CS364B: Frontiers in Mechanism Design
Lecture #8: MIR and MIDR Mechanisms*

Tim Roughgarden†

January 29, 2014

1 The Challenges of DSIC Multi-Parameter Mechanism Design

Last lecture concluded with a $\frac{1}{2}$-approximation algorithm for computing a welfare-maximizing allocation for bidders submodular valuations (scenario #6); the exercises give another such algorithm. How can we extend the allocation rules induced by these algorithms into DSIC mechanisms?

This question turns out to be considerably harder than it looks. We won't even offer any DSIC approximation mechanisms for scenario this lecture. Instead, this lecture and the next consider some simpler scenarios and develop new tools that yield good DSIC approximation mechanisms for them. We'll then circle back to bidders with submodular valuations next week, wielding our expanded design toolbox.

Why is the question so difficult? After all, we design some good DSIC approximation mechanisms last quarter, albeit in a single-parameter setting. A key step was to characterize the relevant design space — the allocation rules that are *implementable*, meaning the rules that can be coupled with a suitable payment rule to yield a DSIC mechanism. We proved Myerson's Lemma (CS364A, Lecture #3), which includes the fact that the implementable allocation rules are precisely the monotone ones, meaning bidding higher (per unit of stuff) can only net a bidder more stuff, holding others' bids fixed. Myerson's Lemma reduces the problem of designing DSIC approximation mechanisms to a problem we could get our head around, namely designing an approximation algorithm that induces a monotone allocation rule. In CS364A (Lecture #4) we used the knapsack problem as our case study: the natural greedy algorithms yield monotone allocation rules; the standard fully-polynomial time approximation schemes do not, but can be tweaked to be monotone. For the knapsack problem,

†Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: `tim@cs.stanford.edu`.

we concluded that the DSIC guarantee is without loss of generality: there is a polynomial-time DSIC mechanism as good as the best-possible (assuming $P \neq NP$) polynomial-time algorithm for the underlying welfare-maximization problem.

To extend this paradigm to multi-parameter settings, like the welfare-maximization problem in combinatorial auctions that we're obsessed with in this course, we need a multi-parameter analog of Myerson's Lemma — a notion of allocation rule monotonicity that characterizes implementability. Because a multi-parameter bid deviation can be higher in some components and lower in others, the multi-parameter analog of monotonicity has to be somewhat complicated.

The good news is that there is such an analog: a multi-parameter allocation rule is implementable if and only if it is "cyclic monotone," a somewhat complicated definition that we'll return to later, when we need it. The bad news is that, in contrast to Myerson's Lemma, this characterization is rarely useful. This is true even in settings where the condition simplifies to "weak monotonicity," another definition we'll return to later. This first difficulty is, given an allocation rule, it can be difficult to check whether or not it satisfies these monotonicity conditions. The second issue, which will become more clear in the forthcoming lectures, is these conditions almost never hold — while many natural single-parameter allocation rules are monotone, cyclic monotone multi-parameter allocation rules seem to be needles in a haystack.[1] Summarizing, for single-parameter problems there is a successful paradigm of (i) characterize the design space of implementable allocation rules and (ii) optimize over this design space. This paradigm has had little success with multi-parameter problems.

An alternative approach to multi-parameter DSIC mechanism design is bottom-up — we start from the DSIC mechanisms that we know and love, and gradually generalize them to solve more and more problems. A majority of the multi-parameter success stories are of this type, and we'll survey some of them in the next few lectures.

Thus far, we've seen exactly one DSIC multi-parameter mechanism: the VCG mechanism. How can we generalize it and retain the DSIC guarantee? One obvious idea is apply the guiding principles of VCG to a different (non welfare-maximizing) allocation rule. That is, let $\mathbf{x}$ denote some allocation rule, such as a rule induced by one of our $\frac{1}{2}$-approximation algorithms for welfare maximization with submodular valuations. The natural analog of the VCG mechanism is to compute the allocation using $\mathbf{x}$ and the payments as the difference in the welfare of others caused by a bidder's participation:

$$p_i(\mathbf{b}) = \sum_{k \neq i} v_k(x_k(0, \mathbf{b}_{-i})) - \sum_{k \neq i} v_k(x_k(\mathbf{b}_{-i})),$$

where we define non-participation as bidding 0 for all items.

[1] If we go beyond combinatorial auctions to mechanism design problems with an abstract outcome space Ω, then Roberts Theorem [9] formalizes this suspicion in a strong way: there are no deterministic DSIC mechanisms other than minor variants of the VCG mechanisms. Analogs of this impossibility theorem hold in somewhat more restricted settings, as well [6]. With combinatorial auctions, however, there can be additional such mechanisms; see [1] for one example. The reason is the special "no externality" assumption in auctions: bidders care only about what they get and not what others get. The extent to which Roberts' Theorem holds in different multi-parameter domains remains poorly understood.

Unfortunately, this idea almost never works. Intuitively, the reason is that the VCG payments are designed to align the utility of a bidder with the welfare of the outcome (up to a constant term that the bidder cannot influence). Thus, if the latter is suboptimal and can be improved, then there should be an opportunity for some bidder to change its reported valuation and increase its utility.[2] This failed experiment provides further intuition that DSIC multi-parameter mechanisms are few and far between.

2 Maximal-in-Range (MIR) Mechanisms

Does plugging an approximation algorithm into the VCG mechanism *ever* work? One obvious sufficient (and almost necessary [5]) condition is pre-commit to a restricted subset Ω' of the outcomes Ω, before seeing the reported valuations, and then run the VCG mechanism as if Ω' was the original outcome space all along. That it, a *maximal-in-range (MIR)* allocation rule computes the welfare-maximizing outcome over the restricted set Ω' (its *range*) with respect to the reported valuations. This mechanism is equivalent to the VCG mechanism for ω' and hence is DSIC.

Why would this ever be useful? The hope is that there is a "sweet spot" Ω' that is small or well-structured enough to enable computationally efficient maximization, yet large enough to contain a near-optimal outcome for every valuation profile.

An approximation algorithm is almost never MIR by accident. Take the Kelso-Crawford auction, for instance, viewed as an approximation algorithm for bidders with submodular valuations. Its range is full, meaning that for every allocation $(S_1, \ldots, S_n)$ there is a valuation profile $\mathbf{v}$ such that the KC auction outputs that allocation. (Just take v_i strictly positive on subsets of S_i, with marginal value 0 for every item outside S_i.) This means that the KC auction is MIR only if it is always optimal (which it is not with submodular valuations, since Walrasian equilibria need not exist).

3 MIR Application: Multi-Unit Auctions

MIR mechanisms are occasionally powerful enough to prove interesting results; we next give one for multi-unit auctions with general valuations (cf., scenario #4).

Scenario #7:

- A set m *identical* items.
- Each bidder i has a private *marginal valuation* $\mu_i(j)$ for a jth item. Thus, bidder i's total valuation for ℓ units is $v_i(\ell) := \sum_{j=1}^{\ell} \mu_i(j)$.

[2]The gap in this reasoning is that the suboptimality of the allocation only immediately implies that a coordinated deviation by *many* bidders can increase the welfare of the computed allocation; it is not clear that some bid can increase the welfare via a *unilateral* deviation. For essentially all allocation rules of interest, though, this reasoning is correct. See [5] for a careful treatment.

- Valuations are *monotone*, meaning that the $\mu_i(j)$'s are non-negative.

Unlike scenario #4 and in the clinching auction, we do not assume that the valuations are downward sloping. For example, if $\mu_i(j) = a_i$ and $\mu_i(\ell) = 0$ for $\ell \neq j$, then bidder i is "single-minded," in that it wants j units and is willing to pay a_i for them, and is not interested in any fewer number of units. Welfare-maximization with such single-minded bidders corresponds to the Knapsack problem. The dynamic programming approach for the Knapsack problem can be extended to compute a welfare-maximizing allocation in scenario #7 in time polynomial in n and m (see Exercises). The VCG mechanism can therefore be implemented in time polynomial in n and m.

An MIR mechanism can be used to dramatically improve over the running time of the VCG mechanism, assuming valuations are treated as "black boxes," at the cost of a constant-factor loss in welfare.

Theorem 3.1 ([2]) *In scenario #7, there is an MIR mechanism that runs in time (including value queries) polynomial in n and $\log m$ and computes an allocation with welfare at least $\frac{1}{2}$ times the maximum possible.*

As an analogy, recall the dynamic programming algorithms for the Knapsack problem run in pseudo-polynomial time — polynomial if either the item sizes or the item values are integers written in unary. The polynomial dependence on m in the VCG mechanism can be thought of as pseudopolynomial time provided valuations are not given explicitly as input — since all m items are identical, only $\log m$ bits are required to describe how many there are. The polynomial dependence on $\log m$ in Theorem 3.1 can be thought of as "truly polynomial" in the input size $\approx n + \log m$.

Proof of Theorem 3.1: Assume that m is a multiple of n^2; this is without loss of generality, as we can round m up to the nearest multiple of n^2 and think of all bidders as having zero marginal value for the additional phantom goods.

We partition the m items into n^2 blocks of m/n^2 items each. We take the restricted range Ω' to be those allocations where all of the items in a block go to the same bidder — that is, allocations in which the number of items received by each bidder is a multiple of m/n^2. The MIR mechanism corresponding to this range can be implemented in time polynomial in n and $\log m$ using the dynamic programming algorithm mentioned above — it is effectively solving a multi-unit auction with n^2 goods. The dependence on $\log m$ is needed to describe quantities when asking the appropriate value queries.

It remains to prove the approximation guarantee. Fix valuations $\mathbf{v}$ and let $(S_1^*, \ldots, S_n^*)$ be an optimal allocation — since items are identical, we can think of each S_i^* as just an integer between 0 and m. Our goal is to exhibit an allocation $(S_1, \ldots, S_n)$ in Ω' with $\sum_{i=1}^n v_i(S_i) \geq \frac{1}{2} \sum_{i=1}^n v_i(S_i^*)$; since our MIR mechanism chooses the welfare-maximizing allocation in Ω', the guarantee follows.

There are two cases. For the first, assume that some bidder i contributes the lion's share of the optimal welfare, meaning $v_i(S_i^*) \geq \frac{1}{2} \sum_{k=1}^n v_k(S_k^*)$. Then, setting S_i to be all items and $S_k = \emptyset$ for all $k \neq i$ yields an allocation in Ω' that, by monotonicity of v_i, has welfare at least $\frac{1}{2} \sum_{k=1}^n v_k(S_k^*)$.

For the second case, assume without loss of generality that all m items are allocated in $(S_1^*, \ldots, S_n^*)$. Let i be a bidder with at least $\frac{m}{n}$ items — n blocks and change — in this allocation. Bidder i will be our sacrificial lamb: for $k \neq i$, we use i's items to round up k's allocation from S_k^* to the nearest multiple S_i of $\frac{m}{n^2}$. S_i is whatever i is left with after topping off all other bidders' allocations. By monotonicity of v_k for every $k \neq i$ we have

$$\sum_{k=1}^{n} v_k(S_k) \geq \sum_{k \neq i} v_k(S_k^*) \geq \tfrac{1}{2} \sum_{k=1}^{n} v_k(S_k^*),$$

where the second inequality follows from the assumption in this case that no single bidder contributes more than half of the welfare in the optimal allocation. ■

4 Maximal-in-Distributional-Range (MIDR) Mechanisms

MIR mechanisms have a few nice applications but, in general, are too limited to be of much use. What else can we do?

We next introduce a randomized variant of MIR mechanisms. Instead of committing up front to a set $\Omega' \subseteq \Omega$ of allocations, we commit to a set $\mathcal{D}$ of *distributions* over allocations Ω. For example, a distribution $D \in \mathcal{D}$ could have the form: randomize 50/50 between allocation ω_1 and allocation ω_2; allocate according to ω, except with 1% probability give every bidder nothing; or assign each item independently to a uniform at random bidder.

Once a set $\mathcal{D}$ of distributions is chosen, the corresponding *maximal-in-distributional-range (MIDR)* allocation rule is uniquely defined (up to tie-breaking): always choose the distribution that is best for the bidders, in the sense of maximizing expected welfare with respect to the reported valuations. Formally, the MIDR (randomized) allocation rule corresponding to $\mathcal{D}$ is:

1. given reported valuations $\mathbf{v}$, let D^* be the distribution that maximizes

$$\mathbf{E}_{\omega \sim D}\left[\sum_{i=1}^{n} v_i(\omega)\right] \tag{1}$$

   over all $D \in \mathcal{D}$.

2. Return an outcome samples at random from D^*.

Generally, the set $\mathcal{D}$ of distributions is chosen to be a compact subset of $\mathcal{R}^{\Omega}$, so that the maximum in (1) exists. MIR allocation rules are MIDR allocation rules in which each distribution $d \in \mathcal{D}$ is a point mass.

Before explaining the corresponding payment rule, let's ask: Why do this? The hope is that allowing distributions increases computational tractability, analogous to how relaxing an intractable integer program can yield a tractable linear program. That said, it is not clear a priori whether or not this idea is useful for any interesting mechanism design problems.[3]

An MIDR allocation rule $\mathbf{x}$ performs exact optimization with respect to distribution range $\mathcal{D}$, and this suggests that we should be able to couple it with a VCG-type payment rule to get a DSIC mechanism. Precisely, define the payment rule $\mathbf{p}$ on reported valuations $\mathbf{v}$ by

$$p_i(\mathbf{v}) = \max_{D \in \mathcal{D}} \mathbf{E}_{\omega \in D}\left[\sum_{k \neq i} v_i(\omega)\right] \mathbf{E}_{\omega \in D^*}\left[\sum_{k \neq i} v_i(\omega)\right], \tag{2}$$

where D^* is the expected welfare-maximizing distribution for the full set $\mathbf{v}$ of reported valuations. That is, bidder i pays the loss of expected welfare to others caused by its participation in the mechanism. Analogous to the VCG mechanism, every bidder i maximizes its expected utility $\mathbf{E}_{\omega \sim \mathbf{x}(\mathbf{v})}[v_i(\omega) - p_i(\mathbf{v})]$ by revealing its true valuation; see Lecture #7 from CS364A and the exercises for more details. The expectation is over the coin flips in the (randomized) allocation rule $\mathbf{x}$, and in claiming that the mechanism $(\mathbf{x}, \mathbf{p})$ is DSIC we are assuming that all bidders are risk-neutral and care only about their expected utility.[4] The payment rule in (2) is deterministic. The assumed risk-neutrality of bidders implies that randomized payment rules also work, provided the expected payment of a bidder conforms to the formula in (2). The exercises give some applications of this flexibility.[5]

Defining MIDR mechanism may feel like grasping at straws, a consequence of the paucity of multi-parameters DSIC mechanisms. This is partly true, but MIDR mechanisms have been applied creatively to a number of interesting problems. For example, we won't have time to talk about the polynomial-time (in n and $\log m$) $(1-\epsilon)$-approximate MIDR mechanism for multi-unit auctions with general monotone valuations (scenario #7 in Section 3), where $\epsilon > 0$ can be as small as inverse polynomial in n [3].[6] This improves over the $\frac{1}{2}$-approximate MIR mechanism we gave in Section 3. No better deterministic polynomial-time DSIC mechanism for the problem is known, and there is no better such mechanism that always allocates all m of the items [2]. Next week we'll design a polynomial-time MIDR mechanism with a constant-factor welfare guarantee for an interesting special case of submodular valuations (scenario #6). The next section gives a polynomial-time $(1-\epsilon)$-approximate MIDR mechanism with general valuations, provided there is at least a logarithmic supply of every item. All of these mechanisms are quite involved, as befits problems that seem to be right at the frontier of

[3]The analogy is also imperfect. Linear relaxations of integer programs include all integer solutions, and generally also fractionally solutions that are strictly better than any integer solution. In contrast, a distributions over allocations cannot be better than the best allocation in its support. Thus, if $\mathcal{D}$ includes all point masses corresponding to Ω, then optimizing over $\mathcal{D}$ is no more tractable than optimizing over Ω. Useful distributional ranges must exclude many or all point masses.

[4]It is interesting and relevant to consider bidders with other attitudes to risk, but as we'll see, the case of risk-neutral bidders is already hard enough for the complex allocation problems that we're studying.

[5]For example, there is a suitable (randomized) payment rule that invokes the allocation rule n times, as in the standard VCG mechanism. Also, for the MIDR mechanism we construct next lecture, randomized payments can be couple with the randomized allocation to ensure that every (truthful) bidder has non-negative utility with probability 1.

[6]That is, the mechanism is a DSIC FPTAS (for "fully polynomial-time approximation scheme).

tractability for DSIC constant-factor approximations.

5 MIDR Application: Supply-k Combinatorial Auctions

This section studies bidders with arbitrary valuation functions but assumes that there are multiple copies of every item.

Scenario #8:

- A set U of m non-identical items.
- k copies of each item.
- Each bidder i wants only one copy of an item, but has an arbitrary private valuation $v_i(S)$ for each bundle $S \subseteq U$.[7]
- The input model is that valuations are given as black boxes that support value and demand queries.

The welfare maximization gets easier as k grows. For example, if $k = n$, then we can just give each bidder its favorite bundle for free. We'll present strong positive results even for moderate values of k. The rest of this lecture consider the algorithmic problem of approximately maximizing welfare in polynomial time. The next lecture shows how to convert this approximation algorithm into a MIDR (and hence DSIC) mechanism with the same approximation guarantee.

The goal in this section is to prove the following theorem.

Theorem 5.1 ([8]) *If $k \geq \frac{c \log m}{\epsilon^2}$ for a sufficiently large constant c, then there is a randomized algorithm with expected polynomial running time that, with probability 1, outputs an allocation with welfare at least $1 - \epsilon$ times the maximum possible.*

The algorithm can be derandomized [7], although we won't explain the details here.

The algorithmic approach is randomized rounding. Recall from lecture #6 the following

[7]For the results in this section, we only need to assume that the v_i's are nonnegative; they don't even need to monotone.

linear programming (LP) relaxation of the welfare maximization problem:

$$(WM-LP) \quad \begin{aligned} \max \quad & \sum_{i=1}^{n} \sum_{S \subseteq U} v_i(S) y_{is} \\ \text{subject to:} \quad & \\ & \sum_{S \subseteq U} y_{iS} \le 1 && \text{for every } i \\ & \sum_{i=1}^{n} \sum_{S\,:\,j \in S} y_{iS} \le k && \text{for every } j \in U \\ & y_{iS} \ge 0 && \text{for every } i \text{ and } S \subseteq U. \end{aligned}$$

Since bidders only one want copy of each item, we can restrict to bundles S that are subsets of U and not worry about multiple copies of items going to the same bidder. The fact that the supply of each item is k is reflected in the right-hand side of the second set of constraints, which was equal to 1 back in Lecture #6.

Every feasible allocation induces an integer solution to (WM-LP) with objective function value equal to its welfare, so the optimal value OPT_{LP} of $(WM-LP)$ is at least as large as the maximum-possible welfare. In Lecture #6 we showed that, assuming valuations support value and demand queries, this linear program can be solved exactly in polynomial time. At the time we had in mind gross substitutes valuations, but never used that hypothesis in solving the LP, only in subsequently concluding that there is optimal solution that is also integral.[8]

The first step is to solve $(WM-LP)$ to obtain an optimal solution $\{y^*_{iS}\}$. Since the v_i's need not satisfy the gross substitutes condition, $\mathbf{y}^*$ need not be integral. Because of its fractional components, $\mathbf{y}^*$ does not correspond to a feasible allocation of the items. The next step in the algorithm is to "round" the fractional values in $\mathbf{y}^*$ to get a good allocation.

There are many ways to round linear programming solutions; here we use one of the simplest and most useful, "randomized rounding". We make an independent random selection for each bidder i, viewing the y^*_{iS}'s as probabilities: bidder i is assigned the bundle S with probability y^*_{iS}. Well not quite: to encourage the feasibility of resulting allocation, we assign bidder i the bundle non-empty bundle S with probability $(1 - \frac{\epsilon}{2}) \cdot y^*_{iS}$, and the empty bundle with any remaining probability. This is well defined because $\sum_{S \subseteq U} y^*_{iS} \le 1$.

[8]Actually, Lecture #6 only explained how to solve the *dual* $(WM-D)$ of $(WM-LP)$ in polynomial time. When (WM-LP) has an integer optimal solution, it can then be reconstructed by a self-reducibility argument. Here, where $(WM-LP)$ need not have an optimal integer solution, we use the following trick instead. Let $\mathcal{C}$ be the set of constraints generated by the ellipsoid algorithm's separation oracle when it is applied to $(WM-D)$. Since the algorithm terminates in polynomial time, $\mathcal{C}$ has polynomial size. The "reduced dual" $(WM-D-red)$, which by definition is only bound by the constraints in $\mathcal{C}$, has the same optimal solution as $(WM-D)$ and has polynomial time. By strong duality, the dual of $(WM-D-red)$ — the "reduced primal" that has only the polynomially decision variables corresponding to $\mathcal{C}$ — has the same optimal objective function value as $(WM-D-red)$, and hence of $(WM-D)$ and $(WM-LP)$. This reduced primal has polynomial size and can be solved efficiently by your favorite polynomial-time method (ellipsoid, interior-point methods, etc.).

To begin the analysis, let's consider the expected welfare of item assignment resulting from randomized rounding (ignoring feasibility issues). By linearity of expectation, this expected welfare is

$$\sum_{i=1}^{n} \mathbf{E}[v_i(S_i)] = \sum_{i=1}^{n} \sum_{S \subseteq U} v_i(S)\mathbf{Pr}[S_i = S] = \sum_{i=1}^{n} \sum_{S \subseteq U} v_i(S)(1-\frac{\epsilon}{2})y^*_{iS} = (1-\tfrac{\epsilon}{2}) \sum_{i=1}^{n} \sum_{S \subseteq U} v_i(S)y^*_{iS}.$$

That is, the expected welfare after randomized rounding is almost as large (up to $1 - \frac{\epsilon}{2}$) as the optimal LP value, which in turn is at least as large as maximum-possible welfare.

The bigger concern is feasibility — given that we're assigning bundles to bidders independently, with no coordination, whose to say that we're respecting the supply constraint of k for each item? To study this, let X_{ij} denote the indicator random variable for whether or not bidder i receives a bundle including the item j. We first observe that the expected number $\sum_{i=1}^{n} X_{ij}$ of copies of item j that get sold is reasonable:

$$\mathbf{E}\left[\sum_{i=1}^{n} X_{ij}\right] = \sum_{i=1}^{n} \sum_{S\,:\,jinS} \mathbf{Pr}[i \text{ gets } S] = (1-\tfrac{\epsilon}{2}) \sum_{i=1}^{n} \sum_{S\,:\,j \in S} y^*_{iS} \le (1-\tfrac{\epsilon}{2}))k,$$

where the first equality follows from linearity of expectation and the inequality follows from the feasibility of $\mathbf{y}^*$ for $(WM - LP)$.

To bound the probability $\sum_{i=1}^{n} X_{ij}$ is significantly larger than its expectation, we recall the following Chernoff bound (e.g., [4, ???]).

Chernoff Bound: If $Y_1, Y_2, \ldots, Y_r$ are independent random variables with range in $[0, 1]$, $\mu = \mathbf{E}[\sum_{ell=1}^{r} Y_\ell]$, and $\delta \in (0, 1]$, then

$$\mathbf{Pr}\left[\sum_{\ell=1}^{r} Y_\ell > (1+\delta)\mu\right] \le e^{-\mu\delta^2/3}. \tag{3}$$

The key point in (3) is that the deviation bound on the right-hand side is decreasing exponentially with the mean μ — this is the source of the $\Omega(\epsilon^{-2} \log m)$ bound for the supply k.

Fix $c > 0$. Applying the Chernoff bound (3), to the sum of the Y_{ij}'s for a fixed item j, with $k \ge c_0\epsilon^{-2} \log m$ for a sufficiently constant c_0 (that depends on c), we obtain

$$\begin{aligned} \mathbf{Pr}\left[\sum_{i=1}^{n} X_{ij} > k\right] &\le \exp\{-\frac{k}{3}\left(\frac{\epsilon}{1-\epsilon}\right)^2\} \\ &\le e^{-c \log m} \qquad \text{since } k = \Omega(\epsilon^{-2} \log m) \\ &= \tfrac{1}{m^c}. \end{aligned}$$

Taking a Union Bound over the m choices of j, we find that randomized rounding yields a feasible allocation with probability at least $1 - \frac{1}{m^{c-1}}$.

Summarizing, provided $\Omega(\epsilon^{-2} \log m)$, randomized rounding yields an allocation with expected welfare $(1 - \frac{\epsilon}{2})$ times the optimal value to $(WM - LP)$ and that is feasible with high probability. An averaging argument and the Union Bound imply that, with probability at least ϵ, randomized rounding yields a "success" — an allocation that is both feasible and that has welfare at least $(1 - \epsilon)$ times the optimal value to $(WM - LP)$.[9] A polynomial number of independent trials yields a success with all but an exponentially small probability. In this very unlikely event, we can just solve the problem in exponential time (by brute force or dynamic programming) — this contributes essentially nothing to the expected running time of the algorithm. The expected running time of the resulting algorithm is polynomial (in n, m, and the number of bits in the valuations), and it returns an allocation with welfare at least $(1 - \epsilon)$ of the optimal value of $(WM - LP)$ with probability 1. This concludes the proof of Theorem 5.1.

References

[1] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 72–87, 2003.

[2] S. Dobzinski and N. Nisan. Mechanisms for multi-unit auctions. *Journal of Artificial Intelligence Research*, 37:85–98, 2010.

[3] S. Dughmi and S. Dobzinski. On the power of randomization in algorithmic mechanism design. *SIAM Journal on Computing*, 42(6):2287–2304, 2013.

[4] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1996.

[5] N. Nisan and A. Ronen. Computationally feasible vcg mechanisms. *Journal of Artificial Intelligence Research*, 29:19–47, 2007.

[6] C. H. Papadimitriou, M. Schapira, and Y. Singer. On the hardness of being truthful. In *Proceedings of the 49th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 250–259, 2008.

[7] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *Journal of Computer and System Sciences*, 37(2):130–143, 1988.

[9] The expected welfare is at least $(1 - \frac{\epsilon}{2})$ times the optimal value of $(WM - LP)$. Infeasible solutions, which occur only a $1/m^{c-1}$ fraction of the time, contribute at most n times the optimal value of $(WM - LP)$ — the most they can contribute is by giving each bidder its favorite bundle, and giving just one bidder its favorite bundle is a feasible solution to $(WM - LP)$. Feasible solutions contribute at most the optimal value of $(WM - LP)$ to this expectation. As long as $n \ll m^{c-1}$, this implies that the probability that randomized rounding outputs an allocation that is both feasible and has welfare at least $(1 - \epsilon)$ times the optimal value of $(WM - LP)$ is at least ϵ.

[8] P. Raghavan and C. D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[9] K. Roberts. The characterization of implementable choice rules. In J.-J. Laffont, editor, *Papers presented at the 1st European Summer Workshop of the Econometric Society*, pages 321–349. North-Holland, 1979.