Bayesian Algorithmic Mechanism Design

SHUCHI CHAWLA
University of Wisconsin - Madison
and
BALASUBRAMANIAN SIVAN
Microsoft Research

This article surveys recent work with an algorithmic flavor in Bayesian mechanism design. Bayesian mechanism design involves optimization in economic settings where the designer possesses some stochastic information about the input. Recent years have witnessed huge advances in our knowledge and understanding of algorithmic techniques for Bayesian mechanism design problems. These include, for example, revenue maximization in settings where buyers have multi-dimensional preferences, optimization of non-linear objectives such as makespan, and generic reductions from mechanism design to algorithm design. However, a number of tantalizing questions remain unsolved.

This article is meant to serve as an introduction to Bayesian mechanism design for a novice, as well as a starting point for a broader literature search for an experienced researcher.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Algorithms; Economics; Theory

Additional Key Words and Phrases: mechanism design, auctions, optimization, approximation, Bayes-Nash equilibrium

1. INTRODUCTION

Mechanism design deals with optimization in strategic settings. The mechanism designer's task is to design a system involving strategic participants, a.k.a. agents, who act in their own self-interest. The system should be designed in a way that agents' selfish behavior leads, at equilibrium, to optimization of a global objective. Any system involving sharing of scarce resources among multiple parties falls within the purview of mechanism design—routing traffic on the Internet, task scheduling in a cloud computing environment, electronic marketplaces involving millions of buyers and thousands of sellers, assigning ad-slots to merchants on a website, are a few examples. Indeed with the growth of commerce on the Internet, mechanism design problems have become nearly as ubiquitous as algorithm design problems.

In the last decade and a half there has been an explosion of work on *algorithmic* mechanism design[1], in other words, mechanism design with an eye towards computational efficiency. Following the convention in computer science, the early part of this work focused predominantly on worst-case analysis and guarantees. A primary challenge that the mechanism designer faces is his lack of information regarding the agents' preferences, which govern their behavior and consequently af-

[1]This term was coined by Nisan and Ronen (2001), who introduced this area to computer scientists.

The first author is supported in part by NSF grants CCF-1101429 and CCF-1320854. Authors' addresses: `shuchi@cs.wisc.edu`, `bsivan@microsoft.com`

fect the system performance. In the worst-case or *prior-free* paradigm, the designer makes no assumptions whatsoever about the agents' preferences, and the system is required to work in all circumstances. Unfortunately, it is often impossible to achieve any interesting guarantees in such a setup. This shortcoming, coupled with emerging applications in data intensive settings[2], motivated the study of *Bayesian* algorithmic mechanism design (henceforth, BAMD).

In the Bayesian setting, the mechanism designer has a stochastic model of the market for which the system is to be designed. He can use this model to inform the design of the system. Once the system has been designed, agents' preferences are instantiated from the stochastic model, and the system is run over these inputs. We study the system's performance at equilibrium: where every agent's actions, given the rules of the system, are in best response to others' actions. Bayesian mechanism design has a rich and large body of literature in economics. Over the last few years, computational and algorithmic approaches have added a new dimension to this literature and have led to solutions for many settings previously considered intractable. These form the topic of this survey article.

We can summarize the central themes in algorithmic mechanism design from several different perspectives:

—The practical computer scientist's perspective: what computational tools can we design and use to efficiently solve a given mechanism design problem?

—The practical economist's perspective: what are the properties of optimal mechanisms? What explains their prevalence or non-prevalence in practice? What kinds of mechanisms arise in practice?

—The theoretician's perspective: does strategizing on part of the agents hurt system performance? To what extent? We can further refine this question. The theoretical economist's perspective: what performance can we obtain under strategic considerations versus the overall optimum? The theoretical computer scientist's perspective: what performance can we obtain *efficiently* in strategic versus non-strategic settings?

We will touch upon all of these themes in this survey. The answers, not surprisingly, depend on several features of the problem, e.g.,: (1) The strength of the equilibrium concept, e.g., dominant strategy equilibrium or Bayes-Nash equilibrium (see Section 2 for definitions);(2) The designer's objective, e.g., linear versus non-linear; (3) The complexity of the agents' preferences, e.g. single-parameter versus multi-parameter, or, linear versus non-linear.

This article focuses on two lines of work. The first deals with the objective of revenue maximization, and is one of the biggest successes of BAMD. Revenue maximization is among the most natural and practically relevant objectives in mechanism design. It is also unique in that it depends on the monetary transfers between the agents, rather than directly on their preferences. While revenue maximization is generally computationally intractable even in the simplest settings (e.g. a single buyer with independent additive values over a set of items), several techniques have been developed for approximating it in a broad class of problems. Another theme

[2]In applications such as ad allocation millions of economic transactions happen every day, and it is possible for the system to collect information about the environment and the market.

has been to understand the power of simple and robust mechanisms, the kinds of mechanisms that we often see in practice, e.g. reserve pricing. We discuss this work in Sections 3 and 4.

In the second part of the article we focus on the gap between strategic and non-strategic optimization by asking: are there generic methods that convert algorithms into mechanisms? By generic, we mean approaches that do not try to understand the inner workings of the algorithm and rely only on its input-output behavior to determine how to make it compatible with agents' preferences. In many settings this is impossible without suffering a large loss in performance. However, surprisingly, there are settings where such *black box reductions* suffer almost no loss in performance. We discuss these results in Section 5.

We briefly survey other directions in BAMD in Section 6 and list some open problems in Section 7.

Other related articles. Krishna (2009) gives an excellent introduction to auction theory. We refer the reader to Hartline (2013a) for a more extensive and nuanced treatment of revenue maximization in Bayesian mechanism design, and Hartline (2013b) for a detailed discussion of approximations in Bayesian and prior-free mechanism design. Nisan (2014) surveys social welfare maximization in prior-free mechanism design.

2. THE BASICS OF BAYESIAN MECHANISM DESIGN

Preferences and mechanisms. The following is a generic setup for a Bayesian mechanism design problem. We have n buyers, a.k.a. agents, and one seller. The seller has different items, resources, or services to offer to the buyers at a price; Henceforth we will refer to these as items. We use the term *allocation* to refer to the set of items that an agent gets. Each agent has a private type that specifies the value that the agent derives from an allocation. Agent i's *type*, denoted t_i, is drawn from a known set T_i. We use $t_i(x_i)$ to denote the value that an agent with type t_i gets from being allocated a (potentially random or fractional) set x_i of items. Agents' types $\mathbf{t}$ are drawn from a publicly known (joint) distribution $\mathbf{F}$. For most of this survey, we will assume that agents' types are drawn from independent distributions, i.e. $\mathbf{F}$ is a product distribution $F_1 \times F_2 \times \cdots \times F_n$.

We will focus on the class of *direct-revelation* mechanisms[3] that ask the agents to reveal their types and then return an outcome that consists of allocations of items to the agents and payments to be made by the agents to the seller. Formally, a direct-revelation mechanism $\mathcal{M}$ is defined by functions $(\mathbf{x}, \mathbf{p})$. $\mathbf{x}$ maps the agents' revealed types to allocations; $\mathbf{p}$ maps the types to payments made by the agents to the mechanism. We use $x_i(\mathbf{t})$ and $p_i(\mathbf{t})$ to denote the allocation and payment respectively for agent i. Recall that the agent derives a value $t_i(x_i(\mathbf{t}))$ from this allocation. We assume that the agent has *quasi-linear* utility, so that his net utility from the outcome $(x_i(\mathbf{t}), p_i(\mathbf{t}))$ is $t_i(x_i(\mathbf{t})) - p_i(\mathbf{t})$.

Ex ante, interim, and ex post functions. The time-line of events in a mechanism design setting proceeds as follows: First, the designer learns of the mech-

[3]This is without loss of generality; See the revelation principle in Krishna (2009).

anism design problem, which includes the distributions from which agents' types are drawn; Based on these inputs the designer designs a mechanism; Then, agents' actual types are instantiated from the respective distributions; Finally, the mechanism is run on the types and an outcome is generated. When we refer to a quantity *ex ante*, we are referring to its expected value before the agents' types have been instantiated, where the expectation is taken over the distribution from which types are drawn. *Interim* refers to the time after the agents' types have been instantiated, but before the mechanism has been run; In particular, at this time, agents know their own types but not each others' instantiated types, and so the final outcome of the mechanism is as yet unknown to them. *Ex post* refers to the final realized value after the mechanism has been run.

We will often be interested in the mechanism's *interim* outcome for an agent i, i.e., the expected outcome for the agent before others' types have been revealed. For convenience, we overload the notation[4] and use $x_i(t_i)$ and $p_i(t_i)$ to denote the agent's interim allocation and payment, respectively. Let $\mathbf{t}_{-i}$ denote the vector of types of all agents other than i, and let $\mathbf{F}_{-i}$ denote the joint distribution from which this vector is drawn. Then, $x_i(t_i) = \mathbf{E}_{\mathbf{t}_{-i}\sim\mathbf{F}_{-i}}[x_i(t_i, \mathbf{t}_{-i})]$ and $p_i(t_i) = \mathbf{E}_{\mathbf{t}_{-i}\sim\mathbf{F}_{-i}}[p_i(t_i, \mathbf{t}_{-i})]$.

Truthfulness. For the most part in this survey we will assume that agents are risk neutral and maximize their expected utility, where the expectation is taken over any randomness in the mechanism, as well as, in some cases, over the randomness in other agents' types. We focus on the class of truthful mechanisms that incentivize agents to reveal their true types. Informally, a mechanism is truthful if every agent is no worse off revealing his type truthfully rather than by lying about it (and is sometimes better off). There are several ways of formalizing this notion, that differ in the amount of information the agent uses in making his decision.

—A mechanism is *Bayesian incentive compatible (BIC)* if truth-telling is in an agent's best interest *before* the agent observes others' types. Formally, for all i, all $t_i \in T_i$, and all $s_i \in T_i$, the agent's expected utility from reporting his true type t_i is no less than his expected utility from reporting a different type s_i when others' report their true types:

$$\mathbf{E}_{\mathbf{t}_{-i}\sim\mathbf{F}_{-i}}[t_i(x_i(t_i, \mathbf{t}_{-i})) - p_i(t_i, \mathbf{t}_{-i})] \geq \mathbf{E}_{\mathbf{t}_{-i}\sim\mathbf{F}_{-i}}[t_i(x_i(s_i, \mathbf{t}_{-i})) - p_i(s_i, \mathbf{t}_{-i})] \tag{1}$$

—A mechanism is *dominant strategy incentive compatible (DSIC)* if truth-telling is in an agent's best interest even *after* the agent observes others' types. Formally, for all i, all $t_i \in T_i$, all $s_i \in T_i$, and all types $\mathbf{t}_{-i}$ of the other agents:

$$t_i(x_i(t_i, \mathbf{t}_{-i})) - p_i(t_i, \mathbf{t}_{-i}) \geq t_i(x_i(s_i, \mathbf{t}_{-i})) - p_i(s_i, \mathbf{t}_{-i}) \tag{2}$$

For mechanisms that are randomized, we can make a finer distinction here between whether the agent remains truthful after knowing the mechanism's coin flips or not. A mechanism is called *truthful in expectation* if the above inequality holds in expectation over the randomness in the mechanism, and is called *ex*

[4]The argument to the allocation or payment function will usually indicate which of the following three cases we refer to—ex ante, interim, or ex post.

post truthful or ex post IC if the inequality continues to hold for every possible instantiation of the mechanism's randomness.

—Sometimes we will need to relax the notion of incentive compatibility to a weaker notion called *ϵ-Bayesian incentive compatibility (ϵ-BIC)*. A mechanism is ϵ-BIC if, informally, for every agent and for all possible true types of the agent, the increase in expected utility that the agent gets from misreporting his type is at most a factor of ϵ times the expected utility he gets from truth-telling.

We also require the mechanism to satisfy *individual rationality (IR)*: each agent always receives non-negative utility. As with incentive compatibility, IR may be satisfied ex post, i.e. after all agent types have been revealed, or interim, i.e. in expectation over other agents' types.

Single and multi-parameter types. A special case for agent preferences is when an agent is interested in just one kind of item. In this case, the agent's type is merely his value for obtaining the item, denoted v_i. The allocation function in this case maps the agent's type to the probability with which the item is provided to the agent, and his value for an allocation probability x_i is given by $t_i(x_i) = v_i x_i$. We call such an agent a *single-parameter* agent.

For a single-parameter agent it is sometimes convenient to express the agent's allocation and payment as functions of the quantile of his value with respect to its distribution. The quantile of an agent is the probability with which a random value drawn from F exceeds the agent's value: $q = 1 - F(v)$; as the agent's value increases his quantile decreases. Since once we specify the distribution F there is a unique mapping from values to quantiles and vice versa, we will use these quantities interchangeably as arguments to the allocation and payment functions.

Multi-parameter agents, in contrast, are interested not just in whether they are served or not but also in what kind of service they receive. For example, a multi-parameter agent may be interested in buying several different items and derive different values from different sets of items. In the most general case, the type of such an agent would specify his value for each possible outcome of the mechanism; often, however, it is possible to express the type more succinctly.

Objectives and constraints. The mechanism designer's goal is to optimize his objective subject to the mechanism being incentive compatible, as well as potentially subject to a feasibility constraint on the allocations. We use $\mathcal{F}$ to denote the feasibility constraint, and require that $\mathbf{x}(\mathbf{t}) \in \mathcal{F}$ for all possible tuples $\mathbf{t}$ of types.[5] In single-parameter settings where the allocation specifies the probability with which each agent is served, $\mathcal{F}$ is simply a subset of $[0,1]^n$. For example, if the seller has a supply constraint that prevents him from allocating to more than k bidders at a time, $\mathcal{F}$ is the set of all vectors $\mathbf{x}$ in $[0,1]^n$ with $\sum_i x_i \leq k$.

Some of the single-parameter results we survey will require the feasibility constraint to be *downwards closed*, that is, for any vector $\mathbf{x} \in \mathcal{F}$ and $\mathbf{y}$ that is component-wise smaller than $\mathbf{x}$, we have $\mathbf{y} \in \mathcal{F}$.

The following are some natural objectives in mechanism design that we will consider in this survey.

[5]In the case of randomized mechanisms, $\mathbf{x}(\mathbf{t})$ is a random vector lying in $\mathcal{F}$.

—*Surplus or social welfare or economic efficiency.* This measures the total utility achieved by all participants in a system; Since payments are internal to the system (passing from the buyers to the seller), the payments cancel and social welfare is equal to the net value created by the allocation: $\mathrm{SW} = \mathbf{E}[\sum_i t_i(x_i)]$. Social welfare can be maximized using the *Vickrey Clarke Groves (VCG)* mechanism, which maximizes social surplus point-wise for every value vector, i.e. $\mathbf{x}(\mathbf{t}) \in \mathrm{argmax}_{\mathbf{z}\in\mathcal{F}} \sum_i t_i(z_i)$. The VCG mechanism is ex post IC. The manifestation of the VCG mechanism in single-parameter settings where the seller has $k \geq 1$ identical items to sell is called the *Vickrey auction.* The Vickrey auction sells the items to the k highest value agents, charging each the value of the $(k+1)$th highest value agent.

—*Seller's revenue.* This measures the total amount of money the seller makes: $\mathrm{Rev} = \mathbf{E}[\sum_i p_i]$. Among the objectives we consider in this survey, revenue is unique in that it depends on the agents' payments, rather than on the allocation.

—*Fairness or load balance.* These objectives capture a notion of fairness among agents: we may want to maximize the share of the agent that is worst off, $\min_i t_i(x_i)$, or minimize the share of the agent that is best off, $\max_i t_i(x_i)$.

Characterization of truthful mechanisms. The mechanism design problem for an objective $\mathcal{O}$ can now be stated as follows. Find a mechanism $\mathcal{M} = (\mathbf{x}, \mathbf{p})$ that maximizes or minimizes $\mathbf{E}_{\mathbf{t}\sim\mathbf{F}}[\mathcal{O}(\mathbf{x}(\mathbf{t}), \mathbf{p}(\mathbf{t}))]$ subject to feasibility and incentive compatibility. We will mostly be interested in optimizing over the class of BIC mechanisms; in many cases it will turn out that a DSIC mechanism is optimal or near-optimal.

What makes mechanism design more challenging compared to traditional algorithm design is that the incentive constraints tie together the mechanism's output at different input type vectors. Fortunately, often times it is possible to greatly simplify the incentive constraints. To get some intuition, let us consider the interim incentive constraints (1) for an agent with a discrete type space T_i. Fixing the agent's allocation rule, the constraints (1) place an upper and lower bound on the difference in the interim payments of the agent at any two types s_i and t_i:

$$\begin{aligned}\mathbf{E}_{\mathbf{t}_{-i}\sim\mathbf{F}_{-i}}[s_i(x_i(t_i,\mathbf{t}_{-i})) - s_i(x_i(s_i,\mathbf{t}_{-i}))] &\leq p_i(t_i) - p_i(s_i) \\ &\leq \mathbf{E}_{\mathbf{t}_{-i}\sim\mathbf{F}_{-i}}[t_i(x_i(t_i,\mathbf{t}_{-i})) - t_i(x_i(s_i,\mathbf{t}_{-i}))]\end{aligned}$$

Consider a complete directed graph over types where the length of any arc (s_i, t_i) is given by the upper bound (RHS) above. If we fix the agent's payment at any one type, we can determine BIC payments at other types by computing shortest paths in this graph. However, in order for shortest paths to exist, the graph must not contain any negative length cycles. In many settings (including continuous type spaces), the absence of negative cycles is sufficient to characterize BIC and can be checked efficiently.

For settings with single-parameter agents with independent types, this approach leads to the following theorem that is referred to in literature as Myerson's lemma. Recall that in this setting, each agent i's type is a single value v_i; the interim allocation x_i for each agent maps his value v_i to a probability $x_i(v_i)$ with which the agent gets the item.

Theorem 2.1 ((Myerson, 1981)). *In a mechanism design setting with single-parameter agents and independent value distributions, a mechanism $\mathcal{M} = (\mathbf{x}, \mathbf{p})$ is Bayesian incentive compatible if and only if for all i,*

*(1) (**monotonicity**) $x_i(v_i)$ is monotone non-decreasing, and,*
*(2) (**payment identity**) $p_i(v_i) = v_i x_i(v_i) - \int_0^{v_i} x_i(z)\,dz + p_i(0)$.*

There are several nice implications of this characterization. First, every weakly monotone allocation function x_i defines a payment function that satisfies condition (2) and is unique up to the additive term $p_i(0)$; $p_i(0)$ is usually 0, and we will omit it hereafter. Second, as an implication, in order to check whether an allocation function can be implemented via a BIC mechanism, it is sufficient to check whether the interim allocation is weakly monotone (condition (1)) for every agent. Third, we remark that we can write the agent's interim utility at value v_i as a function of the allocation rule: $\int_0^{v_i} x_i(z)\,dz$. Consequently, another equivalent characterization of BIC mechanisms is that every agent's interim utility is a convex function of her value. Finally, the same characterization holds for DSIC mechanisms as well, with the interim allocation, payment, and utility replaced by the ex post allocation, payment, and utility respectively.

In general multi-parameter settings, a different characterization of BIC mechanisms can be obtained which, once again, amounts to checking a certain monotonicity condition for the allocation function, and a payment identity (see Rochet (1987) for details). Unfortunately, this condition is generally harder to check algorithmically than the weak monotonicity condition (1) in Theorem 2.1.

Some representative examples. We will repeatedly refer to the following four representative settings to illustrate mechanism design techniques. The first two examples involve single-parameter agents, and the last two involve multi-parameter agents.

Example 1. (Single-item auction.) Here the seller has one item to sell. Each agent has a value for the item. The mechanism asks the agents to report their values, determines who to allocate the item to and what price to charge.

Example 2. (Multi-unit auction.) Here the seller has k identical copies of an item to sell. Each agent has a value for the item. The mechanism asks the agents to report their values, determines a subset of at most k agents, each of which get one copy of the item, and also determines the price to charge to each agent.

Example 3. (Multi-item unit-demand setting.) Here the seller has multiple copies of m different items to sell. Each agent has different values for different items, but wishes to buy at most one of the items. The mechanism asks the agents to report their values, determines which item to allocate to whom (if at all, and satisfying the supply constraint), and what prices to charge.

Example 4. (Multi-item additive setting.) Here the seller has multiple copies of m different items to sell. Each agent has different values for different items. An agent's value for obtaining a bundle (i.e. subset) of items is the sum of values of the items in that bundle. The mechanism asks the agents to report their values, determines which items to allocate to whom (if at all, and satisfying the supply constraint), and what prices to charge.

3. REVENUE MAXIMIZATION FOR SINGLE-PARAMETER AGENTS

In this section we will focus on the objective of revenue for the seller, in other words the optimal mechanism design problem. We will assume that the seller is risk neutral, so that his goal is to maximize his revenue (i.e. the payments made by the buyers to the mechanism) in expectation over the buyers' types as well as any randomness in the mechanism. We will optimize for expected revenue over the class of all BIC mechanisms; in some cases it will turn out that the optimal BIC mechanism is actually DSIC.

In this section, we consider settings where all of the agents are single-parameter, that is, they only care about the probability with which they get allocated or served, but don't distinguish between how they are served. In the following section we will consider settings where agents have different values for different allocations, i.e. the multi-parameter setting.

3.1 One singe-parameter agent

We begin with the simplest case of a single buyer interested in a single item. This buyer's type is given simply by his value v for the item, drawn from a known distribution F. As we mentioned earlier, in this setting, there is a unique mapping between the agent's value v and the quantile $q = 1-F(v)$ of this value with respect to the distribution F. We will, accordingly, write allocation and payment rules for mechanisms interchangeably as functions of quantile or value. A deterministic mechanism for this setting either allocates the item to the agent or not, that is, $x(q) \in \{0, 1\}$. Recall that in order for this mechanism to satisfy BIC, the allocation rule should be weakly increasing in v, or equivalently, weakly decreasing in q. In other words, there is some threshold quantile above which $x(q)$ is 0 and below which it is 1. We can thus refer to deterministic mechanisms in this setting by their threshold quantile. Let τ_q denote the mechanism (and allocation rule) with a threshold q. Note that this mechanism will allocate the item to the agent if his value exceeds $F^{-1}(1-q)$, and will charge the agent $F^{-1}(1-q)$. The expected revenue obtained by τ_q is therefore $qF^{-1}(1-q)$.

Let us now talk about randomized mechanisms. The allocation rule of a randomized mechanism maps the quantile of an agent to the probability with which the agent gets the item, that is, $x(q) \in [0, 1]$. For BIC, we require $x(q)$ to be a weakly decreasing function of q. Can randomized mechanisms obtain more revenue than deterministic mechanisms in this setting? Myerson (1981) and Riley and Zeckhauser (1983) were the first to show that they do not: in this simple setting, deterministic mechanisms are in fact optimal for revenue. We now present an elementary proof of this.

First, for intuition, consider the example where $x(q)$ is a function with two steps: $x(q) = 1$ for $q \leq 0.3$, $x(q) = 0.5$ for $0.3 < q \leq 0.6$, and $x(q) = 0$ for $q > 0.6$. Then, we can equivalently write x as the average of two threshold allocation rules: $\tau_{0.3}$ and $\tau_{0.6}$. That is, $x(q) = \frac{1}{2}\tau_{0.3}(q) + \frac{1}{2}\tau_{0.6}$ for all $q \in [0, 1]$. More generally, any weakly monotone allocation function $x(q)$ can be written as a distribution over deterministic allocation rules:

$$x(q) = \int_{z=0}^{z=1} (-x'(z))\tau_z(q)\,dz,$$

where $x'(\cdot)$ is the derivative of the allocation function with respect to the quantile and $\int_{z=0}^{z=1} -x'(z)\, dz = 1$.

What does this imply about the expected revenue obtained by x? Since the agent's payment is linearly related to his allocation probability via the payment identity ((2) in Theorem 2.1), the expected revenue generated by x is likewise a weighted average of the revenues generated by deterministic allocation rules:

$$\mathrm{Rev}(x) = \int_{z=0}^{z=1} (-x'(z))\, \mathrm{Rev}(\tau_z)\, dz.$$

Henceforth, we will use $R(q)$ as shorthand for $\mathrm{Rev}(\tau_q)$; the function $R(q)$ is called the *revenue curve* corresponding to the distribution F. Then, the revenue of *any* mechanism in this setting is a convex combination of points on the revenue curve. This immediately implies that the mechanism maximizing revenue is the deterministic mechanism τ_q corresponding to the quantile q that maximizes $R(q)$. We call the threshold price corresponding to this quantile the *monopoly reserve price* for the distribution F.

3.2 Many single-parameter agents

Next we consider a setting with $n > 1$ buyers, each of whom is interested in a single item. Buyer i's type is given by his value v_i drawn from independent distribution F_i. The seller faces a feasibility constraint specified by a set system $\mathcal{F} \subseteq 2^{[n]}$; the (randomized) allocation made by the seller at any value vector must lie in $\mathcal{F}$ (with probability 1). Once again the seller's goal is to maximize his expected revenue, or the expected total payment of all buyers.

As mentioned earlier, a challenge in mechanism design is that the incentive constraints bind together the mechanism's outcomes at different input value vectors. Myerson (1981) presented a solution to the revenue maximization problem for single-parameter agents that decomposes the overall optimization problem into many smaller optimization problems, one for each value vector $\mathbf{v}$; Each of these problems can then be solved without taking incentive constraints into account. We will now describe Myerson's approach.

Consider any BIC mechanism $(\mathbf{x}, \mathbf{p})$. Note that for any agent i, the interim allocation and payment rules (x_i, p_i) define a BIC single-agent mechanism for i. As described above, we can write the revenue of this single agent mechanism as $\mathbf{E}_q[-{x_i}'(q)R_i(q)]$, where $R_i(q)$ is the revenue curve corresponding to agent i's value distribution F_i. This is precisely agent i's contribution to the revenue of the mechanism $(\mathbf{x}, \mathbf{p})$.

Myerson's insight was that we can express this term as a linear function of the allocation x_i by integrating by parts:

$$\begin{aligned}\int_0^1 (-{x_i}'(q))R_i(q)\, dq &= x_i(0)R_i(0) - x_i(1)R_i(1) + \int_0^1 x_i(q)R_i'(q)\, dq \\ &= \mathbf{E}_q[x_i(q)R_i'(q)]\end{aligned}$$

where $R_i'(q)$ denotes the derivative of the revenue curve. The last equality follows by the convention $R_i(0) = R_i(1) = 0$, and by noting that the quantile of any agent is distributed uniformly at random between 0 and 1.[6]

Let us denote $R_i'(\cdot)$ by $\phi_i(\cdot)$, and note that this function depends only on agent i's distribution F_i and not on others' values or distributions. By convention, we express ϕ_i as a function of the agent's value (although it is equal to the derivative of the revenue curve with respect to the quantile). We can then sum up the contributions of all of the agents and write the revenue of a mechanism $\mathbf{x}$ as:

$$\text{Rev}(\mathbf{x}) \;= \mathbf{E}_{\mathbf{v}}[\textstyle\sum_i x_i(\mathbf{v})\phi_i(v_i)] \tag{3}$$

The function $\phi_i(v_i)$ is called the *virtual value* function, and is given by the following expression, where f_i is the probability density function associated with the distribution F_i.

$$\phi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

Correspondingly, the quantity $\sum_i x_i(\mathbf{v})\phi_i(v_i)$ is called the *virtual surplus* of the allocation $\mathbf{x}$ at vector $\mathbf{v}$. Note that virtual values are always smaller than the corresponding values; furthermore they can some times be negative. Bulow and Roberts (1989) argue that the difference $v_i - \phi_i(v_i)$ can be interpreted as an informational rent that a mechanism has to pay the agent in order to induce truth-telling.

Now let us get back to the problem of finding the mechanism $\mathbf{x}$ that maximizes the expression (3) subject to being BIC. A natural approach for doing so is to choose the allocation rule $\mathbf{x}$ that maximizes the virtual surplus $\sum_i x_i(\mathbf{v})\phi_i(v_i)$ pointwise, i.e. for each valuation vector $\mathbf{v}$, subject to feasibility. Ignoring incentive constraints, this clearly maximizes the overall expected revenue $\text{Rev}(\mathbf{x})$. When the virtual value functions ϕ_i are weakly increasing functions of v_i, this also produces a weakly monotone allocation rule and is therefore BIC. In other words, the optimal mechanism for this special case is a virtual surplus maximizer.

Regularity. Distributions F_i that generate a weakly increasing virtual value function are called *regular*. Several common distributions, e.g., the uniform, exponential, and normal distributions, as well as several other unimodal distributions, are regular. Since $\phi_i(\cdot)$ is the derivative of the revenue curve $R_i(\cdot)$, regularity is equivalent to the revenue curve being concave as a function of the quantile.

Ironing. In nonregular settings, maximizing the virtual surplus pointwise may lead to a non-BIC allocation rule. To form further intuition about this setting, let us go back to the case of a single-agent. Recall that in a single-agent setting, the revenue of any mechanism is a convex combination of points on the revenue curve. When the agent's value distribution is regular, the revenue curve $R(q)$ is concave, and so the revenue of any randomized mechanism lies below the curve. When the agent's distribution is nonregular, the curve is not concave everywhere, and there are some randomized mechanisms with revenue that lies above the curve. Informally, we can capture the revenues of all optimal single-agent randomized mechanisms by taking the concave upper envelope of $R(q)$; denote this $\bar{R}(q)$. (See Figure 1.) We call this the *ironed revenue curve*. Just as the derivative of the revenue curve is the virtual

[6]By the definition of the term "quantile", every quantile is equally likely.

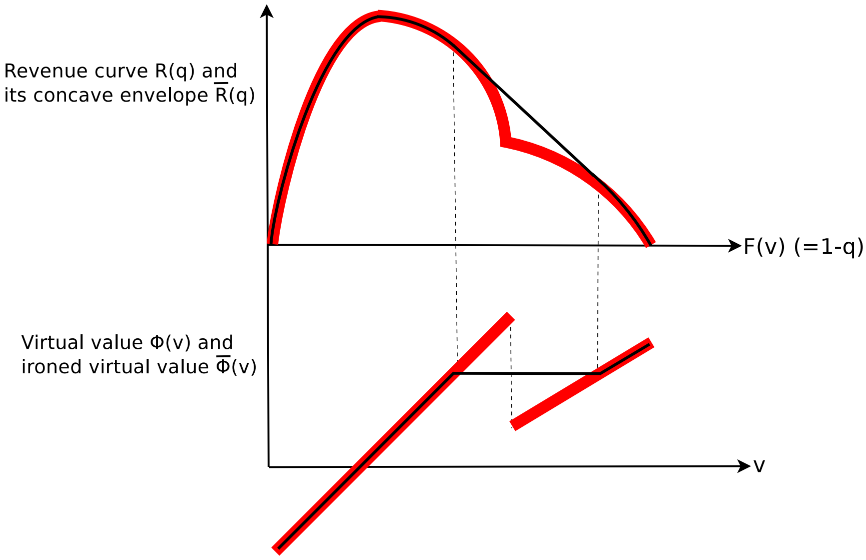

Fig. 1. An example of ironing: the ironed revenue curve is the concave upper envelope of the revenue curve. Notice that over any contiguous interval of quantiles where $R(q)$ and $\bar{R}(q)$ are not equal, $\bar{R}(q)$ is linear, and so, $\bar{\phi}(q)$ is constant.

value of the agent, we call the derivative of the ironed revenue curve his *ironed virtual value*, denoted $\bar{\phi}$.

Now, let us get back to the multi-agent setting and consider the following thought experiment. Pretend that agent i's revenue curve is $\bar{R}_i(q)$ rather than $R_i(q)$, and let us choose the allocation rule that pointwise maximizes the agents' ironed virtual surplus rather than their virtual surplus, subject to feasibility. Then, since the ironed revenue curves are concave, the ironed virtual values are weakly increasing functions and the resulting mechanism is BIC. What about the expected revenue it achieves? Since $\bar{R}_i(q)$ is always larger than $R_i(q)$, the expected ironed virtual surplus of the mechanism is no smaller than the optimal achievable revenue, but its expected virtual surplus may be smaller. We need to account for each agent i for the difference $\mathbf{E}\big[(-x_i{}'(q))(\bar{R}_i(q) - R_i(q))\big]$. Now comes a sleight of hand. With Figure 1 in mind, note that over any contiguous interval where $\bar{R}_i(q)$ and $R_i(q)$ differ, the agent's ironed virtual value is constant. In other words, our proposed mechanism treats different values in this interval the same, and the interim allocation function stays constant over this "ironed" interval[7]. In other words, $x_i{}' = 0$ over such an interval. Consequently, the contribution of this interval to the above difference is 0: the true revenue of $\mathbf{x}$ is the same as its ironed revenue. We therefore obtain the following description of the optimal mechanism: pick an allocation to maximize ironed virtual surplus, breaking ties consistently. We call this *Myerson's*

[7]We need to be a little careful in situations where it is necessary to break ties between two agents with the same ironed virtual value. As long as the tie breaking is also independent of the actual values of the agents, the statement that the allocation function is constant over the ironed interval holds true.

mechanism.

Theorem 3.1 (Myerson (1981)). *The optimal BIC mechanism for single-parameter agents computes the ironed virtual value for each agent and allocates to a subset of agents that maximizes the ironed virtual surplus subject to feasibility, breaking ties consistently and charging each agent her threshold value for getting served.*

An important point to notice about Myerson's mechanism is that it is deterministic and dominant strategy incentive compatible. In other words, in single-parameter settings, the optimal BIC mechanism is in fact DSIC! As we will see later, this property does not hold generally for multi-parameter settings.

Let us apply this theorem to Examples 1 and 2 from Section 2. First, consider a single-item auction setting, and assume that the agents' values for the item are drawn independently from identical regular distributions. Then the virtual value functions ϕ_i for all of the agents are identical. The optimal mechanism allocates the item to the agent with the highest virtual value as long as his virtual value is non-negative. Observe: (1) The agent with the highest virtual value is also the agent with the highest value; (2) The condition that an agent's virtual value is non-negative is equivalent to his value being above the monopoly reserve price for his value distribution that we defined in Section 3—the value at which the revenue curve attains its maximum. In other words, the optimal mechanism is a *Vickrey auction with a reserve price.* For example, if the agents' values are drawn uniformly from the interval $[0, 1]$, then the optimal auction serves the highest value agent if his value is at least $1/2$. Likewise, in a multi-unit auction setting with k items and i.i.d. regular agent values, the optimal mechanism is a k-unit Vickrey auction with a reserve price. On the other hand, the optimal mechanism for the single-item setting with i.i.d. *nonregular* values is not necessarily a Vickrey auction with a reserve price. This is because when two agents are tied for the highest ironed virtual value, the Vickrey auction breaks ties in favor of the higher value agent, whereas the optimal auction breaks ties independently of the agents' values.

What happens when the agents' values are not identically distributed? Suppose we have two agents with v_1 drawn from Unif$[0, 1]$ and v_2 drawn from Unif$[0.5, 1.5]$. Then, the reader can verify that the optimal single-item mechanism serves the second agent if his value is at least $\max\{0.75, v_1 + 0.25\}$ and serves the first agent if his value is at least $\max\{0.5, v_2 - 0.25\}$. The two agents face different reserve prices, and moreover, sometimes the lower value agent is served. In settings with a more general feasibility constraint, even when agents' values are identically distributed, the optimal mechanism can be similarly complex.

Assigning payments. In our discussion so far, we have focused on the form of the allocation function and ignored how to assign payments to agents. For Myerson's mechanism this turns out to be simple, because the mechanism is deterministic: for every fixed value vector of other agents, the allocation rule for an agent is a step function; We assign the agent a payment corresponding to the threshold for that step function.

3.3 Simple near-optimal mechanisms for single-parameter agents.

While in the regular i.i.d. single-item auction setting (as in the example above) the optimal mechanism is the Vickrey auction with a reserve price, in general the optimal mechanism can be quite complex to understand and implement. Hartline and Roughgarden (2009) initiated an investigation into whether simple mechanisms can approximate the performance of the optimal mechanism. There are several approaches for designing simple approximately-optimal mechanisms.

VCG based mechanisms. Recall that the VCG mechanism chooses at any type vector the feasible allocation maximizing surplus, whereas Myerson's optimal mechanism chooses the feasible allocation maximizing virtual surplus. Chawla et al. (2007) first observed that in single-item regular settings, if we first remove all agents with values below their respective monopoly reserve prices r_i^* and run the VCG mechanism over the remaining, we obtain an expected revenue that is at least half of the optimal expected revenue. The intuition behind this is simple: if VCG allocates the item to an agent i who is different from the agent, say j, with the highest virtual value, then agent i needs to pay at least agent j's value which is at least j's virtual value. In other words, either VCG gets a high virtual surplus (by allocating to the highest virtual value agent), or it gets a high revenue (by charging at least the highest virtual value). Setting reserve prices for each agent ensures that VCG never serves an agent with negative virtual value. This analysis is tight. In particular, consider the following one-item two-agent example due to Hartline and Roughgarden (2009). One of the agents has a fixed value of 1, whereas the other has a value drawn from the c.d.f.[8] $F(v) = 1 - 1/v$ for $v \geq 1$, then the VCG mechanism with any reserve price gets a revenue of at most 1. On the other hand, the optimal mechanism obtains a revenue close to 2, by first offering the item to the second agent at a very high price (say $1/\epsilon$ for some $\epsilon > 0$), and if he refuses, offering the item to the first agent at a price of 1.

Hartline and Roughgarden showed that this *VCG with monopoly reserves* mechanism provides a 2-approximation to the optimal revenue in fairly general settings: in regular settings when the feasibility constraint is a matroid[9] and for arbitrary downwards closed feasibility constraints when the value distributions have a monotone hazard rate[10]. The approximation factor of 2 is tight in both cases. On the other hand, no constant factor approximation holds in nonregular settings even with a single item, or in general downwards closed settings even with regular distributions. They further showed that in single-item regular settings, an even simpler

[8]This distribution, known as the *equal revenue distribution* has the special property that its revenue curve is a constant function. In other words, for a single-agent setting with the agent's value drawn from this distribution, all threshold mechanisms get the same expected revenue. This distribution is particularly handy in constructing bad examples for revenue maximization.

[9]A matroid is a set system with the nice property that a greedy algorithm can optimize over it: selecting elements greedily in order of decreasing value produces a maximum value feasible set. Matroid environments include as special cases single-item and multi-item settings. We refer the reader to Brualdi (1969) for further details.

[10]The hazard rate of a distribution with c.d.f. F and density function f is the function $f(v)/(1 - F(v))$. Roughly speaking, distributions with monotone non-decreasing hazard rate (a.k.a. MHR) have tails no heavier than that of an exponential distribution. MHR distributions are regular.

mechanism gives a 4-approximation: VCG with *anonymous* reserves, namely where the same reserve price is used for each of the agents even when their value distributions differ. This result does not extend beyond single-item settings, however: even in k-item regular settings, the gap between the optimal revenue and the revenue of VCG with an anonymous reserve can be as large as $\Omega(\log k)$.

Lookahead auctions. As mentioned earlier, VCG with monopoly reserve prices can perform very poorly when the value distributions are nonregular. Consider, for example, a single-item auction setting where every agent has a value of 1 with probability $1-1/H$ and $H/2$ with probability $1/H$, where $H > 0$ is sufficiently large. The monopoly reserve price for this distribution is 1. The Vickrey auction with a monopoly reserve price therefore gets a revenue of roughly $1 + (H/2)\binom{n}{2}(1/H)^2 = 1 + O(n^2/H)$. The optimal auction on the other hand sets a reserve price of $H/2$, and gets a revenue of $(H/2)(n)(1/H) = n/2$. With $H = n^2$, we get a gap of $\Omega(n)$ between the revenues of the two auctions. In this case, if only we set the reserve price for the Vickrey auction correctly, we would be able to obtain all of the optimal revenue. Ronen (2001) introduced a VCG-type auction for single-item settings that he called the *Lookahead auction*. This auction serves the item to the highest value agent, but charges the agent the monopoly reserve price for his value distribution *conditioned on being the highest value*. When the agent's value distribution is regular, this conditioning simply sets a price for the agent that is the maximum of the second highest value and the monopoly reserve price for his original value distribution. When the value distribution is nonregular, as in the example above, the conditional reserve price can be different. Ronen showed that this mechanism achieves a 2-approximation for arbitrary single-item settings (in fact, even when the agents' values are correlated). Chawla et al. (2014) generalized Ronen's argument to general single-parameter matroid settings with correlated agent values, again obtaining a 2-approximation.

Posted pricing. The third approach for simple near-optimal mechanisms does away with auctions altogether and just uses posted prices. In several settings, auctions are not an option and posted pricing is the only available method for sale. Furthermore, posted-price mechanisms reduce the strategic burden on the bidder by not requiring him to reveal his private value and instead requiring only to accept or reject a posted price. We will see in Section 4 that they also give an approach for approximation in certain multi-parameter settings. Chawla et al. (2010) defined the notion of a sequential posted price mechanism: the mechanism fixes reserve prices for serving each of the agents, and serves any feasible subset of the agents willing to pay their corresponding reserve price. They showed that these posted price mechanisms achieve constant factor approximations in many environments, such as uniform or partition matroid environments, for regular and nonregular value distributions. Furthermore, a constant factor approximation can be obtained for general matroid as well as matroid-intersection environments if the mechanism is allowed to serve the best feasible subset over agents willing to pay their reserve: the subset that maximizes the sum of reserves. Yan (2011) explained the reason behind these results by relating the performance of posted price mechanisms to the concept of *correlation gap*, and also presented tighter analyses of these mechanisms. Chakraborty et al. (2010) developed a PTAS for computing the optimal posted price mechanism in certain multi-unit auction environments.

Single sample mechanisms. Another kind of simple mechanisms do away entirely with dependence on the agents' value distributions. This line of work began with Bulow and Klemperer (1996) who proved that in a single-item setting when all of the agents have identically distributed regular values, the VCG mechanism over $n + 1$ agents obtains a higher expected revenue than the optimal mechanism over n agents. In other words, recruiting an extra bidder is more beneficial than setting the right reserve price. Inspired by this, Dhangwatnotai et al. (2010) showed that in regular i.i.d. settings with a matroid feasibility constraint, the following *single sample* mechanism obtains a 2-approximation: pick a random agent and use his value as a reserve price; run the VCG mechanism with that reserve price over the remaining agents[11]. This mechanism also gives a 4-approximation in downwards closed MHR settings. Note that the single sample mechanism does not need to know about the agents' value distributions at all; all it needs to obtain a good approximation is that the agents have identically distributed values. A variant of the mechanism obtains a good approximation even in non-i.i.d. settings as long as for every agent there is at least one other with an identically distributed value. Sivan and Syrgkanis (2013) show that these results extend also to a large class of nonregular value distributions.

Other mechanisms. Another kind of mechanisms inspired by Bulow and Klemperer (1996) are *supply limiting* mechanisms. These work in multi-item (and some times multi-parameter) settings, and obtain a good approximation by artifically limiting the number of items available to be sold. The reader is referred to Roughgarden et al. (2012); Devanur et al. (2011) for details. Finally, Azar et al. (2013) present approximately-optimal mechanisms that rely only on a few statistics such as the means of the value distributions.

4. REVENUE MAXIMIZATION FOR MULTI-PARAMETER AGENTS

We now consider the revenue maximization problem in settings with multi-parameter agents. The optimal mechanism design problem becomes more complex when buyers do not just care about whether or not they are served, but have different preferences over different outcomes. Revenue maximization in single-parameter contexts is very well understood as we describe in Section 3: We understand properties of the optimal mechanism (it is a virtual value maximizer, and is deterministic, DSIC); We can find the optimal mechanism efficiently (as long as we can optimize for virtual surplus); We understand the power of simple and robust mechanisms. Unfortunately, many of these properties fail to extend to multi-parameter domains.

[11]The intuition behind why this works is as follows. Consider a single agent with a regular value distribution, in other words, a concave revenue curve, and consider picking a random quantile q and running the threshold mechanism τ_q for this agent. The revenue of this randomized mechanism is $\mathbf{E}_q[R(q)]$. Since the revenue curve is concave, this expectation turns out to be at least half of $\max_q R(q)$. But running τ_q with a random q is the same as setting a reserve price for the agent drawn from his own value distribution. With some work, this argument can be made to extend to arbitrary matroid i.i.d. settings.

Optimal multi-dimensional mechanisms are not always deterministic, and do not seem to permit succinct descriptions (Rochet and Chone, 1998; Manelli and Vincent, 2007).

Our focus in this section will primarily be computational: how do we efficiently find optimal or near optimal mechanisms? As a side effect, we will be able to prove interesting properties of optimal or near-optimal mechanisms in some settings. For example, we will show that the optimal mechanism can always be written as a convex combination of polynomially[12] many virtual value maximizers. Furthermore, in some settings, posted pricing based mechanisms obtain constant factor approximations.

We will focus on settings where the seller has many different items to offer, and the buyers have different values for different bundles of items. A natural assumption for the buyers' value functions is that they are monotone increasing in the item bundles: allocating an extra item to an agent weakly increases his value. However, we will need further assumptions in order to be able to approximate the optimal mechanism. Note that with m items under sale, the agent's value function is a 2^m-dimensional vector. At a minimum the buyer should be able to efficiently communicate this vector to the seller. Two approaches have been considered in the literature to address this[13]: (1) Each agent's type space is small and known to the seller, so that the agent merely needs to communicate his type to the seller, and not necessarily explicitly specify his value for each possible outcome. Here we will design mechanisms in time polynomial in the sizes of the type spaces, $\sum_i |T_i|$; (2) Each agent's type is succinctly representable, e.g. the agent is only interested in buying singleton items, or his value is additive over items in which case it suffices to specify the per-item values. Here we will design mechanisms in time polynomial in the size of the representation.

Randomized versus deterministic mechanisms. An important difference between the single-parameter and multi-parameter mechanism design problems is that in the latter case mechanisms can benefit from using a new kind of randomization, namely selling random allocations. Let us consider the mechanism design problem with a single agent. A deterministic mechanism for a single agent prices each outcome[14] at a certain price, and lets the buyer choose which outcome he prefers. A utility maximizing buyer chooses the outcome that maximizes his value for the outcome minus the corresponding price. A randomized mechanism can set a random price for each possible outcome. But, interestingly, in addition it can also price random allocations or *lotteries*. A lottery is a distribution over outcomes. For example,

[12] We mean polynomial in the natural parameters of the problem: the number of agents, the number of items, the sizes of the type spaces, and the description complexity of the types.

[13] In prior-free welfare maximization literature, a third approach has been studied, which uses demand or value queries to elicit relevant parts of the agents' value functions; we refer the reader to Blumrosen and Nisan (2009).

[14] Here an outcome is a bundle of items allocated to the buyer. A deterministic mechanism assigns an outcome and a price to each type of the agent. For incentive compatibility, different types mapped to the same outcome must pay the same price. Furthermore, no type should prefer the (outcome, price) pair at another type to its own. Therefore, we can interpret the mechanism as creating a menu of all the (outcome, price) pairs that it wants to allocate at some type, and letting the agent choose which one he prefers most at his given type.

for a buyer interested in obtaining one of two different items, a $(1/2, 1/2)$ lottery would allocate to the buyer the first item with probability $1/2$ and the second item with probability $1/2$. When a buyer buys a lottery, we assume that the outcome of the lottery is determined *after* the sale is completed. In particular, his value for a lottery is his expected value for an outcome drawn from the corresponding distribution (assuming that the buyer is risk neutral). A randomized mechanism is a menu of lotteries along with a price for each; The buyer is offered the menu and can choose which lottery he wants to buy.

We emphasize that mechanisms that sell lotteries cannot necessarily be expressed as distributions over deterministic mechanisms[15]. Lotteries are therefore to be thought of as new outcomes that are distinct from (albeit derived from) the original set of deterministic outcomes. This is in contrast to single-parameter settings where all randomized BNE mechanisms are distributions over deterministic BNE mechanisms. A consequence is that selling lotteries can improve the seller's expected revenue. Pavlov (2011a) gives a simple example exhibiting this. Consider a setting with two items for sale; The unit-demand buyer has three equally likely types with values $(1, 0)$, $(0, 1)$, or $(1/2, 1/2)$ for the two items. In other words, the buyer either values only one of the items at 1, or is indifferent between the two items at $1/2$. Then, an optimal item pricing for this setting is to price both items at 1. A buyer with the first or the second type buys an item; A buyer with the third type does not. So, the expected revenue of the item pricing is $2/3$. Now consider adding a lottery to this menu that allocates one of the two items with equal probability, and is priced at $1/2$. A buyer with the first or the second type will not want to switch to such a lottery, because it does not bring him any benefit. A buyer with the third type, however, will now be willing to buy this lottery, as opposed to not buying anything at all. So, the new mechanism obtains a revenue of $5/6$, which is 25% higher than that of the item pricing. Such a gap between deterministic and randomized revenue exists also when the buyer's values for the two items are independently distributed (Thanassoulis (2004) presents an example).

From the revenue maximization viewpoint, there are three questions of interest:

(1) What do optimal randomized mechanisms look like and how do we optimize for revenue over this class?
(2) How do we optimize for revenue over the class of deterministic mechanisms?
(3) What is the power of randomization, i.e., how much better are randomized mechanisms over deterministic mechanisms with respect to revenue?

[15] One "proof" of this statement is as follows. Consider any deterministic mechanism for selling one of two items to a single unit-demand buyer. This mechanism sets a price for each item and lets the agent buy his favorite item. Note that fixing the item prices, if the buyer's value for item 1 decreases, he is equally or more likely to buy item 2. Now consider selling a $(1/2, 1/2)$ lottery over the items at some fixed price. Suppose the buyer has a high value for item 1 and low value for item 2, so that he is willing to pay the price for the lottery. As the buyer's value for item 1 decreases, at some point, he stops buying the lottery and no longer receives item 2. In other words, decreasing the buyer's value for item 1 *decreases* his allocation for item 2, and we cannot express the outcome of the lottery mechanism as a distribution over outcomes of the pricing mechanisms. Mathematically, the lottery mechanism charges a single price for a distribution over the items; It is not possible to distribute this price across the individual items in a way that preserves the purchasing behavior for all possible types of buyers.

We begin by discussing these questions for agents with succinctly represented type spaces.

4.1 Multi-parameter agents with succinctly representable types

Two kinds of settings with succinctly representable types are known to be tractable for revenue maximization: unit-demand agents, and agents with additive values. A *unit-demand agent* is only interested in buying a single item: his value for a set of items is equal to the value for the most desirable item in that set; This is the setting in Example 3. An *additive value agent* is interested in buying as many items as he can: his value for a set of items is the sum of the values of items in that set; This is the setting in Example 4. In both cases, the agent's type can be described succinctly using m numbers, one per item, where m is the number of items. Both of these settings are quite natural. Unit-demand preferences apply, for example, when the seller is selling houses or cars or a luxury item. Additive preferences may apply when the items for sale have very different uses from each other, for example, a buyer may buy a pair of shoes or a box of cereal or both and buying one does not increase or decrease her value for the other. As we will see, the revenue maximization problem for a unit-demand buyer can be quite different from that for an additive buyer, although similar approximations are known in either case.

One unit-demand agent with correlated item values. We have previously seen from Pavlov's example that randomized mechanisms can be more powerful than item pricings in this setting. What is the worst-case ratio between the maximum expected revenue of a randomized mechanism and that of an item pricing? Recall that a randomized mechanism is a menu of lotteries, each associated with a price; once again, the agent is free to select any lottery from the menu at the given prices. It is easy to see that the gap between lottery pricings and item pricings can be no larger than the number of distinct types of the buyer. In fact, it is no larger than the number of distinct lotteries sold by the optimal mechanism, the so-called *menu size complexity* of the optimal mechanism[16]. In settings with an unbounded type space, can we bound it in terms of the number of items? Surprisingly, the answer is no: this gap can be unbounded even when there are just two distinct items (Briest et al., 2014; Hart and Nisan, 2013); In particular, there is a setting with just two items where a lottery pricing obtains infinite expected revenue, whereas the revenue of any item pricing is bounded by a constant. Briest (2008) showed that the optimal deterministic mechanism a.k.a. item pricing cannot be approximated to within any subpolynomial factor.

One unit-demand agent with independent item values. These results motivate considering special cases where the item pricing and lottery pricing problems are tractable. One such setting was considered by Chawla, Hartline, and Kleinberg (2007) and Chawla, Malec, and Sivan (2012). Recall that the unit-demand agent's

[16]To see this, note that for a mechanism selling k distinct lotteries, there is another mechanism selling just one lottery with at least a $1/k$ fraction of the former's revenue. But we can obtain the revenue of a single lottery by selling each item individually at the same price as that of the lottery; A unit-demand buyer that wants to buy the lottery would want to buy at least one item at the same price.

type is given by a value for each different item. These papers assume that the agent's values for different items are independent. Under this assumption, the first paper presented an algorithm to compute an item pricing that obtains at least half the revenue of the optimal item pricing. The second showed that the same pricing is a 4-approximation to the optimal lottery pricing, and as a consequence the revenue of the optimal lottery pricing is no more than a factor of 4 times the revenue of the optimal item pricing, regardless of the number of items.

We will first describe Chawla, Hartline, and Kleinberg (2007)'s approach for designing an approximately optimal item pricing for a single unit-demand agent. The key element in this approach is to interpret the unit-demand item pricing problem as a single-parameter mechanism design problem with colluding agents. In particular, let $\mathcal{I}$ denote an instance of the unit-demand pricing problem where the agent's value v_i for item i is drawn from independent distribution F_i. On the other hand, let $\hat{\mathcal{I}}$ denote an instance of the single-item auction problem (i.e. Example 1) with m agents, where agent i has a value v_i drawn from independent distribution F_i. Note that the two instances are quite similar in the type vectors they take as input. What makes them different is how the agents in the two settings react to the rules of the mechanism. Let us suppose, as a thought experiment, that the unit-demand agent in the instance $\mathcal{I}$ has m representatives that he sends to participate on his behalf in a mechanism for the instance $\hat{\mathcal{I}}$. Representative i wants to buy item i on behalf of the unit-demand agent, and is not interested in any other item. However, in order to represent the unit-demand agent faithfully, the representatives collude so as to maximize their collective utility rather than their individual utilities. Then it holds that any mechanism for the instance $\hat{\mathcal{I}}$ with colluding agents can be converted to a mechanism for the instance $\mathcal{I}$ with the same equilibrium outcome. In other words, we can think of deterministic mechanisms for the unit-demand instance $\mathcal{I}$ as collusion-robust mechanisms for the single-parameter instance $\hat{\mathcal{I}}$ and vice versa[17]. On the other hand, the optimal mechanism (namely, Myerson's mechanism) for $\hat{\mathcal{I}}$ is not collusion robust (in the sense that collusion among agents can hurt its revenue guarantee), and gets more expected revenue. To summarize, the optimal revenue achievable by an item pricing for $\mathcal{I}$ is: (1) no more than the revenue of Myerson's mechanism for $\hat{\mathcal{I}}$, and, (2) no less than the revenue of a collusion robust mechanism for $\hat{\mathcal{I}}$. This motivates designing a collusion robust mechanism for $\hat{\mathcal{I}}$ that gets nearly as much revenue as Myerson's mechanism for the same setting.

Before we describe such a mechanism, we will take a detour to describe a related problem from optimal stopping theory called the *gambler's problem*): a gambler is presented with m boxes, each of which has a reward in it. The reward in box i is drawn from a known distribution. The gambler opens the boxes in succession. Upon opening box i, the gambler either decides to keep it, in which case the game ends and no more boxes are opened, or he decides to leave it, in which case he can never reclaim that reward. The gambler competes against a prophet who knows the instantiations of all of the rewards in advance and can therefore obtain the

[17]This correspondence does not work with randomized mechanisms, because, for example, if a randomized mechanism for $\mathcal{I}$ allocates a lottery over some set of items to the buyer at some price, there is no good way to distribute this price across the representatives corresponding to the set of items.

largest one. The gambler's problem is to determine a strategy that obtains nearly as much reward as the prophet. A priori it is not clear whether any approximation to the prophet's expected reward is achievable. Samuel-Cahn (1984) presented a very simple threshold strategy for this problem that obtains half of the prophet's expected reward in expectation: the gambler picks a threshold r such that the probability that at least one of the rewards exceeds this threshold is *exactly* $1/2$;[18] He then accepts the first reward that exceeds this threshold, if any. This result is called a *prophet inequality.* At first it is a little strange that the gambler achieves a 2-approximation to the prophet's reward given that with $1/2$ probability he does not accept anything at all. This is because when the gambler fails to obtain any reward, the prophet's reward is also low; On the other hand, when the gambler does win the reward, with a good probability, this is the only reward that exceeds r and is therefore equal to the reward that the prophet obtains. A very nice property of this threshold strategy that is especially useful for us in the mechanism design context is that the solution does not depend upon the ordering of the rewards; Indeed the gambler does not even need to know the ordering in advance. In fact we may as well assume that the gambler obtains the smallest reward that exceeds the threshold, and the 2-approximation continues to hold.

Let us now get back to the problem of designing a collusion robust mechanism for the single-item auction problem. Recall from Section 3.2 (Equation (3)) that the expected revenue of any BIC mechanism is equal to its expected virtual surplus, or in the single-item setting, the expected virtual value of the agent that wins the item. Consider an instance of the gambler's problem described above with the rewards being the virtual values corresponding to the agents in the single-item auction. The optimal mechanism for the single-item auction behaves like the prophet: it collects the maximum virtual value as long as it is nonnegative. We argue, on the other hand, that we can convert the threshold strategy of Samuel-Cahn into a collusion robust mechanism that obtains as much revenue as that strategy. Let r denote the gambler's threshold. Precisely, we allocate the item to the agent corresponding to the reward picked by the gambler. We charge the agent the value at which his virtual value equals the threshold r. Note that this mechanism is BIC: the higher that the virtual value of an agent, the higher is his probability of getting the item. It obtains as much expected revenue as the gambler's expected reward. Since the threshold r depends only on the reward distributions, and not the actual rewards, the price paid by an agent is also independent of others' reported values. Finally, by colluding, the agents can merely affect which agent with a virtual value above the threshold gets the item but cannot affect the price that this agent pays. As noted earlier, the gambler's reward, equivalently the mechanism's revenue, is robust to such a collusion. Therefore, the mechanism corresponding to Samuel-Cahn's threshold strategy is a collusion-robust 2-approximation to $\hat{\mathcal{I}}$. What does this mechanism look like? It picks a threshold on virtual values; This threshold corresponds to a price p_i, one for each agent i. Of the agents that have values exceeding their corresponding prices, the agents can collude and decide who wins

[18]We assume the reward distributions are continuous so that r always exists. Other thresholds work when this assumption does not hold. In fact there is a continuum of thresholds that gives the same approximation.

the item. In the unit demand setting $\mathcal{I}$, this corresponds to an item pricing $\mathbf{p}$ offered to the unit-demand agent; The agent can then pick whichever item he desires. We therefore obtain a 2-approximation to the unit-demand item pricing problem[19].

While this approach leads to a 2-approximation for item pricing in the unit-demand setting, it does not immediately give an approximation to the optimal lottery pricing. The problem is that the expected revenue of a lottery pricing for $\mathcal{I}$ can exceed that of the optimal mechanism for $\hat{\mathcal{I}}$. Chawla, Malec, and Sivan (2012) show, however, that the difference between the two can be charged to the expected revenue of the Vickrey auction over representatives in $\hat{\mathcal{I}}$. This leads to a bound of 4 on the gap between the revenue of the optimal randomized and the optimal deterministic mechanism in this setting.

Cai and Daskalakis (2011) consider the same unit-demand pricing problem and design a PTAS to the optimal pricing scheme when the distributions are independent and satisfy the monotone hazard rate (MHR) condition, and obtain a quasi-PTAS when the distributions are independent and regular.

New prophet inequalities. The connection between collusion robust mechanism design and the gambler's problem extends to more general feasibility constraints, and has spurred an interest in prophet inequalities and led to new results for the gambler's problem. See, e.g., Chawla et al. (2010); Alaei (2014); Kleinberg and Weinberg (2012); Azar et al. (2014).

Many unit-demand agents with independent item values. Chawla, Hartline, Malec, and Sivan (2010) generalize the approach of Chawla et al. (2007) to multi-agent settings where agents are unit-demand and their values for different items are independent. In this setting, they assume that the seller faces a general downwards closed feasibility constraint on the allocations that he can make. As in the single-agent setting, they upper bound the revenue of the optimal deterministic mechanism in the multi-parameter unit-demand setting by the revenue in a related single-parameter setting where every multi-parameter agent is replaced by several single-parameter representatives. The representatives corresponding to a single multi-parameter agent collude among themselves. Any mechanism that is collusion-robust can then be ported back into the original multi-parameter setting without loss in performance. It remains then to design collusion robust mechanisms for the corresponding single-parameter setting. They introduce sequential posted price mechanisms, that we previously discussed in Section 3.3, and show that these mechanisms are both collusion robust and provide constant factor approximations to the optimal expected revenue for many kinds of feasibility constraints. As for the single-agent unit-demand setting, Chawla, Malec, and Sivan (2012) extend the reach of posted-price mechanisms and show that they also approximate the optimal randomized mechanism's revenue.

A different approach for extending the single-agent unit-demand result to multi-agent settings is given by Alaei (2014). He considers the following generalized single-agent problem: given a vector $\mathbf{y}$ of probabilities, find the revenue optimal

[19] While the item pricing that we describe is due to Chawla et al. (2007), that paper only proved that it achieves a 3-approximation; The connection to the gambler's problem and the improved approximation factor of 2 is due to Chawla et al. (2010).

mechanism for a single unit-demand agent such that for all items j, the ex ante probability with which the mechanism allocates item j to the agent is no more than the probability y_j. He argues that if it is possible to solve the above single-agent problem efficiently, then under a large class of downwards closed feasibility constraints, it is possible to approximate the multi-agent problem. Let $R_i(\mathbf{y}_i)$ denote the optimal revenue that can be obtained from agent i given the ex ante constraint $\mathbf{y}_i$. Now we can proceed as follows: (1) We find ex ante allocations to each agent that maximize the sum of the corresponding revenues. In other words, Alaei shows how to solve the program: maximize $\sum_i R_i(\mathbf{y}_i)$ subject to $\sum_j y_{ij} \leq 1$ for all items j;[20] (2) Given the solution to this program, we solve the per-agent mechanism design problem for the optimal ex ante allocations, that is, for each agent i, given $\mathbf{y}_i$ find single-agent mechanisms M_i that obtain revenue $R_i(\mathbf{y}_i)$ subject to the constraint of selling an item j with ex ante probability at most y_{ij}; (3) Finally, we convert the per agent mechanisms to an overall multi-agent mechanism satisfying supply constraints. That is, if at some type vector, the mechanisms $\{M_i\}$ collectively allocate more than one copy of some item, then we need to deallocate some items without losing too much revenue. Alaei shows that this approach leads to improved constant factor approximations in some settings.

One additive agent. Next we consider the case of an additive buyer and assume as before that the buyer's values for different items are independent. As in the unit-demand case, a deterministic mechanism for an additive buyer may set prices, one for each individual item, and allow the agent to buy the set he desires. Since the agent's value for sets of items is additive over the items, his buying decisions and therefore also the mechanism's pricing is independent across items. One may suspect that the optimal deterministic mechanism is of this format. Surprisingly, this is not true. Suppose, for example, that the buyer's value for each item is distributed uniformly between 0 and 1. Then an item pricing prices each item at 0.5 and obtains on average 0.25 per item. On the other hand, when the number of items m is large enough, selling the entire set of items at a price of $m/2 - \epsilon$ obtains a revenue of nearly 0.5 per item, doubling the seller's revenue. Selling the items together as a set reduces the uncertainty in the buyer's value, leading to better revenue extraction.

The optimal deterministic mechanism may in fact specify a price for each possible *bundle* of items, allowing the agent to select his favorite bundle. Consequently, it may not permit a succinct representation. We therefore further ask: is there a succinctly representable mechanism that approximates the optimal deterministic or the optimal randomized mechanism?

Early work on this problem (e.g., Manelli and Vincent, 2006, 2007; Pavlov, 2011a,b) focused on specific value distributions. Daskalakis et al. (2013) and Wang and Tang (2014) studied the special case of only two items. The first general approximation to the optimal mechanism was provided by Hart and Nisan (2012). They showed that selling all items separately yields an $O(\log^2 m)$ approximation to the optimal randomized revenue, where m is the number of items. Further improvements were obtained by Li and Yao (2013) and Babaioff et al. (2014) culminating in

[20]Here we assume, for simplicity of exposition that we have one copy of each item available.

a constant factor approximation to the optimal revenue. Babaioff et al. essentially showed that the only reason that pricing each item independently may not be a good idea is when the sum of values for all the items is concentrated. In particular, one of the following two simple mechanisms obtains a constant factor approximation: (1) price each item separately, and, (2) price just the "grand bundle" of all items.

To summarize, in both unit-demand and additive settings, when the agent's values for individual items are independent, a simple item pricing or bundling approach gives a constant factor approximation to the optimal randomized revenue. Does this property extend to other linear value functions with independent item values, e.g. values that are additive up to bundles of size k? We believe the answer should be yes, but no results are known for such settings.

Many additive agents and beyond. Extending the work of Babaioff et al. (2014) to more than one agent or to more general value functions (e.g. additive up to k items) remains open. Cai and Huang (2013) tackle a special case of many additive agents: i.i.d. agents and all value distributions are MHR. Using properties of MHR distributions along the lines of Cai and Daskalakis (2011), they design a PTAS for this problem using second-price and VCG allocation rules.

4.2 Multi-parameter agents with small type spaces

Next we consider settings where the agents' type spaces are discrete and small in size. The key to optimal mechanism design in this case is to express the problem in the form of a compact linear or convex program. For concreteness, we will consider a seller with m items. A buyer's type will then consist of 2^m different values, one for each bundle of items. We will make the simplifying assumption that each buyer's type is linear over the items. We address the setting of non-linear types towards the end of this section. Formally, an agent i's type as well as allocation will be m-dimensional vectors, $\mathbf{v}_i$ and $\mathbf{x}_i$ respectively, with the jth component v_{ij} specifying the value that the agent derives from item j alone, and the jth component x_{ij} specifying the probability with which item j is allocated to the agent. For allocations that are feasible according to a given feasibility constraint, $\mathbf{x}_i \in \mathcal{F}$, the agent's value will be given by $\mathbf{v}_i(\mathbf{x}_i) = \mathbf{v}_i \cdot \mathbf{x}_i = \sum_j v_{ij} x_{ij}$. The feasibility constraint allows us to encode agents of different kinds, e.g. unit-demand agents that only desire one item ($\mathcal{F}$ includes all vectors with ℓ_1 norm at most 1), and agents that desire up to k items but have additive values over them ($\mathcal{F}$ includes all vectors with ℓ_1 norm at most k).

Recall that we use T_i to denote the set of types for agent i. Our goal will be to find a (near-)optimal mechanism in time polynomial in $\sum_i |T_i|$.

The special case of a single agent. For a single agent, the optimal mechanism design problem can be expressed as a linear program. The variables in the program are the allocations $\mathbf{x}(\mathbf{v})$ and prices $p(\mathbf{v})$ at each type $\mathbf{v}$. The revenue objective can be written as: $\sum_{\mathbf{v} \in T} \Pr[\mathbf{v}] p(\mathbf{v})$. The following constraints encode Bayesian IC and

IR respectively.

$$
\begin{aligned}
\mathbf{v}\cdot\mathbf{x}(\mathbf{v}) - p(\mathbf{v}) &\geq \mathbf{v}\cdot\mathbf{x}(\mathbf{v}') - p(\mathbf{v}') && \text{for all types } \mathbf{v},\mathbf{v}' \in T && (4)\\
\mathbf{v}\cdot\mathbf{x}(\mathbf{v}) - p(\mathbf{v}) &\geq 0 && \text{for all types } \mathbf{v} \in T && (5)\\
\mathbf{x}(\mathbf{v}) &\in \mathcal{F} && \text{for all types } \mathbf{v} \in T
\end{aligned}
$$

Here we are implicitly assuming that the feasibility constraint $\mathcal{F}$ can be expressed via (a few) linear constraints. The optimal mechanism can therefore be found in time polynomial in the size of the type space. We remark that the linear program optimizes over randomized BIC mechanisms. An implication is that the size of the menu of the optimal mechanism is no larger than the number of distinct types of the buyer.

Multiple agents: reduced forms. In principle, we can write a linear program for revenue maximization with multiple agents just as we did for a single agent. The problem is that we now need to select outcomes for each possible tuple of values of all agents, the number of which is $\prod_i |T_i|$, and not $\sum_i |T_i|$. To get around this issue, we will write the linear program in terms of agents' *interim* allocations and payments, rather than their ex post allocations and payments. For BIC it suffices to express the constraints (4) and (5) in terms of the interim quantities. We call these interim functions the *reduced form* of the mechanism.

Going from ex post allocations to interim allocations necessarily involves some loss in information about the mechanism. Given an interim allocation function, what is the (distribution over) ex post allocation rule(s) that implements it? Can every interim allocation rule be expressed as a convex combination over ex post rules? In other words, given some interim allocation rule $\mathbf{x}(\cdot)$, can we find ex post allocations $\mathbf{y}(\mathbf{v})$ at every tuple of values $\mathbf{v}$ that have the right "marginal sums" $x_{ij}(\mathbf{v}_i) = \mathbf{E}_{\mathbf{v}_{-i}}[y_{ij}(\mathbf{v})]$ for all i, j?

The answer to this question is no, even in single-item settings. Consider for example a single-item auction setting with two agents, and suppose that both agents have two equally likely values v_1 and v_2. We want to implement the following interim allocation function: $x_1(v_1) = 0.8$, $x_1(v_2) = 0.4$, $x_2(v_1) = 0.6$, and, $x_2(v_2) = 0.3$. Our only constraint is that at any vector of values, the ex post probabilities with which we allocate the item to the two agents sum up to no more than 1. The reader can verify through trial and error (or by setting up linear equations over 8 variables, each corresponding to the ex post allocation for one of the two agents for one of the four possible value vectors) that this is not possible. An easy way to see why is that the ex ante probability of allocation to agent 1 is $(0.8 + 0.4)/2 = 0.6$, and the ex ante probability of allocation to agent 2 is $(0.6+0.3)/2 = 0.45$, and these two probabilities sum up to more than 1. That is, we are trying to allocate more than one item on average, and so any ex post allocation matching these marginals must become infeasible at some value vector.

We can now summarize our reduced form program (RF-LP) as follows.

$$\text{maximize } \mathbf{E}_{\mathbf{v}}\left[\sum_i p_i(\mathbf{v}_i)\right] \text{ subject to} \qquad \text{(RF-LP)}$$

$$\mathbf{v}_i \cdot \mathbf{x}_i(\mathbf{v}_i) - p_i(\mathbf{v}_i) \geq \mathbf{v}_i \cdot \mathbf{x}_i(\mathbf{v}_i{}') - p_i(\mathbf{v}_i{}') \qquad \forall i, \mathbf{v}_i, \mathbf{v}_i{}' \in T_i$$

$$\mathbf{v}_i \cdot \mathbf{x}_i(\mathbf{v}_i) - p_i(\mathbf{v}_i) \geq 0 \qquad \forall i, \mathbf{v}_i \in T_i$$

$$\exists \text{ feasible rule } \mathbf{y} \text{ such that, } x_{ij}(\mathbf{v}_i) = \mathbf{E}_{\mathbf{v}_{-i}}[y_{ij}(\mathbf{v})] \qquad \forall i, j, \mathbf{v}_i \in T_i \qquad (6)$$

Letting $\mathcal{P}$ be the set of all interim allocation rules[21] that are both feasible with respect to $\mathcal{F}$ and expressible using feasible ex post allocation rules, we can write constraint (6) more succinctly as:

$$\mathbf{x}(\cdot) \in \mathcal{P} \qquad (7)$$

The challenge now is to solve the above program with the constraint (7). A priori it is not clear whether the above program is a linear or even convex program. We describe two approaches for checking constraint (7) for a given interim allocation rule $\mathbf{x}$. The first approach is combinatorial but has limited applicability. The second argues that $\mathcal{P}$ is a convex polytope and uses the ellipsoid method to solve the program. We briefly mention a third approach for solving (RF-LP) directly towards the end of this section.

Border's approach for reduced form feasibility in a single-item auction. As mentioned above, in a single item auction setting, at a minimum, the interim allocation rule $\mathbf{x}$ should satisfy the property that the ex ante allocation probabilities for all of the agents sum up to no more than 1. More generally, consider a set of values $S_i \subseteq T_i$ for agent i. The probability that the agent has a value that belongs to S_i and is allocated the item is $\sum_{v_i \in S_i} \Pr[v_i] x_i(v_i)$. The probability that the item goes to some agent i with a value in S_i is the sum of these quantities over all agents: $\sum_i \sum_{v_i \in S_i} \Pr[v_i] x_i(v_i)$. It must be true, trivially, that this sum is no more than the total probability that some agent i has a value in S_i. The latter probability is just $1 - \prod_i \Pr[v_i \notin S_i] = 1 - \prod_i (1 - \sum_{v_i \in S_i} \Pr[v_i])$. Border (2007) and Che et al. (2013) independently proved that checking this condition for all possible choices of the sets S_i is, in fact, sufficient to guarantee the feasibility of an interim allocation rule in the single-item setting[22]:

$$\sum_i \sum_{v_i \in S_i} \Pr[v_i] x_i(v_i) \leq 1 - \prod_i \left(1 - \sum_{v_i \in S_i} \Pr[v_i]\right)$$

These conditions reduce the feasibility checking for a reduced form from solving a set of equations on $n \prod_i |T_i|$ variables to checking merely $\sum_i 2^{|T_i|}$ equations. For large type spaces, this is still undesirable. Cai et al. (2012a) and Alaei et al.

[21] We remark that this set depends on any feasibility constraints the designer faces, such as a supply constraint on the items, and also on the distribution over values. As such it should be written as a function of those two arguments. For simplicity of exposition, we do not specify the arguments.

[22] Necessary and sufficient conditions for agents with i.i.d. values were previously given by Border (1991)

(2012) further showed that it is sufficient to check these conditions for only polynomially many different tuples of sets $\{S_i\}$. They also gave efficient approaches for decomposing a feasible reduced form into a convex combination over ex post allocation rules. However, beyond single-item settings, it is not clear whether Border's approach can lead to efficient verification of feasibility constraints.

Feasibility checking beyond a single item. The second approach to checking the feasibility of a reduced form was developed in a series of papers by Cai, Daskalakis, and Weinberg (2012b; 2013a; 2013b). Recall that given a reduced form $\mathbf{x}$, our goal is to determine whether $\mathbf{x} \in \mathcal{P}$ (constraint (7)). Observe that the set $\mathcal{P}$ is a convex set: if two reduced forms are feasible, any convex combination of them is also feasible. In particular, $\mathcal{P}$ is just the convex hull of the reduced forms of all feasible ex post allocation rules.

Cai et al. essentially show that if we can efficiently (approximately) maximize social welfare over $\mathcal{P}$, then we can use the social welfare maximization algorithm as a subroutine to (approximately) check for the membership of $\mathbf{x}$ in $\mathcal{P}$. This allows us to use the ellipsoid algorithm to solve (RF-LP).

We stress that this approach gives a reduction from revenue maximization to social welfare maximization that is black-box (as defined in Section 5) in the sense that the reduction is "generic" and does not need to understand the inner functioning of the algorithm for social welfare. The approach also has similarities with Myerson's approach for single-parameter revenue maximization in that it queries the social welfare algorithm at value vectors that are not the agents' real values, but are functions of the real values; these "fake" values can be thought of as virtual values.

We will now describe this approach in detail.

Checking for membership in $\mathcal{P}$ using a virtual welfare maximizing algorithm. Henceforth we will assume access to an efficient algorithm for maximizing social welfare over feasible ex post allocation rules, equivalently over the set $\mathcal{P}$. The algorithm takes as input a distribution over agents' values and returns a feasible ex post allocation rule that maximizes social welfare in expectation over the values. In other words, it solves a problem of the form: maximize $\mathbf{E}_{\mathbf{v}}\left[\sum_{i,j} w_{ij} x_{ij}(\mathbf{v}_i)\right]$ subject to $\mathbf{x} \in \mathcal{P}$. Succinctly, we can write this as: maximize $\mathbf{w} \cdot \mathbf{x}$ subject to $\mathbf{x} \in \mathcal{P}$. We can interpret the quantities w_{ij} as "virtual values", so that, the algorithm maximizes expected virtual surplus subject to feasibility.

Having a virtual surplus maximizing algorithm as above allows us to maximize any linear function over the polytope $\mathcal{P}$. We can relate this to checking for the feasibility of a reduced form $\mathbf{x}$ in $\mathcal{P}$ by noting that a reduced form $\mathbf{x}$ is feasible if and only if for all vectors $\mathbf{w}$, we have $\mathbf{w} \cdot \mathbf{x} \leq \max_{\mathbf{y} \in \mathcal{P}} \mathbf{w} \cdot \mathbf{y}$.

In particular, given a reduced form $\mathbf{x}$, we can determine whether or not it is feasible (and find a violating constraint $\mathbf{w}$, if not) by solving the following program. Here $\mathbf{x}$ is fixed and $\mathbf{w}$ are the variables we are optimizing over.

$$\text{maximize} \quad \mathbf{w} \cdot \mathbf{x} - W \quad \text{subject to} \qquad \text{(Separation-LP)}$$
$$W \geq \mathbf{w} \cdot \mathbf{y} \quad \text{for all } \mathbf{y} \in \mathcal{P}$$

(Separation-LP) can again be solved using the ellipsoid algorithm as follows: given a vector $\mathbf{w}$ we can find $W = \max_{\mathbf{y} \in \mathcal{P}} \mathbf{w} \cdot \mathbf{y}$ using the virtual surplus maximizing algorithm given to us; This gives us a separation oracle for (Separation-LP) which we can use to solve the program.

To summarize, given a reduced form $\mathbf{x}$, we first solve (Separation-LP) using the virtual welfare maximization algorithm as a black-box. The solution to the program either tells us that $\mathbf{x} \in \mathcal{P}$, or gives us a violating constraint $\mathbf{w}$. In other words, we get a separation oracle for $\mathcal{P}$.

Solving the linear program (RF-LP)**.** We described how to implement a separation oracle for the constraint (7). Armed with this, we can run the ellipsoid algorithm over (RF-LP) to find an optimal feasible reduced form. Next we need to figure out how to decompose this reduced form into a convex combination over ex post allocation rules. What if the reduced form does not have a succinct decomposition? Fortunately, since reduced forms are $(m \sum_i |T_i|)$-dimensional vectors (specifying the interim allocation probability for each item at each possible value of each agent), Caratheodory's theorem implies that they can be written as a convex combination over at most $m \sum_i |T_i|$ extreme points of $\mathcal{P}$. Each extreme point, on the other hand, is the solution to a virtual welfare maximization problem, and can therefore be succinctly represented by simply specifying the corresponding virtual values. This implies that every feasible reduced form has a succinct decomposition, and can also be used to find one.

We omitted a couple of details in the above description that we now address:

—In some settings we only have an approximate virtual welfare maximizing algorithm, and not an exact one. Cai et al. (2013a) show that this is in fact sufficient to obtain an approximate separation oracle for (7) satisfying some nice properties, that further lead to an approximate solution to the linear program. The separation oracle they construct can also be used to decompose feasible points into convex combinations of the extreme points of $\mathcal{P}$.

—In order to solve (Separation-LP), we don't just need access to an algorithm that produces a virtual welfare maximizing allocation function, but also need to know the expected virtual welfare that the allocation achieves, in other words, the quantity $\max_{\mathbf{y} \in \mathcal{P}} \mathbf{w} \cdot \mathbf{y}$ in addition to the allocation $\mathbf{y}$ that maximizes it. Estimating this quantity requires sampling from the agents' value spaces and leads to both a small loss in our overall expected revenue and a mechanism that is ϵ-BIC instead of BIC.

Beyond linear types. Recall that in order to be able to express the BIC constraints as linear constraints in (RF-LP), we made the assumption that agents' values are linear (additive) over the items. Cai et al. (2013b) show how to move beyond this assumption. Their key insight is to expand the reduced form of a mechanism to specify in addition to the interim allocation function the following two quantities: (1) the value that the agent receives at a particular type from the interim allocation at that type; (2) the value that the agent receives at a particular type by misreporting a different type and obtaining the interim allocation corresponding to the latter type. These quantities are sufficient to encode BIC. The rest of the approach can then be modified to work with the expanded reduced form.

An alternate approach to solving (RF-LP). Bhalgat et al. (2013) give an alternative approach for solving the program (RF-LP) by including in the program the exponentially many ex post allocation and payment variables, and expanding the constraint (7) into exponentially many constraints. They then use the multiplicative weights update framework to solve this program in pseudo polynomial time. The advantage of this approach is that it is relatively easy to incorporate variations such as ex post budget constraints and risk aversion for the buyers and the seller. The disadvantage (other than the running time) is that it does not generalize beyond linear value functions and objectives.

4.3 Multi-parameter agents with a single-parameter feasibility constraint

Alaei et al. (2013) consider settings where agents have multi-parameter preferences, i.e. they have different values for different kinds of items or services, but the externality they impose on other agents depends only on the probability with which they get served at all. As a representative example, consider a buyer buying a car at a dealership. The dealership offers to paint the car in one of a variety of different colors. The buyer likes some colors more than others, and as such the dealership can charge different amounts for cars of different colors. However, in terms of how this affects the inventory of cars at the dealership, all that matters is whether or not the buyer buys the car, but not which color he ends up selecting. In other words, the feasibility constraint that the dealership faces depends on the total number of cars they sell, but not the total number for each color. In this example, the buyer has multi-parameter preferences but the feasibility constraint essentially treats him like a single-parameter agent. Alaei et al. show that for such a setting many of the salient features of Myerson (1981)'s approach for single-parameter settings continue to hold. In particular, it is possible to define a revenue curve for each agent specifying the maximum revenue that can be obtained from the agent given an ex ante probability of sale[23]. Then, the optimal mechanism turns out to be a marginal revenue maximizer as in Myerson's setting.

5. BLACK-BOX REDUCTIONS FROM MECHANISM TO ALGORITHM DESIGN

We now turn to a central theme in algorithmic mechanism design, namely whether incentives make mechanism design harder than algorithm design and to what extent. In the best case scenario, we would be able to "add on" incentive constraints to algorithms in an oblivious manner – without understanding or modifying the underlying algorithm – and without much loss in performance., From a theoretical viewpoint this would imply that truthfulness is cheap in a computational sense. From a practical viewpoint, this would allow us to design a mechanism as a layer around a previously optimized algorithm. Imagine, for instance, that a practitioner designs an algorithm for a certain setting and optimizes it to over time to suit his specific application. If this practitioner later wants to port his algorithm to a strategic setting, having a generic tool to convert algorithms into incentive compatible mechanisms allows him to avoid re-doing all of the design and optimization work again. In fact, the generic tool would also allow a different practitioner to convert

[23]In this respect, this approach has similarities with Alaei (2014)'s approach for unit-demand agents discussed previously.

the algorithm into a mechanism without having to understand all of the implicit constraints that the algorithm satisfies. Can such a generic tool exist? Although it seems too good to be true, we will show that it is indeed possible in certain Bayesian settings[24].

Before we elaborate, let us formalize what we mean by an oblivious or *black-box* reduction from mechanism design to algorithm design. As an example, suppose the given algorithm's output satisfies monotonicity (Theorem 2.1 and its multi-parameter generalization in Rochet (1987)). Then to convert it into a BIC mechanism we simply need to use the payment identity to define BIC payments. Note that the payment identity needs to know the algorithm's output at various different inputs (type vectors), but it does not need to understand how the algorithm generates the output or what constraints it satisfies. It can generate the right payments[25] to accompany the algorithm's output by simply querying the algorithm at multiple inputs. This is what we mean by a black-box reduction.

Using this example as a guide, we will say that a *strong black-box reduction* from mechanism design to algorithm design for a given setting is a transformation procedure that uses black-box access to an algorithm for the *same* setting, and produces an output that is incentive compatible, with performance that is comparable to that of the given algorithm. There are several points worth noting about this definition: (1) We have intentionally left the definitions of incentive compatibility and performance ambiguous; We will strive to obtain the strongest possible guarantees in both cases. (2) We want the transformation to be "generic" in the sense that it does not need to understand the constraints that the underlying algorithm satisfies; It will satisfy these constraints automatically by returning an output it has seen the algorithm generate (at a potentially different input). (3) This is a "strong" reduction in the sense that it reduces the mechanism design problem in a certain domain to the algorithm design problem in the *same* domain. The second and third points, in particular, address the practical motivation for a generic tool described at the beginning of this section.

Taking cue from the last point, we define a *weak black-box reduction* from mechanism design to algorithm design as a reduction that employs black-box access to an algorithm for (potentially) a different objective and under an additional set of constraints. The objective and new constraints for the underlying algorithm typically would make this problem harder than the non-strategic version of the original problem. The reduction can nevertheless be quite fruitful: the new algorithmic problem may be much simpler than one with a complete set of incentive constraints; as in the case of strong black-box reductions, here the transformation procedure does

[24]There is an extensive literature on the separation between algorithm design and mechanism design in non-Bayesian settings. See Dobzinski and Vondrak (2012) and references there in for the welfare objective, and Nisan and Ronen (2001) and Ashlagi et al. (2009) for the makespan objective. See Dughmi and Roughgarden (2010) and Huang et al. (2011) for black-box reductions for welfare in certain non-Bayesian settings.

[25]We are ignoring the issue that the payment identity requires computing an integral. From a computational viewpoint, this would necessitate discretizing the type space and we may not obtain an exact answer, although a really good approximation would suffice to give an ϵ-BIC mechanism for some small $\epsilon > 0$ (see, e.g., Hartline and Lucier (2010), and the definition of ϵ-BIC in Section 2).

not need to understand or even know the underlying feasibility constraints that the given algorithm handles. While this approach does not quite result in the sort of generic tool described at the beginning of this section, it gives us an approach for using the vast theory of algorithm design "as-is" in designing mechanisms. From a theoretical viewpoint it also lets us relate and therefore understand the approximability of a mechanism design problem in terms of the approximability of a related algorithm design problem.

Hartline and Lucier (2010) first showed that strong black-box reductions are possible in Bayesian settings. In particular, they considered the social welfare maximization objective for single-parameter agents and designed a transformation that is BIC and suffers a small additive loss in expected social welfare relative to the algorithm.Hartline, Kleinberg, and Malekian (2011) and Bei and Huang (2011) extended this result to social welfare maximization in an arbitrary multi-parameter setting with samplable types. Chawla et al. (2012) showed that these approaches for social welfare do not extend to non-linear objectives. In particular, for the objective of minimizing makespan (which is similar to the load balancing objective; see Section 2) no strong black-box reduction can preserve the average-case performance of the underlying algorithm to within a subpolynomial factor (in the number of agents). Fortunately, weak reductions provide an avenue for obtaining positive results for nonlinear objectives. Cai et al. (2013b) provide one such approach that is an extension of their work on revenue maximization described in Section 4.

In the remainder of this section we elaborate on the results and techniques of Hartline et al. (2011), Chawla et al. (2012), and Cai et al. (2013b).

5.1 Black-box reductions for social welfare

We now focus on the social welfare maximization problem and describe the approach of Hartline and Lucier (2010), Bei and Huang (2011) and Hartline et al. (2011) at a high level. Recall that we are given an algorithm $\mathcal{A}$ that maps type vectors of agents to feasible allocations for the agents. We are also given access to the distribution on agents' types; in particular, we can obtain samples from this distribution. Our goal is to design a transformation procedure that queries the algorithm and the distribution, and produces a mechanism mapping type vectors to feasible allocations and payments; The mechanism should be ϵ-BIC, and obtain expected social welfare close to that of the algorithm $\mathcal{A}$. We will focus on the problem of generating a monotone allocation function and ignore the supporting payments. We will ensure the feasibility of the allocation function produced by the transformation by requiring that at any input type vector $\mathbf{t}$ the transformation returns an allocation that it has seen the algorithm output at some (potentially different) type vector $\mathbf{s}$. In doing so, the transformation will not need to understand the underlying feasibility constraint at all, and will satisfy it trivially. As such we will interpret the behavior of the transformation as mapping input type vectors $\mathbf{t}$ to other (potentially random) type vectors $\mathbf{s}$; We call the latter the *surrogate* types.

Let us now focus on this mapping from types to surrogate types. Call this mapping σ. Then, at a type vector $\mathbf{t}$, the mechanism returned by the transformation will generate the allocation $\mathbf{x}(\mathbf{t}) = \mathcal{A}(\sigma(\mathbf{t}))$. Formally, we want to find a (potentially random) mapping σ, such that the function $\mathcal{A}(\sigma(\mathbf{t}))$ satisfies the monotonicity condition from Theorem 2.1 or its generalization from Rochet (1987), as applicable. Furthermore, the expected social welfare of the new allocation function $\mathcal{A}(\sigma(\mathbf{t}))$ should be close to or better than that of the original allocation function $\mathcal{A}(\mathbf{t})$. Finally, finding such a mapping σ should be computationally efficient.

The key to computational efficiency is that we will construct σ from agent-specific mappings σ_i, that is, we will find a mapping from types to surrogate types for each agent individually based solely on the interim allocations that the algorithm makes to these agents, and ignoring the actual types of other agents. Two things enable this: (1) in order to obtain BIC, it suffices to look at an agent's interim allocation, rather than their ex post allocation; (2) the social welfare objective is separable across agents, so that if the individual contribution of every agent to the expected social welfare is preserved by the transformation, then the overall performance is also preserved. There is one potential challenge, however. We will construct a mapping σ_i from types to types for an agent i by looking at his interim allocation function, assuming that all other agents' types stay the same. However, when we likewise apply the mapping for other agents' types, this could alter agent i's interim allocation, potentially undoing our work in improving i's allocation. In order to ensure that this does not happen, we need one other property from the mapping σ_i: for every agent i, the distribution of types $\sigma_i(t_i)$ should be identical to the original distribution over types t_i. This would imply that the mapping for any agent preserves the interim allocation function of any other agent.

Let us now restate our new goal for an individual agent i. Let $\mathcal{A}_i(t_i)$ denote the interim allocation that the algorithm makes to agent i at type t_i when other agents' types are distributed according to their underlying distribution. We want to construct a mapping σ_i from i's type space to itself with the following properties: (1) $\mathcal{A}_i(\sigma_i(t_i))$ satisfies the monotonicity condition from Theorem 2.1 or its generalization from Rochet (1987), as applicable; (2) The expected value that agent i obtains from the new interim allocation $\mathcal{A}_i(\sigma_i(t_i))$ is (nearly) as large as the expected value that the agent obtains from the old interim allocation $\mathcal{A}_i(t_i)$; (3) The mapping preserves the agent's type distribution, that is, $\sigma_i(t_i)$ is distributed identically to t_i. Because the first two properties are properties of the interim allocation, $\mathcal{A}_i(\sigma_i(t_i))$, resulting from the mapping, we can equivalently think of σ as mapping types to interim allocations (that we will simply call the algorithm's output), while preserving the distribution of the output (i.e. property (3)).

Why does such a mapping exist? In short, the answer is that social welfare maximization is highly compatible with monotonicity. To understand this, let us consider the simple case of a single-parameter agent with a uniform distribution over k different values $v_1 \geq v_2 \geq \cdots \geq v_k$. Let $a_j = \mathcal{A}(v_j)$ denote the algorithm's interim allocation to the agent at value v_j. Suppose that this interim allocation function does not satisfy weak monotonicity, that is, there are indices $j_1 < j_2$ with $a_{j_1} < a_{j_2}$. We want to find a mapping from values to outputs satisfying the three properties above. Since the distribution over types and outputs is the uniform distribution by assumption, the mapping is going to be a matching between the v_js and the a_js. Now let us consider the first two properties of the mapping separately. First, which matching between the v_js and the a_js achieves monotonicity? The answer is: the matching that matches v_j to the jth highest allocation in the set $\{a_1, \cdots, a_k\}$.

Second, which matching between the v_js and the a_js maximizes expected social welfare? Note that the expected social welfare of the agent is $1/k$ times the total weight of the matching where the weight of a single matched pair $(v_j, a_{j'})$ is the product $v_j a_{j'}$. So the answer is again: the matching that matches v_j to the jth highest allocation in the set $\{a_1, \cdots, a_k\}$. The two answers are identical. In other words, monotonizing an interim allocation rule automatically (weakly) improves the social welfare. The reader can convince herself that this property continues to hold in any single-parameter setting, and not just for the uniform distribution over a discrete type space. In fact it turns out that this property holds also in arbitrary multi-parameter settings because the multi-parameter monotonicity condition of Rochet (1987) can be expressed in terms of maximum weight matchings. This interpretation of the mapping for an agent in terms of maximum weight matchings also gives us a computationally efficient procedure for finding the mapping.

Our discussion ignores many crucial details: the type space of each agent may be prohibitively large, or even unbounded; computing the interim allocations of agents and the corresponding expected values may be challenging. These can be overcome via sampling and by allowing the final mechanism (in some cases) to satisfy ϵ-BIC rather than exact BIC.

5.2 Strong black-box reductions for nonlinear objectives

Can the approach of Hartline and Lucier be applied to objectives other than social welfare? In other words, is there a mapping σ from type vectors to type vectors such that $\mathcal{A}(\sigma(\mathbf{t}))$ satisfies monotonicity and preserves the algorithm's performance in expectation with respect to some nonlinear objective? At a minimum, in order to be able to construct the mapping on a per-agent basis, the objective function would have to be additively separable across agents. Suppose that is indeed the case, can we solve the per-agent mapping problem as described in the previous section but with respect to some arbitrary nonlinear objective? Chawla et al. (2012) show that this is not possible: for every nonlinear objective, there is an example with one single-parameter agent where any mapping from the agent's type space to itself that satisfies monotonicity and preserves the distribution over types suffers an unbounded loss in the objective. They further show that for certain nonlinear objectives, no approaches can lead to a strong black-box reduction with small loss in performance.

We first note that not all nonlinear objectives can be aligned with incentive compatibility. Consider, for example, the fairness objective in a single-parameter setting. We have n agents with values v_i and a single item that can be distributed fractionally among the agents. Our objective is to maximize the minimum value: $\min_i(v_i x_i)$ subject to $\sum_i x_i \leq 1$. Then it is easy to see that optimal or near optimal allocation rules are highly non-monotone: if an agent raises his value, the objective forces us to reduce his allocation in order to balance the distribution of welfare across all agents; Conversely, no monotone allocation rule can obtain a good approximation. This motivates the definition of *monotone objectives*: objectives for which exact optimization leads to a monotone allocation function. Social welfare is one example. For monotone objectives, the black-box reduction question is essentially about computational efficiency: if we could exactly optimize the objective, then we would have a monotone allocation rule, but we cannot necessarily do so in a computationally efficient manner.

Chawla et al.'s negative result concerns the makespan minimization problem. Informally, this is the min-max variant of the fairness objective. The setup is that of n selfish machines, with private speeds $\mathbf{v}$. The mechanism's goal is to allocate work to the machines, with $x_i(\mathbf{v})$ being the total amount of work allocated to machine i. The mechanism's makespan is then the time at which the last machine finishes its work: $\max_i(x_i(\mathbf{v})/v_i)$; we want to minimize this quantity. Note that machines get paid to do work, so this a "reverse auction" setting; the reader can check that BIC in this setting is equivalent to the interim allocation to each machine being monotone weakly increasing in the machine's speed or value v_i. Chawla et al. showed that there is no strong black-box reduction for this problem where the expected makespan of the mechanism is at most subpolynomially (in n) worse than that of the underlying algorithm. We will now describe the high-level intuition behind this result.

Recall that a black-box reduction is a transformation procedure that queries the algorithm at multiple inputs and then creates a new allocation function; this new allocation function should have the property that the output (allocation) at any type vector is an output that the transformation procedure obtained by querying the algorithm at some input. In the case of the makespan minimization problem, the transformation procedure upon querying the algorithm may determine that it needs to change the output at a certain type vector, for example, it may want to lower the allocation to some low speed machines or increase the allocation to some high speed machines. In doing so it must avoid with reasonable probability outputs that lead to a makespan much higher than that of the algorithm. Here is where the problem arises. Since the makespan objective is highly nonlinear (it is the maximum over n agent-specific makespans), there are many ways to generate a bad output with a high makespan, and few ways to generate a good one. The transformation procedure therefore faces an uphill task: it needs to find an input at which the algorithm produces an output with the desired changes to allocations while avoiding the many ways of increasing the makespan. In fact, there exists an example where in order to do so the transformation would need to make exponentially many queries to the algorithm.

5.3 Weak black-box reductions for nonlinear objectives

Cai, Daskalakis, and Weinberg (2013b) consider arbitrary concave objectives[26] in settings with small type spaces (as in Section 4.2) and provide a weak black-box reduction that runs in time polynomial in $\sum_i |T_i|$. Specifically, they reduce the mechanism design problem for an objective $\mathcal{O}$ in such settings to an algorithm design problem for a *different* objective $\mathcal{O}'$. Given access to such an algorithm, their transformation produces a mechanism that is ϵ-BIC, and obtains an approximation factor for $\mathcal{O}$ related to the algorithm's approximation factor for $\mathcal{O}'$. Here we define the approximation factor for the mechanism as well as the algorithm with respect to their objective function value in expectation over the type distribution.

[26]Concave objectives are not always monotone in the sense of the definition in Section 5.2; That is, exact optimization of these objectives does not always lead to monotonicity of the allocation function.

There is an important distinction between the weak reduction of Cai et al. for makespan minimization and the strong reduction of Hartline and Lucier (2010) for welfare maximization. In the latter work, the mechanism produced by the transformation achieves nearly the same absolute performance as the underlying algorithm. In the former work, the mechanism achieves nearly the same *approximation ratio* as the underlying algorithm, where the mechanism's approximation ratio is measured against the performance of the optimal BIC mechanism, while the algorithm's approximation ratio is measured against the performance of the optimal (non-strategic) algorithm. In particular, the mechanism's absolute performance may be far worse than that of the algorithm.

Cai et al.'s approach is similar to and inspired by their solution to the revenue maximization problem in settings with small type spaces that we described in Section 4.2. Recall that for the revenue maximization problem the idea is to set up a compact linear program over the agents' interim allocation and payment rules. While the BIC and IR constraints as well as the revenue objective can be expressed linearly over these interim quantities, the key challenge is to encode and enforce the constraint that the interim allocations are feasible in that they arise from (a distribution over) feasible ex post allocations (see (RF-LP) and constraint (7) in Section 4.2). For non-linear objectives another challenge is that the objective cannot necessarily be expressed as a function of the interim allocations and payments. We will now outline the changes that need to be made to this approach to make it work for arbitrary concave objectives.

Cai et al.'s main insight is to expand the reduced form of the mechanism (which previously included only the interim allocation rule) to include the expected objective function value that the mechanism achieves. Armed with this extra variable, call it o, we can once again express the BIC and IR constraints, and the objective function in the form of a compact linear program. It remains to verify that the reduced form $(\mathbf{x}, o)$ is feasible in that there is a distribution over feasible ex post allocations that implements this reduced form. As in the revenue setting, they show that as long as $\mathcal{O}$ is a concave objective, the set of all feasible reduced forms $(\mathbf{x}, o)$ is a convex polytope. Linear functions over this polytope now have a component corresponding to the objective function, in addition to a virtual welfare component. Optimizing over the polytope therefore is equivalent to optimizing the objective $\mathcal{O}' = \mathcal{O}$ plus virtual welfare. Given an algorithm for this generalized objective, they show how to adapt the machinery described in Section 4.2 to obtain an ϵ-BIC mechanism for the objective $\mathcal{O}$.

An important point is worth mentioning here. The new objective $\mathcal{O}' = (\mathcal{O} +$ virtual welfare) may be much harder to approximate than the original objective $\mathcal{O}$ (in a non-strategic setting). This is especially so because the virtual values corresponding to the welfare component can be positive or negative. In follow-up work, Daskalakis and Weinberg (2014) apply this framework to the makespan minimization problem and observe that the resulting algorithmic problem of minimizing the makespan plus virtual welfare is hard to approximate to within any finite factor. Nevertheless, they show that it is possible to leverage the convex optimization approach via a slightly different approximation. In particular, they show that if the algorithm produces a solution for the objective $\mathcal{O}'$ such that for some $\beta > 1$,

($\beta\mathcal{O}$ + virtual welfare) is an α-approximation to the maximum value for $\mathcal{O}'$, then the machinery of Cai et al. (2013b) can be used to obtain an $\alpha\beta$-approximation mechanism for $\mathcal{O}$. They instantiate this approach for the makespan minimization objective, obtaining an approximation factor of 2, which, remarkably, matches the best known approximation for the same objective in non-strategic settings.

6. FURTHER DIRECTIONS

We will now briefly survey some other directions of interest in BAMD. The references provided are not meant to be exhaustive; interested readers should use these as a starting point for a literature search into the topic.

Other objectives. We have previously discussed work on social welfare and revenue maximization. Tradeoffs among these objectives have also been studied: Daskalakis and Pierrakos (2011) study the design of mechanisms that simultaneously approximate both objectives; Hartline and Roughgarden (2008) consider a residual surplus objective with the goal of obtaining high social surplus with low monetary transfers. Another objective related to revenue is the so-called maximum payment objective that arises in the context of contest design and crowdsourcing (see, e.g., Chawla, Hartline, and Sivan (2012); Ghosh and Kleinberg (2014)). As mentioned earlier, in a series of works Cai et al. develop general techniques for designing approximation mechanisms for concave objectives. This bounds the approximability of these objectives in strategic settings. A related question is whether the performance of the optimal mechanism for these objectives is close to that of the optimal (non-strategic) algorithm. Chawla, Hartline, Malec, and Sivan (2013) study this question for the makespan objective and present mechanisms that approximate the non-strategic optimum. Finally, new applications in private data analysis have motivated the use of mechanisms for buying private information with the designer's objective being the accuracy of the estimate based on the bought information (see, e.g., Ghosh and Roth (2011)).

Correlated or interdependent agents. Throughout this article we have assumed that agents' types are distributed independently of each other. In the real world this is usually not true. Oftentimes agents' values are informed by a common source or derive from a resale market, as is the case with houses or art. When agents' values are correlated, the revenue maximization problem is computationally intractable even with just 3 agents (Papadimitriou and Pierrakos, 2011). However, good approximations can be obtained in many settings (Ronen, 2001; Ronen and Saberi, 2002; Dobzinski et al., 2011; Chen et al., 2011; Babaioff et al., 2012; Minooei and Swamy, 2013; Fu et al., 2014; Chawla et al., 2014).

A more interesting situation arises when the agents do not know their values precisely. Milgrom and Weber (1982) introduced the following model, which is now standard for this setting: Agents' information about their own value is captured by a *signal* that they receive from the environment. Agents' signals may be correlated. Further, their true values may depend not only on their own signal, but also on others' signals. Since signals are private, agents may not know their values precisely, and base their actions on the stochastic information they have about others' signals. During the course of a mechanism, agents may learn others' signals, update their own estimates of their values, and act according to these updated estimates. A classic example of this setting is an auction for the right to drill for oil in a certain location Wilson (1969). The value of an oil lease depends on how much oil there actually is, and the different bidders may have access to different assessments about this. Consequently, a bidder might change her own estimate of the value of the oil lease given access to the information another bidder has. An interesting consequence of this is the breakdown of the revelation principle: mechanisms that don't reveal the agents' signals to each other cannot necessarily implement the same outcome as those that reveal all of the information. Work in this model has consequently focused on ex post equilibria, rather than Bayes-Nash equilibria.

A series of papers (e.g., Chung and Ely, 2006; Vohra, 2011; Csapó and Müller, 2013; Roughgarden and Talgam-Cohen, 2013) characterize optimal ex post IC mechanisms, obtaining in some cases a "revenue equals virtual surplus" type of characterization. Syrgkanis et al. (2013) and Abraham et al. (2013) study the special case of common value settings, that is, where all of the agents have the same (unknown) value. Roughgarden and Talgam-Cohen (2013) and Li (2013) provide prior-independent approximations in certain agent-symmetric settings. Chawla, Fu, and Karlin (2014) provide approximately-optimal mechanisms in quite general settings: they show, in particular, that a simple variant of the VCG mechanism with reserve prices and random admission provides constant factor approximations in matroid environments with few assumptions.

Budgeted agents. The mechanism design problem changes considerably when agents have budgets. Budgets play a role distinct from the value an agent has for an item or service. For example, an agent may value a piece of art greatly but not have the means to pay for it. How a mechanism deals with budget constraints depends on how it handles individual rationality. For mechanisms that are interim individually rational, it suffices to satisfy the budget constraints also in an interim sense, because the agent's payment at a particular type can be redistributed as a function of others' types in order to satisfy the budget constraint ex post. Consequently, optimal budget-feasible interim-IR mechanisms are typically *all-pay* mechanisms, where an agent makes a payment based on his type to the mechanism regardless of whether or not (or what) he gets allocated. Bhattacharya et al. (2010) and Alaei (2014) present approximation mechanisms for this setting. The linear programming based framework of Cai et al. (2012b) can also easily handle budgets with interim-IR. Mechanisms that are ex post individually rational are far weaker and necessitate different design techniques. Chawla et al. (2011) present simple approximations in this context for succinct types, while Bhalgat et al. (2013) present near-optimal pseudo-polynomial time mechanisms for small type spaces.

Risk aversion. Most work in BAMD assumes that both the agents and the designer are risk neutral, that is, they care about their objective in expectation, and not about how it is distributed. On the other hand, individuals or entities in the real world are often risk averse and would, e.g., prefer a certain reward to a lottery that gives twice the reward with 50% probability, and no reward otherwise. Many of the mechanisms we design use the randomness in the agents' types as well as their own randomness to their advantage. These mechanisms would potentially perform quite poorly if one or more of the participants is risk averse. Fu et al. (2013) develop a theory of mechanism design with risk-averse agents. Sundararajan and Yan (2010) and Bhalgat et al. (2012) consider the mechanism design problem faced by a risk-averse seller.

Hardness and inapproximability results. Hardness results for algorithm design in a particular setting immediately carry over to mechanism design in the same setting, as mechanisms face extra constraints. Can mechanisms be hard to approximate in settings where the algorithmic problem is easy? This is indeed the case for revenue maximization in multi-parameter settings. Briest (2008) shows that computing revenue optimal prices for a single-agent unit-demand setting with m items is hard to approximate within $O(m^\epsilon)$ when the agent's values for different items are correlated and the distribution is explicitly specified. When the distribution is samplable but not described explicitly, Briest shows hardness of approximation within $\log(|T|^\epsilon)$ where $|T|$ is the size of the type space. Daskalakis et al. (2012) show that this problem remains hard when the agent's values for different items are independent, and the distributions are succinctly described. Chen et al. (2014) settle the complexity of this setting by proving that it is NP-hard to compute the revenue optimal pricing even when the values are independent and every value distribution has a support of size 3.

Moving to additive settings, Daskalakis et al. (2014) show that computing the optimal randomized auction is hard in the single-agent m-item setting with independently distributed values, even when each item has just two possible values. Papadimitriou and Pierrakos (2011) show that the optimal auction is NP-hard to compute or approximate in single item auctions with 3 agents and correlated distributions. Cole and Roughgarden (2014) study the sample complexity of computing a $(1-\epsilon)$ approximation for the revenue optimal single item auction and show that poly $(n, 1/\epsilon)$ samples are necessary and sufficient for MHR distributions; The sample complexity degrades as we move away from MHR distributions to regular distributions. The necessity of a polynomial dependence on n is remarkable considering that there is no such dependence when the agents' valuations are i.i.d. regular (Dhangwatnotai et al., 2010), or when only a constant factor approximation is sought (Hartline and Roughgarden, 2009).

Prior independence. A criticism of the Bayesian assumption in mechanism design is that it places a large informational burden on the designer. This designer must somehow obtain an accurate Bayesian prior in order to carry out the optimization. Dhangwatnotai et al. (2010) observed that in many settings the Bayesian prior is not needed to obtain a good approximation, although performance is still measured in expectation with respect to the underlying (unknown) prior. Further techniques in this direction have been developed by Devanur et al. (2011), Roughgarden et al. (2012), and Sivan and Syrgkanis (2013).

Information as a tool. In some settings, the seller or designer may possess more information than the agents, and can wield this knowledge as a tool to further his objective. Miltersen and Sheffet (2012); Emek et al. (2012); Dughmi et al. (2014) develop approaches for sharing information selectively with the agents so as to maximize social welfare or revenue.

Non-truthful mechanisms. We have focused exclusively on direct revelation incentive compatible mechanisms. There is an extensive literature on optimization via non-truthful mechanisms, that studies the performance of these mechanisms in Bayes-Nash equilibria (via the Bayesian prices of anarchy and stability). We refer the reader to Syrgkanis and Tardos (2013) for a brief survey of this literature.

Dynamic auctions. Finally, many of the applications of BAMD are in dynamic online settings where auctions are run repeatedly, and agents arrive and leave over time. The buyer typically has a deadline beyond which he is not interested in buying, and the seller has a fixed supply of items which expires over time. This includes, for example, ad auctions, airline ticket sales, hotel accommodation sales, task scheduling in a cloud computing environment, etc. For a concrete example, consider a seller with k identical items available over T time periods. Each buyer is interested in one item and either wants it in the same time period he arrives in, or can wait till the end of the process. If the seller knows the joint distribution over the agents' values and deadlines, what selling mechanism should he use to maximize revenue? Deb and Pai (2013) analyze this simple setting and characterize revenue optimal mechanisms. Several aspects of dynamic settings make them especially challenging: both the buyers and the seller can strategize by reasoning about future events; some of the seller's decisions must be made online, that is without full knowledge of what is to come; buyers may enter or leave as they please; buyers may lie about their deadlines. We refer the reader to Parkes and Duong (2007); Constantin and Parkes (2009); Parkes (2009); Deb and Pai (2013) for a preliminary treatment of this subject.

7. A FEW OPEN PROBLEMS

Below we list a few major open directions in BAMD.

Revenue maximization with succinct type spaces. Designing approximately revenue optimal mechanisms for large but succinctly representable type spaces is an area of active research with several open questions. As discussed in Section 4.1, the "easy" cases are the extreme cases where the agent valuations are unit-demand or additive. Even here, for additive valuations, extending the result of Babaioff et al. (2014) to multiple agents is open. Moving beyond the two extremes, e.g., to additive valuations up to k items is also open, even in a single-agent setting. On a different dimension, moving beyond independence across items for large type spaces has received little attention so far. Coming up with natural models of correlation that can be succinctly represented is the first step here. However, we can't proceed too far in this direction: Cai et al. (2013b) show that unless NP=RP, even with a single agent and submodular valuations, the optimal revenue cannot be approximated to any polynomial factor in time polynomial in the size of type spaces.

Revenue maximization with oracle access to valuations. A challenge for mechanism design in multi-parameter settings is efficiently extracting information from the agents about their (exponentially large) value functions. As we discussed in Section 4, BAMD literature has so far focused on settings with succinct descriptions of value functions or small type spaces. An alternate approach is to have oracle access to the value function via, e.g., value or demand queries; Several positive results are known for prior-free welfare maximization in such models (see, e.g., Blumrosen and Nisan (2009)). Are there similar models for the Bayesian setting that enable positive results for revenue maximization?

Simple mechanisms for settings with explicit type spaces. Even when type spaces are explicitly given (i.e., the small type spaces setting), developing mechanisms that are simple to understand and implement is still open for several objectives. The framework of Cai, Daskalakis, and Weinberg (2012a,b, 2013a,b) provides a generic solution for arbitrary concave objectives, however, this framework is quite complex and may result in algorithmic problems that are hard to solve. Specific objectives may permit simpler and faster solutions.

Interdependent valuations. As mentioned earlier, most work on settings involving interdependence among agents' types deals with ex post IC mechanisms. BIC mechanisms can in some settings perform much better than EPIC mechanisms (see, e.g., Cremer and McLean (1988); McAfee and Reny (1992)). Can we approximate over this class of mechanisms? Approximation over ex post IC mechanisms is also open for settings beyond matroid constraints for both the social welfare and revenue objectives. Interdependent or correlated settings with multi-parameter agents have not been explored much. While Jehiel et al. (2006) show strong impossibility results for ex-post implementation in generic multi-parameter settings, reasonable restrictions on value functions may give room for positive results.

Dynamic mechanism design. As mentioned earlier, many of the applications of BAMD are in dynamic or online settings. Characterizing optimal mechanisms in simple dynamic auction settings is a wide open area for research.

Acknowledgements

We are grateful to Anna Karlin and Seeun Umboh for comments on an earlier draft of this article.

REFERENCES

Abraham, I., Athey, S., Babaioff, M., and Grubb, M. 2013. Peaches, lemons, and cookies: Designing auction markets with dispersed information. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*. EC '13. ACM, New York, NY, USA, 7–8.

Alaei, S. 2014. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing 43,* 2, 930–972.

Alaei, S., Fu, H., Haghpanah, N., and Hartline, J. D. 2013. The simple economics of approximately optimal auctions. In *IEEE Symp. on Foundations of Computer Science*. 628–637.

Alaei, S., Fu, H., Haghpanah, N., Hartline, J. D., and Malekian, A. 2012. Bayesian optimal auctions via multi- to single-agent reduction. In *ACM Conf. on Electronic Commerce*. 17.

Ashlagi, I., Dobzinski, S., and Lavi, R. 2009. An optimal lower bound for anonymous scheduling mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce*. EC '09. 169–176.

Azar, P. D., Daskalakis, C., Micali, S., and Weinberg, S. M. 2013. Optimal and efficient parametric auctions. In *ACM Symp. on Discrete Algorithms*. 596–604.

Azar, P. D., Kleinberg, R., and Weinberg, S. M. 2014. Prophet inequalities with limited information. In *ACM Symp. on Discrete Algorithms*. 1358–1377.

Babaioff, M., Immorlica, N., Lucier, B., and Weinberg, S. M. 2014. A simple and approximately optimal mechanism for an additive buyer. arXiv:1405.6146.

Babaioff, M., Kleinberg, R., and Leme, R. P. 2012. Optimal mechanisms for selling information. In *ACM Conf. on Electronic Commerce*. 92–109.

Bei, X. and Huang, Z. 2011. Bayesian incentive compatibility via fractional assignments. In *ACM Symp. on Discrete Algorithms*. 720–733.

Bhalgat, A., Chakraborty, T., and Khanna, S. 2012. Mechanism design for a risk averse seller. In *Workshop on Internet and Network Economics (WINE)*. 198–211.

Bhalgat, A., Gollapudi, S., and Munagala, K. 2013. Optimal auctions via the multiplicative weight method. In *ACM Conf. on Electronic Commerce*. 73–90.

Bhattacharya, S., Goel, G., Gollapudi, S., and Munagala, K. 2010. Budget constrained auctions with heterogeneous items. In *ACM Symp. on Theory of Computing*. 379–388.

Blumrosen, L. and Nisan, N. 2009. On the computational power of demand queries. *SIAM J. Comput. 39,* 4, 1372–1391.

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, 1175–87.

Briest, P. 2008. Uniform budgets and the envy-free pricing problem. In *ICALP*. 808–819.

Briest, P., Chawla, S., Kleinberg, R., and Weinberg, S. M. 2014. Pricing randomized allocations. *Journal of Economic Theory*. To appear.

Brualdi, R. A. 1969. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society 1*, 161–167.

Bulow, J. and Klemperer, P. 1996. Auctions vs negotiations. *American Economic Review 86,* 1, 180–194.

Bulow, J. and Roberts, J. 1989. The simple economics of optimal auctions. *The Journal of Political Economy 97*, 1060–90.

Cai, Y. and Daskalakis, C. 2011. Extreme-value theorems for optimal multidimensional pricing. In *IEEE Symp. on Foundations of Computer Science*. 522–531.

Cai, Y., Daskalakis, C., and Weinberg, S. M. 2012a. An algorithmic characterization of multi-dimensional mechanisms. In *ACM Symp. on Theory of Computing*. 459–478.

CAI, Y., DASKALAKIS, C., AND WEINBERG, S. M. 2012b. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *IEEE Symp. on Foundations of Computer Science*. 130–139.

CAI, Y., DASKALAKIS, C., AND WEINBERG, S. M. 2013a. Reducing revenue to welfare maximization: Approximation algorithms and other generalizations. In *ACM Symp. on Discrete Algorithms*. 578–595.

CAI, Y., DASKALAKIS, C., AND WEINBERG, S. M. 2013b. Understanding incentives: Mechanism design becomes algorithm design. In *IEEE Symp. on Foundations of Computer Science*. 618–627.

CAI, Y. AND HUANG, Z. 2013. Simple and nearly optimal multi-item auctions. In *ACM Symp. on Discrete Algorithms*. 564–577.

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*. 158–169.

CHAWLA, S., FU, H., AND KARLIN, A. 2014. Approximate revenue maximization in interdependent value settings. In *ACM Conf. on Electronic Commerce*. To appear.

CHAWLA, S., HARTLINE, J. D., AND KLEINBERG, R. D. 2007. Algorithmic pricing via virtual valuations. In *ACM Conf. on Electronic Commerce*. 243–251.

CHAWLA, S., HARTLINE, J. D., MALEC, D. L., AND SIVAN, B. 2010. Multi-parameter mechanism design and sequential posted pricing. In *ACM Symp. on Theory of Computing*. 311–320.

CHAWLA, S., HARTLINE, J. D., MALEC, D. L., AND SIVAN, B. 2013. Prior-independent mechanisms for scheduling. In *ACM Symp. on Theory of Computing*. 51–60.

CHAWLA, S., HARTLINE, J. D., AND SIVAN, B. 2012. Optimal crowdsourcing contests. In *ACM Symp. on Discrete Algorithms*. 856–868.

CHAWLA, S., IMMORLICA, N., AND LUCIER, B. 2012. On the limits of black-box reductions in mechanism design. In *ACM Symp. on Theory of Computing*. 435–448.

CHAWLA, S., MALEC, D. L., AND MALEKIAN, A. 2011. Bayesian mechanism design for budget-constrained agents. In *ACM Conf. on Electronic Commerce*. 253–262.

CHAWLA, S., MALEC, D. L., AND SIVAN, B. 2012. The power of randomness in bayesian optimal mechanism design. *Games and Economic Behavior*. http://dx.doi.org/10.1016/j.geb.2012.08.010.

CHE, Y.-K., KIM, J., AND MIERENDORFF, K. 2013. Generalized reduced-form auctions: A network-flow approach. *Econometrica 81,* 6, 2487–2520.

CHEN, X., DIAKONIKOLAS, I., PAPARAS, D., SUN, X., AND YANNAKAKIS, M. 2014. The complexity of optimal multidimensional pricing. In *ACM Symp. on Discrete Algorithms*. 1319–1328.

CHEN, X., HU, G., LU, P., AND WANG, L. 2011. On the approximation ratio of k-lookahead auction. In *Workshop on Internet and Network Economics (WINE)*. 61–71.

Chung, K. and Ely, J. C. 2006. Ex-post incentive compatible mechanism design. Manuscript.

Cole, R. and Roughgarden, T. 2014. The sample complexity of revenue maximization. In *ACM Symp. on Theory of Computing*. To appear.

Constantin, F. and Parkes, D. C. 2009. Self-correcting sampling-based dynamic multi-unit auctions. In *ACM Conf. on Electronic Commerce*. 89–98.

Cremer, J. and McLean, R. P. 1988. Full Extraction of the Surplus in Bayesian and Dominant Strategy Auctions. *Econometrica 56,* 6 (November), 1247–57.

Csapó, G. and Müller, R. 2013. Optimal mechanism design for the private supply of a public good. *Games and Economic Behavior 80,* C, 229–242.

Daskalakis, C., Deckelbaum, A., and Tzamos, C. 2012. Optimal pricing is hard. In *Workshop on Internet and Network Economics (WINE)*. 298–308.

Daskalakis, C., Deckelbaum, A., and Tzamos, C. 2013. Mechanism design via optimal transport. In *ACM Conf. on Electronic Commerce*. 269–286.

Daskalakis, C., Deckelbaum, A., and Tzamos, C. 2014. The complexity of optimal mechanism design. In *ACM Symp. on Discrete Algorithms*. 1302–1318.

Daskalakis, C. and Pierrakos, G. 2011. Simple, optimal and efficient auctions. In *Workshop on Internet and Network Economics (WINE)*. 109–121.

Daskalakis, C. and Weinberg, S. M. 2014. Bayesian truthful mechanisms for job scheduling from bi-criterion approximation algorithms. arXiv:1405.5940.

Deb, R. and Pai, M. 2013. Ironing in dynamic revenue management: Posted prices & biased auctions. In *ACM Symp. on Discrete Algorithms*. 620–631.

Devanur, N. R., Hartline, J. D., Karlin, A. R., and Nguyen, C. T. 2011. Prior-independent multi-parameter mechanism design. In *Workshop on Internet and Network Economics (WINE)*. 122–133.

Dhangwatnotai, P., Roughgarden, T., and Yan, Q. 2010. Revenue maximization with a single sample. In *ACM Conf. on Electronic Commerce*. 129–138.

Dobzinski, S., Fu, H., and Kleinberg, R. D. 2011. Optimal auctions with correlated bidders are easy. In *ACM Symp. on Theory of Computing*. 129–138.

Dobzinski, S. and Vondrak, J. 2012. The computational complexity of truthfulness in combinatorial auctions. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. EC '12. ACM, New York, NY, USA, 405–422.

Dughmi, S., Immorlica, N., and Roth, A. 2014. Constrained signaling in auction design. In *ACM Symp. on Discrete Algorithms*. 1341–1357.

Dughmi, S. and Roughgarden, T. 2010. Black-box randomized reductions in algorithmic mechanism design. In *FOCS*. 775–784.

Emek, Y., Feldman, M., Gamzu, I., Leme, R. P., and Tennenholtz, M. 2012. Signaling schemes for revenue maximization. In *ACM Conf. on Electronic Commerce*. 514–531.

Fu, H., Haghpanah, N., Hartline, J., and Kleinberg, R. 2014. Optimal auctions for correlated bidders with sampling. In *ACM Conf. on Electronic Commerce*. To appear.

Fu, H., Hartline, J. D., and Hoy, D. 2013. Prior-independent auctions for risk-averse agents. In *ACM Conf. on Electronic Commerce*. 471–488.

Ghosh, A. and Kleinberg, R. 2014. Optimal contest design for simple agents. In *ACM Conf. on Electronic Commerce*. To appear.

Ghosh, A. and Roth, A. 2011. Selling privacy at auction. In *Proceedings of the 12th ACM Conference on Electronic Commerce*. EC '11. ACM, New York, NY, USA, 199–208.

Hart, S. and Nisan, N. 2012. Approximate revenue maximization with multiple items. In *ACM Conf. on Electronic Commerce*. 656.

Hart, S. and Nisan, N. 2013. The menu-size complexity of auctions. In *ACM Conf. on Electronic Commerce*. 565–566.

Hartline, J. D. 2013a. Bayesian mechanism design. *Foundations and Trends in Theoretical Computer Science 8,* 3, 143–263.

Hartline, J. D. 2013b. Mechanism design and approximation. `http://jasonhartline.com/MDnA/`.

Hartline, J. D., Kleinberg, R., and Malekian, A. 2011. Bayesian incentive compatibility via matchings. In *ACM Symp. on Discrete Algorithms*. 734–747.

Hartline, J. D. and Lucier, B. 2010. Bayesian algorithmic mechanism design. In *ACM Symp. on Theory of Computing*. 301–310.

Hartline, J. D. and Roughgarden, T. 2008. Optimal mechanism design and money burning. In *ACM Symp. on Theory of Computing*. 75–84.

Hartline, J. D. and Roughgarden, T. 2009. Simple versus optimal mechanisms. In *ACM Conf. on Electronic Commerce*. 225–234.

Huang, Z., Wang, L., and Zhou, Y. 2011. Black-box reductions in mechanism design. In *Intl. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 254–265.

Jehiel, P., Vehn, M. M.-t., Moldovanu, B., and Zame, W. R. 2006. The limits of ex post implementation. *Econometrica 74,* 3 (May), 585–610.

Kleinberg, R. and Weinberg, S. M. 2012. Matroid prophet inequalities. In *ACM Symp. on Theory of Computing*. 123–136.

Krishna, V. 2009. *Auction theory*. Academic press.

Li, X. and Yao, A. C.-C. 2013. On revenue maximization for selling multiple independently distributed items. *Proceedings of the National Academy of Sciences*.

Li, Y. 2013. Approximation in mechanism design with interdependent values. In *ACM Conf. on Electronic Commerce*. 675–676.

Manelli, A. and Vincent, D. 2007. Multidimensional mechanism design: Revenue maximization and the multiple-good monopoly. *Journal of Economic Theory 137,* 1, 153–185.

Manelli, A. M. and Vincent, D. R. 2006. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory 127,* 1 (March), 1–35.

McAfee, R. P. and Reny, P. J. 1992. Correlated Information and Mechanism Design. *Econometrica 60,* 2 (March), 395–421.

Milgrom, P. R. and Weber, R. J. 1982. A Theory of Auctions and Competitive Bidding. *Econometrica 50,* 5 (September), 1089–1122.

Miltersen, P. B. and Sheffet, O. 2012. Send mixed signals: earn more, work less. In *ACM Conf. on Electronic Commerce*. 234–247.

Minooei, H. and Swamy, C. 2013. Near-optimal and robust mechanism design for covering problems with correlated players. In *Conference on Web and Internet Economics (WINE)*. 377–390.

Myerson, R. 1981. Optimal auction design. *Mathematics of Operations Research 6*, 58–73.

Nisan, N. 2014. Algorithmic mechanism design (through the lens of multi-unit auctions). To appear in the Handbook of Game Theory, IV.

Nisan, N. and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic Behavior 35,* 1-2, 166–196.

Papadimitriou, C. H. and Pierrakos, G. 2011. On optimal single-item auctions. In *ACM Symp. on Theory of Computing*. 119–128.

Parkes, D. C. 2009. When analysis fails: Heuristic mechanism design via self-correcting procedures. In *SOFSEM*. 62–66.

Parkes, D. C. and Duong, Q. 2007. An ironing-based approach to adaptive online mechanism design in single-valued domains. In *AAAI*. 94–101.

Pavlov, G. 2011a. Optimal mechanism for selling two goods. *The B.E. Journal of Theoretical Economics 11,* 1.

Pavlov, G. 2011b. A property of solutions to linear monopoly problems. *The B.E. Journal of Theoretical Economics 11,* 4.

Riley, J. and Zeckhauser, R. 1983. Optimal Selling Strategies: When to Haggle, When to Hold Firm. *The Quarterly Journal of Economics 98,* 2 (May), 267–89.

Rochet, J.-C. 1987. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics 16,* 2 (April), 191–200.

Rochet, J.-C. and Chone, P. 1998. Ironing, Sweeping, and Multidimensional Screening. *Econometrica 66,* 4, 783–826.

Ronen, A. 2001. On approximating optimal auctions. In *ACM Conf. on Electronic Commerce*. 11–17.

Ronen, A. and Saberi, A. 2002. On the hardness of optimal auctions. In *IEEE Symp. on Foundations of Computer Science*. 396–405.

Roughgarden, T. and Talgam-Cohen, I. 2013. Optimal and near-optimal mechanism design with interdependent values. In *ACM Conf. on Electronic Commerce*. 767–784.

Roughgarden, T., Talgam-Cohen, I., and Yan, Q. 2012. Supply-limiting mechanisms. In *ACM Conf. on Electronic Commerce*. 844–861.

Samuel-Cahn, E. 1984. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *The Annals of Probability 12,* 4, 1213–1216.

Sivan, B. and Syrgkanis, V. 2013. Vickrey auctions for irregular distributions. In *Conference on Web and Internet Economics (WINE)*. 422–435.

Sundararajan, M. and Yan, Q. 2010. Robust mechanisms for risk-averse sellers. In *ACM Conf. on Electronic Commerce*. 139–148.

SYRGKANIS, V., KEMPE, D., AND TARDOS, E. 2013. Information asymmetries in common-value auctions with discrete signals. In *SSRN*.

SYRGKANIS, V. AND TARDOS, E. 2013. Composable and efficient mechanisms. In *ACM Symp. on Theory of Computing*. 211–220.

THANASSOULIS, J. E. 2004. Haggling Over Substitutes. *Journal of Economic Theory 117,* 2, 217–245.

VOHRA, R. 2011. *Mechanism design: A linear programming approach (Econometric society monographs).* Cambridge University Press.

WANG, Z. AND TANG, P. 2014. Optimal mechanisms with simple menus. In *ACM Conf. on Electronic Commerce*. To appear.

WILSON, R. 1969. Competitive bidding with disparate information. *Management Sciences 15*.

YAN, Q. 2011. Mechanism design via correlation gap. In *ACM Symp. on Discrete Algorithms*. 710–719.