CS364B: Frontiers in Mechanism Design
Lecture #9: MIDR Mechanisms via Scaling Algorithms*

Tim Roughgarden†

February 5, 2014

1 Recap: Welfare Maximization with Logarithmic Supply

Last lecture we introduced the problem of welfare maximization when there are many copies of every item.

Scenario #8:

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

Last lecture we discussed the purely algorithmic problem of approximately maximizing welfare in polynomial time, deferring incentive issues to the present lecture. Our main algorithmic result was the following strong approximation guarantee.

Theorem 1.1 *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 a feasible allocation with welfare at least $1 - \epsilon$ times the maximum possible.*

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

To recap the proof: we solved the LP relaxation of the welfare-maximization problem (using the ellipsoid algorithm, and demand queries to implement the dual separation oracle) to obtain an optimal LP solution $\mathbf{y}^*$; we then repeatedly applied randomized rounding to $(1-\frac{\epsilon}{2})\mathbf{y}^*$ until it produced a feasible solution with welfare at least $(1-\epsilon)$ times the objective function value of $\mathbf{y}^*$. Linearity of expectation and an averaging argument show that randomized rounding is likely to produce a solution with good welfare; Chernoff bounds and our scaling by $(1-\frac{\epsilon}{2})$ imply that it is likely to produce a feasible allocation. This argument did not even require the valuations v_i to be monotone (just nonnegative).

The same rounding algorithm can be applied to any feasible solution $\mathbf{y}$ — our proof never used the fact that $\mathbf{y}^*$ was the optimal solution to the linear program.

Theorem 1.2 *Under the assumptions of Theorem 1.1, there is a randomized algorithm with expected polynomial running time that takes as input a feasible LP solution* $\mathbf{y}$ *and, with probability 1, outputs a feasible allocation* $(S_1, S_2, \ldots, S_n)$ *with welfare* $\sum_{i=1}^n v_i(S_i)$ *at least* $(1-\epsilon)\sum_{i=1}^n \sum_{S \subseteq U} v_i(S) y_{iS}^*$.

The rounding algorithm can be derandomized without affecting the guarantees in Theorem 1.1 and 1.2. This argument is not unduly difficult but it is outside the scope of these notes; see [2] for details.

2 Recap: The Search for DSIC Multi-Parameter Mechanisms

Let's also recall where we are in the bigger picture of the course. Two lectures ago we took a seemingly small step beyond gross substitutes — our final tractable special case — to submodular valuations, for which welfare-maximization is NP-hard. The good news is that submodular valuations are still well-structured enough to admit good approximation algorithms, with the Kelso-Crawford auction providing a $\frac{1}{2}$-approximation. The question was then how to extend such good approximation algorithms into good polynomial-time DSIC approximation mechanisms. This is a challenging problem because there seem to be very few DSIC multi-parameter mechanisms (useful or otherwise).

Our approach is bottom up, in the sense that we're gradually expanding our portfolio of DSIC mechanism, and Scenario #8 is a good challenge problem for this purpose. The starting point is the VCG mechanism. What else can we do? Our first baby step was to consider maximal-in-range (MIR) mechanisms, where the mechanism's range is restricted up front (independent of the valuation profile), and then the VCG mechanism is applied to the restricted range. We saw an application of MIR mechanisms last lecture, with a DSIC 2-approximation mechanism for welfare-maximization in multi-unit auctions. This mechanism applies even when bidders' valuations are not downward-sloping and runs in time polynomial in n and $\log m$ (cf., the clinching auction of Lecture #4). The idea was to bundle the m items into $\approx \frac{m}{n^2}$ blocks of $\approx n^2$ items each, and run the VCG mechanism over the restricted range that only allocates items in blocks.

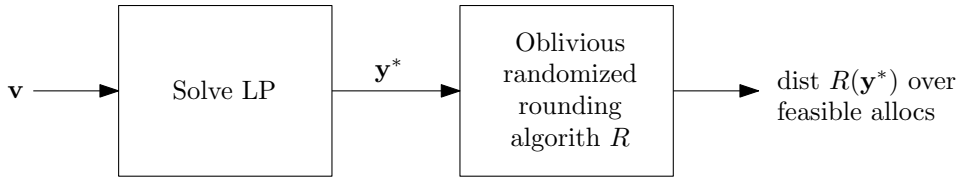

Figure 1: An algorithm for randomized rounding that is created by composition of a *relaxation* and a *rounding* algorithm.

The limited applicability of MIR mechanisms motivated the definition of *maximal-in-distributional-range (MIDR)* mechanisms. Recall from last lecture that an MIDR allocation rule fixes up front a compact set $\mathcal{D}$ of distributions over outcomes and works as follows:

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^*.

Recall also that coupling an MIDR allocation rule with an analog of VCG payments yields a DSIC mechanism (assuming the bidders are risk neutral and care only about their expected utility).

MIDR allocation rules are obviously more general than MIR rules, but are they useful? The MIDR condition is still require restrictive: hoping that your favorite allocation rule happens to be MIDR is delusional, so rules must be explicitly design to meet the requirement. What kind of algorithms seem closest to MIDR rules? Randomized rounding algorithms, like the one seen last lecture, share much of the spirit of MIDR allocation rules. We exploit this connection in this lecture and the next.

3 Scaling Algorithms

We can think of a randomized rounding algorithm as the composition of two algorithms (Figure 1). The first algorithm takes as input a valuation profile $\mathbf{v}$, perhaps as black boxes that support various queries, and returns a fractional allocation $\mathbf{y}^*$, such as the optimal solution to a LP relaxation with an objective function that depends on the provided valuations (like welfare). The second algorithm takes $\mathbf{y}^*$ as input and produces a random feasible allocation, distributed according to some distribution $R(\mathbf{y}^*)$.[1] We call these the *relaxation* and *rounding* algorithms, respectively. The randomized rounding algorithm we designed last lecture for

[1]We are assuming that the algorithm R depends only on $\mathbf{y}^*$ and does not depend directly on $\mathbf{v}$. Such *oblivious* randomized rounding procedures are the ones that are useful in DSIC mechanism design.

Scenario #8 provides a concrete example of this more abstract viewpoint: the relaxation step maximizes welfare over all fractionally feasible allocations, while the rounding step randomly generates a feasible allocation from a given fractional one. In these notes we consider only relaxation algorithms that optimally solve a relaxation of the welfare-maximization problem.

A randomized rounding algorithm of the above from induces a randomized allocation rule $\mathbf{x}(\mathbf{v})$, where the output is a random sample from the distribution $R(\mathbf{y}^*)$ induced the output $\mathbf{y}^*$ of the relaxation algorithm (given input $\mathbf{v}$). Is such a rule MIDR? Not necessarily. The issue that a rule if MIDR only if its maximizes expected welfare over all distributions in its range. While the relaxation algorithm maximizes welfare over fractional solutions $\mathbf{y}$, this optimality guarantee need not be preserved by the rounding subroutine. Indeed, since $R(\mathbf{y}^*)$ is oblivious to the profile $\mathbf{v}$ that generated $\mathbf{y}^*$, the requirement translates to the distribution $R(\mathbf{y}^*)$ being simultaneously optimal (among distributions in the range) for every profile $\mathbf{v}$ that generates $\mathbf{y}^*$. Since many different profiles $\mathbf{v}$ generally lead to the same optimal fractional solution $\mathbf{y}^*$ — just as many different linear objective functions are maximized by the same corner of a polytope — this is an intimidating constraint. See the Exercises for a concrete example.

Can we think of any special types of rounding algorithms R for which optimizing explicitly over fractional solutions $\mathbf{y}$ optimizes implicitly over the resulting distributions $R(\mathbf{y})$? That is, could there be a optimality-preserving procedure for compiling fractional solutions into distributions over integer solutions? For the welfare-maximization linear relaxation introduced last lecture (with x_{iS}'s as variables), a simple sufficient condition is the following.

Definition 3.1 For $\alpha \in [0,1]$, an oblivious rounding algorithm R for the welfare-maximization linear program of Lecture #6 is an *α-scaling algorithm* if for every input $\mathbf{y}$, every bidder i, and every bundle $S \subseteq U$,

$$\mathbf{Pr}_{R(\mathbf{y})}[i \text{ gets S}] = \alpha \cdot y_{iS}.$$

The key property of an α-scaling algorithm is that, no matter what the valuation profile $\mathbf{v}$ is, the expected welfare of the algorithm's output is exactly α times the welfare of the fractional allocation $\mathbf{y}$. That is, for every $\mathbf{v}$ and $\mathbf{y}$,

$$\mathbf{E}_{\omega \sim R(\mathbf{y})}\left[\sum_{i=1}^{n} v_i(\omega)\right] = \sum_{i=1}^{n}\sum_{S \subseteq U} v_i(S)\mathbf{Pr}_{R(\mathbf{y})}[i \text{ gets } S] = \alpha \cdot \sum_{i=1}^{n}\sum_{S \subseteq U} v_i(S) y_{iS}, \tag{2}$$

where the equalities follow from linearity of expectation and the α-scaling condition, respectively. Thus, maximizing the left-hand side (over $\mathbf{y}$) is the same problem as maximizing the right-hand side (over $\mathbf{y}$).

Equation (2) immediately implies that composing a relaxation algorithm with a α-scaling algorithm yields a MIDR rule. Formally, the range $\mathcal{D}$ of the allocation rule is a subset of $\{R(\mathbf{y}) : \mathbf{y} \in Y\}$, where Y denotes the feasible region of the LP relaxation. By definition, for every valuation profile $\mathbf{v}$, the relaxation algorithm computes the feasible solution $\mathbf{y}^* \in Y$ that maximizes the right-hand side of (2). Equation (2) implies that, for every valuation profile $\mathbf{v}$, $R(\mathbf{y}^*)$ maximizes $R(\mathbf{y})$ over $\mathbf{y} \in Y$ and hence over the rule's distributional range.

An α-scaling algorithm provides an approximation guarantee in addition to a DSIC guarantee: by the definition of a relaxation algorithm, the right-hand side of (2), and hence the expected welfare of the allocation returned by an α-scaling algorithm, is at least α times the optimal welfare.

4 Non-Scaling Algorithms

Consider the following relaxed version of Definition 3.1: for every input $\mathbf{y}$, every bidder i, and every bundle $S \subseteq U$,

$$\mathbf{Pr}_{R(\mathbf{y})}[i \text{ gets S}] \geq \alpha \cdot y_{iS}. \tag{3}$$

Observe that this weaker condition remains sufficient to imply a welfare approximation guarantee of α. This issue is that composing such a rounding algorithm R with a relaxation algorithm need not produce an MIDR allocation rule. Intuitively, bidders might misreport valuations to guide the relaxation algorithm to a fractional solution that the rounding algorithm has an unusually easy time with. To be concrete, suppose $\alpha = .8$ and consider two valuation profiles $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$. Suppose $R(\mathbf{y}^{(1)})$ and $R(\mathbf{y}^{(2)})$ are two distributions in the range of the allocation rule, with $\mathbf{y}^{(1)}$ optimizing $\sum_{i=1}^{n}\sum_{S\subseteq U} v^{(1)}(S)y_{iS}$ over $\mathbf{y} \in Y$ but with $\sum_{i=1}^{n}\sum_{S\subseteq U} v^{(1)}(S)y^{(2)}_{iS}$ only slightly less than $\sum_{i=1}^{n}\sum_{S\subseteq U} v^{(1)}(S)y^{(1)}_{iS}$. Suppose the rounding algorithm R satisfies (3) with $\alpha = .8$ for $\mathbf{y}^{(1)}$ and with $\alpha = .9$ for $\mathbf{y}^{(2)}$. Then the expected welfare under $R(\mathbf{y}^{(2)})$ is higher than under the computed distribution $R(\mathbf{y}^{(1)})$, and this violates the MIDR condition.

Let's return to Scenario #8. Recall from Theorem 1.1 that we have a randomized algorithm for welfare-maximization that, with probability 1, outputs a feasible allocation with welfare at least $1-\epsilon$ times the maximum possible. Moreover, the rounding algorithm sure seems like a scaling algorithm, with $\alpha = 1-\frac{\epsilon}{2}$, since each random trial of the algorithm picks a bundle S for bidder i according to the probability distribution induced by $\{(1-\frac{\epsilon}{2})y_{iS}\}_{S\subseteq U}$. There is a subtle issue, however: the distribution over feasible allocations induced by the rounding algorithm is *conditioned on a successful trial*. That is, if we ignore our imposed constraints that the computed allocation should be feasible and have welfare at least $(1-\epsilon)$ times that of the fractional solution $\mathbf{y}^*$, then $\mathbf{Pr}[i \text{ gets } S]$ is indeed $(1-\frac{\epsilon}{2})y^*_{iS}$ for every i and S. Unfortunately, conditioning on the event that these constraints are met, warps the distribution in a hard-to-understand way. For an informal example, consider a fractional LP solution $\mathbf{y}$ in which $y_{iS} > 0$ for a big bundle S. It's possible that whenever i actually receives the big bundle S, it blocks many other bidders and thus it becomes less likely that the rest of the allocation will both be feasible and have high welfare. In such a scenario, the probability that i gets S conditioned on a successful random trial would be lower than the unconditional probability of this event.

5 Computing Good Scaling Algorithms

We'll salvage an MIDR allocation rule for Scenario #8 by following the novel approach of Lavi and Swamy [1]: we use the *existence* of a $(1-\epsilon)$-approximate (non-MIDR) randomized rounding algorithm to compute, in polynomial time, a $(1-\epsilon)$-approximate allocation rule that *is* MIDR. We'll accomplish this through a general technique for "smoothing out" the error in an approximation algorithm, whose applications are not limited to Scenario #8.

Theorem 5.1 ([1]) *In scenario #8 with $k \geq \frac{c \log m}{\epsilon^2}$ for a sufficiently large constant c, there is a $(1-\epsilon)$-scaling algorithm that runs in polynomial time.*

The next corollary follows immediately from Theorem 5.1 and the discussion in Section 3.

Corollary 5.2 *In scenario #8 with $k \geq \frac{c \log m}{\epsilon^2}$ for a sufficiently large constant c, there is a $(1-\epsilon)$-approximate DSIC mechanism that runs in polynomial time.*

As in all our discussion of scenario #8, Corollary 5.2 assumes the provided valuations support computationally efficient demand queries. It offers one of the most compelling DSIC guarantees known for combinatorial auctions — the valuation class is very general, the approximation factor of $(1-\epsilon)$ is excellent and just as good as the state-of-the-art approximation guarantees for polynomial-time algorithms (without a DSIC constraint). The primary drawback — in addition to the logarithmic supply requirement — is that the mechanism is quite complicated. We covered the non-trivial ingredients needed for a good approximation algorithm last lecture; the next section covers the additional ideas needed for a good DSIC approximation mechanism.

6 Proof of Theorem 5.1

We build on two tools developed last lecture. First, we assume that, given the valuations $\mathbf{v}$, we can compute the welfare-maximizing fractional allocation $\mathbf{y}^*$ in polynomial time. Second, we'll use the rounding algorithm of Theorem 1.2 as a subroutine; for simplicity, we use the derandomized variant from [2].

The high-level idea is to use linear programming (again!) to compute the output distribution $R(\mathbf{y})$ of a $(1-\epsilon)$-scaling algorithm given the input $\mathbf{y}$. In more detail, let $\mathbf{y}^*$ denote a possible output of the relaxation algorithm. Since the relaxation algorithm runs in polynomial time (in n, m, and number of bits needed to describe a valuation) and $\mathbf{y}^*$ is its explicit output, $\mathbf{y}^*$'s description has polynomial length. It follows that the support size of $\mathbf{y}^*$ has polynomial size — that is, the cardinality of $I = \{(i,S) : y^*_{is} > 0\}$ is polynomial. By contrast, the total number of variables in exponential in m.[2]

[2]If you prefer, here's a more formal argument. Since $\mathbf{y}^*$ maximizes welfare with respect to some valuations $\mathbf{v}$ over the polytope Y of fractional allocations, and the welfare objective function is linear in $\mathbf{y}$, we can assume $\mathbf{y}^*$ is a vertex of Y. The dimension of Y is $n(2^m-1)$, so at a vertex of Y at least $n(2^m-1)$ of its constraints are satisfied with equality. Y has only $n+m$ constraints other than the nonnegativity constraints, so all but at most $n+m$ of the nonnegativity constraints hold with equality. Thus, the support size of $\mathbf{y}^*$ is at most

We next observe that the allowable distributions of a $(1-\epsilon)$-scaling algorithm on the input $\mathbf{y}^*$ can be encoded as a system of linear equations inequalities. Let Ω denote the allocations in which each bidder i receives a bundle S_i with either $S_i = \emptyset$ or with $(i, S_i) \in I$. For $(i, S) \in I$, let $\Omega_{iS} \subseteq \Omega$ denote the allocations in Ω where bidder i gets the bundle S. Consider the following linear system:

$$(LP2) \qquad \begin{aligned} &\sum_{\omega \in \Omega_{iS}} \lambda_\omega = (1-\epsilon) y^*_{iS} && \text{for every } (i,S) \in I \\ &\sum_{\omega \in \Omega} \lambda_\omega = 1 \\ &\lambda_\omega \geq 0 && \text{for every } \omega \in \Omega. \end{aligned}$$

The feasible solutions to (LP2) correspond to the distributions D over the allocations in Ω that satisfy $\mathbf{Pr}_{\omega \in D}[i \text{ gets } S] = (1-\epsilon) \cdot y^*_{iS}$ for every bidder i and non-empty bundle S. (Observe this holds also for the pairs $(i, S) \notin I$, with $y^*_{iS} = 0$.) Our next task is show that (LP2) is feasible and that we can compute a feasible solution in polynomial time. We'll ultimately want to sample from this distribution, so we're also hoping to compute one with polynomial support.

The linear system (LP2) has an exponential number $|\Omega|$ of variables, but a polynomial number $|I| + 1$ for constraints other than the non-negativity constraints. Thus, the dual linear system has a polynomial number of variables, and is a candidate for the application of the ellipsoid method. We next develop intuition for this dual linear system.

How could I convince you that the system (LP2) is infeasible? Suppose I presented you with *pseudo-valuations* $z_i(S)$ for each $(i, S) \in I$, which need not be nonnegative or monotone but which satisfy

$$\underbrace{\max_{(S_1, S_2, \ldots, S_n) \in \Omega} \sum_{i=1}^{n} z_i(S_i)}_{:= Z^*} < (1-\epsilon) \sum_{(i,S) \in I} z_i(S) y^*_{iS}. \tag{4}$$

Then, multiplying each constraint in (LP2) corresponding to $(i, S) \in I$ by $z_i(S)$ and the second constraint by $-Z^*$ and adding the results yields N equations in which the right-hand side is the scalar

$$(1-\epsilon) \sum_{(i,S) \in I} z_i(S) y^*_{iS} - Z^* > 0.$$

Also by (4), the coefficient of each (nonnegative) variable λ_ω with $\omega = (S_1, S_2, \ldots, S_n) \in \Omega$ in the left-hand side of this equation is $\sum_{i=1}^{n} z_i(S_i) - Z^* \leq 0$. This inequality cannot be satisfied — the left-hand side is non-positive no matter that the λ_ω's are, while the right-hand side is strictly positive — which certifies the infeasibility of the linear system (LP2).

The linear system dual to (LP2) is simply the set of proofs of infeasibility of the form (4):

$n + m$.

$$
\begin{array}{lll}
(D2) & \displaystyle\sum_{i=1}^{n} z_i(S_i) \leq Z^* & \text{for every } \omega = (S_1, S_2, \ldots, S_n) \in \Omega \\
& \displaystyle(1-\epsilon) \sum_{(i,S)\in I} z_i(S) y^*_{iS} > Z^* & \\
& Z^*, z_i(S) \text{ unrestricted} & \text{for every } (i,S) \in I.
\end{array}
$$

By definition, if (D2) is feasible, then (LP2) is infeasible. Strong duality — also known as "Farkas's Lemma" of the "Theorem of the Alternatives" for the feasibility problems we're currently considering — implies the converse, so exactly one of the systems (LP2), (D2) is feasible.[3]

We now show that (D2) is infeasible, and hence (LP2) is feasible. For the proof, consider running the ellipsoid method on the dual linear system (D2). It has a polynomial number $|I| + 1$ of variables, so we can solve it in polynomial time — meaning we either exhibit a feasible solution or conclude that no feasible solution exists — as long as we can provide a polynomial-time separation oracle. This separation oracle takes as input an alleged feasible solution $(Z^*, \mathbf{z})$. The last constraint can be checked directly in polynomial time; if it is violated, we return it as the violated constraint.

If the last constraint is not violated, then

$$
(1-\epsilon) \sum_{(i,S)\in I} z_i(S) y^*_{iS} > Z^* \tag{5}
$$

and we proceed as follows. Recall the $z_i(S)$'s need not be nonnegative, so define $\hat{z}_i(S) = \max\{0, z_i(S)\}$. Since $\mathbf{y} \geq 0$, the inequality (5) continues to hold for $\hat{\mathbf{z}}$. Interpret the $\hat{z}_i(S)$'s as valuations (with $\hat{z}_i(S) = 0$ whenever $(i,S) \notin I$), and invoke the (derandomized version of the) algorithm in Theorem 1.2 on $\mathbf{y}^*$ to produce a feasible allocation $(S_1, S_2, \ldots, S_n)$ with

$$
\sum_{i=1}^{n} \hat{z}_i(S_i) \geq (1-\epsilon) \sum_{i=1}^{n} \sum_{S \subseteq U} \hat{z}_i(S) y^*_{iS} > Z^*. \tag{6}
$$

Recall that modulo the derandomization, nothing complicated is going on here: in each independent random trial, we independently give each bidder a bundle chosen at random from the distribution $\{(1-\frac{\epsilon}{2}) y^*_{iS}\}_{S \subseteq U}$, and we stop the first time a random trial yields a feasible allocation that satisfies (6).

Now obtain $(T_1, T_2, \ldots, T_n)$ from $(S_1, S_2, \ldots, S_n)$ by setting $T_i = \emptyset$ if $z_i(S_i) < 0$ (and hence $\hat{z}_i(S_i) = 0$) and $T_i = S_i$ otherwise. We then have

$$
\sum_{i=1}^{n} z_i(T_i) = \sum_{i=1}^{n} \hat{z}_i(S_i) > Z^*,
$$

[3] A proof of Farkas's Lemmas is beyond the scope of these notes; hopefully you've seen it in an optimization class.

which shows that the alleged feasible solution $(Z^*, \mathbf{z})$ violated the constraint of (D2) corresponding to the allocation $(T_1, T_2, \ldots, T_n)$.

Summarizing, we've shown that, no matter which alleged solution $\mathbf{z}$ we're given, we can find a violated inequality in polynomial time. This implies that (D2) has no feasible solution, and hence (LP2) is feasible.

To compute efficiently a feasible solution of (LP2), we proceed as follows. First, we run the ellipsoid method on the dual linear system (D2), generating violated inequalities as needed. After a polynomial number of iterations, the ellipsoid method correctly concludes that the system (D2) is infeasible. Each of the polynomially many violated inequalities corresponds to an allocation. Let $\mathcal{C}$ denote the family of generated inequalities and $\Omega^* = \{\omega_1, \ldots, \omega_\ell\}$ the corresponding allocations.

The "reduced dual" (RD), which has only the constraints in $\mathcal{C}$, is already infeasible (otherwise the ellipsoid method could terminate incorrectly given the input (RD)). The corresponding "reduced primal" (RP) is simply the original linear system (LP2), except with variables corresponding only to Ω^*. Strong duality implies that (RP) has a feasible solution. Since (RP) has a polynomial number of variables and constraints, we can solve it efficiently using any polynomial-time linear programming algorithm (e.g., the ellipsoid method or an interior-point method).

7 Recap

Summarizing, here is the polynomial-time MIDR $(1 - \epsilon)$-approximate allocation rule that we've constructed for Scenario #8:

1. Given valuations as black-boxed that support value and demand queries, solve the linear programming relaxation of Lecture #8 using the ellipsoid method. Recall from Lecture #8 that the relevant separation oracle reduces to a demand query. Let $\mathbf{y}$ denote the optimal solution, and I the indices of the polynomially many non-zero variables in $\mathbf{y}$.

2. Apply the ellipsoid method to the dual linear system (D2); as argued above, the relevant separation oracle reduces to the algorithm in Theorem 1.2. Let Ω^* denote the polynomially many allocations that correspond to the violated constraints generated by the ellipsoid method en route to proving the infeasibility of (D2).

3. Solve the reduced primal (RP), with decision variables corresponding only to Ω^*, using any polynomial-time linear programming algorithm. Let $\lambda_1, \ldots, \lambda_\ell$ denote the computed feasible solution to (RP).

4. Return an allocation ω^* chosen at random from Ω^* according to the distribution $\lambda_1, \ldots, \lambda_\ell$.

References

[1] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. *Journal of the ACM*, 58(6), 2011. Article 25.

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