[2]It has to be continuous and decreasing in the price of the slot and should become non-positive for a high enough price.

if search engine and advertisers have different estimates of clickthrough rates. This also answers an open question raised by Aggarwal et al. (2009). Furthermore, our mechanism is welfare maximizing in the sense that it maximizes the combined welfare of the search engine and any group of advertisers who have quasilinear utilities and agree with the search engine on clickthrough rates and assuming that the search engine has the correct clickthrough rates. In particular, if all advertisers have quasilinear utilities and agree with the search engine on clickthrough rates, then the outcome of our mechanism coincides with the VCG outcome.

6.4 Model and Main Results

In this section, we consider competitive equilibria in two sided markets with goods on one side and buyers on the other side. Later in section 6.5, we consider the more general model with agents on both sides and show that it can be reduced to the simpler buyer/good model. In subsection 6.4.1, we formally define the problem and our notation. In subsection 6.4.2, we explain the main challenges of dealing with non-quasilinear utilities and explain why it is much harder to prove these results for non-quasilinear utilities compared to their quasilinear counterparts. In subsection 6.4.3, we present our main general theorems.

6.4.1 Model

In this subsection, we formally define the problem and our notation.

We denote by $M = (I, J, \{u_i^j\})$, a market M with the set of unit demand buyers I and the set of goods J such that the utility of buyer i for receiving good j at price x is given by the monotonically decreasing function $u_i^j(x)$ which is privately known by buyer i. We assume that for a large enough x, $u_i^j(x)$ becomes zero or negative[3]. We will use $p_i^j(\cdot)$ to denote the inverse of $u_i^j(\cdot)$. Next, we formally define a *Competitive*

[3] This is to ensure that $u_i^j(\cdot)$ is invertible. We also require the domain and the range of $u_i^j(\cdot)$ to cover the whole $\mathbb{R}$. Since at an equilibrium both x and $u_i^j(x)$ are positive, we can easily extend the domain of any $u_i^j(\cdot)$ to the whole $\mathbb{R}$ to meet this requirement.

Equilibrium.

Definition 38 (Competitive Equilibrium)**.** *Given a market* $M = (I, J, \{u_i^j\})$*, a* "Competitive Equilibrium" *of* M *is an assignment of prices to goods together with a feasible matching of goods to buyers such that each buyer receives her most preferred good at the assigned prices and every unmatched good has a price of* 0*. Formally, we say that* $W = (\mathbf{p}, \mathbf{u})$ *is a competitive equilibrium of* M *with price vector* $\mathbf{p}$ *and payoff vector* $\mathbf{u}$ *if and only if there exists a* "Supporting Matching" $\boldsymbol{\mu}$ *such that the following conditions hold. We use* $\boldsymbol{\mu}(i)$ *to denote the good that is matched to buyer* i*:*

$$\forall i \in I, \forall j \in J: \quad \begin{cases} \mathbf{u}_i = u_i^j(\mathbf{p}^j) & j = \boldsymbol{\mu}(i) \\ \mathbf{u}_i \geq u_i^j(\mathbf{p}^j) & j \neq \boldsymbol{\mu}(i) \end{cases} \tag{6.1}$$

$$\forall i \in I: \quad \boldsymbol{\mu}(i) = \emptyset \Rightarrow \mathbf{u}_i = 0 \tag{6.2}$$

$$\forall j \in J: \quad \boldsymbol{\mu}^{-1}(j) = \emptyset \Rightarrow \mathbf{p}^j = 0 \tag{6.3}$$

$$\forall i \in I, \forall j \in J: \quad \mathbf{u}_i \geq 0, \mathbf{p}^j \geq 0 \tag{6.4}$$

We denote an unmatched buyer or good by $\boldsymbol{\mu}(i) = \emptyset$ *or* $\boldsymbol{\mu}^{-1}(j) = \emptyset$*. Throughout this chapter, instead of explicitly writing* $W = (\mathbf{p}, \mathbf{u})$ *, we use* $p(W)$ *and* $u(W)$ *to denote the price vector* $\mathbf{p}$ *and the payoff vector* $\mathbf{u}$ *at* W*. We also use* $\mu(W)$ *to denote a supporting matching for* W*. Note that there could be more than one supporting matching for a given* W *so we assume* $\mu(W)$ *may return any one of them. We will denote the set of all competitive equilibria for a market* M *by* $\mathbb{W}(M)$*.*

6.4.2 The Main Challenges of Non-Quasilinear Utilities

In this subsection, we explain the main challenges of dealing with non-quasilinear utilities. First, it is helpful to explain the connection between VCG and competitive equilibria in unit-demand markets with quasilinear utilities.

VCG is based on maximizing the social welfare which is defined as the sum of the utilities. Taking the sum of quasilinear utility functions makes sense because they are measured in the same units and the payment terms cancel out. However, with general utilities, social welfare is not well-defined since different agents' utilities are not measured in the same units and are therefore non-transferrable (i.e. transferring \$1 from one agent to another does not transfer the same amount of utility). In our problem, with quasilinear utilities, the utility functions would be of the form $u_i^j(x) = v_i^j - x$ where v_i^j is the value of the agent i for good j. We could then construct a complete bipartite graph with agents and goods in which each edge (i, j) has a weight of v_i^j. A social welfare maximizing mechanism like VCG would pick a maximum weight matching in this graph which is also captured by the following LP:

$$
\begin{array}{rrrr}
\textbf{Primal:} & \max \cdot \sum_{i \in I} \sum_{j \in J} v_i^j \mathbf{x}_i^j & \textbf{Dual:} & \min \cdot \sum_{i \in I} \mathbf{u}_i + \sum_{j \in J} \mathbf{p}^j \\
\forall i \in I: & \sum_{j \in J} \mathbf{x}_i^j \leq 1 & \forall i \in I, \forall j \in J: & \mathbf{u}_i + \mathbf{p}^j \geq v_i^j \\
\forall j \in J: & \sum_{i \in I} \mathbf{x}_i^j \leq 1 & & \mathbf{u}_i \geq 0 \\
& \mathbf{x}_i^j \geq 0 & & \mathbf{p}^j \geq 0
\end{array}
\tag{6.5}
$$

Notice that there is a one-to-one correspondence between solutions of the dual program and the competitive equilibria (observe that by complementary slackness, if $\mathbf{x}_i^j > 0$ then $\mathbf{u}_i = v_i^j - \mathbf{p}^j$). It is not hard to show that the prices at the lowest competitive equilibrium (the one that has the lowest prices) correspond to the VCG payments. Furthermore, any competitive equilibrium of the market leads to a social welfare maximizing allocation (this follows from strong duality). To compute a maximum weight matching in this graph we can use the *Hungarian Method* Kuhn (1956). Interestingly, the Hungarian method is equivalent to the following ascending price auction proposed by Demange, Gale and SotomayorDemange et al. (1986):

Definition 39 (Ascending Price Auction). *Set all the prices equal to* 0. *Find a minimally over demanded subset of goods, i.e. a subset T of goods such that there is a subset S of the buyers who strictly prefer the goods in T at the current prices and $|S| > |T|$. Increase the prices of goods in T at the same rate until one of the buyers in S becomes indifferent between a good outside of T and her preferred good in T. At that point, recompute the minimally over-demanded subset and repeat this process until there is no over demanded subset of goods.*

In fact, all of the existing methods for computing the lowest competitive equilibrium, that we are aware of, are based on running an ascending auction of the above form or a similar ascending auction. Furthermore, in all of the related work that are based on such ascending auctions, the proofs are heavily tied with the way the ascending auction proceeds and the fact that it stops in finite time. Essentially, all of these auctions work as follows: They advance the prices at some rate to the next point at which there is a change in the demand structure [4]. Then, they recompute the rates and repeat. For quasilinear utility functions, the ascending auction stop after $O(|I| + |J|^2)$ iterations (Each time the combinatorial structure of the demand changes we start a new iteration).

Unfortunately, these ascending auctions may not even terminate in finite time if utilities are not quasilinear. The problem occurs when we try to raise the prices of goods in set T. With quasilinear utilities, when we raise the prices of all the goods in T at the same rate, for buyers in S, the relative preferences over the goods in T do not change. However, that is not true for general utility functions. For general utilities, we may need to raise the prices of goods in T at different and possibly non-constant rates and even then the preferences of buyers in S over goods in T may change an unbounded number of times. We demonstrate this in the following example:

Example 1. *Suppose there are 3 goods and 4 buyers with utility functions as given in the following table in which $V \geq 2$ is some constant and x is the price of the corresponding good:*

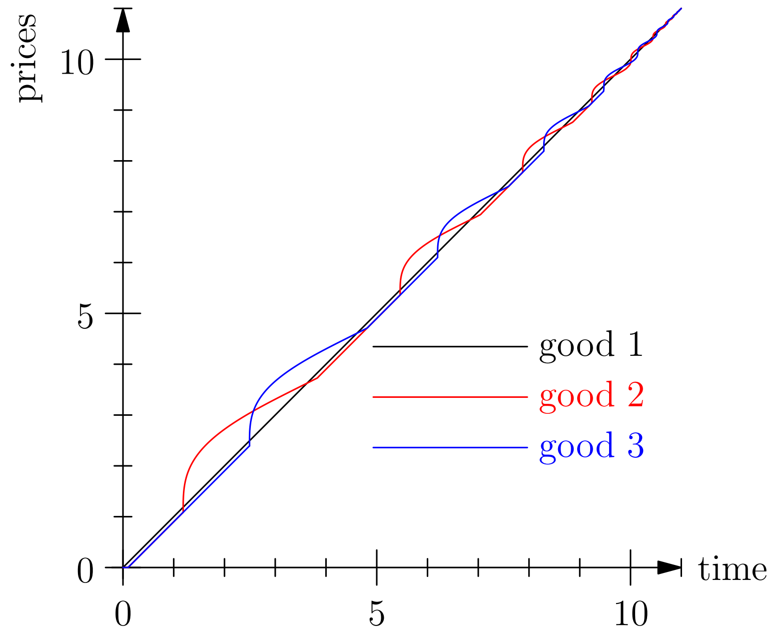
[4]Note that since there are no structural changes, the prices are essentially jumped discretely to the next point at which there is a change in the demand sets

	good 1	*good 2*	*good 3*
buyer 1	$V+1-x$	$V+1-x$	$V+1-x$
buyer 2	0	$V+1-x$	0
buyer 3	0	0	$V+1-x$
buyer 4	$V-x$	$V-c(x)$	$V-c'(x)$

All buyers have quasilinear utilities except buyer 4 *for whom* $c(x) = x + \frac{V-x}{V}\sin(V\log(V-x))$ *and* $c'(x) = x + \frac{V-x}{V}\cos(V\log(V-x))$*. Notice that both* $c(\cdot)$ *and* $c'(\cdot)$ *are strictly increasing in* x *if* $V \geq 2$*, so all utility functions are strictly decreasing in prices. Figure 6.1 shows the prices of goods during the ascending auction. We should emphasis that in this particular example, the ascending path of prices is unique. The ascending auction can only increases the prices of goods that are over demanded, i.e., demanded by at least two buyers. Furthermore, it can only raise the price of a good to the point where the demand of that good is about to drop to* 1*. Therefore, for every good with a positive price during the auction there should be at least a demand of* 2*. Observe that the demand set of buyer* 1 *and* 4 *changes an infinite number of times during the ascending auction. Specifically, the demand set of both buyer* 1 *and* 4 *include good* 1 *at all times. However, the demand set of buyer* 1 *includes good* 2 *and/or good* 3 *only at the times in which the price curves of those goods overlap with the price curve of good* 1*. Similarly, the demand set of buyer* 4 *includes good* 2 *and/or good* 3 *only at the times in which the price curves of those goods do not overlap with the price curve of good* 1*. Observe that the demand structure changes an infinite number of times as the price of the goods approach* V*. So an ascending auction does not stop in finite time.*

The previous example, although contrived, illustrates what could go wrong with ascending auctions and constructive proofs that are based on them. In general, ascending auctions are very sensitive to the structure of utility functions. Later, in Theorem 45, we present a direct way of computing the lowest competitive equilibrium without running an ascending auction.

Hard budgets vs. Smooth budgets: Notice that with hard budget constrains, utility functions are quasilinear except at the point where buyers hit their budget limits. Therefore, the issue that was outlined in Example. 1 does not arise with



The blue and red curves have been slightly shifted down to make the black curve visible.

Assume $V = 11$ and that the price of good 1 is increased at the rate of 1.

Figure 6.1: Prices of goods in the ascending auction of Example. 1.

quasilinear utilities and hard budget constraints. In fact, ascending auctions with hard budget constraints converge almost as fast as ascending auction with quasilinear utilities because each buyer may hit her budget limits at most $|J|$ times (once per each good) and beyond that they never demand that good again. This is what makes general non-quasilinear utilities much harder to work with compared to quasilinear utilities with hard budgets. It is worth mentioning that the related work of Aggarwal et al. (2009) and Ashlagi et al. (2010) are based on such ascending auctions.

6.4.3 Main Results

In this subsection, we state our main theorems that capture the important properties of competitive equilibria. Our main contributions in this section are Theorem 45 which characterizes the prices/payoffs at the higest/lowest competitive equilibria and

Theorem 46 that establishes the group strategyproofness of the mechanism that selects the lowest competitive equilibrium. Our characterization reveals a deep connection between the way VCG computes its payments and the way prices can be computed at the lowest competitive equilibria. We start by showing that the set of competitive equilibria form a lattice.

Theorem 44 (Equilibrium Lattice)**.** *For a given market $M = (I, J, \{u_i^j\})$, with the set of competitive equilibria $\mathbb{W}(M)$, we define a partial ordering as follows. For any two competitive equilibria $W, W' \in \mathbb{W}(M)$, we say $W \leq W'$ iff $p(W) \leq p(W')$ (or equivalently $u(W) \geq u(W')$)*[5]*. The partially ordered set $(\mathbb{W}(M), \leq)$ is a complete lattice. The* inf *and* sup *operators on the lattice are defined as follows. Let $W_{\text{inf}} = \inf(W, W')$ and $W_{\text{sup}} = \sup(W, W')$. Both W_{inf} and W_{sup} are valid competitive equilibria (we provide a supporting matching for each one):*

$$W_{\text{inf}} : \begin{cases} p^j(W_{\text{inf}}) = \min(p^j(W), p^j(W')) \\ u_i(W_{\text{inf}}) = \max(u_i(W), u_i(W')) \\ \boldsymbol{\mu}_{\text{inf}}(i) = \begin{cases} \boldsymbol{\mu}(i) & \mathbf{u}_i \geq \mathbf{u}'_i \\ \boldsymbol{\mu}'(i) & \mathbf{u}_i < \mathbf{u}'_i \end{cases} \end{cases} \qquad W_{\text{sup}} : \begin{cases} p^j(W_{\text{sup}}) = \max(p^j(W), p^j(W')) \\ u_i(W_{\text{sup}}) = \min(u_i(W), u_i(W')) \\ \boldsymbol{\mu}_{\text{sup}}(i) = \begin{cases} \boldsymbol{\mu}(i) & \mathbf{u}_i < \mathbf{u}'_i \\ \boldsymbol{\mu}'(i) & \mathbf{u}_i \geq \mathbf{u}'_i \end{cases} \end{cases} \tag{6.6}$$

In particular, the lattice has a unique minimum which we refer to as the lowest competitive equilibrium *(i.e. has the lowest prices and the highest payoffs) and a unique maximum which we refer to as the* highest competitive equilibrium *(i.e. has the highest prices and the lowest payoffs)*[6] *.*

Throughout the rest of the chapter, we use the lattice structure of the set of competitive equilibria without making explicit references to Theorem 44.

Before we present our theorems, we define the following notation. Note that we can fully specify a competitive equilibrium W by just specifying either $u(W)$ or $p(W)$.

[5]A vector is considered less than or equal to another vector if it is less than or equal to the other vector in every component

[6]It is not hard to show that $\mathbb{W}(M)$ is a closed compact set

Given either the price vector or the payoff vector, we can compute the other one by taking the induced prices/induced payoffs as defined next.

Definition 40 (Induced Payoffs $u(\mathbf{p})$, Induced Prices $p(\mathbf{u})$)**.** *We use $u(\mathbf{p})$ to denote the* "Induced Payoffs" *of buyers from price vector $\mathbf{p}$ which is the best payoff that each buyer can possibly get given the prices $\mathbf{p}$. Similarly, we use $p(\mathbf{u})$ to denote the* Induced Prices *of goods from the payoff vector $\mathbf{u}$. The formal definition is as follows (remember that $p_i^j(\cdot)$ is the inverse of $u_i^j(\cdot)$):*

$$u_i(\mathbf{p}) = \max(\{u_i^j(\mathbf{p}^j) | j \in J\} \cup \{0\}) \tag{6.7}$$

$$p^j(\mathbf{u}) = \max(\{p_i^j(\mathbf{u}_i) | i \in I\} \cup \{0\}) \tag{6.8}$$

It is easy to see that if W is a competitive equilibrium then $u(W) = u(p(W))$ and $p(W) = p(u(W))$ [7]. Throughout this chapter, we use bold letters $\mathbf{p}$ and $\mathbf{u}$ to denote variables representing price/payoff vectors and non-bold letters p and u to denote functions returning price/payoff vectors. The next theorem states the main result of this chapter. In what follows, $u_i(\mathbf{p})$ and $p^j(\mathbf{u})$ denote the induced payoff and induced price as defined in (6.7) and (6.8) respectively.

Theorem 45 (Inductive Equilibrium)**.** *Given a market $M = (I, J, \{u_i^j\})$, a competitive equilibrium always exists. Furthermore, the lowest and the highest competitive equilibria can be computed inductively as follows. Let $\underline{W}$ be the lowest and $\overline{W}$ be the highest competitive equilibrium of the market M. For an arbitrary buyer i and an arbitrary good j, let $\overline{W}_{-i}$ be the highest competitive equilibrium of the market M_{-i} (i.e. the market without buyer i) and let $\underline{W}^{-j}$ be the lowest competitive equilibrium of the market M^{-j} (i.e. the market without good j). The following inductive statements fully characterize the prices/payoffs at the lowest/highest competitive equilibrium of M:*

I. $u_i(\underline{W}) = u_i(p(\overline{W}_{-i}))$.

II. $p^j(\overline{W}) = p^j(u(\underline{W}^{-j}))$.

[7] The inverse is not true.

Furthermore, the following inequalities always hold:

resume $p^j(\underline{W}) \leq p^j(\overline{W}_{-i})$, *in particular, if* $j = \boldsymbol{\mu}(i)$ *then* $p^j(\underline{W}) = p^j(\overline{W}_{-i})$.

resume $u_i(\overline{W}) \leq u_i(\underline{W}^{-j})$, *in particular, if* $j = \boldsymbol{\mu}(i)$ *then* $u_i(\overline{W}) = u_i(\underline{W}^{-j})$.

Note that just 45.I and 45.II are enough to fully characterize the lowest/highest competitive equilibria because we can fully specify any competitive equilibrium by specifying either the prices or the payoffs. Intuitively, we can interpret them as the following:

- (45.I) We can compute the payoff of any buyer i at the lowest competitive equilibrium of M by doing the following. Remove i from the market. Compute the prices at the highest competitive equilibrium of the rest of the market. Then, bring buyer i back to the market. The payoff that buyer i gets from her most preferred good at these prices is equal to her payoff at the lowest competitive equilibrium of the market M.

- (45.II) We can compute the price of any good j at the highest competitive equilibrium of the market M by doing the following. Remove good j from the market. Compute the buyers' payoffs at the lowest competitive equilibrium of the rest of the market. Then, bring good j back to the market. Ask each of the buyers to name a price for good j that would give them the same payoff as what they get in the lowest competitive equilibrium of the market without good j. Take the maximum among the named prices and that will be the price of good j at the highest competitive equilibrium of the whole market.

Next, we combine the above two characterization to reveal a striking similarity between the prices of the lowest competitive equilibrium and VCG payments. Note that social welfare is not even well-defined for a market with general utilities so VCG is inapplicable.

By combining the (45.I) and (45.II) we get the following interpretations for the prices of goods at the lowest competitive equilibrium. WLOG, we give the interpretation for some arbitrary good j which is allocated to buyer i at the lowest competitive equilibrium. Notice the striking similarity between these interpretation and the VCG payments *"The price that buyer i pays for good j is the lowest price at which the rest of the market becomes indifferent between buying or not buying good j. i.e., the lowest price for good j such that there is a competitive equilibrium for the market without i and j such that no buyers would strictly prefer good j to her current allocation. In other words, the price that buyer i has to pay to get good j is equal to how much good j is worth to the rest of the market."*

Theorem 46 (Group Strategyproofness)**.** *A mechanism that uses the allocations/prices of the lowest competitive equilibrium is group strategyproof for buyers, meaning that there is no coalition of buyers that can collude and misreport their $u_i^j(\cdot)$ such that all of them get strictly higher payoffs (assuming that there are no side payments).*

In the rest of this section, we give a sketch of the proof of Theorem 45. We start by defining a *Tight Alternating Path*.

Definition 41 (Tight Alternating Path)**.** *Given a market* $M = (I, J, \{u_i^j\})$ *and a competitive equilibrium* W *of* M *with a supporting matching* $\mu(W)$*, we define a* Tight Alternating Path *with respect to* W *and* $\mu(W)$ *as follows. Consider the complete bipartite graph of buyers/goods. We say the edge* (i, j) *is tight iff* $u_i(W) = u_i^j(p^j(W))$*. A tight alternating path is a path consisting of tight edges where every other edge on the path belongs to* $\mu(W)$*. A tight alternating path may start at either a buyer or a good and may end at either a buyer or a good. In particular, the end points of the path may be unmatched in* $\mu(W)$*. For example consider a tight alternating path* $(i_1, j_1, i_2, j_2, i_3)$ *where* i_1 *is matched with* j_1 *and* i_2 *is matched with* j_2 *and* i_3 *is unmatched in* $\mu(W)$*. In particular that means* $u_{i_3}(W) = 0 = u_{i_3}^{j_2}(p^{j_2}(W))$*.*

Definition 42 (Demand Sets)**.** *Given a market* $M = (I, J, \{u_i^j\})$*, we denote the demand set of a subset of buyers* S *at prices* $\mathbf{p}$ *by* $D_S(\mathbf{p})$*. Similarly, we denote the demand set of a subset of goods* T *at payoffs* $\mathbf{u}$ *by* $D^T(\mathbf{u})$*. Formally:.*

$$D_S(\mathbf{p}) = \{j \in J | \exists i \in S : u_i^j(\mathbf{p}^j) = u_i(\mathbf{p})\} \tag{6.9}$$

$$D^T(\mathbf{u}) = \{i \in I | \exists j \in T : p_i^j(\mathbf{u}_i) = p^j(\mathbf{u})\} \tag{6.10}$$

For a competitive equilibrium W *of* M*, we use* $D_S(W)$ *to denote* $D_S(p(W))$ *and* $D^T(W)$ *to denote* $D^T(u(W))$*.*

Lemma 11 (Tightness)**.** *Given a market* $M = (I, J, \{u_i^j\})$*, and a competitive equilibrium* W *of* M*:*

- W *is the lowest competitive equilibrium of* M *iff for every subset* T *of the goods with strictly positive prices we have* $|D^T(W)| \geq |T| + 1$*, i.e. at least* $|T| + 1$ *buyers are interested in* T*.*
- W *is the highest competitive equilibrium of* M *iff for every subset* S *of the buyers with strictly positive payoffs we have* $|D_S(W)| \geq |S| + 1$*, i.e. the buyers in* S *are interested in at least* $|S| + 1$ *goods.*

To see why Lemma 11 is true intuitively, assume that W is the lowest competitive equilibrium of a market M but there is a subset T of goods such that $|D^T(W)| = |T|$.

Then, we could decrease the prices of goods in T down to the point where either a buyer out of $D^T(W)$ becomes indifferent between her current allocation and some good in T; or one of the goods in T hit the price of 0. But then, we get a competitive equilibrium less than W which is a contradiction. Although this lemma seems very intuitive, its formal proof turns out to be quite challenging. Most of the appendix is devoted to proving this lemma. The next lemma states a critical property of the lowest/highest competitive equilibria.

Lemma 12 (Critical Alternating Paths)**.** *Given a market $M = (I, J, \{u_i^j\})$, and a competitive equilibrium W of M with a supporting matching $\boldsymbol{\mu}$:*

- *Iff W is the highest competitive equilibrium of M then for any good j there exists a tight alternating path from j to a buyer with a payoff of 0 or to an unmatched good. The alternating path must start with a matching edge or be of length 0.*
- *Iff W is the lowest competitive equilibrium of M then for any buyer i there exists a tight alternating path from i to a good with a price of 0 or to an unmatched buyer. The alternating path must start with a matching edge or be of length 0.*

We refer to such a tight alternating path as a "Critical Alternating Path"*.*

Proof. We only prove the first statement since the second one is similar (completely symmetric): If either j is unmatched or the payoff of buyer who is matched to j is 0 we are done. Otherwise, we run the following algorithm while maintaining a subset T of goods and a subset S of buyers with strictly positive payoffs such that there is a tight alternating path from j to each good in T and each buyer in S and $\boldsymbol{\mu}(S) = T$. Initially, we set $T \leftarrow \{j\}$ and $S \leftarrow \{\boldsymbol{\mu}^{-1}(j)\}$. The repeating step is as follows: Since all buyers in S have strictly positive payoffs, we can apply Lemma 11 and argue that $D_S(W) \geq |S| + 1 = |T| + 1$. So, there must be a buyer i^* in S that has a tight edge to some good j' not in T. If either j' is unmatched or $i' = \boldsymbol{\mu}^{-1}(j')$ has a payoff of 0 then we are done. Otherwise, add j' to T and add i' to S and repeat. Note that we always find a critical alternating path in at most $|I| - 1$ iterations. The "only if" direction is trivial by applying Lemma 11. $\square$

Next, we give a sketch the proof of our main theorem. The complete proof can be found in the appendix.

Proof sketch of Theorem 45. We only give a sketch of the proof of (45.I) and (?). The proofs of (45.II) and (?) are completely symmetric to the other two.

The plan of the proof is as follows. We remove an arbitrary buyer i from the market and compute the highest competitive equilibrium of the rest of the market. We then show that the prices at the highest competitive equilibrium of the market without i leads to a valid competitive equilibrium for the whole market (including buyer i) but with a possibly different matching. We also show that the induced payoff of buyer i from these prices is the same as her payoff at the lowest competitive equilibrium of the whole market. The detail of the construction is as follows.

Choose an arbitrary buyer $i \in I$. Let M_{-i} denote the market without buyer i and let $\overline{W}_{-i}$ be the highest competitive equilibrium of the market M_{-i}. Note that the market M_{-i} is of size $|I| + |J| - 1$ so by inductively applying Theorem 45 to M_{-i} we can argue that there exists a competitive equilibrium for the market M_{-i}, so the highest competitive equilibrium of M_{-i} is well-defined. Let $\mathbf{p} = p(\overline{W}_{-i})$ be the prices at $\overline{W}_{-i}$. We claim that using the prices $\mathbf{p}$ for the market M leads to a valid competitive equilibrium W. In particular, all the prices/payoffs in W are the same as the prices/payoffs in $\overline{W}_{-i}$ and also the payoff of buyer i is $u_i(\mathbf{p})$ however the matching might be different. To obtain a supporting matching for W, we start with a supporting matching for $\overline{W}_{-i}$ and modify it as follows. If $u_i(\mathbf{p}) = 0$ then we can leave buyer i unmatched and the matching does not need to be changed. Otherwise, if $u_i(\mathbf{p}) > 0$ then let j be the good from which buyer i achieves her highest payoff at the current prices, i.e. $u_i(\mathbf{p}) = u_i^j(\mathbf{p}^j)$. By applying Lemma 12 to the market M_{-i}, we can argue that there is a tight alternating path from j either to an unmatched good with a price of 0 or to a buyer with a payoff of 0. In both cases, we can match good j to buyer i and then switch the matching edges along the alternating path to get a new matching that supports W. Note that if the alternating path ends in a buyer with a payoff of 0 then the last edge of the alternating path was a matching edge and that buyer is now unmatched in the new matching, but she still has a payoff of 0. The complete proof can be found in the appendix.

Observe that from the view point of buyer i, this is a posted price mechanism with posted price vector $p(\overline{W}_{-i})$ which does not depend on i's reported utility (note that the choice of i was arbitrary).

□

6.5 More General Models

In this section, we consider competitive equilibria in markets where both sides consist of agents with general utility function. We show a reduction from this model to the simpler model with goods and buyers. We also characterize a class of price discriminating truthful mechanisms based on these markets.

Consider the matching markets of the form $M = (I, J, \{u_i^j\}, \{q_i^j\})$ with two sets of agents I and J such that if $i \in I$ and $j \in J$ are matched and x amount of money is transferred from i to j, then the utility of i is given by $u_i^j(x)$ and the utility of j is given by $q_i^j(-x)$ (note that x might be negative). We assume u_i^j's and q_i^j's have the same properties we assumed in the buyer/good model (e.g. continuous, decreasing, etc). A competitive equilibrium is also defined similarly. Despite its apparent generality, this model can be reduced to the buyer/good model as the following theorem states:

Corollary 6. *Given a market $M = (I, J, \{u_i^j\}, \{q_i^j\})$ with agents on both sides, we can construct a market $M' = (I, J, \{u'^j_i\})$ with buyers/goods in which $u'^j_i(\cdot) = u_i^j(-q_i^{j(-1)}(\cdot))$. Then, every competitive equilibrium in M' corresponds to a competitive equilibrium in M and vice versa with the exact same payoffs. Therefore, all of the results that we proved in the previous sections carry over to these markets. Furthermore, the mechanism that selects the lowest competitive equilibrium of M' (which is also the lowest competitive equilibrium of M) is group strategyproof for agents of type I. Note that we could change the role of I and J and get a similar results for agents of type J. Observe that the lowest competitive equilibrium of agents of type J is the highest competitive equilibrium of agents of type I and vice versa.*

Next, we present a class of price discriminating truthful mechanisms based on the same idea:

Corollary 7. *Given a market $M = (I, J, \{u_i^j\})$, the seller(s) can personalize the price for each good/buyer by applying an arbitrary continuous and increasing function $g_i^j(\cdot)$ to the primary price of the good j. In other words, if the price of a good $j \in J$ at the equilibrium is $\mathbf{p}^j$ then the price observed by agent $i \in I$ is $g_i^j(\mathbf{p}^j)$. Note that $g_i^j(\cdot)$'s should be fixed in advance and should not depend on the reports of buyers. It is easy to see that every competitive equilibrium in this market correspond to a competitive equilibrium in the market $M' = (I, J, \{u'^j_i\})$ where $u'^j_i(\cdot) = u_i^j(g_i^j(\cdot))$ and vice versa. Consequently, all of the results that we proved in the previous sections carry over to these markets/mechanisms.*

Intuitively, the above two theorems suggest that we can write the utility functions of the agents in set I in terms of the payoffs of the agents in set J (or in terms of the primary prices in the case of personalized prices). We can then treat the agents on set J as goods and their payoffs as the prices of these goods. Note that to maintain the group strategyproofness, it is crucial that $g_i^j(\cdot)$'s be fixed in advance and not depend on the reports of buyers. In the next section, we present a practical application of this idea.

6.6 Application to Ad-Auctions

In this section, we present a truthful mechanism for Ad-auctions that combines pay per click (a.k.a charge per click or CPC) and pay per impression (a.k.a CPM) advertisers with general utility functions. In particular, our mechanism is group strategyproof regardless of whether the search engine uses the correct clickthrough rates or whether advertisers agree with the search engine on clickthrough rates.

We formally define our model as follows. Given a set of advertisers I and a set of slots J, we assume that utility of advertiser i from slot j is given by $u_i^j(x)$[8] where x is payment per click for CPC advertisers and payment per impression for CPM advertisers. We say that a CPC advertiser i has standard utility function if for all slots j: $u_i^j(x) = c_i^j(v_i^j - x)$ in which v_i^j is the advertiser's value for a click on slot j and

[8] $u_i^j(x)$ must be continuous and decreasing in x and for a high enough x it should become non-positive

c_i^j is the advertiser's belief about her clickthrough rate (CTR). We also say that a CPM advertiser i has standard utility function if for all slots j: $u_i^j(x) = v_i^j - x$ in which v_i^j is the advertiser's value for a click on slot j. Furthermore, we assume that search engine believes that the CTR of advertiser i on slot j is $\hat{c}_i^j$ which might be different from c_i^j(i.e., advertisers and search engine could disagree). Furthermore, we assume that v_i^j and c_i^j are advertiser's private information but $\hat{c}_i^j$ is publicly announced.

Before we explain our mechanism, let us consider what happens if we applied VCG to this setting, assuming that we only have CPC advertisers with standard utility function. We get the following LPs. The primal computes the social welfare maximizing allocation while the dual computes prices/payoffs:

$$\begin{array}{llcll}
\textbf{Primal:} & \max \cdot \sum_{i \in I} \sum_{j \in J} c_i^j v_i^j \mathbf{x}_i^j & \qquad & \textbf{Dual:} & \min \cdot \sum_{i \in I} \mathbf{u}_i + \sum_{j \in J} \mathbf{p}^j \\
\forall i \in I: & \sum_{j \in J} \mathbf{x}_i^j \leq 1 & & \forall i \in I, \forall j \in J: & \mathbf{u}_i + \mathbf{p}^j \geq c_i^j v_i^j \\
\forall j \in J: & \sum_{i \in I} \mathbf{x}_i^j \leq 1 & & & \mathbf{u}_i \geq 0 \\
& \mathbf{x}_i^j \geq 0 & & & \mathbf{p}^j \geq 0
\end{array} \tag{6.11}$$

The set of solutions to the dual program would be the set of competitive equilibria of the market and the one with the lowest prices would correspond to the lowest competitive equilibrium which would also coincide with the VCG payments/payoffs. However, the problem is that payments should be charged *per click* while $\mathbf{p}^j$ represents the expected payment, i.e., payment *per impression.* So, per click payments are given by $\mathbf{p}^j/\hat{c}_i^j$. However, by dividing by $\hat{c}_i^j$, we lose the strategyproofness guarantee if c_i^j and $\hat{c}_i^j$ are not the same (i.e., if advertisers and search engine have different estimates about clickthrough rates). Note that we cannot use $\mathbf{p}^j/c_i^j$ either because then advertisers have the incentive to untruthfully report a higher c_i^j which would give them a higher chance of winning a better slot and at the same time would lower their payment. Next, we present a mechanism that also addresses this problem.

Mechanism 1. *Compute the lowest competitive equilibrium of the market* $M = (I, J, \{u_i^j\})$ *using the following personalized prices by applying Corollary 7. For each* *advertiser i and slot j we define a personalized price $g_i^j(\cdot)$ as follows. If i is a CPC advertiser then we define $g_i^j(x) = x/\hat{c}_i^j$ in which $\hat{c}_i^j$ is the estimate of the search engine for the clickthrough rate of advertiser i on slot j. If i is a CPM advertiser then we define $g_i^j(x) = x$.*

Remark 4 (Interpretation of mechanism 1)**.** *We can conceptually reinterpret this mechanism as an ascending auction as follows. Initially, we assign a primary price of 0 to every slot and during the auction whenever the demand for a slot is more that one, we increase the primary price of that slot. At any time during the auction, each advertiser demands one of the slots at the current prices. However, different advertisers see different prices. At any point during the auction, advertiser i observes a price of $g_i^j(\mathbf{p}^j)$ for slot j in which $\mathbf{p}^j$ is the primary price of the slot j. The auction stops when there is no over demanded slot. Intuitively, $\mathbf{p}^j$ denotes the expected revenue of the search engine from slot j and $g_i^j(\mathbf{p}^j)$ is the price that advertiser i has to pay for each click so that the search engine makes $\mathbf{p}^j$ in expectation.*

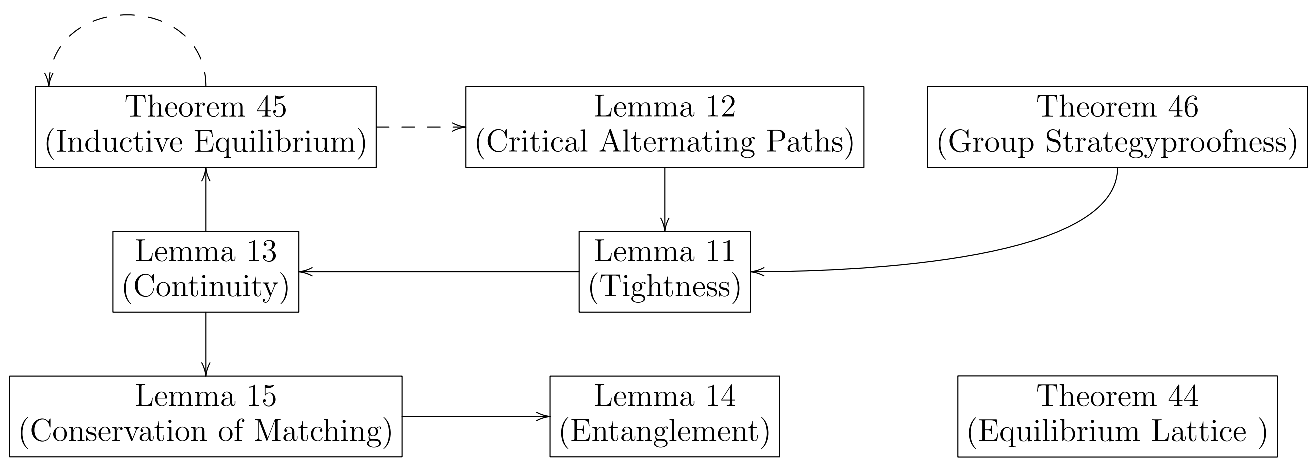
Next theorem summarizes the important properties of the above mechanism.

Theorem 47. *Mechanism 1 is group strategyproof and also maximizes the social welfare in the following sense. Let A be a group of advertisers with standard utility functions who also agree with the search engine on the CTRs and let A' be the rest of the advertisers. Let s denote the search engine. Then mechanism 1 maximizes the welfare of $\{s\} \cup A$ and also the presence of A' may not decrease the welfare of $\{s\} \cup A$. In particular, if all advertisers have standard utility functions and agree with the search engine on the CTRs then the outcome of this mechanism coincides exactly with the VCG outcome.*

Notice that mechanism 1 is group strategyproof regardless of whether the search engine and advertisers have the same estimates about the clickthrough rates. This answers an open question raised by Aggarwal et al. (2009). As for existing CPC ad-auction mechanisms, that we are aware of, incentive compatibility relies on everyone agreeing on clickthrough rate estimates made by the search engine.

6.7 Other Results & Omitted Proofs

In this section, we present the missing proofs and several other results. Before we proceed, we should mention that some of our theorems/lemmas mutually depend on each other. However, this does not create a cycle in the proofs. The following diagram illustrates the dependencies in the proofs. A solid arrow from A to B means that when A is invoked on a market of size n, the proof of A invokes B on a market of the same size. A dashed arrow from A to B means that A invokes B on a market of strictly smaller size. We can then prove all of the lemmas/theorems by induction on the size of the market. i.e. we assume that all of the lemmas/theorem are true for markets of size less than n and then all of the lemmas/theorems can be proved for markets of size n.



(6.12)

Note that Theorem 44 is implicitly used by most of the other lemmas/theorems so we didn't display the dependencies on it in the above diagram.

We proceed by defining *Bounded Competitive Equilibrium*:

Definition 43 ($(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-Bounded Competitive Equilibrium)**.** *Given a market $M = (I, J, \{u_i^j\})$ and a lower bound price vector $\underline{\mathbf{p}} \geq \mathbf{0}$ and a lower bound payoff vector $\underline{\mathbf{u}} \geq \mathbf{0}$, we say that W is a* $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium *of M iff W is a competitive equilibrium of M and $p(W) \geq \underline{\mathbf{p}}$ and $u(W) \geq \underline{\mathbf{u}}$.*

Note that for a given $\underline{\mathbf{u}}$ and $\underline{\mathbf{p}}$, the $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibria of M form a complete lattice which is a complete sublattice of all of the competitive equilibria of M. In particular, there is a lowest and a highest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M. Notice that for arbitrary $\underline{\mathbf{u}}$ and $\underline{\mathbf{p}}$, a $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium does not necessarily exist.

The next lemma provides the basic ingredient for the proof of Lemma 11.

Lemma 13 (Continuity)**.** *Assume a market $M = (I, J, \{u_i^j\})$ with $|I| = |J|$ and lower bounds $\underline{\mathbf{p}} \geq \mathbf{0}$ and $\underline{\mathbf{u}} \geq \mathbf{0}$ on the prices/payoffs, such that a $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium exists. Then, at the lowest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium, there exists at least one good $j^* \in J$ whose price is exactly equal to its lower bound (i.e. $\underline{\mathbf{p}}^{j^*}$). Similarly, at the highest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium, there exists at least one buyer $i^* \in I$ whose payoff is exactly equal to her lower bound (i.e. $\underline{\mathbf{u}}_{i^*}$).*

To give more intuition on the above statements, consider the following immediate corollary which we can derive by setting no lower bounds (i.e. a lower bound of $\mathbf{0}$) on the prices/payoff:

Corollary 8. *Given a market $M = (I, J, \{u_i^j\})$ with $|I| = |J|$ the following statements are always true:*

- *At the lowest competitive equilibrium, there is at least one good that has a price of* 0.
- *At the highest competitive equilibrium there is at least one buyer that achieves a payoff of* 0.

As another immediate corollary, Lemma 13 shows that there is a continuum of equilibria between the lowest and the highest competitive equilibria:

Corollary 9. *Given a market $M = (I, J, \{u_i^j\})$ with $|I| = |J|$, with $\underline{W}$ and $\overline{W}$ being the lowest and the highest competitive equilibria respectively, there is a continuum of equilibria between $\underline{W}$ and $\overline{W}$.*

Proof. Define $\underline{p}(t) = (1 - t)p(\underline{W}) + tp(\overline{W})$. Now by applying Lemma 13 we can get a continuum of equilibria by computing the lowest $(0, \underline{p}(t))$-bounded competitive equilibrium for each $t \in [0, 1]$. □

Before we can proceed further, we need two more lemmas. The next two lemmas show very basic properties of competitive equilibria in unit-demand markets:

Lemma 14 (Entanglement)**.** *Given a market $M = (I, J, \{u_i^j\})$ and a competitive equilibria W of M. If buyer i is matched with good j at W then the price of good j and the payoff of buyer i are entangled in any other competitive equilibrium of M which means at any other competitive equilibrium like W' if the price of good j is higher then the payoff of buyer i must be lower and vice versa. Note that this claim is true regardless of wether buyer i and good j are actually matched to each other in W'.*

Proof. Let W' be any other competitive equilibrium of M. Partition the buyers to S, S' and S'' such that buyers in S have a higher payoff at W, buyers in S' have a higher payoff at W' and buyers in S'' have the same payoff at both W and W'. Similarly, partitions the goods to T, T' and T'' such that goods in T have a higher price at W, goods in T' have a higher price and W' and goods in T'' have the same price at both W and W'. It is easy to show the following statements are true using the definition of competitive equilibria and the fact that both W and W' are competitive equilibria:

- At W, all buyers in S must be matched to goods in T' so $|S| \leq |T'|$.
- At W', all goods in T' must be matched to buyers in S so $|T'| \leq |S|$.

From the above statement, we can conclude $|S| = |T'|$ and buyers in S and goods in T' must be matched to each other at both equilibria. Similarly:

- At W, all goods in T must be matched to buyers in S' so $|T| \leq |S'|$.

- At W', all buyers in S' must be matched to goods in T so $|S'| \leq |T|$.

So, we can conclude $|S'| = |T|$ and buyers in S' and goods in T must be matched to each other at both equilibria. Furthermore, we can then conclude that buyers in S'' and goods in T'' may only be matched to each other. That proves the claim of the lemma. □

Lemma 15 (Conservation of Matching)**.** *Given a market $M = (I, J, \{u_i^j\})$, for any $i \in I$, if there exists a competitive equilibrium W of M at which buyer i has a strictly positive payoff (i.e. $u_i(W) > 0$) then buyer i is never unmatched in any competitive equilibrium of M. Similarly, for any $j \in J$, if there exists a competitive equilibrium W of M at which good j has a strictly positive price (i.e. $p^j(W) > 0$) then good j is never unmatched at any competitive equilibrium of M.*

Proof. Let W' be any other competitive equilibrium of the market. Partition the buyers to S, S', S'' and partition the goods to T, T', T'' as in the proof of Lemma 14. In the proof of that lemma, we showed that the matching pairs can only be from S and T' or S' and T or S'' and T''. Let i be any buyer with strictly positive payoff at W, if i has strictly positive payoff at W' then it must also be matched at W'. Otherwise if the payoff of buyer i is 0 at W' then i must belong to set S. We know that $|S| = |T'|$ and that all the goods in T' must have strictly positive prices so they must all be matched and therefore all buyers in S, including buyer i, must also be matched. The second statement can be proved by a similar argument for good j . □

Proof of Lemma 13. We only prove the first claim. The proof of the second claim is similar (completely symmetric). The plan of the proof is as follows:

First, we define a transformed market $M' = (I, J, \{u'^j_i\})$ with $u'^j_i(x) = u_i^j(x + \underline{\mathbf{p}}^j) - \underline{\mathbf{u}}_i$. We claim that there is a one-to-one mapping between $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibria of the original market and the competitive equilibria of the transformed market. Formally, a $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium W of M corresponds to a competitive equilibrium W' of M' such that $p(W') = p(W) - \underline{\mathbf{p}}$ and $u(W') = u(W) - \underline{\mathbf{u}}$ and $\mu(W') = \mu(W)$. We then show that there is competitive equilibrium of M' in which there is a good with a price of 0 which then means in the corresponding competitive equilibrium of the original market the price of that good is equal to its lower bound and therefore at the lowest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M the price of that good must also be equal to its lower bound which proves the claim.

We now prove that there is a good with a price of 0 at the lowest competitive equilibrium of M'. We choose an arbitrary buyer i from M' and remove it from the market. Let $\overline{W}_{-i}$ be the highest competitive equilibrium of the remaining market. By the assumption of the lemma, we know $|I| = |J|$ and so in $\overline{W}_{-i}$ there are more goods than there are buyers so there must be an unmatched good which we denote by j^*. Note that the price of j^* in $\overline{W}_{-i}$ must be 0. On the other hand, by applying Theorem 45 to M' and using (?) we have $p^j(\underline{W}) \leq p^j(\overline{W}_{-i})$ for every good j. Therefore, it must be that the price of j^* in $\underline{W}$ is also 0 and that completes the proof.

There is a subtlety that we should point out about the one-to-one mapping between the competitive equilibria of the original market and those of the transformed market. It is clear that every $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M can be transformed to a competitive equilibrium of M'. However, for the other direction, we need to show that all goods/buyers are matched, otherwise after applying the inverse transform we may end up with an unmatched good/buyer that has a positive price/payoff. To show that all buyers/goods are matched in every competitive equilibrium of M', we can apply Lemma 15. To apply that lemma, we only need to show that there is a competitive equilibrium of M' in which all goods have strictly positive prices and then by that lemma all the goods must always be matched (and so do all buyers because $|I| = |J|$). Notice that if there is no competitive equilibrium for M' in which all goods have strictly positive prices then either in every $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M there is a good whose price is equal to its lower bound or M has no $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium at all which either way trivially proves the claim of this lemma. □

Proof of Lemma 11. We only prove the first statement. The proof of the second statement is similar (completely symmetric).

First, we prove the "only if" direction. Assume that W is the lowest competitive equilibrium of M. For every subset T of goods with strictly positive prices, we prove that $D^T(W) \geq |T| + 1$, i.e. there are at least $|T| + 1$ buyers who are interested in some good in T. The proof is as follows. Since all the goods in T have strictly positive prices, they must all be matched. Let S be the subset of buyers that are matched to T. Notice that $S \subset D^T(W)$ and $|S| = |T|$. So, to complete the proof we only need to show that there is one more buyer not in S who is also interested in a good in T. Let $\underline{\mathbf{p}}$ be the prices induced by payoffs of buyers not in S, i.e. $\underline{\mathbf{p}}^j = \max_{i \in I-S} p_i^j(u_i(W))$. Similarly, let $\underline{\mathbf{u}}$ be the payoffs induced by the prices of goods not in T, i.e. $\underline{\mathbf{u}}_i = \max_{j \in J-T} u_i^j(p^j(W))$. Notice that W is a $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibria of the market $M' = (S, T, \{u_i^j\})$. Furthermore, if we replaced the part of W that corresponds to S and T with any other $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M', we would get a valid competitive equilibrium for M which implies that W must be the lowest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M' as well because otherwise we could replace the part of W corresponding to S and T with the lowest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M' and get a lower competitive equilibrium for M which would contradict W being the lowest competitive equilibrium of M. Because W is the lowest $(\underline{\mathbf{u}}, \underline{\mathbf{p}})$-bounded competitive equilibrium of M', by applying Lemma 13 to the market M', we can argue that there is a good $j^* \in T$ such that $p^{j^*}(W) = \underline{\mathbf{p}}^{j^*}$. Since all the goods in T, including j^*, have strictly positive prices, $\underline{\mathbf{p}}^{j^*}$ must also be strictly positive and because of the way we defined $\underline{\mathbf{p}}$ there must be a buyer i^* not in set S such that $\underline{\mathbf{p}}^{j^*} = p_{i^*}^{j^*}(\mathbf{u}_{i^*})$. That means i^* must be interested in good j^* and therefore $\{i^*\} \cup S \subset D^T(W)$ which proves that there are at least $|T| + 1$ buyers interested in the goods in T.

The proof of the "if" direction is trivial. The proof is by contradiction. Let W be a competitive equilibrium of M such that for every subset T of goods with strictly positive prices we have $D^T(W) \geq |T| + 1$. Let $\underline{W}$ be the lowest competitive equilibrium of M and assume that W and $\underline{W}$ are not the same. Let T consist of all the goods that have a higher price at W compared to $\underline{W}$. We know that there are at least $|T| + 1$ buyers interested in T at W and these buyers must have higher payoffs at $\underline{W}$ because the prices of the goods in T are strictly lower. Therefore, the goods assigned to these buyers at $\underline{W}$ must have lower prices and so there are at least $|T| + 1$ goods that have higher prices at W compared to $\underline{W}$ which contradicts the assumption that T was the set of all the goods that had higher prices at W. □

Next, we present the complete proof of Theorem 45:

Proof of Theorem 45. We only prove (45.I) and (?). The proofs of (45.II) and (?) are completely symmetric to the other two.

The plan of the proof is as follows. We remove an arbitrary buyer i from the market and compute the highest competitive equilibrium of the rest of the market. We then show that the prices at the highest competitive equilibrium of the market without i leads to a valid competitive equilibrium for the whole market (including buyer i) but with a possibly different matching. We also show that the induced payoff of buyer i from these prices is the same as her payoff at the lowest competitive equilibrium of the whole market. The detail of the construction is as follows.

Choose an arbitrary buyer $i \in I$. Let M_{-i} denote the market without buyer i and let $\overline{W}_{-i}$ be the highest competitive equilibrium of the market M_{-i}. Note that the market M_{-i} is of size $|I| + |J| - 1$ so by inductively applying Theorem 45 to M_{-i} we can argue that there exists a competitive equilibrium for the market M_{-i}, so the highest competitive equilibrium of M_{-i} is well-defined. Let $\mathbf{p} = p(\overline{W}_{-i})$ be the prices at $\overline{W}_{-i}$. We claim that using the prices $\mathbf{p}$ for the market M leads to a valid competitive equilibrium W. In particular, all the prices/payoffs at W are the same as the prices/payoffs at $\overline{W}_{-i}$ and also the payoff of buyer i is $u_i(\mathbf{p})$, however the matching might be different. To obtain a supporting matching for W, we start with a supporting matching for $\overline{W}_{-i}$ and modify it as follows. If $u_i(\mathbf{p}) = 0$ then we can leave buyer i unmatched and the matching does not need to be changed. Otherwise, if $u_i(\mathbf{p}) > 0$ then let j be the good from which buyer i achieves her highest payoff at the current prices, i.e. $u_i(\mathbf{p}) = u_i^j(\mathbf{p}^j)$. By applying Lemma 12 to the market M_{-i}, we can argue that there is a tight alternating path from j either to an unmatched good with a price of 0 or to a buyer with a payoff of 0. In both cases, we can match good j to buyer i and then switch the matching edges along the alternating path to get a new matching that supports W. Note that if the alternating path ends in a buyer with a payoff of 0 then the last edge of the alternating path was a matching edge and that buyer is now unmatched in the new matching, but she still has a payoff of 0.

Next, we prove each one of our claims:

- Proof of **Existence**: By induction, we assumed that M_{-i} must have a competitive equilibrium. We then took the highest competitive equilibrium of M_{-i} and constructed a competitive equilibrium for M. So M has a competitive equilibrium.

- Proof of (?) $p^j(\underline{W}) \leq p^j(\overline{W}_{-i})$: Notice that we constructed a competitive equilibrium W of the market M which has the same prices as the $\overline{W}_{-i}$. The prices at the lowest competitive equilibrium of M are no more than the prices at W so $p(\underline{W}) \leq p(W) = p(\overline{W}_{-i})$.

- Proof of (?) $p^j(\underline{W}) = p^j(\overline{W}_{-i})$ **when** $\boldsymbol{\mu}(i) = j$: Notice that since i and j are matched, if we remove both of them the rest of $\underline{W}$ is still a valid competitive equilibrium for M_{-i}^{-j}. Let $\underline{W}_{-i}^{-j}$ denote the lowest competitive equilibrium of M_{-i}^{-j}. Note that both $\underline{W}$ and $\underline{W}_{-i}^{-j}$ are valid competitive equilibria for M_{-i}^{-j} but $\underline{W}_{-i}^{-j}$ is the lowest, so the prices of goods $J - \{j\}$ might only be lower at $\underline{W}_{-i}^{-j}$ and so the payoffs of buyers $I - \{i\}$ might only be higher at $\underline{W}_{-i}^{-j}$ and so the price induced by buyers $I - \{i\}$ on good j might only be lower at $\underline{W}_{-i}^{-j}$ than the price induced by them on good j at $\underline{W}$. However, by applying Theorem 45 inductively on market M_{-i} and using (45.II), we get that $p^j(\overline{W}_{-i})$ is exactly the induced price of buyers $I - \{i\}$ on good j at $\underline{W}_{-i}^{-j}$. Therefore, $p^j(\overline{W}_{-i})$ must be less than or equal to the induced price on good j at $\underline{W}$ which is itself less that or equal to $p^j(\underline{W})$. On the other hand, from the previous paragraph we have $p^j(\underline{W}) \leq p^j(\overline{W}_{-i})$, so the two must be equal.

- Proof of (45.I) $u_i(\underline{W}) = u_i(p(\overline{W}_{-i}))$: If i is matched with j in $\underline{W}$ then $u_i = u_i^j(p^j(\underline{W}))$ and by the previous statement $p^j(\underline{W}) = p^j(\overline{W}_{-i})$. Therefore $u_i(\underline{W}) = u_i^j(p^j(\overline{W}_{-i})) = u_i(p(\overline{W}_{-i}))$. The last equality follows from the fact that we chose j to be the good from which buyer i obtains her highest payoff at prices $p(\overline{W}_{-i})$.

□

Proof of Theorem 46. To prove that the mechanism that uses the lowest competitive equilibrium for allocations/payments is group strategyproof for buyers, we must show that there is no coalition of buyers who can collude such that all of them achieve strictly higher payoffs (without making side payments). The proof is by contradiction. Let S be the largest subset of buyers who can collude and possibly misreport their u_i^j's and all of them achieve strictly higher payoffs. Let $\underline{W}$ be the lowest competitive equilibrium of M with respect to the true utility functions and let W' be the lowest competitive equilibrium with respect to the reported utility functions assuming that buyers is S have colluded. Let T be the subset of the goods that are matched to S at W'. Since all the buyers in S are achieving strictly higher payoffs at W', they cannot be unmatched at W' (i.e. $|T| = |S|$) and the prices of the goods in T should be strictly lower at W'. That means the goods in T must have had strictly positive prices in $\underline{W}$. By applying Lemma 11, we argue that there must have been a subset S' of buyers of size at least $|T| + 1$ who were interested in some good in T at $\underline{W}$. Observe that all of the buyers in S' must be getting a strictly higher payoff at W' because the prices of all the goods in T are strictly lower. But S' is larger than S which contradicts our assumption that S was the largest set of buyers who could all benefit from collusion. □

Proof of Theorem 47. The group strategyproofness follows from Theorem 46. So we only prove the second part: Assuming the mechanism has computed a lowest competitive equilibrium W as the outcome with price vector $\mathbf{p}$, the expected utility of advertiser i from slot j is given by $u_i^j(g_i^j(\mathbf{p}^j))$ where $\mathbf{p}^j$ is the base price of good j and $g_i^j(x) = x/\hat{c}_i^j$ is the personalized price of slot j for advertiser i. So for each advertiser $i \in A$ we have $u_i^j(\mathbf{p}^j) = c_i^j(v_i^j - \mathbf{p}^j/\hat{c}_i^j)$. Furthermore, since $c_i^j = \hat{c}_i^j$, we can simplify the utility function and get $u_i^j(\mathbf{p}^j) = c_i^j v_i^j - \mathbf{p}^j$. Now, consider the complete bipartite graph G with advertisers and slots. Let the weight of each edge (i, j) be $c_i^j v_i^j$. Note that for each advertiser $i \in A$ we have $u_i(W) + p^j(W) \geq c_i^j v_i^j$ in which W is the outcome of the mechanism. Therefore, the total expected welfare of the coalition $\{s\} \cup A$ is at least as much as the weight of the maximum weight matching in the absence of A'. Furthermore, if A' is empty (i.e. everyone agrees on the CTRs), the mechanism computes the efficient allocation (i.e., a maximum weight matching) and the outcome is the same as the VCG outcome. □

Next, we present the proof of the lattice structure. The proof does not use any other lemma.

Proof of Theorem 44. To simplify the proof, we add $|J|$ dummy buyers and $|I|$ dummy goods so as to make sure that we can always get a perfect matching. We set $u_i^j(x) = -x$ whenever either i or j or both are dummy. By doing this we can always make sure that for every competitive equilibrium W there is a perfect matching $\boldsymbol{\mu}$ that supports the equilibrium. Note that (6.2) and (6.3) ensure that for any unmatched buyer i, $\mathbf{u}_i = 0$ and for any unmatched good j, $\mathbf{p}^j = 0$ so we can arbitrarily match the unmatched buyers/goods to the new dummy buyers/goods and then match the remaining dummy buyers/goods together. Observe that by adding dummy buyers/goods we don't need to be concerned about (6.2) and (6.3) anymore [9].

- First, we prove that W_{inf} is a valid competitive equilibrium :
  - We first show that $\boldsymbol{\mu}_{\text{inf}}$ is a valid matching. The proof is by contradiction. Suppose it is not. Then, there should be $i, i' \in I$ such that $\boldsymbol{\mu}_{\text{inf}}(i) = \boldsymbol{\mu}_{\text{inf}}(i') = j$. Since both $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ are valid matchings, j should be matched to i in one of them and to i' in the other one. WLOG, assume that $\boldsymbol{\mu}(i) = j$ and $\boldsymbol{\mu}'(i') = j$. From the definition of $\boldsymbol{\mu}_{\text{inf}}$ and because $\boldsymbol{\mu}_{\text{inf}}(i) = \boldsymbol{\mu}(i)$, we can argue $\mathbf{u}_i \geq \mathbf{u}'_i$. So we have $u_i^j(\mathbf{p}^j) = \mathbf{u}_i \geq \mathbf{u}'_i \geq u_i^j(\mathbf{p}'^j)$ which means $\underline{\mathbf{p}^j \leq \mathbf{p}'^j}$. On the other hand, by repeating the same argument for i' instead of i, we can conclude that $\mathbf{u}_{i'} < \mathbf{u}'_{i'}$ (note that according to the definition of $\boldsymbol{\mu}_{\text{inf}}$, $\boldsymbol{\mu}_{\text{inf}}(i') = \boldsymbol{\mu}'(i')$ if $\mathbf{u}_{i'} < \mathbf{u}'_{i'}$) and so we have $u_{i'}^j(\mathbf{p}^j) \leq \mathbf{u}_{i'} < \mathbf{u}'_{i'} = u_{i'}^j(\mathbf{p}'^j)$ which means $\underline{\mathbf{p}_j > \mathbf{p}'^j}$. We have a contradiction because we just proved $\mathbf{p}^j \leq \mathbf{p}'^j$ and $\mathbf{p}_j > \mathbf{p}'^j$. Therefore, $\boldsymbol{\mu}_{\text{inf}}$ must be a valid matching.
  - We show that W_{inf} satisfies the (6.1). WLOG, assume $\boldsymbol{\mu}_{\text{inf}}(i) = \boldsymbol{\mu}(i) = j$. So $u_i^j(\mathbf{p}^j) = \mathbf{u}_i \geq \mathbf{u}'_i \geq u_i^j(\mathbf{p}'^j)$ which means $\mathbf{p}^j \leq \mathbf{p}'^j$. Therefore, $p^j(W_{\text{inf}}) = \mathbf{p}^j$. Together with the fact that $u_i(W_{\text{inf}}) = \mathbf{u}_i$ and $\boldsymbol{\mu}_{\text{inf}}(i) = \boldsymbol{\mu}(i)$

[9] Remember that for non-dummy buyers, $u_i^j(0)$ might be negative so we may not be able to match the 0 priced items and 0 payoff buyers. This is a technicality that arises when we transform the market in the proof of Lemma 13

we can argue that W_{inf} satisfies the first part of (6.1) because W satisfies it. Similarly, for any j', $\mathbf{u}_i \geq u_i^{j'}(\mathbf{p}'^{j'})$ and $\mathbf{u}'_i \geq u_i^{j'}(\mathbf{p}'^{j'})$. Therefore, $u_i(W_{\text{inf}}) \geq \max(u_i^{j'}(\mathbf{p}^{j'}), u_i^{j'}(\mathbf{p}'^{j'})) = u_i^{j'}(\min(\mathbf{p}^{j'}, \mathbf{p}'^{j'})) = u_i^{j'}(p^{j'}(W_{\text{inf}}))$ so the second part of (6.1) is also satisfied.

- Next, we prove that W_{sup} is a valid competitive equilibrium :
  - We first show that $\boldsymbol{\mu}_{\text{sup}}$ is a valid matching. The proof is by contradiction. Suppose it is not. Then, there should be $i, i' \in I$ such that $\boldsymbol{\mu}_{\text{sup}}(i) = \boldsymbol{\mu}_{\text{sup}}(i') = j$. Since $|I| = |J|$ and there are two buyers that are matched to the same good, there should be another good j' to which no buyer is matched. On the other hand, both $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ are valid perfect matchings so j' must be matched in both of them. Let $s = \boldsymbol{\mu}^{-1}(j')$ and $s' = \boldsymbol{\mu}'^{-1}(j')$. Notice that $s \neq s'$ otherwise $\boldsymbol{\mu}_{\text{sup}}(s)$ would be j' as well. Because j' is not matched in $\boldsymbol{\mu}_{\text{sup}}$, it should be that $\boldsymbol{\mu}_{\text{sup}}(s) \neq j'$ which means $\boldsymbol{\mu}_{\text{inf}}(s) = j'$. Similarly, it should be that $\boldsymbol{\mu}_{\text{sup}}(s') \neq j'$ which means $\boldsymbol{\mu}_{\text{inf}}(s') = j'$. However, that means $\boldsymbol{\mu}_{\text{inf}}(s) = \boldsymbol{\mu}_{\text{inf}}(s') = j'$ which means $\boldsymbol{\mu}_{\text{inf}}$ is not a valid matching which is a contradiction since we already proved $\boldsymbol{\mu}_{\text{inf}}$ is a valid perfect matching.
  - We show that W_{sup} satisfies the (6.1). WLOG, assume $\boldsymbol{\mu}_{\text{sup}}(i) = \boldsymbol{\mu}(i) = j$. Let $i' = \boldsymbol{\mu}_{\text{inf}}^{-1}(j)$. It must be that $\boldsymbol{\mu}'(i') = j$ (because we already know that $\boldsymbol{\mu}(i) = j$ so $\boldsymbol{\mu}(i') \neq j$, but i' must be matched to j in $\boldsymbol{\mu}$ or $\boldsymbol{\mu}'$). That means $u_{i'}^j(\mathbf{p}'^j) = \mathbf{u}'_{i'} \geq \mathbf{u}_{i'} \geq u_{i'}^j(\mathbf{p}^j)$, so $\mathbf{p}'^j \leq \mathbf{p}^j$ and therefore $p^j(W_{\text{sup}}) = \mathbf{p}^j$. We also have $u_i(W_{\text{sup}}) = \mathbf{u}_i$ and $\boldsymbol{\mu}_{\text{sup}}(i) = \boldsymbol{\mu}(i)$ so W_{sup} must satisfy the first part of (6.1) because W satisfies it. On the other hand, since we assumed $\boldsymbol{\mu}_{\text{sup}}(i) = \boldsymbol{\mu}(i) = j$, for any j', we have $u_i(W_{\text{sup}}) = \mathbf{u}_i \geq u_i^{j'}(\mathbf{p}^{j'}) \geq u_i^{j'}(\max(\mathbf{p}^{j'}, \mathbf{p}'^{j'})) = u_i^{j'}(p^{j'}(W_{\text{sup}}))$. So W_{sup} satisfies second part of (6.1) as well.

□

Chapter 7

Conclusion

In chapter 2 and chapter 4, we presented optimal reductions and approximately optimal reductions from a multi agent Bayesian mechanism deign problem to single agent subproblems. We showed that an exponential increase in the complexity of the underlying optimization problem can be avoided by such reductions. chapter 2 presented reductions by decomposing the problem using the interim allocation rule. Such decomposition allows computation of the optimal mechanism in polynomial time, however the approach may not be practical as it makes use of the ellipsoid method. chapter 4 presented reductions by decomposing the problem using the ex ante allocation rule. Such decomposition allows efficient computation of approximately optimal mechanisms and also yields practical algorithms. It also leads to the following conclusions.

- **Market size.** As the ratio of the maximum demand to supply (e.g., $\frac{1}{k}$) decreases, less coordination is required on decisions made for different agents; i.e., as $\frac{1}{k} \to 0$, the optimal mechanism treats each agent almost independently of other agents. Observe that all of the approximation factors in this thesis only depend on k (i.e., $1 - \frac{1}{\sqrt{k+3}}$) and not on the number of agents. It suggests that, for characterizing asymptotic properties of such markets, the right parameter to consider is perhaps the ratio of the maximum demand to supply; in particular, notice that the number of agents is irrelevant.

- **Computational hardness.** For mechanism design problems in a variety of settings, the difficulty of making coordinated optimal decisions for multiple agents can be avoided by losing a small constant factor in the objective (i.e., losing only a $\frac{1}{\sqrt{k+3}}$ fraction of the objective), therefore the main difficulty of constructing constant factor approximation mechanisms in multi dimensional settings stems from the difficulty of designing single agent mechanisms, which ultimately stems from enforcing the incentive compatibility constraints in the single agent problem.

  In chapter 5, we presented a generalization of the magician's problem from chapter 4. We also presented applications of the magician's problem in prophet inequalities (section 5.3) and online stochastic generalized assignment problem(section 5.4).

  In chapter 6, we studied the class of competitive equilibria in two sided matching markets with general utility functions. We presented an exact inductive characterization of the competitive equilibria and gave a constructive proof for its existence and various other properties. All of the previous known proofs were non-constructive. Our characterization provides a deep insight into the structure of the equilibria and reveals striking similarities between the payments at the lowest competitive equilibrium for general utilities and VCG payments for quasilinear utilities. We also presented a social-welfare maximizing truthful mechanism for pay per click Ad-auctions where the search engine and the advertisers may disagree on the clickthrough rates (VCG is inapplicable if payments are per click).

  Our characterizations raise the question of whether it is possible to generalize this result to more general matchings (e.g., the many-to-one matchings) with general utilities. Another challenge is to find more efficient algorithms for computing the competitive equilibria.

Bibliography

Aggarwal, G., Muthukrishnan, S., Pál, D., and Pál, M. (2009). General auction mechanism for search advertising. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 241–250, New York, NY, USA. ACM.

Alaei, S. (2011). Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 512–521.

Alaei, S., Fu, H., Haghpanah, N., Hartline, J. D., and Malekian, A. (2012a). Bayesian optimal auctions via multi- to single-agent reduction. In *ACM Conference on Electronic Commerce*, page 17.

Alaei, S., Hajiaghayi, M., and Liaghat, V. (2012b). Online prophet-inequality matching with applications to ad allocation. In *ACM Conference on Electronic Commerce*, pages 18–35.

Alaei, S., Jain, K., and Malekian, A. (2011). Competitive equilibrium in two sided matching markets with general utility functions. *SIGecom Exchanges*, 10(2):34–36.

Armstrong, M. (1996). Multiproduct nonlinear pricing. *Econometrica*, 64(1):51–75.

Ashlagi, I., Braverman, M., Hassidim, A., Lavi, R., and Tennenholtz, M. (2010). Position auctions with budgets: Existence and uniqueness. *The B.E. Journal of Theoretical Economics*, 10(1).

Ausubel, L. M. and Milgrom, P. (2002). Ascending auctions with package bidding. Working Papers 02004, Stanford University, Department of Economics.

Babaioff, M., Lavi, R., and Pavlov, E. (2006). Single-value combinatorial auctions and implementation in undominated strategies. In *SODA*, pages 1054–1063.

Bhattacharya, S., Goel, G., Gollapudi, S., and Munagala, K. (2010). Budget constrained auctions with heterogeneous items. In *44th ACM Symposium on Theory of Computing*, pages 379–388.

Blumrosen, L. and Holenstein, T. (2008). Posted prices vs. negotiations: an asymptotic analysis. In *ACM Conference on Electronic Commerce*, page 49.

Border, K. (2007). Reduced form auctions revisited. *Economic Theory*, 31(1):167–181.

Border, K. C. (1991). Implementation of reduced form auctions: A geometric approach. *Econometrica*, 59(4):pp. 1175–1187.

Briest, P., Chawla, S., Kleinberg, R., and Weinberg, S. M. (2010). Pricing randomized allocations. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 585–597.

Bulow, J. and Roberts, J. (1989). The simple economics of optimal auctions. *Journal of Political Economy*, 97(5):1060–90.

Cai, Y. and Daskalakis, C. (2011). Extreme-value theorems for optimal multidimensional pricing. In *IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 522–531.

Cai, Y., Daskalakis, C., and Weinberg, M. (2012). An algorithmic characterization of multi-dimensional mechanisms.

Chakraborty, T., Even-Dar, E., Guha, S., Mansour, Y., and Muthukrishnan, S. (2010). Approximation schemes for sequential posted pricing in multi-unit auctions. In *WINE*.

Chawla, S., Hartline, J. D., and Kleinberg, R. D. (2007). Algorithmic pricing via virtual valuations. In MacKie-Mason, J. K., Parkes, D. C., and Resnick, P., editors, *ACM Conference on Electronic Commerce*, pages 243–251. ACM.

Chawla, S., Hartline, J. D., Malec, D. L., and Sivan, B. (2010). Multi-parameter mechanism design and sequential posted pricing. In *42nd ACM Symposium on Theory of Computing*, pages 311–320.

Chawla, S., Malec, D., and Malekian, A. (2011). Bayesian mechanism design for budget-constrained agents. In *ACM Conference on Electronic Commerce*.

Che, Y. and Gale, I. (1995). The optimal mechanism for selling to budget-constrained consumers. Technical report.

Chekuri, C., Vondrák, J., and Zenklusen, R. (2010). Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*, pages 575–584.

Demange, G. and Gale, D. (1985). The strategy structure of two-sided matching markets. *Econometrica*, 53(4):873–88.

Demange, G., Gale, D., and Sotomayor, M. (1986). Multi-item auctions. *Journal of Political Economy*, 94(4):863–72.

Dhangwatnotai, P., Roughgarden, T., and Yan, Q. (2010). Revenue maximization with a single sample. In *ACM Conference on Electronic Commerce*, pages 129–138.

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

Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. volume 69, pages 9–15.

Grötschel, M., Lovász, L., and Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197. 10.1007/BF02579273.

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

Hajiaghayi, M. T., Kleinberg, R. D., and Sandholm, T. (2007). Automated online mechanism design and prophet inequalities. In *AAAI*, pages 58–65.

Hartline, J. D., Kleinberg, R., and Malekian, A. (2011). Bayesian incentive compatibility and matchings. In *SODA*.

Hartline, J. D. and Lucier, B. (2010). Bayesian algorithmic mechanism design. In *STOC*, pages 301–310.

Hartline, J. D. and Roughgarden, T. (2009). Simple versus optimal mechanisms. In *ACM Conference on Electronic Commerce*, pages 225–234.

Hatfield, J. W. and Kominers, S. D. (2010). Matching in networks with bilateral contracts: extended abstract. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10.

Hatfield, J. W. and Milgrom, P. R. (2005). Matching with contracts. *American Economic Review*, 95(4):913–935.

Henzinger, M. and Vidali, A. (2011). Multi-parameter mechanism design under budget and matroid constraints. In *ESA*, pages 192–202.

Hill, T. P. and Kertz, R. P. (1992). A survey of prophet inequalities in optimal stopping theory. In *Contemporary Mathematics*.

Kelso, Alexander S, J. and Crawford, V. P. (1982). Job matching, coalition formation, and gross substitutes. *Econometrica*, 50(6):1483–1504.

Knaster, B., Kuratowski, C., and Mazurkiewicz, S. (1929). Ein beweis des fixpunktsatzes fr n-dimensionale simplexe. *Fund. Math*, 14:132–137.

Kuhn, H. W. (1956). Variants of the hungarian method for assignment problems. page 253258.

Lemke, C. E. and Howson, J. T., J. (1964). Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423.

Leonard, H. B. (1983). Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–79.

Mamer, J.and Bikhchandani, S. (1997). *Journal of Economic Theory*, 74:385–413.

Manelli, A. M. and Vincent, D. R. (2010). Bayesian and dominant-strategy implementation in the independent private-values model. *Econometrica*, 78(6):1905–1938.

Maskin, E. and Riley, J. (1984). Optimal auctions with risk averse buyers. *Econometrica*, 52(6):pp. 1473–1518.

Maskin, E. S. (2000). Auctions, development, and privatization: Efficient auctions with liquidity-constrained buyers. *European Economic Review*, 44(4-6):667–681.

Matthews, S. (1983). Selling to risk averse buyers with unobservable tastes. *Journal of Economic Theory*, 30(2):370–400.

Matthews, S. A. (1984). On the implementability of reduced form auctions. *Econometrica*, 52(6):pp. 1519–1522.

McAfee, R. P. and McMillan, J. (1988). Multidimensional incentive compatibility and mechanism design. *Journal of Economic Theory*, 46(2):335–354.

Mierendorff, K. (2011). Asymmetric reduced form auctions. *Economics Letters*, 110(1):41–44.

Mirman, L. J. and Sibley, D. (1980). Optimal nonlinear prices for multiproduct monopolies. *Bell Journal of Economics*, 11(2):659–670.

Myerson, R. B. (1981). Optimal auction design. *Mathematics of Operations Research*, 6(1):pp. 58–73.

Pai, M. M. and Vohra, R. (2008). Optimal auctions with financially constrained bidders. Discussion papers, Northwestern University, Center for Mathematical Studies in Economics and Management Science.

Quinzii, M. (1984). Core and competitive equilibria with indivisibilities. *International Journal of Game Theory*, 13:41–60.

Roberts, K. W. S. (1979). Welfare considerations of nonlinear pricing. *Economic Journal*, 89(353):66–83.

Rochet, J.-C. and Chone, P. (1998). Ironing, sweeping, and multidimensional screening. *Econometrica*, 66(4):783–826.

Scarf, H. E. (1967). The core of an n person game. *Econometrica*, 35(1).

Schrijver, A. (2003). *Combinatorial Optimization : Polyhedra and Efficiency (Algorithms and Combinatorics)*. Springer.

Shapley, L. and Shubik, M. (1971). The assignment game i: The core. 1(1):111130.

Spence, A. M. (1980). Multi-product quantity-dependent prices and profitability constraints. *Review of Economic Studies*, 47(5):821–41.

Wikipedia (2012). http://en.wikipedia.org/wiki/mana.

Wilson, R. (1994). Nonlinear pricing. *Journal of Political Economy*, 102(6):1288–1291.

Yan, Q. (2011). Mechanism design via correlation gap. In *SODA*.