2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

Optimal Multi-Dimensional Mechanism Design: Reducing Revenue to Welfare Maximization

Yang Cai[1]
EECS, MIT
ycai@csail.mit.edu

Constantinos Daskalakis[1]
EECS, MIT
costis@mit.edu

S. Matthew Weinberg[1,2]
EECS, MIT
smw79@mit.edu

***Abstract*—We provide a reduction from revenue maximization to welfare maximization in multi-dimensional Bayesian auctions with arbitrary—possibly combinatorial—feasibility constraints and independent bidders with arbitrary—possibly combinatorial—demand constraints, appropriately extending Myerson's single-dimensional result [21] to this setting. We also show that every feasible Bayesian auction—including in particular the revenue-optimal one—can be implemented as a distribution over *virtual VCG allocation rules*. A virtual VCG allocation rule has the following simple form: Every bidder's type t_i is transformed into a virtual type $f_i(t_i)$, via a bidder-specific function. Then, the allocation maximizing virtual welfare is chosen. Using this characterization, we show how to find and run the revenue-optimal auction given only black-box access to an implementation of the VCG allocation rule. We generalize this result to arbitrarily correlated bidders, introducing the notion of a *second-order* VCG allocation rule.**

Our results are computationally efficient for all multi-dimensional settings where the bidders are additive, or can be efficiently mapped to be additive, albeit the feasibility and demand constraints may still remain arbitrary combinatorial. In this case, our mechanisms run in time polynomial in the number of items and the total number of bidder types, but *not* type profiles. This is polynomial in the number of items, the number of bidders, and the cardinality of the support of each bidder's value distribution. For generic correlated distributions, this is the natural description complexity of the problem. The runtime can be further improved to polynomial in only the number of items and the number of bidders in item-symmetric settings by making use of techniques from [15].

I. Introduction

The *multi-dimensional mechanism design* problem has received much attention from the economics community, and recently the computer science community as well. The problem description is simple: a seller has a limited supply of several heterogenous items for sale and many interested buyers. The goal is for the seller to design an auction for the buyers to play that will maximize her revenue. In order to make this problem tractable (not just computationally, but at all), some assumptions must be made. First, we assume that the seller has some Bayesian prior $\mathcal{D}$ on the *types* of buyers that will show up to the auction. Second, we assume that the buyers have the same prior as the seller, and that they will play any auction at a *Bayes-Nash Equilibrium*. We also assume that all buyers are *quasi-linear* and *risk-neutral*, terms that are defined formally in Section II. Finally, we say that the goal of the seller is to maximize her *expected revenue* over all auctions when played at a Bayes-Nash Equilibrium (BNE). All these assumptions have become standard for this problem. Indeed, all were made in Myerson's seminal paper on revenue-maximizing mechanism design where this problem is solved for a single item and product distributions over bidders' types [21]. In addition, Myerson introduces the *revelation principle*, showing that every auction played at a Bayes-Nash Equilibrium is strategically equivalent to a *Bayesian Incentive Compatible* (BIC) *direct revelation* mechanism. In a direct revelation mechanism, each bidder reports a bid for each possible subset of items they may receive. A direct revelation mechanism is called BIC if it is a BNE for each bidder to bid exactly their value for each subset. In essence, Myerson's revelation principle says that one only needs to consider BIC direct revelation mechanisms rather than arbitrary auctions played at a BNE to maximize revenue (or any other objective for that matter).

As we depart from Myerson's single-item setting, the issue of *feasibility* arises. With only a single item for sale, the feasibility constraints are simply that the item is always awarded to at most a single bidder. With many heterogenous items, there are many natural scenarios that we would like to model. Here are some examples:

1) Maybe the items are houses. In this case, a feasible allocation awards each house to at most one bidder, and to each bidder at most one house.
2) Maybe the items are appointment slots with doctors. Then, a feasible allocation does not award the same slot to more than one bidder, and does not award a bidder more than one slot with the same doctor, or overlapping slots with different doctors.
3) Maybe the items are bridges built at different

[1]Supported by a Sloan Foundation Fellowship, a Microsoft Research Faculty Fellowship, and NSF Award CCF-0953960 (CAREER) and CCF-1101491.
[2]Supported by a NSF Graduate Research Fellowship.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.88

locations. In this case, a feasible allocation awards each bridge to everyone or to no one.

Sometimes, feasibility constraints are imposed by the supply side of the problem: a doctor cannot meet with two patients at once, and a bridge cannot be built for one bidder but not another. Other times, feasibility constraints are imposed by the demand side of the problem: no bidder wants two houses or two appointments with the same doctor. Without differentiating where feasibility constraints come from, we model them in the following way: let $\mathcal{A} = [m] \times [n]$ denote the space of assignments (where (i, j) denotes that bidder i is assigned item j), and let $\mathcal{F}$ be a set system on $\mathcal{A}$ (that is, a subset of $2^{\mathcal{A}}$). Then in a setting with feasibility constraints $\mathcal{F}$, it is possible for the seller to simultaneously make any subset of assignments in $\mathcal{F}$. $\mathcal{F}$ may be a truly arbitrary set system, *it need not even be downward-closed.*

As we leave the single-dimensional setting, we also need to consider how a bidder values a bundle of multiple items. In general, a bidder may have arbitrarily complicated ways of evaluating bundles of items, and this information is encoded into the bidder's type. For the problem to be computationally meaningful, however, one would want to either assume that the auctioneer only has oracle access to a bidder's valuation, or impose some structure on the bidders' valuations allowing them to be succinctly described. Indeed, virtually every recent result in revenue-maximizing literature [2], [3], [6], [7], [9], [10], [11], [15], [17] assumes that bidders are *capacitated-additive*.[1] In fact, most results are for unit-demand bidders. It is easy to see that, if we are allowed to incorporate arbitrary demand constraints into the definition of $\mathcal{F}$, such bidders can be described in our model as simply additive. In fact, far more complex bidders can be modeled as well, as demand constraints could instead be some arbitrary set system. Because $\mathcal{F}$ is already an arbitrary set system, we may model bidders as simply additive and still capture virtually every bidder model studied in recent results, and more general ones as well. In fact, we note that every multi-dimensional setting can be mapped to an additive one, albeit not necessarily computationally efficiently.[2] So while we focus our discussion to additive bidders throughout this paper, our results apply to every auction setting, without need for any additivity assumption. In particular, our characterization result (Informal Theorem 2) of feasible allocation rules holds for any multi-dimensional setting, and our reduction from revenue to welfare optimization (Informal Theorem 1) also holds for any setting, and we show that it can be carried out computationally efficiently for any additive setting. We proceed to state our main results.

Optimal Multi-dimensional Mechanism Design: With the above motivation in mind, we formally state the revenue optimization problem we solve. We remark that virtually every known result in the multi-dimensional mechanism design literature (see references above) tackles a special case of this problem, possibly with budget constraints on the bidders (which can be easily incorporated in all results presented in this paper as discussed in Appendix H of the full version [8]), and possibly replacing BIC with IC. We explicitly assume in the definition of the problem that the bidders are additive, recalling that this is not a restriction if computational considerations are absent.

Revenue-Maximizing Multi-Dimensional Mechanism Design Problem (MDMDP): Given as input m distributions (possibly correlated across items) $\mathcal{D}_1, \ldots, \mathcal{D}_m$ over valuation vectors for n heterogenous items and feasibility constraints $\mathcal{F}$, output a BIC mechanism M whose allocation is in $\mathcal{F}$ with probability 1 and whose expected revenue is optimal relative to any other, possibly randomized, BIC mechanism when played by m additive bidders whose valuation vectors are sampled from $\mathcal{D} = \times_i \mathcal{D}_i$.

We provide a poly-time black-box reduction from the MDMDP with feasibility constraints $\mathcal{F}$ to implementing VCG subject to feasibility constraints $\mathcal{F}$, by introducing the notion of a *virtual VCG allocation rule*. A virtual VCG allocation rule is defined by a collection of functions f_i for each bidder i. f_i takes as input bidder i's reported bid vector and outputs a virtual bid vector. When the reported bid vectors are $\vec{v}_1, \ldots, \vec{v}_m$, the virtual VCG allocation rule with functions $\{f_i\}_{i \in [m]}$ simply implements the VCG allocation rule (subject to feasibility constraints $\mathcal{F}$) on the virtual bid vectors $f_1(\vec{v}_1), \ldots, f_m(\vec{v}_m)$. We remark that implementing VCG for additive bidders is in general *much* easier than implementing VCG for arbitrary bidders.[3] Our

[1] A bidder is capacitated-additive if for some constant c her value for any set S of at most c goods is equal to the sum of her values for each item in S, and her value for any set S of more than c goods is equal to her value for her favorite $S' \subseteq S$ of at most c goods.

[2] The generic transformation is to introduce a meta-item for every possible subset of the items, and have the feasibility constraints (which are allowed to be arbitrary) be such that an allocation is feasible if and only if each bidder receives at most one meta-item, and the corresponding allocation of real items (via replacing each meta-item with the subset of real items it represents) is feasible in the original setting. Many non-additive settings allow much more computationally efficient transformations than the generic one.

[3] When the bidders are additive, implementing VCG is solving the following problem, which is well understood for a large class of feasibility constraints: Every element of $\mathcal{A} = [m] \times [n]$ has a weight. The weight of any subset of $\mathcal{A}$ is equal to the sum of the weights of its elements. Find the max-weight subset of $\mathcal{A}$ that is in $\mathcal{F}$.

solution to the MDMDP is informally stated below, and is formally given as Theorem 4 of Section VI:

Informal Theorem 1. *For a given $\mathcal{F}$, let $A_{\mathcal{F}}$ denote an implementation of the VCG allocation rule with respect to $\mathcal{F}$ (i.e. $A_{\mathcal{F}}$ takes as input a profile of bid vectors and outputs the VCG allocation in $\mathcal{F}$). Then for all $\mathcal{D}_1, \ldots, \mathcal{D}_m$ with finite support and all $\mathcal{F}$, given $\mathcal{D}_1, \ldots, \mathcal{D}_m$ and black-box access to $A_{\mathcal{F}}$ (and without need of knowledge of $\mathcal{F}$), there exists a fully polynomial-time randomized approximation scheme*[4] *for the MDMDP whose runtime is polynomial in n, the number of bidder types (and* not *type profiles), and the runtime of $A_{\mathcal{F}}$. Furthermore, the allocation rule of the output mechanism is a distribution over virtual VCG allocation rules.*

We remark that the functions defining a virtual VCG allocation rule may map a bidder type to a vector with negative coordinates. Therefore, our given implementation of the VCG allocation rule should be able to handle negative weights. This is not a restriction for arbitrary downwards-closed $\mathcal{F}$ as any implementation of VCG that works for non-negative weights can easily be (in a black-box way) converted into an implementation of VCG allowing arbitary (possibly negative) inputs.[5] But this is not necessarily true for non downwards-closed $\mathcal{F}$'s. If the given $A_{\mathcal{F}}$ cannot accommodate negative weights, we need to replace it with an algorithm that can in order for our results to be applicable.

Several extensions are stated and discussed in Section 6 of the full version [8], including solutions for distributions of infinite support, and improved runtimes for item-symmetric settings, making use of techniques from [15]. We also extend all our solutions to accommodate strong budget constraints by the bidders in Appendix H of [8].

So how does our solution compare to Myerson's single-dimensional result? One interpretation of Myerson's optimal auction is the following: First, he shows that the allocation rule used by the optimal auction is just the Vickrey allocation rule, but on virtual bids instead of true bids. Second, he provides a closed form for each virtual transformation using (ironed) virtual values. And finally, he provides a closed form pricing rule that makes the entire mechanism BIC (in fact, IC). In the multi-dimensional setting, it is known that randomness is necessary to achieve optimal revenue, even with a single bidder and two items [5], [12], so we cannot possibly hope for a solution as clean as Myerson's. However, we have come quite close in a very general setting: Our allocation rule is just a distribution over virtual VCG allocation rules. And instead of a closed form for each virtual transformation and the pricing rule, we provide a computationally efficient algorithm to find them.

Characterization of Feasible Interim Allocation Rules: In addition to our solution of the MDMDP, we provide a characterization of feasible interim allocation rules of multi-dimensional mechanisms in all (not necessarily additive) settings.[6] We show the following informal theorem, which is stated formally as Theorem 1 in Section III. Recall that a virtual VCG allocation rule is associated with a collection of functions f_i that map types t_i to virtual types $f_i(t_i)$ for each bidder i, and allocates the items as follows: for a given type vector $(t_1, ..., t_m)$, the bidders' types are transformed into virtual types $(f_1(t_1), \ldots, f_m(t_m))$; then the virtual welfare optimizing allocation is chosen.

Informal Theorem 2. *Let $\mathcal{F}$ be any set system of feasibility constraints, and $\mathcal{D}$ any (possibly correlated) distribution over bidder types. Then the interim allocation rule of any feasible mechanism can be implemented as the interim rule of a distribution over virtual VCG allocation rules.*

A. Related Work

1) Structural Results: Structural results for the allocation rule of optimal auctions were already known prior to this work for special cases of the MDMDP and its extension to correlated bidders. As we have already discussed, Myerson showed that the revenue-optimal auction for selling a single item is a virtual Vickrey auction: bids are transformed to virtual bids, and the item is awarded to the bidder with the highest non-negative virtual value [21]. It was later shown that this approach also applies to all single-dimensional settings (i.e. when bidders can't tell the difference between different houses, appointment slots, bridges, etc) as long as bidders' values are independent. In this setting, bids are transformed to virtual bids (via Myerson's transformation), and the virtual-welfare-maximizing feasible allocation is chosen. These structural results are strong, but only hold for the single-dimensional setting and are therefore of very limited applicability.

On the multi-dimensional front, it was recently shown that similar structure exists in restricted settings. It is shown in [7] that when selling multiple heterogenous

[4]This is often abbreviated as FPRAS, and we provide its formal definition in Section II.

[5]The following simple black-box transformation achieves this: first zero-out all negative coordinates in the input vectors; then run VCG; in the VCG allocation, un-allocate item j from bidder i if the corresponding coordinate is negative; this is still a feasible allocation as the setting is downwards-closed.

[6]For non-additive settings, the characterization is more usable for the purposes of mechanism design when applied to meta-items (see discussion above), although it still holds when directly applied to items as well.

items to additive bidders with *no* demand constraints (i.e. $\mathcal{F}$ only ensures that each item is awarded to at most one bidder), the optimal auction *randomly* maps bids to virtual bids (according to some function that depends on the distributions from which bidders' values are drawn), then *separately* allocates each item to the highest virtual bidder.[7] It is shown in [2] that when there are many copies of the same customizable item and a matroid constraint on which bidders can simultaneously receive an item (i.e. $\mathcal{F}$ only ensures that at most k items are awarded, subject to a matroid constraint on the served bidders), that the optimal auction randomly maps bids to virtual bids (according to some function that depends on the distributions from which bidders' values are drawn), then allocates the items to maximize virtual surplus (and customizes them after). Both results, while quite strong for their corresponding settings, are extemely limited in the settings where they can be applied. In particular, neither says anything about the simple setting of selling houses to unit-demand bidders (i.e. when $\mathcal{F}$ ensures that each house is awarded at most once and each bidder receives at most one house: Example 1, Section I). Selling houses to unit-demand bidders is on the easy side of the settings solved in this paper, as we provide a characterization of allocation rules in multi-dimensional settings with arbitrary feasibility constraints. We do not even assume that $\mathcal{F}$ is downward-closed.

For correlated bidders, the series of results by Cremer and McLean [13], [14] and McAfee and Reny [20] solve for arbitrary feasibility constraints subject to a non-degeneracy condition on the bidder correlation (that is not met when bidders are independent). Under this assumption, they show that the optimal auction extracts full surplus (i.e. has expected revenue equal to expected welfare) and simply uses the VCG allocation rule (the prices charged are not the VCG prices, but a specially designed pricing menu based on the bidder correlation). Our characterization of feasible allocation rules (Informal Theorem 2) applies, in fact, to *arbitrary* feasibility constraints as well as *arbitrary* bidder correlation, without need of the non-degeneracy assumption. Nevertheless, we argue that it is not directly usable for the purposes of optimizing over BIC auctions when the bidders are correlated (see discussion in Section 7 of the full paper [8]). To rectify this, we provide an algorithmically usable generalization of our characterization result for correlated bidders that remains rather simple: Any feasible auction randomly maps pairs of actual bids and possible alternative bids to second-order bids. Then, the second-order bids are combined (based on the underlying bidder correlation) to form virtual bids, and the virtual-welfare-maximizing allocation is chosen.

2) Algorithmic Results: The computer science community has contributed computationally efficient solutions to special cases of the MDMDP in recent years. Many are constant factor approximations [1], [3], [10], [11], [17]. These results cover settings where the bidders are unit-demand (or capacitated-additive) and the seller has matroid or matroid-intersection constraints on which bidders can simultaneously receive which items. All these settings are special cases of the MDMDP framework solved in this paper.[8] In even more restricted cases near-optimal solutions have already been provided. Tools are developed in [6], [9], [15] that yield solutions for simple cases with one or few bidders. Cases with many asymmetric independent bidders are considered in [7] and [2]. As discussed above, in [7], the case where $\mathcal{F}$ ensures that each item is awarded at most once is solved. In [2], the case where $\mathcal{F}$ ensures that at most k items are awarded, subject to a matroid constraint on the served bidders is solved. Our computational results push far beyond existing results, providing a computationally efficient solution for multi-dimensional settings with arbitrary feasibility constraints, as long as the welfare-optimization problem for the same constraints is tractable.

B. Our Approach and Intermediate results

Since receiving attention from computer scientists, several special cases of the MDMDP have been solved computationally efficiently by linear programming [9], [15]. Simply put, these algorithms explictly store a variable for every possible bidder profile denoting the probability that bidder i receives item j on that profile, and write a linear program to maximize expected revenue subject to feasibility and BIC constraints. Unfortunately, the number of variables required for such a program is exponential in the number of bidders, making such an explicit description prohibitive. More recent solutions have used the *reduced form* of an auction [2], [7] to sidestep this curse. The reduced form of an auction was first studied in [4], [18], [19] and contains, for every bidder i, for every type A of bidder i, and every item j, the probability that bidder i receives item j when truthfully reporting type A over the randomness of the auction *and the randomness in the other bidders' types, assuming they report truthfully*. Indeed, the reduced form contains all the necessary information to verify that an auction is BIC when bidders are independent,

[7]In fact, the allocation rule of [7] has even stronger structure in that each item is independently allocated to the bidder whose virtual value for that item is the highest, and moreover the random mapping defining virtual values for each item simply irons a *total ordering* of all bidder types that depends on the underlying distribution.

[8]Again, in some of these results [1], [3] bidders may also have budget constraints, which can be easily incorporated to the MDMDP framework without any loss, as is shown in Appendix H of the full version [8], and some replace BIC with IC [1], [10], [11], [17].

although verifying its feasibility (i.e. whether a feasible mechanism exists matching its marginal allocation probabilities) appears difficult. Despite this difficulty, computationally efficient separation oracles were discovered for independent bidders and a single item [2], [7]. The techniques of [2] also accomodate many copies of the same item and a matroid constraint on which bidders may simultaneously be served. In this paper, we go far beyond these results considering reduced forms in settings with arbitrary feasibility constraints. Surprisingly, we are able to give a simple proof of a strong characterization result: for arbitrary feasibility constraints, every feasible reduced form can be implemented by a distribution over virtual VCG allocation rules. Our proof is in Section III and follows the spirit of [4], [7]: we examine the region of feasible reduced forms (we show it must be a convex polytope) and identify special structure in its extreme points.

In Section IV we provide a separation oracle for checking feasibility of reduced forms, as well as a decomposition algorithm for explicitly writing feasible reduced forms as distributions over virtual VCG allocation rules for arbitrary feasibility constraints $\mathcal{F}$, given only black-box access to an implementation of VCG subject to the same constraints $\mathcal{F}$. To make our algorithms exact, we need time polynomial in $|\mathcal{D}|$, making them practically unusable. In Section V, we show instead how to ϵ-implement both the separation oracle and the decomposition algorithm in time polynomial in $\sum_{i=1}^m |\mathcal{D}_i|$ and $1/\epsilon$ with high probability. By ϵ-implementing a separation oracle and a decomposition algorithm for some polytope P, we mean: (i) computing a polytope P' such that every point in P is within $\epsilon/\text{poly}(n\sum_{i=1}^m |\mathcal{D}_i|)$ (in ℓ_∞ distance) of a point in P' and vice versa; and (ii) exactly implementing a separation oracle and a decomposition algorithm for P'. We show that (i) and (ii) are sufficient for computationally efficiently deciding whether a reduced form that is ϵ-far (in ℓ_∞) from the boundary of P lies inside P, as well as for computing a distribution over virtual VCG allocation rules that is within $\epsilon/\text{poly}(n\sum_{i=1}^m |\mathcal{D}_i|)$ (in ℓ_∞ distance) of any given feasible reduced form that is ϵ-far from the boundary of P.

In Section VI, we show how to combine the linear programs from [7], [15] with our algorithms for reduced forms to obtain an FPRAS for MDMDP using only black-box access to an implementation of VCG for $\mathcal{F}$. In generic cases, the runtime is polynomial in the number of items and $\sum_{i=1}^m |\mathcal{D}_i|$ (but importantly not $|\mathcal{D}|$). In many settings (e.g. when there is correlation among item values, or when the value distributions have sparse supports) this is the natural description complexity of the problem, and several recent algorithms [1], [2], [3], [7], [16] have the same computational complexity, namely polynomial in the number of bidders, the number of items and the cardinality of the support of each bidder's value distribution. Additionally, by using results of [15] we can reduce the runtime to polynomial in only the number of items and number of bidders in item-symmetric settings, as well as extend our solution to distributions with infinite support. These extensions are discussed in Section 6 of the full version [8]. Our mechanisms can be made interim or ex-post individually rational without any difference in revenue. We are also able to naturally accommodate hard budget constraints in our solutions. The simple modification that is necessary is given in Appendix H of [8].

Finally, in Section 7 of [8], we provide our characterization result for correlated bidders. To do this, we introduce the notion of a second-order reduced form, and show that every second-order reduced form can be implemented as a distribution over second-order VCG allocation rules. With this modification, all related techniques of Section IV also apply to correlated bidders. All discussion of correlated bidders is omitted due to space constraints. See the full version of our paper [8].

II. Preliminaries and notation

We denote the number of bidders by m, the number of items by n, and the type space of bidder i by T_i. To ease notation, we sometimes use A (B, C, etc.) to denote the *type* of a bidder, without emphasizing whether it is a vector or a scalar. The elements of $\times_i T_i$ are called *type profiles*, and specify a type for every bidder. We assume type profiles are sampled from a distribution $\mathcal{D}$ over $\times_i T_i$. We denote by $\mathcal{D}_i$ the marginal of $\mathcal{D}$ over bidder i's type, and by $\mathcal{D}_{-i}$ the marginal of $\mathcal{D}$ over the types of all bidders, except bidder i. Finally, we use t_i for the random variable representing the type of bidder i. So when we write $\Pr[t_i = A]$, we mean the probability that bidder i's type is A. In Appendix B of the full version [8], we also discuss how our algorithms access distribution $\mathcal{D}$.

We let $\mathcal{A} = [m] \times [n]$ denote the set of possible *assignments* (i.e. the element (i, j) denotes that bidder i was awarded item j). We call (distributions over) subsets of $\mathcal{A}$ (randomized) *allocations*, and functions mapping type profiles to (possibly randomized) allocations *allocation rules*. We call an allocation combined with a price charged to each bidder an *outcome*, and an allocation rule combined with a pricing rule a (direct revelation) *mechanism*. As discussed in Section I, we may also have a set system $\mathcal{F}$ on $\mathcal{A}$ (that is, a subset of $2^{\mathcal{A}}$), encoding constraints on what assignments can be made simultaneously by the mechanism. $\mathcal{F}$ may be incorporating arbitrary demand constraints imposed by each bidder, and supply constraints imposed by the seller, and will be referred to as our *feasibility* *constraints*. In this case, we restrict all allocation rules to be supported on $\mathcal{F}$.

The *reduced form* of an allocation rule (also called the *interim allocation rule*) is a vector function $\pi(\cdot)$, specifying values $\pi_{ij}(A)$, for all items j, bidders i and types $A \in T_i$. $\pi_{ij}(A)$ is the probability that bidder i receives item j when truthfully reporting type A, where the probability is over the randomness of all other bidders' types (drawn from $\mathcal{D}_{-i}$) and the internal randomness of the allocation rule, assuming that the other bidders report their types truthfully. We sometimes want to view the reduced form as a $n\sum_{i=1}^{m}|T_i|$-dimensional vector, and may write $\vec{\pi}$ to emphasize this view.

Given a reduced form π, we will be interested in whether the form is "feasible", or can be "implemented." By this we mean designing a feasible allocation rule M (i.e. one that respects feasibility constraints $\mathcal{F}$ on every type profile with probability 1 over the randomness of the allocation rule) such that the marginal probability $M_{ij}(A)$ that bidder i receives item j when truthfully reporting type A is exactly $\pi_{ij}(A)$, where the probability is computed with respect to the randomness in the allocation rule and the randomness in the types of the other bidders, assuming that the other bidders report their types truthfully. While viewing reduced forms as vectors, we will denote by $F(\mathcal{F}, \mathcal{D})$ the set of feasible reduced forms when the feasibility constraints are $\mathcal{F}$ and bidders are sampled from $\mathcal{D}$.

Throughout this paper we assume that the bidders are additive, keeping in mind that this is not a restriction computational considerations aside (see Section I). A bidder is *additive* if her value for a bundle of items is the sum of her values for the items in that bundle. If bidders are additive, to specify the preferences of bidder i, we can provide a valuation vector $\vec{v}_i$, with the convention that v_{ij} represents her value for item j. Even in the presence of arbitrary demand constraints, the *value* of additive bidder i of type $\vec{v}_i$ for a randomized allocation that respects the bidder's demand constraints with probability 1, and whose expected probability of allocating item j to the bidder is π_{ij}, is just the bidder's expected value, namely $\sum_j v_{ij}\cdot\pi_{ij}$. The *utility* of bidder i for the same allocation when paying price p_i is just $\sum_j v_{ij} \cdot \pi_{ij} - p_i$. Such bidders whose value for a distribution of allocations is their expected value for the sampled allocation are called *risk-neutral*. Bidders subtracting price from expected value are *quasi-linear*.

Throughout this paper, we denote by OPT the expected revenue of an optimal solution to MDMDP. Also, most of our results for this problem construct a *fully polynomial-time randomized approximation scheme*, or FPRAS. This is an algorithm that takes as input two additional parameters $\epsilon, \eta > 0$ and outputs a mechanism (or succinct description thereof) whose revenue is at least OPT $-\ \epsilon$, with probability at least $1-\eta$ (over the coin tosses of the algorithm), in time polynomial in $n\sum_i |T_i|, 1/\epsilon$, and $\log(1/\eta)$.

Finally, some arguments will involve reasoning about the *bit complexity* of a rational number. We say that a rational number has bit complexity b if it can be written with a binary numerator and denominator that each have at most b bits. In Appendix A of [8] we provide the standard notions of Bayesian Incentive Compatibility (BIC) and Individual Rationality (IR) of mechanisms.

III. Characterization of Feasible Reduced Forms

In this section, we provide our characterization result, showing that every feasible reduced form can be implemented as a distribution over virtual VCG allocation rules. For space considerations, all proofs of this section are in Appendix C of the full version of this paper [8]. In the following definition, $VCG_{\mathcal{F}}$ denotes the allocation rule of VCG with feasibility constraints $\mathcal{F}$. That is, on input $\vec{v} = (\vec{v}_1, \ldots, \vec{v}_m)$, $VCG_{\mathcal{F}}$ outputs the allocation that VCG selects when the reported types are $\vec{v}$.

Definition 1. *A* ***virtual VCG*** *allocation rule is defined by a collection of weight functions $\{f_i : T_i \rightarrow \mathbb{R}^n\}_i$ where f_i maps a type of bidder i to a* virtual type *of bidder i. On any type profile $\vec{v}$, the virtual VCG allocation rule with functions $\{f_i\}_{i\in[m]}$ runs $VCG_{\mathcal{F}}$ on input $(f_1(\vec{v}_1), \ldots, f_m(\vec{v}_m))$.*[9] *$VVCG_{\mathcal{F}}(\{f_i\}_{i\in[m]})$ denotes the virtual VCG allocation rule with feasibility constraints $\mathcal{F}$ and weight functions $\{f_i\}_{i\in[m]}$.*

In other words, a virtual VCG allocation rule is simply a VCG allocation rule, but maximizing virtual welfare instead of true welfare. It will be convenient to introduce the following notation, viewing the weight functions as a (scaled) $n\sum_{i=1}^{m}|T_i|$-dimensional vector. Below, f_{ij} denotes the j^{th} component of f_i.

Definition 2. *Let $\vec{w} \in \mathbb{R}^{n\sum_{i=1}^{m}|T_i|}$. Define f_i so that $f_{ij}(A) = \frac{w_{ij}(A)}{\Pr[t_i = A]}$. Then $VVCG_{\mathcal{F}}(\vec{w})$ is the virtual VCG allocation rule $VVCG_{\mathcal{F}}(\{f_i\}_{i\in[m]})$.*

It is easy to see that every virtual VCG allocation rule can be defined using the notation of Definition 2 by simply setting $w_{ij}(A) = f_{ij}(A) \cdot \Pr[t_i = A]$. We scale the weights this way only for notational convenience (which first becomes useful in Lemma 1). We say that a virtual VCG allocation rule is simple iff, for all $\vec{v}_1, \ldots, \vec{v}_m$, $VCG_{\mathcal{F}}(f_1(\vec{v}_1), \ldots, f_m(\vec{v}_m))$ has a unique max-weight allocation. We now state the main

[9] If there are multiple VCG allocations, break ties arbitrarily, but consistently. A consistent lexicographic tie-breaking rule is discussed in Section III-A. For concreteness, the reader can use this rule for all results of this section.

theorem of this section, which completely characterizes all feasible reduced forms.

Theorem 1. *Let $\mathcal{F}$ be any set system of feasibility constraints, and $\mathcal{D}$ be any (possibly correlated) distribution over bidder types with finite support. Then every feasible reduced form (with respect to $\mathcal{F}$ and $\mathcal{D}$) can be implemented as a distribution over at most $n\sum_{i=1}^{m}|T_i|+1$ simple virtual VCG allocation rules.*[10]

The proof of Theorem 1 begins with a simple observation and proposition, proved in [8].

Observation 1. *An allocation rule is feasible if and only if it is a distribution over feasible deterministic allocation rules.*

Proposition 1. *If $|\mathcal{D}|$ is finite, $F(\mathcal{F},\mathcal{D})$ is a convex polytope.*

Now that we know that $F(\mathcal{F},\mathcal{D})$ is a convex polytope, we want to look at the extreme points by examining, for any $\vec{w}$, the allocation rule of $F(\mathcal{F},\mathcal{D})$ whose reduced form maximizes $\vec{\pi}\cdot\vec{w}$. Lemma 1 and Proposition 2 characterize the extreme points of $F(\mathcal{F},\mathcal{D})$, allowing us to prove Theorem 1. All three proofs are simple, and provided in Appendix C of the full paper [8].

Lemma 1. *Let $\vec{\pi}$ be the reduced form of $VVCG_{\mathcal{F}}(\vec{w})$ (using an arbitrary tie-breaking rule) when bidders are sampled from $\mathcal{D}$. Then, for all $\vec{\pi}'\in F(\mathcal{F},\mathcal{D})$, $\vec{\pi}\cdot\vec{w}\geq\vec{\pi}'\cdot\vec{w}$.*

Proposition 2. *Every corner (i.e. vertex) of $F(\mathcal{F},\mathcal{D})$ can be implemented by a simple virtual VCG allocation rule, and the reduced form of any simple virtual VCG allocation rule is a corner of $F(\mathcal{F},\mathcal{D})$.*

We conclude with a necessary and sufficient condition for feasibility of a reduced form. The proof is simple and can be found in Appendix C of [8]. For the following statement, for any weight vector $\vec{w}\in\mathbb{R}^{n\sum_{i=1}^{m}|T_i|}$, $W_{\mathcal{F}}(\vec{w})$ denotes the total expected weight of items awarded by $VVCG_{\mathcal{F}}(\vec{w})$ (where we assume that the weight of giving item j to bidder i of type A is $f_{ij}(A)=w_{ij}(A)/\Pr[t_i=A]$). The proof of Lemma 1 implies that the tie-breaking rule used in $VVCG_{\mathcal{F}}(\vec{w})$ does not affect the value of $W_{\mathcal{F}}(\vec{w})$, and that no feasible allocation rule can possibly exceed $W_{\mathcal{F}}(\vec{w})$. The content of the next corollary is that this condition is also sufficient.

Corollary 1. *A reduced form $\vec{\pi}$ is feasible (with respect to $\mathcal{F}$ and $\mathcal{D}$) if and only if, for all $\vec{w}\in[-1,1]^{n\sum_{i=1}^{m}|T_i|}$, $\vec{\pi}\cdot\vec{w}\leq W_{\mathcal{F}}(\vec{w})$.*

[10]Note that we are *not* claiming that every feasible *allocation rule* can be implemented as a distribution over virtual VCG allocation rules. See a brief example illustrating the content of Theorem 1 in Section 3 of the full version [8].

A. Tie-breaking

Any given $\vec{w}$ can be modified to some $\vec{w}'$ so that (i) $VVCG_{\mathcal{F}}(\vec{w}')$ maximizes $\vec{\pi}\cdot\vec{w}$ over $\vec{\pi}\in F(\mathcal{F},\mathcal{D})$, and (ii) $VVCG_{\mathcal{F}}(\vec{w}')$ is simple. The modification can be carried out in polynomial time and implements a lexicographic tie-breaking among allocations with equal virtual welfare under $\vec{w}$. For details see Section 3.1 of [8]. From now on, whenever we use $VVCG_{\mathcal{F}}(\vec{w})$, we will implicitly assume that this tie-breaking modification has been applied to $\vec{w}$. Sometimes we will explicitly state so, if we want to get our hands on $\vec{w}'$.

IV. Algorithms for Reduced Forms

The characterization result of Section III hinges on the realization that $F(\mathcal{F},\mathcal{D})$ is a convex polytope whose corners can be implemented by especially simple allocation rules, namely simple virtual VCG allocation rules. To compute the reduced form of an optimal mechanism, we would like to be able to optimize a linear objective (expected revenue) over $F(\mathcal{F},\mathcal{D})$. So we need a separation oracle for this polytope. Additionally, once we have found the revenue-optimal reduced form in $F(\mathcal{F},\mathcal{D})$, we need some way of implementing it. As we know that every corner of $F(\mathcal{F},\mathcal{D})$ can be implemented by an especially simple allocation rule, we would like to decompose a given feasible reduced form into an explicit convex combination of corners (which then corresponds to the reduced form of a distribution over simple virtual VCG allocation rules). Without worrying about computational complexity, in this section we provide a generic framework for obtaining both algorithms. In Section V, we discuss how to approximately implement these algorithms efficiently with high probability obtaining an FPRAS with only black-box access to an implementation of the VCG allocation rule.

A. Separation Oracle

We know from Corollary 1 that if a reduced form $\vec{\pi}$ is infeasible, then there is some weight vector $\vec{w}\in[-1,1]^{n\sum_{i=1}^{m}|T_i|}$ such that $\vec{\pi}\cdot\vec{w}>W_{\mathcal{F}}(\vec{w})$. Finding such a weight vector explicitly gives us a hyperplane separating $\vec{\pi}$ from $F(\mathcal{F},\mathcal{D})$, provided we can also compute $W_{\mathcal{F}}(\vec{w})$. So consider the function:

$$g_{\vec{\pi}}(\vec{w})=W_{\mathcal{F}}(\vec{w})-\vec{\pi}\cdot\vec{w}.$$

We know that $\vec{\pi}$ is feasible if and only if $g_{\vec{\pi}}(\vec{w})\geq 0$ for all $\vec{w}\in[-1,1]^{n\sum_{i=1}^{m}|T_i|}$. So the goal of our separation oracle SO is to minimize $g_{\vec{\pi}}(\vec{w})$ over the hypercube, and check if the minimum is negative. If negative, the reduced form is infeasible, and the minimizer bears witness. Otherwise, the reduced form is feasible. To write a linear program to minimize $g_{\vec{\pi}}(\vec{w})$, recall that $W_{\mathcal{F}}(\vec{w}) = \max_{\vec{x}\in F(\mathcal{F},\mathcal{D})}\{\vec{x}\cdot\vec{w}\}$, so $g_{\vec{\pi}}(\vec{w})$ is a piece-wise linear function. Using standard techniques, we could add a variable, t, for $W_{\mathcal{F}}(\vec{w})$, add constraints to guarantee that $t \geq \vec{x}\cdot\vec{w}$ for all $\vec{x}\in F(\mathcal{F},\mathcal{D})$, and minimize $t - \vec{\pi}\cdot\vec{w}$. As this is a burdensome number of constraints, we will use an internal separation oracle $\widehat{SO}$, whose job is simply to verify that $t \geq \vec{x}\cdot\vec{w}$ for all $\vec{x}\in F(\mathcal{F},\mathcal{D})$, otherwise output a violating hyperplane.

To implement $\widehat{SO}$, let $R_{\mathcal{F}}(\vec{w})$ denote the reduced form of $VVCG_{\mathcal{F}}(\vec{w})$. Then we know that $R_{\mathcal{F}}(\vec{w})\cdot\vec{w} \geq \vec{x}\cdot\vec{w}$ for all $\vec{x}\in F(\mathcal{F},\mathcal{D})$. So if any equation of the form $\vec{x}\cdot\vec{w}\leq t$ is violated, then certainly $R_{\mathcal{F}}(\vec{w})\cdot\vec{w}\leq t$ is violated. Therefore, for an input $\vec{w}, t$, we need only check a single constraint of this form. So let $\widehat{SO}(\vec{w},t)$ output "yes" if $R_{\mathcal{F}}(\vec{w})\cdot\vec{w}\leq t$, otherwise output the violated hyperplane $R_{\mathcal{F}}(\vec{w})\cdot\vec{z} - y \leq 0$. $\widehat{SO}$ allows us to reformulate the linear program and minimize $g_{\vec{\pi}}(\vec{w})$ efficiently using Ellipsoid. Our LP is explicitly given in Figure 1 of Appendix D of [8].

So our separation oracle SO for checking whether $\vec{\pi}\in F(\mathcal{F},\mathcal{D})$ works as follows: first we run the afore-described LP to minimize $g_{\vec{\pi}}(\vec{w})$. Let the optimum output by the LP be $t^*, \vec{w}^*$. If the value of the LP is negative, we know that $\vec{w}^*\cdot\vec{\pi} > t^* = W_{\mathcal{F}}(\vec{w}^*)$, and we obtain our violated hyperplane. Otherwise, the reduced form is feasible, so we output "yes."

We conclude this section with a lemma relating the bit complexity of the corners of $F(\mathcal{F},\mathcal{D})$ to the bit complexity of the output of our separation oracle. This is handy for efficiently implementing our algorithms in the next section. The proof is simple, and provided in Appendix D of [8]. We make use a standard property of the Ellipsoid algorithm (see Theorem 10 of [8]).

Lemma 2. *If all coordinates of each corner of $F(\mathcal{F},\mathcal{D})$ are rational numbers of bit complexity ℓ, then every coefficient of any hyperplane output by SO is a rational number of bit complexity poly$(n\sum_{i=1}^m |T_i|, \ell)$.*

B. Decomposition Algorithm via a Corner Oracle

We provide an algorithm for writing a feasible reduced form as a convex combination of corners of $F(\mathcal{F},\mathcal{D})$, i.e. reduced forms of simple virtual VCG allocation rules. A decomposition algorithm for arbitrary polytopes P is already given in [7], and the only required ingredients for the algorithm is a separation oracle for P, a *corner oracle* for P, and a bound b on the bit complexity of the coefficients of any hyperplane that can possibly be output by the separation oracle. The goal of this section is to define both oracles and determine b for our setting. But let us first recall the result of [7]. Before stating the result, let us specify the required functionality of the corner oracle.

The corner oracle for polytope P takes as input k (where k is at most the dimension, in our case $n\sum_i |T_i|$) hyperplanes $H_1,\dots,H_k$ (whose coefficients are all rational numbers of bit complexity b) and has the following behavior: If no hyperplane intersects P in its interior and there is a corner of P that lies in all hyperplanes, then such a corner is output. Otherwise, the behavior may be arbitrary. Below is the theorem from [7].

Theorem 2. *([7]) Let P be a d-dimensional polytope with corner oracle CO and separation oracle SO such that each coefficient of every hyperplane ever output by SO is a rational number of bit complexity b. Then there is an algorithm that decomposes any point $\vec{x}\in P$ into a convex combination of at most $d+1$ corners of P. Furthermore, if ℓ is the maximum number of bits needed to represent a coordinate of $\vec{x}$, then the runtime is polynomial in d, b, ℓ and the runtimes of SO and CO on inputs of bit complexity poly(d, b, ℓ).*

So all we need to do is define CO and SO, and provide a bound on the bit complexity of the hyperplanes output by SO. We've already defined SO and bounded the bit complexity of hyperplanes output by it by poly$(n\sum_{i=1}^m |T_i|, \ell)$, where ℓ is the maximum number of bits needed to represent a coordinate in a corner of $F(\mathcal{F},\mathcal{D})$ (see Lemma 2 of Section IV-A). So now we define CO and state its correctness in Theorem 3 whose proof is in Appendix D of [8]. In the last line, CO outputs the weights $\vec{w}'$ as well so that we can actually implement the reduced form that is output.

Algorithm 1 Corner Oracle for $F(\mathcal{F},\mathcal{D})$

1: Input: Hyperplanes $(\vec{w}_1, h_1),\dots,(\vec{w}_a,\ h_a)$, $a \leq n\sum_{i=1}^m |T_i|$.
2: Set $\vec{w} = \sum_{j=1}^a \frac{1}{a}\vec{w}_j$.
3: Use the tie-breaking rule of Section III-A (stated formally in Lemma 4 in [8]) on $\vec{w}$ to obtain $\vec{w}'$.
4: Output the reduced form of $VVCG_{\mathcal{F}}(\vec{w}')$, and $\vec{w}'$.

Theorem 3. *The Corner Oracle of Algorithm 1 correctly outputs a corner of $F(\mathcal{F},\mathcal{D})$ contained in $\cap_{j=1}^a H_j$ whenever the hyperplanes $H_1,\dots,H_a$ are boundary hyperplanes of $F(\mathcal{F},\mathcal{D})$ and $\cap_{j=1}^a H_j$ contains a corner. Furthermore, if all coordinates of all H_j are rational numbers of bit complexity b, and* $\Pr[t_i = A]$ *is a rational number of bit complexity ℓ for all $i, A\in T_i$, then every coordinate of the weight vector $\vec{w}'$ is a rational number of bit complexity poly$(n\sum_{i=1}^m |T_i|, b, \ell)$.*

V. Efficient Implementation of Algorithms for Reduced Forms

In this section, we show how to approximately implement the separation oracle (SO) of Section IV-A and the corner oracle (CO) of Section IV-B efficiently with high probability, thereby obtaining also an approximate decomposition algorithm for $F(\mathcal{F}, \mathcal{D})$. We begin by bounding the runtime of the exact implementations, showing that they are especially good when $\mathcal{D}$ is a uniform (possibly non-product) distribution of small support. As above, $A_{\mathcal{F}}$ denotes an algorithm that implements the VCG allocation rule with respect to feasibility constraints $\mathcal{F}$, and we use $rt_{\mathcal{F}}(b)$ to denote the runtime of $A_{\mathcal{F}}$ when each input weight has bit complexity b.

A. *Exact Implementation*

The only tricky step in implementing SO and CO is computing $R_{\mathcal{F}}(\vec{w})$ for a given $\vec{w}$. A simple approach is to just enumerate every profile in the support of $\mathcal{D}$ and check if $VVCG_{\mathcal{F}}(\vec{w})$ awards bidder i item j. This can be done in time polynomial in the cardinality $|\mathcal{D}|$ of the support of $\mathcal{D}$, the bit complexity ℓ of the probabilities used by $\mathcal{D}$ and $rt_{\mathcal{F}}(\text{poly}(b, \ell))$, where b is the bit complexity of $\vec{w}$'s coordinates. So, if b is an upper bound on the bit complexity of the coordinates of the weight vectors $\vec{w}$ for which $R_{\mathcal{F}}(\vec{w})$ is computed in an execution of SO (CO), then SO (CO) can be implemented in time polynomial in $n \sum_i |T_i|, |\mathcal{D}|$, ℓ, b, c, and $rt_{\mathcal{F}}(\text{poly}(b, \ell))$, where c is the bit complexity of the numbers in the input of SO (CO). Alone, this result is not very helpful as we can do much more interesting computations in time polynomial in $|\mathcal{D}|$, including exactly solve MDMDP [15]. The interesting corollary is that when $\mathcal{D}$ is a (possibly correlated) uniform distribution over a collection of profiles (possibly with repetition) whose number is polynomial in $n \sum_i |T_i|$, the runtime of all algorithms of Section IV becomes polynomial in $n \sum_i |T_i|$, c, and $rt_{\mathcal{F}}(\text{poly}(n \sum_i |T_i|, c))$, where c is the bit complexity of the numbers in the input to these algorithms. In Appendix E of [8], we quantify this claim precisely, enabling efficient approximations for arbitrary distributions in the next section.

B. *Approximate Implementation*

We discuss how to "approximately implement" both algorithms in time polynomial in $\sum_{i=1}^m |\mathcal{D}_i|$, where $|\mathcal{D}_i|$ is the cardinality of the support of $\mathcal{D}_i$, using the results of Section V-A. But we need to use the right notion of approximation. Simply implementing both algorithms approximately, e.g. separating out reduced forms that are not even approximately feasible and decomposing reduced forms that are approximately feasible, might not get us very far, as we could lose the necessary linear algebra to solve LPs. So we use a different notion of approximation. We compute a "simple" polytope P' that, with high probability, is a "good approximation" to $F(\mathcal{F}, \mathcal{D})$ in the sense that we can optimize over P' instead of over $F(\mathcal{F}, \mathcal{D})$. Then we implement both the separation and the decomposition algorithms for P' exactly so that their running time is polynomial in n, $\sum_{i=1}^m |T_i|$, c and $rt_{\mathcal{F}}(\text{poly}(n \sum_{i=1}^m |T_i|, c))$, where c is the number of bits needed to describe a coordinate of the input to these algorithms.

Approach: So how can we compute an approximating polytope? Our starting point is natural: Given an arbitrary distribution $\mathcal{D}$, we can sample profiles $P_1, \ldots, P_k$ from $\mathcal{D}$ independently at random and define a new distribution $\mathcal{D}'$ that samples a profile uniformly at random from $P_1, \ldots, P_k$ (i.e. chooses each P_i with probability $1/k$). Clearly as $k \to \infty$ the polytope $F(\mathcal{F}, \mathcal{D}')$ should approximate $F(\mathcal{F}, \mathcal{D})$ better and better. The question is how large k should be taken for a good approximation. If taking k polynomial in $n \sum_{i=1}^m |T_i|$ suffices, then Section V-A implies that we can implement both the separation and the decomposition algorithms for $F(\mathcal{F}, \mathcal{D}')$ in the desired running time.

However this approach fails, as some types may very well have $\Pr[t_i = A] << \frac{1}{\text{poly}(n \sum_{i=1}^m |T_i|)}$. Such types likely wouldn't even appear in the support of $\mathcal{D}'$ if k is taken polynomial in $n \sum_{i=1}^m |T_i|$. So then how would the proxy polytope $F(\mathcal{F}, \mathcal{D}')$ inform us about $F(\mathcal{F}, \mathcal{D})$ in the corresponding dimensions? To cope with this, for each bidder i and type $A \in T_i$, we take an additional k' samples from $\mathcal{D}_{-i}$ and set $t_i = A$ on those samples. $\mathcal{D}'$ is defined to choose uniformly at random from all $k + k' \sum_{i=1}^m |T_i|$ sampled profiles (original plus additional).

Now here is what we can guarantee. Taking k and k' both polynomial in $n \sum_{i=1}^m |T_i|$, we show that with high probability every $\vec{\pi}$ in $F(\mathcal{F}, \mathcal{D})$ has some $\vec{\pi}' \in F(\mathcal{F}, \mathcal{D}')$ with $|\vec{\pi} - \vec{\pi}'|_\infty$ small. This is done by taking careful concentration and union bounds. We also show the converse: with high probability every $\vec{\pi}' \in F(\mathcal{F}, \mathcal{D}')$ has some $\vec{\pi} \in F(\mathcal{F}, \mathcal{D})$ with $|\vec{\pi} - \vec{\pi}'|_\infty$ small. This requires a little more care as the elements of $F(\mathcal{F}, \mathcal{D}')$ are not fixed a priori (i.e. before taking samples from $\mathcal{D}$ to define $\mathcal{D}'$), but depend on the choice of $\mathcal{D}'$, which is precisely the object with respect to which we want to use the probabilistic method. We resolve this apparent circularity by appealing to some properties of the algorithms of Section IV (namely, bounds on the bit complexity of any output of SO and CO). We put these results together to obtain our efficient approximation algorithms. All details of this section can be found in Appendix F of the full version [8].

VI. Revenue-Maximizing Mechanisms

In this section we make Informal Theorem 1 precise, stating how the algorithms of Section V-B combined with the LPs of [7], [15] provide a computationally efficient, nearly-optimal solution to MDMDP using only black-box access to an implementation of the VCG allocation rule. In the following statement, the allocation rule of the output mechanism is a distribution over simple virtual VCG allocation rules. On the other hand, there is no special structure in the pricing rule—it is just the output of a linear program. Recall that $A_\mathcal{F}$ denotes an algorithm implementing the VCG allocation rule with feasibility constraints $\mathcal{F}$, and $rt_\mathcal{F}(b)$ is the runtime of $A_\mathcal{F}$ when each input weight has bit complexity b.

Theorem 4. *For all $\epsilon, \eta > 0$, all $\mathcal{D}$ of finite support in $[0,1]^{nm}$, and all $\mathcal{F}$, given $\mathcal{D}$ and black-box access to $A_\mathcal{F}$ there is an additive FPRAS for MDMDP. In particular, the FPRAS obtains expected revenue OPT $- \epsilon$, with probability at least $1-\eta$, in time polynomial in ℓ, $m, n, \max_{i\in[m]}\{|T_i|\}, 1/\epsilon, \log(1/\eta)$ and $rt_\mathcal{F}(poly(n\sum_{i=1}^{m}|T_i|, \log 1/\epsilon, \log\log(1/\eta), \ell))$, where ℓ is an upper bound on the bit complexity of the coordinates of the points in the support of $\mathcal{D}$, as well as of the probabilities assigned by $\mathcal{D}_1, \ldots, \mathcal{D}_m$ to the points in their support. The output mechanism is ϵ-BIC, its allocation rule is a distribution over simple virtual VCG allocation rules, and it can be implemented in the afore-stated running time.*

The assumption that $\mathcal{D}$ is supported in $[0,1]^{mn}$ as opposed to some other bounded set is w.l.o.g., as we could just scale the values down by a multiplicative $v_{\max}$, causing the additive approximation error to be $\epsilon v_{\max}$. We also remark that the output mechanism can be made interim or ex-post individually rational without any difference in revenue, and we can accommodate bidders with hard budget constraints. Finally, the running time can be improved to polynomial in just the number of items and bidders in item-symmetric settings. These variants of our result are presented in Section 6 of the full paper [8], while all proofs can be found in Appendices G and H.

References

[1] Saeed Alaei. Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers. In *the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

[2] Saeed Alaei, Hu Fu, Nima Haghpanah, Jason Hartline, and Azarakhsh Malekian. Bayesian Optimal Auctions via Multi- to Single-agent Reduction. In *the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

[3] Sayan Bhattacharya, Gagan Goel, Sreenivas Gollapudi, and Kamesh Munagala. Budget Constrained Auctions with Heterogeneous Items. In *the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.

[4] Kim C. Border. Implementation of reduced form auctions: A geometric approach. *Econometrica*, 59(4):1175–1187, 1991.

[5] Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S. Matthew Weinberg. Pricing Randomized Allocations. In *the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[6] Yang Cai and Constantinos Daskalakis. Extreme-Value Theorems for Optimal Multidimensional Pricing. In *the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

[7] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. An Algorithmic Characterization of Multi-Dimensional Mechanisms. In *the 44th Annual ACM Symposium on Theory of Computing (STOC)*, 2012.

[8] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. Optimal Multi-Dimensional Mechanism Design: Reducing Revenue to Welfare Maximization. *arxiv report*, 2012. http://arxiv.org/abs/1207.5518.

[9] Yang Cai and Zhiyi Huang. Simple and Nearly Optimal Multi-Item Auctions. *Manuscript*, 2012.

[10] Shuchi Chawla, Jason D. Hartline, and Robert D. Kleinberg. Algorithmic Pricing via Virtual Valuations. In *the 8th ACM Conference on Electronic Commerce (EC)*, 2007.

[11] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-Parameter Mechanism Design and Sequential Posted Pricing. In *the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.

[12] Shuchi Chawla, David L. Malec, and Balasubramanian Sivan. The Power of Randomness in Bayesian Optimal Mechanism Design. In *the 11th ACM Conference on Electronic Commerce (EC)*, 2010.

[13] Jacques Cremer and Richard P. McLean. Optimal selling strategies under uncertainty for a discriminating monopolist when demands are interdependent. *Econometrica*, 53(2):345–361, 1985.

[14] Jacques Cremer and Richard P. McLean. Full extraction of the surplus in bayesian and dominant strategy auctions. *Econometrica*, 56(6):1247–1257, 1988.

[15] Constantinos Daskalakis and S. Matthew Weinberg. Symmetries and Optimal Multi-Dimensional Mechanism Design. In *the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

[16] Shahar Dobzinski, Hu Fu, and Robert D. Kleinberg. Optimal Auctions with Correlated Bidders are Easy. In *the 43rd ACM Symposium on Theory of Computing (STOC)*, 2011.

[17] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *the 44th Annual ACM Symposium on Theory of Computing (STOC)*, 2012.

[18] Eric Maskin and John Riley. Optimal Auctions with Risk Averse Buyers. *Econometrica*, 52(6):1473–1518, 1984.

[19] Steven Matthews. On the Implementability of Reduced Form Auctions. *Econometrica*, 52(6):1519–1522, 1984.

[20] R. Preston McAfee and Philip J. Reny. Correlated information and mechanism design. *Econometrica*, 60(2):395–421, 1992.

[21] Roger B. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6(1):58–73, 1981.