CPSC 536F: Algorithmic Game Theory
Lecture: MIDR Mechanisms And An Application To The Generalized Assignment Problem

Hu Fu

October 21, 2016

1 MIR and MIDR mechanisms

As we discussed in class, VCG mechanisms require optimizing the social welfare exactly, which is in general NP-hard. Replacing the welfare computations in a VCG mechanism by outputs of approximation algorithms, however, does not yield incentive compatible mechanisms.

One way to remedy this is to restrict the range of allocations in our computation of optimal social welfare. That is, if we restrict ourselves upfront to a certain set of allocations, over which we could perform exact optimization of social welfare efficiently, then using the welfare-maximizing allocations restricted to this set in a "VCG-based" mechanism would preserve the incentive compatibility. (Try showing this by yourself.)

Definition 1. A mechanism is *maximal-in-range* if there is a range of allocations $\mathcal{R}$ independent of the bidders' valuations, such that for all reported valuations $v_1, \ldots, v_n$, the mechanism returns an allocation $(S_1, \cdots, S_n)$ in $\arg\max_{(T_1,\cdots,T_n)\in\mathcal{R}} \sum_i v_i(T_i)$, and the payment for bidder i is equal to $\sum_{j\neq i} v_j(S_j) - \max_{(T_1,\cdots,T_n)\in\mathcal{R}} \sum_{j\neq i} v_j(T_j)$.

Proposition 1. *An MIR mechanism is DSIC.*

The crux of an MIR mechanism is a tradeoff in the design of the range $\mathcal{R}$: it should be, on the one hand, large and complex enough so that for any profile of valuations (maybe restricted to a certain class), there exists an allocation in $\mathcal{R}$ that gives a desired approximation for the optimal social welfare; on the other hand, $\mathcal{R}$ needs to be small or structured enough so that we have computational tools to optimize over it efficiently.

Example 1. Let the range $\mathcal{R}$ consist of n allocations, where the i-th allocation gives the whole bundle M to the i-th bidder, and all other bidders get nothing. Enumeration suffices to optimize over this range, and for all monotone valuations this trivially gives an n-approximation to the optimal social welfare.

Example 2. A 2-approximation MIR mechanism for multi-unit auctions (Dobzinski and Nisan, 2010) (see Section 3 of Tim Roughgarden's lecture notes).

There is no reason to restrict ourselves to consider only deterministic mechanisms. One may naturally generalize the definition of an MIR mechanism to a *maximal-in-distributional-range* mechanism, where elements in the range $\mathcal{R}$ are not allocations, but *distributions* over allocations. As long as the bidders are risk-neutral (recall our discussion from earlier lectures), this preserves the incentive compatibility of the mechanism.

Definition 2. A mechanism is *maximal-in-distributional-range* if there is a range of distributions over allocations, $\mathcal{R}$, independent of the bidders' valuations, such that for all reported valuations $v_1, \ldots, v_n$, the mechanism returns an allocation $(S_1, \cdots, S_n)$ drawn from a distribution $D^* \in \mathcal{R}$, where D^* is in

$$\arg\max_{D \in \mathcal{R}} \mathbf{E}_{(T_1, \cdots, T_n) \sim D} \left[\sum_i v_i(T_i) \right],$$

and the payment of bidder i is $\sum_{j \neq i} v_j(S'_j) - \sum_{j \neq i} v_j(S_j)$ where $(S'_1, \cdots, S'_n)$ is drawn from distribution $D^*_i \in \mathcal{R}$ and D^*_i is in

$$\arg\max_{D \in \mathcal{R}} \mathbf{E}_{(T_1, \cdots, T_n) \sim D} \left[\sum_{j \neq i} v_j(T_j) \right].$$

Proposition 2. *An MIDR mechanism is DSIC for risk-neutral bidders.*

2 Using LP relaxations to construct MIDR mechanisms

It may not be clear why extending the notion of MIR mechanisms to MIDR should be helpful, and therefore it may come surprising that several creative ways have been discovered to leverage the additional power of randomization allowed by MIDR mechanisms. In this lecture we will see one of these, an ingenious method proposed by Lavi and Swamy (2011), looking at LP relaxations and certain rounding schemes as a way to construct MIDR mechanisms.

Consider some high-dimensional representation of all feasible allocations. The convex body they span is then a polytope whose vertices are all integral. Let's call that polytope I. When the social welfare maximization problem is NP-hard, the number of its hyperfaces and the number of such vertices are both exponential. An LP relaxation is often formed by naming a different set of hyperplanes which "surround" the integral vertices. The polytope formed by these hyperplanes, denoted by P, contains all the original integral vertices and hence I, but is usually larger, and has its own vertices. The configuration LP we have seen is one such relaxation. It contains all vectors representating valid allocations, but has new vertices which do not correspond to valid allocations. Recall that the *integrality gap* of P is the largest ratio between the value of a linear objective realizable on a vertex of P and the value of the same objective realizable on a vertex of I.

Example 3. Consider three *single-minded* bidders and three items. Bidder 1 has positive value of 2 only for the bundle $\{1, 2\}$ and doesn't care for anything not containing both these items; bidder 2 is similar, but her desired bundle is $\{2, 3\}$, and bidder 3 desires only $\{1, 3\}$. It is obvious then that the optimal social welfare is 2, because we cannot satisfy more than one bidder. However, a valid solution to the configuration LP is to give $\{1, 2\}$ to bidder 1 with probability $\frac{1}{2}$, and $\{2, 3\}$ to bidder 2 with probability $\frac{1}{2}$, and $\{1, 3\}$ to bidder 3 with probability $\frac{1}{2}$. (Check that this satisfies the configuration LP.) This "fractional" allocation generates a social welfare of 3. (Note that the valuations here are not gross substitute!)

In general, the solution to an LP relaxation gives an upper bound to the problem we are originally interested in. An important class of approxamation algorithms try to turn these infeasible fractional solutions to feasible integral solutions, while losing as little as possible in the objective function. This approach is called *rounding*. In fact, this is one of the most important techniques in the modern toolbox for approximation algorithm design.

An observation by Lavi and Swamy (2011) is the following: if a rounding scheme works by shrinking an LP relaxation solution $\vec{s}$ by a fixed ratio α, and somehow showing that the shrunk solution $\vec{s}'$ must lie in the original integral polytope I, then by expressing $\vec{s}'$ as a convex decomposition of vertices of I, one can see the shrunk solution $\vec{s}'$ as a *distribution over allocations*. In other words, if one can show that the polytope P, after being shrunk by a factor of α to a smaller polytope P', is fully contained in the integral polytope I, then the smaller polytope P' may as well be seen as a *range of distributions over allocations*. Since we were doing exact optimization over the relaxed polytope P, the shrunk solution we return as a distribution over allocations is maximum in P', the fixed range of distributions over allocations. We then get an MIDR mechanism!

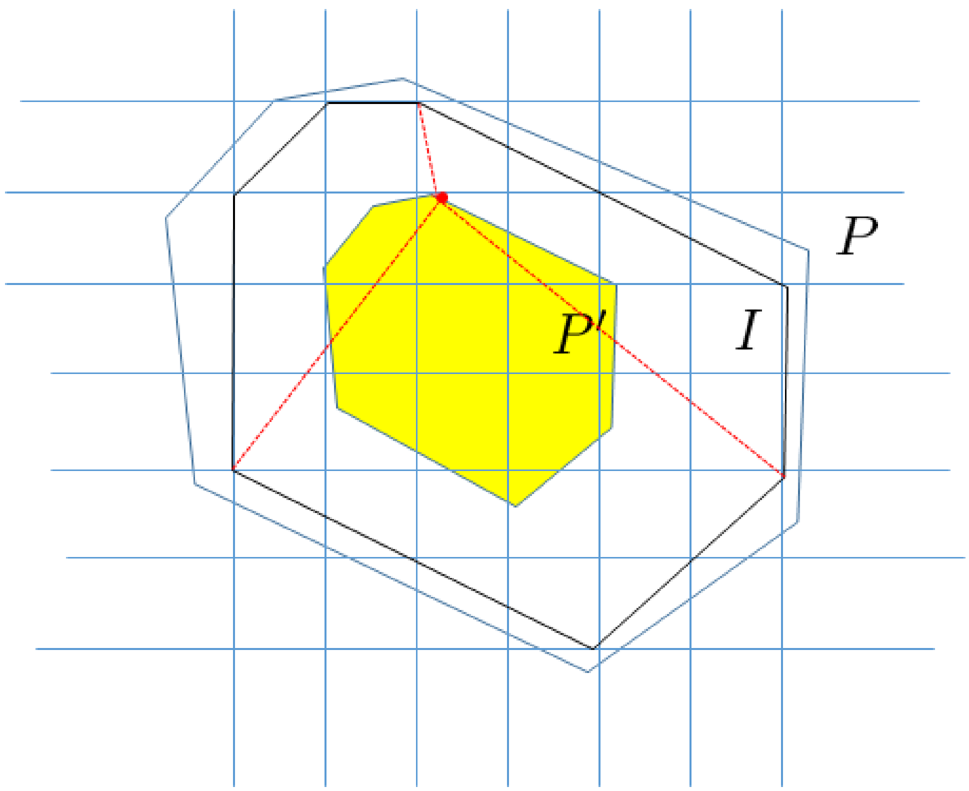

Figure 1: An over-simplified illustration of an LP relaxation, the shrunk polytope, and the convex decomposition in terms of vertices of the integral polytope.

There are a few caveats to this approach. First, MIDR mechanisms require that the range must be fixed before any valuation is seen. Therefore, the polytope P must not depend on any information privately held by the bidders. For instance, the configuration LP satisfies this condition. Second, in the rounding procedure, all solutions should be shrunk by the same factor. This is to guarantee that we have exact *maximum* in the range. Third, we need to be able to compute a valid convex decomposition of the shrunk solution in polynomial time. In fact, Lavi and Swamy (2011) showed a way to do this decomposition for the configuration LP via the ellipsoid method, where the separation oracle is given by an *integrality gap verifying* approximation algorithm. For the sake of time, we will not be able to cover this part of their result, and interested students are encouraged to consult Chapter 12.3 of the AGT book, or the original paper. Instead, we will in the next section look at a specific example, in which a famous approximation algorithm for scheduling developed in the 1990s (Shmoys and Tardos, 1993) applied via this framework to give a 2-approximation MIDR mechanism for the Generalized Assignment Problem (GAP).

3 A 2-approximation MIDR mechanism for the Generalized Assignment Problem (GAP)

In a generalized assignment problem, we have n bidders and m resources. Each bidder i has value $v_i(j)$ for resource j, but has only limited capacity to hold the resources. In particular, the capacity of bidder i is B_i, and each resource j will occupy c_{ij} from her capacity. Therefore, when allocated a bundle S of resources, her value is

$$\max_{T \subseteq S: \sum_{j \in T} c_{ij} \leq B_i} \sum_{j \in T} v_i(j).$$

Importantly, only the values $v_i(j)$'s are private information, whereas the bidders' capacities B_i's and the weights c_{ij}'s are all public information. Our goal is design an MIDR mechanism that maximizes the social welfare.

Remark 1. Here communicating the whole valuation function is easy (there are only m real numbers to report), whereas computing a value query (or a demand oracle) is NP-hard as it is a knapsack problem. It is easy to see then that the social welfare maximization problem is NP-hard also. In fact, the latter is strictly harder than the former. As we know, the knapsack problem has a pseudo-polynomial time algorithm, whereas the welfare maximization problem is hard even when the B_i's are polynomially bounded, which can be seen by a reduction from the bin-packing problem (Roughgarden and Talgam-Cohen, 2015). In this case, the value/demand oracle is said to be *weakly NP-hard*, whereas the social welfare maximization problem *strongly NP-hard*.

The main result we are to prove here, through the framework explained in the previous section, is the following theorem.

Theorem 1. *There is a polynomial time computable, MIDR mechanism that gives a* 2*-approximation for the generalized assignment problem.*

Let's first see that there is a natural LP relaxation for the problem. Let x_{ij} denote whether item j is allocated to bidder i. Consider the following LP:

$$\max \sum_{i,j} v_i(j) x_{ij} \tag{GAP}$$

$$\text{s.t.} \sum_i x_{ij} \leq 1, \quad \forall j \in M, \tag{1}$$

$$\sum_j c_{ij} x_{ij} \leq B_i, \quad \forall i \in N, \tag{2}$$

$$x_{ij} \geq 0, \forall i, j. \tag{3}$$

The first set of constraints prevent any item to be over allocated. The second set of constraints say no bidder gets unnecessary items once her capacity is hit. It is true that not all feasible allocations are enclosed in the polytope specified by these constraints, but once we take away from bidders the items that are rendered useless by the capacity constraints, we see that the remaining allocation does lie in the feasible region of (GAP).

It is important to note that the feasible region given by (GAP) does not depend on any private information held by the bidders.

As mentioned, the following rounding scheme is the one given by Shmoys and Tardos (1993), who studied the minimization version of the problem in the context of scheduling; Chekuri and Khanna (2000) pointed out the 2-approximation algorithm implicit in it for the maximization problem. The applicability of this algorithm to the design of MIDR mechanism via Lavi and Swamy was first observed by Dughmi and Ghosh (2010).

Our mechanism first solves the LP relaxation (GAP). Let x^* be the solution vector. We will find a convex composition of x^* into integral allocations, whose expected social welfare is exactly equal to $\frac{1}{2} \sum_{ij} v_i(j) x^*_{ij}$. We will use the following well-known theorem, whose proof can be found in many resources.

Theorem 2 (Birkhoff - von Neumann). *The matching polytope for a bipartite graph (L, R, E) defined by*

$$\sum_{i \in L : (i,j) \in E} x_{ij} \leq 1, \quad \forall j \in R, \tag{4}$$

$$\sum_{j \in R : (i,j) \in E} x_{ij} \leq 1, \quad \forall i \in L, \tag{5}$$

$$x_{ij} \geq 0, \quad \forall i \in L, j \in R \tag{6}$$

has integral vertices representing matchings. Moreover, a convex decomposition of a point in this polytope into its vertices can be found in polynomial time.

Remark 2. If $n = m$ and if we tighten all the inequalities in (4) and (5) into equalities, we can view the vector x_{ij}'s as an n by n matrix with nonnegative entries, whose rows and columns all sum to 1. Such matrices are called *doubly stochastic matrices*. A *permutation matrix* is a square matrix with $0, 1$ elements, and each row and each column of it has exactly one nonzero element (that is, 1). The Birkhoff-von Neumann then says that every doubly stochastic matrix can be written as a convex combination of permutation matrices. This is the usual way in which the theorem is stated.

Our attempt to decompose x^* will be to place it in a matching polytope and then invoke Birkhoff-von Neumann theorem. If we look at the constraints of (GAP) which x^*_{ij} respects, (1) already looks like one side of the matching constraints. On the other side, however, $\sum_j x^*_{ij}$ can certainly be more than 1, as a bidder is allowed to receive more than one item. The idea is to split each bidder into several nodes, such that each node is given only part of the x^*_{ij}'s, and the sum of each part does not exceed 1. In this form, we would have a feasible point in a matching polytope, and we can invoke the Birkhoff-von Neumann theorem to get a distribution of matchings. Given a matching, an allocation can then be read off: item j goes to bidder i if the item node is matched to one of the bidder nodes.

However, we need to do this carefully, because our ultimate goal is that the expected welfare should remain unchanged. As long as we respect the capacity constraints, everything is linear, the expectation of allocations being equal to x^* guarantess the expected social welfare being equal to the LP value of x^*. But once the allocation hits a budget B_i for any bidder i, the valuation is capped and we no longer have linearity — this would be so messy that we never want to run into the situation.

We now construct a bipartite graph using x^*. On the right hand side are m nodes corresponding to the items. On the left hand side, for each bidder i, we create $k_i = \lceil \sum_j x^*_{ij} \rceil$ nodes. Name them $L_{i1}, \cdots, L_{ik_i}$. Edges are added in the following way. For each bidder i, rank the items by her value in a non-increasing order. Without loss of generality, assume $c_{i1} \geq c_{i2} \geq \cdots \geq c_{im}$. We then add an edge connecting the first bidder i node, L_{i1}, with item 1 with weight $x'_{i_1 1} = x^*_{i1}$, another edge connecting L_{i1} with item 2 with weight $x'_{i_1 2} = x^*_{i2}$, etc., until the point where the total weight of the edges incident to L_{i1} hits 1. At that point, we say L_{i1} is saturated and we move on to L_{i2}. Suppose L_{i1} was saturated when we were adding item j, we let the edge connecting L_{i1} with item j carry a weight $x'_{i_1 j}$ such that the total weight of edges incident to L_{i1} is equal to 1, and whatever weight is left in x^*_{ij} (that is, $x^*_{ij} - x'_{i_1 j}$), we put on an edge connecting L_{i2} and item j. We then continue this with item $j+1$ and so on until L_{i2} gets saturated and we move on to the L_{i3}. We repeat this till all x^*_{ij}'s are distributed to the weights x''s in this bipartite graph. We do this for every bidder i.

Now by construction, the edge weights x' in this bipartite graph respect the constraints of the matching polytope, so we can readily apply Birkhoff-von Neumann theorem to get decomposition into matchings, except that we want to make sure that the capacity constraint for any bidder is not violated in the allocation.

Claim 1. *In any matching in the constructed bipartite graph, if bidder i's capacity is violated, then removing at most one item from her allocation would restore her capacity constraint.*

Proof. Suppose item j_k is matched to the node L_{ik}, for $k = 1, \cdots, k_i$. (If node L_{ik} is not matched, let j_k be $\perp$, a null item of capacity 0.) We show that $\sum_{k=2}^{k_i} c_{ij_k} \leq B_i$.

Let us number the items to make the presentation easier. Suppose items $1, 2, \cdots, \ell_1$ are connected to L_{i1}, and items $\ell_1, \ell_1 + 1, \ldots, \ell_2$ are connected to L_{i2}, and so on. First, recall that, from the construction, $\sum_{k=1}^{k_i} \sum_{j=\ell_{k-1}}^{\ell_k} c_{ij} x'_{kj} = \sum_j c_{ij} x^*_{ij} \leq B_i$. Secondly, recall that, because the items were ordered by their weights for bidder i, any item connected to the node L_{ik}, including item j_k, has weight at most that of the item ℓ_{k-1}, the lightest item incident to the previous bidder node. That is, $c_{ij_k} \leq c_{i\ell_{k-1}}$. Then for all $k < k_i$, $\sum_{j=\ell_{k-1}}^{\ell_k} x'_{i_k j} = 1$, and hence $c_{i\ell_k} \leq \sum_{j=\ell_{k-1}}^{\ell_k} x'_{i_k j} c_{ij}$.

Combining all these conditions,

$$\sum_{k=2}^{k_i} c_{ij_k} \leq \sum_{k=1}^{k_i-1} c_{i\ell_{k-1}} \leq \sum_{k=1}^{k_i-1} \sum_{j=\ell_{k-1}}^{\ell_k} c_{ij} x'_{i_k j} \leq \sum_j c_{ij} x^*_{ij} \leq B_i.$$

□

Given the claim, our final algorithm works as follows. Solve the linear program (GAP) to get a fractional solution x^*. Construct the bipartite graph and decompose x^* into a convex combination of matchings in the graph. Draw a matching from the distribution defined by this convex combination. With probability half, give each bidder i the item matched to the node L_{i1} in the matching (if L_{i1} is not matched, bidder i gets nothing in this case.) With the remaining probability half, give each bidder i the items matched to the nodes $L_{i2}, \cdots, L_{ik_i}$. As we have argued, all such allocations respect the capacity constraints, and the expected social welfare is exactly half of the LP value of x^*.

4 Budget Additive Valuations

Recall the budget additive valutions: a valuation $v : 2^M \to \mathbb{R}$ is budget additive if there is a budget B and for every $S \subseteq M$, $v(S) = \min\{\sum_j v(\{j\}), B\}$. The social welfare maximization can obviously be computed via a similar LP rounding approach, by replacing the capacities in (GAP) by the values. In fact, since the LP becomes more special, the resulting LP has an integrality gap of $\frac{4}{3}$. We known this is tight because of an algorithm given by Chakrabarty and Goel (2010) using iterative rounding. It may be instructive to look at the integrality gap example:

Example 4 (Integrality gap)**.** There are two bidders and three items. For $i = 1, 2$, $v_i(1) = v_i(2) = 1$, $v_i(3) = 2$, $B_i = 2$. The optimal social welfare is 3, whereas the allocation vector $x_{11} = x_{22} = 1$, $x_{13} = x_{23} = \frac{1}{2}$, whichsatisfies all the LP constraints, gives welfare 4.

However, we cannot turn the algorithm by Chakrabarty and Goel (or that by Shmoys and Tardos) directly into an MIDR mechanism, because now the polytope depends on the values and the budgets, which are private information possessed by the bidders.

Open question: Does there exist a polynomial time computable, truthful, $O(1)$-approximation mechanism for budget additive valutions?

References

Chakrabarty, D. and Goel, G. (2010). On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP. *SIAM J. Comput.*, 39(6):2189–2211.

Chekuri, C. and Khanna, S. (2000). A PTAS for the multiple knapsack problem. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA.*, pages 213–222.

Dobzinski, S. and Nisan, N. (2010). Mechanisms for multi-unit auctions. *J. Artif. Intell. Res. (JAIR)*, 37:85–98.

Dughmi, S. and Ghosh, A. (2010). Truthful assignment without money. In *Proceedings 11th ACM Conference on Electronic Commerce (EC-2010), Cambridge, Massachusetts, USA, June 7-11, 2010*, pages 325–334.

Lavi, R. and Swamy, C. (2011). Truthful and near-optimal mechanism design via linear programming. *J. ACM*, 58(6):25.

Roughgarden, T. and Talgam-Cohen, I. (2015). Why prices need algorithms. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 19–36.

Shmoys, D. B. and Tardos, É. (1993). An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474.