Communication Complexity	13 Sep, 2011 (@ IMSc)

6(a). Applications to auctions and linear programming

Lecturer: Prahladh Harsha *Scribe: Nikhil Balaji*

We have seen some basic protocols and lower bounds in communication complexity, and some of their applications to VLSI Design, time-space tradeoffs in the Turing machine model, streaming algorithms, and depth lower bounds for monotone circuits. We will continue in the same spirit and see two more applications of communication complexity. The agenda for today's lecture is as follows: We will study application of communication complexity to:

- combinatorial auctions [BN07], and
- expressing combinatorial optimization problems as linear programs [Yan91].

6(a).1 Combinatorial Auctions

Consider the following scenario: There are n bidders for an auction of m items. The auction is conducted by a single auctioneer who is in possession of the m items. Each bidder has a predetermined (private) valuation of all the m items. The auction is carried out with the following aims:

- The auctioneer wants to maximize his total revenue.
- The bidders want to maximize their total value, which is governed by a valuation function private to the bidder, and neither the auctioneer, nor the other bidders possess any knowledge of a particular bidder's valuation of the items.
- The whole process is expected to happen in a setting of social welfare, i.e., A bidder who wants an item the most hopefully, should be able to get it.

Let $\{\mathcal{B}_i\}_{i=1}^n$ denote each bidder. The auction model that we consider here is that, rather than bidding for individual items, the bidders are interested in "bundles" of items. In order to reflect this, we will define the private valuation functions as assigning weights to all 2^m possible subsets of m items, rather than indivdually to each item. We have, $\mathcal{V}_i : 2^{[m]} \to \mathbb{R}^+$, $\forall i \in [n]$ satisfying the following properties:

1. *Monotonicity:* $\forall S \subseteq T$, $\mathcal{V}(S) \leq \mathcal{V}(T)$.
2. *Free-Disposal:* $\mathcal{V}(\emptyset) = 0$.

Observe that we do not assume that the valuation $\mathcal{V}$ satisfies sub-additivity, i.e., $\mathcal{V}(S \cup T) \neq \mathcal{V}(S) + \mathcal{V}(T)$.

6(a).1.1 The allocation problem

We are interested in the problem of allocating m items to the n bidders[1]

INPUT: The $\{\mathcal{V}_i\}_{i\in[n]}$ valuation function of all the n bidders.
OUTPUT: A partition $(S_1,\ldots,S_n)$ of the m items and a corresponding price vector $(P_1,\ldots,P_n)$ so as to maximize the social welfare, i.e, $\sum_{i\in[n]}\mathcal{V}_i(S_i)$.

The above objective function maximized the social welfare. However, different parties may want to maximize different quantities, based on their interests. For instance,

1. Each *bidder* would want to maximize her utility, ie., $u_(S_i) = \mathcal{V}_i(S_i) - P_i$ for the i-th bidder.
2. The *auctioneer* would want to maximize her revenue, i.e., $\sum_{i\in[n]} P_i$.
3. The *society* would like to maximize social welfare, i.e., maximize $\sum_{i\in[n]}\mathcal{V}_i(S_i)$ or maximize the value of the least happy person, i.e., maximize $\min_{i\in[n]}\mathcal{V}_i(S_i)$.

In this lecture, we will be dealing with allocations that aim to maximize social welfare.[2]

Remark 6(a).1. *Observe that even writing down the valuation functions $\mathcal{V}_i$'s takes exponential time (and space) in the number of items. We will assume that the bidders know their valuations implicitly, i.e., there are small circuits that compute the valuations. In general, we will be interested in algorithms that run in time polynomial in m, the number of items and n, the number of bidders. In other words, the algorithm does not have sufficient time to (explicitly) read the entire valuation.*

6(a).1.2 The single-minded bidder

As noted in Remark 6(a).1, the allocation problem does not get to see the valuation functions explicitly, but only implicitly. In this section, we will consider a special case, wherein the valuation function can be expressed very succinctly, namely the single-mided bidder. A valuation function is called single minded if for all bidders $\mathcal{B}_i$, there exists a $S_i^* \subseteq [m]$ and $v^* \in \mathbb{R}$ such that, $\forall S \subseteq [m]$,

$$\mathcal{V}_i(S) = \begin{cases} v_i^*, & \text{if } S_i^* \subseteq S \\ 0, & \text{otherwise} \end{cases}$$

The pair (S_i^*, v_i^*) is called a single-minded bid of bidder $\mathcal{B}_i$. So, in this case, the auctioneer could either allocate to bidder $\mathcal{B}_i$, the subset $S = S_i^*$ or not allocate anything at all, i.e., $S = \emptyset$. The allocation problem in this case would be:

[1] Such auctions are of practical interest-for example, in spectrum auctions, where licenses are sold for the use of a certain band of the electromagnetic spectrum in a certain geographic area. This could be modeled as a *combinatorial auction*, where the mobile service providers are the bidders and the bandwidth allocation being the item in auction.

[2] A mechanism is said to be *incentive compatible* if it is in the best interest of each bidder (provably) to answer $\mathcal{V}_i(S_i)$ truthfully.

INPUT: $\{S_i^*, v_i^*\}_{i\in[n]}$
OUTPUT: An allocation $T \subseteq [n]$, such that $\forall i, j \in T$, we have $S_i^* \cap S_j^* = \emptyset$ and $\sum_{i\in T} v_i^*$ is maximized.

We show that even for this simple case, the allocation problem is NP-hard.

Theorem 6(a).2. *The allocation problem for single-minded bidders maximizing social welfare (Max-Social Welfare(MSW)) is* NP*-hard. Or more precisely, the decision problem of whether the optimal allocation has a social welfare of atleast k is* NP*-complete.*

Proof. Our allocation problem is an optimization problem where we have to maximize social welfare. We will reduce from the *Maximal Independent Set Problem (MIS)* which is the following: Given a graph $G = (V, E)$ and $k \in \mathbb{Z}^+$, we want our algorithm to output a set $U \subseteq V$ and $|U| \geq k$, such that $\forall x, y \in U, (x, y) \notin E(G)$. Given an instance of the Independent Set Problem, we will build an allocation problem instance from it, such that, V will be the bidders in the auction, and E will be the set of items to be auctioned. So, for each $i \in V, (S_i^*, v_i^*) = (\{e \in E : e \text{ is incident on } i\}, 1)$. Notice that if we have an algorithm for the allocation problem which outputs the set T, we can directly use it to solve the Maximal Independent Set Problem : T has the property that $\forall i, j \in T$, we have $S_i^* \cap S_j^* = \emptyset$ *if and only if* the set of vertices T is an independent set for the graph G, and the social welfare obtained from this auction is exactly $\sum_{i\in T} v_i^* = k$. Hence, the single-minded bidder allocation problem is NP-hard. □

Since the above reduction is from MIS which exhibits very high inapproximability, we can further strengthen the conclusion of the above theorem.

Corollary 6(a).3. *Even $m^{\frac{1}{2}-\varepsilon}$-approximating MSW is* NP*-hard, even in the simple case of the single-minded bidder allocation.*[3]

6(a).1.3 The communication complexity perspective

Suppose the auctioneer and bidders had unlimited computational power, does this make the allocation problem any simpler? The single-minded bidder case then becomes easy to solve (for the objective of maximizing social welfare). But what about the allocation problem for general valuation functions? Note, that the auctioneer can solve the allocation problem by obtaining the valuation functions from all the bidders. However, this requires the bidders to send exponentially long messages to the auctioneers. Is it feasible for the auctioneer and bidders to arrive at a social-welfare maximizing allocation without exchanging too many messages? We will show, using communication complexity, that this is not possible. More precisely, even ignoring computational hardness of the problem (i.e., assuming the auctioneer and bidders have unlimited computational power), we will show any protocol between the bidders and auctioneer that achieves maximum social welfare, requires exponential communication between the various parties in the worst case.

As a first step towards simplifying the current scenario, we will eliminate the need for an auctioneer (since anyway, the goal of the parties is to maximize social welfare), and treat

[3]This approximation, albeit poor, can be achieved by a polynomial time algorithm, which also turns out to be *incentive-compatible*.

the problem as one of communication between the bidders. Let's further assume that there are just two bidders (call them *Alice* and *Bob*) with a 0/1-valuation function.

Theorem 6(a).4. *Any protocol (even randomized) between Alice and Bob (with* $0/1$*-valuations) that maximizes social welfare requires at least* $\Omega(\binom{m}{m/2})$*- bits of communication in the worst case.*

Proof. Since there are only two bidders (Alice and Bob), we can assume wlog. that the allocation problem is equivalent to finding a set $S \subseteq [m]$ such that $\mathcal{V}_A(S) + \mathcal{V}_B(S^c)$ is maximized (where $S^c = [m] \setminus S$). Recall that $\mathcal{V}_A$ and $\mathcal{V}_B$ are monotone functions and since they are $0/1$ valuations, we have that $\max_{S\subseteq[m]}[\mathcal{V}_A(S) + \mathcal{V}_B(S^c)] \in \{0, 1, 2\}$.

We will prove the theorem by reducing from the Disjointness of subsets of $\left[\binom{m}{m/2}\right]$ (i.e, $\mathsf{DISJ}_{\binom{m}{m/2}}$). Let's assume m to be even. Consider an instance of $\mathsf{DISJ}_{\binom{m}{m/2}}$, where Alice and Bob get as inputs $X, Y \subseteq \left[\binom{m}{m/2}\right]$ (or equivalently $X, Y \in \{0,1\}^{\binom{m}{m/2}}$). We will view the strings X and Y as assigning $0/1$-values to each of the $m/2$-sized subsets of $[m]$. Alice and Bob then construct valuations functions $\mathcal{V}_A$ and $\mathcal{V}_B$ as follows. For sets S of size $m/2$, we let

$$\mathcal{V}_A(S) = 1 \text{ iff } X(S) = 1$$
$$\mathcal{V}_B(S) = 1 \text{ iff } Y(S^c) = 1$$

For sets S such that $|S| < m/2$, we have $\mathcal{V}_A = \mathcal{V}_B = 0$ and for sets S such that $|S| > m/2$, we set $\mathcal{V}_A(S) = \mathcal{V}_B(S) = 1$. This ensures that both $\mathcal{V}_A$ and $\mathcal{V}_B$ are monotone.

Observe that $\max_{S\subseteq[m]}[\mathcal{V}_A(S) + \mathcal{V}_B(S^c)] \in \{1, 2\}$ and furthermore if $\max_{S\subseteq[m]}[\mathcal{V}_A(S) + \mathcal{V}_B(S^c)]$, then both S and S^c and $m/2$-sized subsets.

$$\begin{aligned} \mathcal{V}_A(S) + \mathcal{V}_B(S^c) = 2 \quad &\Leftrightarrow \quad X(S) = 1 \text{ and } Y(S) = 1 \\ &\Leftrightarrow \quad X \cap Y \neq \emptyset \end{aligned}$$

Hence, $max_{S\subseteq[m]}[\mathcal{V}_A(S) + \mathcal{V}_B(S^c)] = 2$ if and only if X and Y are not disjoint. We can straightaway use the disjointness lower bound and conclude that atleast $\binom{m}{m/2}$ bits have to be communicated in order for the protocol to find the optimal allocation.

□

6(a).2 Expressing combinatorial optimization problems as Linear Programs

We will now see an application of communication complexity in combinatorial optimization due to Yannakakis [Yan91]. In the mid-1980's, there were several attempts to give polynomial sized linear programming formulations for various NP -complete problems including, the *Hamiltonian cycle problem*[4]. Yannakakis observed a common theme in these (failed)

[4]Since a linear program on polynomially many variables and polynomially many constraints is solvable in polynomial time (using the ellisoid algorithm) and Hamiltonian cycle is NP-hard, these attempts if successful would imply that P = NP! Check, http://www.win.tue.nl/~gwoegi/P-versus-NP.htm for more details.

attempts and in the process of refuting them, identified a combinatorial parameter of the underlying optimization problems and showed that the LP formulation could be polynomial sized if and only if this parameter was small. He then used techniques from communication complexity to show that this parameter is large typically for NP-complete problems. For instance, one such result proved by Yannakakis [Yan91] is that expressing the Traveling Salesperson Problem (TSP) by a *symmetric linear program* requires exponential size. We will build towards this result in the rest of this lecture, and sketch the proof in the next lecture.

6(a).2.1 Combinatorial Optimization Problems

A typical combinatorial optimization problem is of the form:

$$\begin{aligned} & \max c^T x \\ \text{subject to} \quad & x \in S \end{aligned}$$

where S is a set of feasible solutions, membership in which can be checked efficiently. For example, S could be the set of all perfect matchings in a graph (i.e., $x \in S \subseteq \{0,1\}^{\binom{n}{2}}$ if x is the characteristic vector of a perfect matching). Another example is the set of all possible travelling salesperson tours. For most of these problems, $\max\{c^T x : x \in S\}$ is equivalent to $\max\{c^T x : x \in \mathrm{conv}(S)\}$, where $\mathrm{conv}(S)$ is the convex hull of the points in S. In other words, for most of these problems the maximum is unchanged whether one optimized over the set S or the convex hull of S. Note that $\mathrm{conv}(S)$ is a polytope and $c^T x$ is a linear function. Hence, we can apply LP solvers to solve such an optimization problem, provided the LP is "small"[5]. As a concrete example, consider the TSP problem.

INPUT: A distance function $d : [n] \times [n] \to \mathbb{R}^{\geq 0}$ between the n vertices of a complete graph K_n.

OUTPUT: A tour $v_0, v_1, \ldots, v_n$, such that, $v_0 = v_n$ and each vertex in the set $\{v_0, v_1, \ldots, v_n\}$ appears exactly once.

OBJECTIVE: Minimize $\sum_{i=0}^{n-1} d(v_i, v_{i+1})$, over all possible tours of the n vertex graph.

Let $S \subseteq \{0,1\}^{\binom{n}{2}}$ be the set of feasible tours in K_n, i.e, $x \in S$ iff x is the characteristic vector of a feasible tour, $x_{i,j}$ is 1 if the edge (i,j) is in the tour and 0 if the edge (i,j) is not in the tour. The convex hull $\mathrm{conv}(S)$ is called the TSP polytope and it can be checked that the TSP polytope has exponentially many facets (constraints). Observe, that solving the TSP problem, i.e., finding $\min\{\sum_{(i,j)\in E} d(i,j)x_{i,j} : x \in S\}$ is equivalent to finding $\min\{\sum_{(i,j)\in E} d(i,j)x_{i,j} : x \in \mathrm{conv}(S)\}$. The latter is a LP and would have been solvable in polynomial time if the TSP polytope had only polynomially many constraints. Is it possible that the TSP polytope has an alternate representation which has only polynomially many variables and constraints?

[5] for the purpose of this lecture, we will limit ourselves to "small" LPs in the sense that they have only polynomially many variables and constraints, in which case it is known that the LP is polynomial time solvable. Polynomial time solvability is implied by more general conditions (eg., existance of a polynomial time *separation oracle* for membership in the polytope), which we will not consider in this lecture.

More generally, suppose we have a polytope $P = \{x \in \mathbb{R}^n : Cx \leq d\}$ where the number of constraints (i.e, number of rows of C) is exponential. One natural question to ask is whether there is an alternate polytope P' of polynomial size that "expresses P". We say that a polytope $P' = \{(x, y) \in \mathbb{R}^{m+n} : C'x + D'y \leq d'\}$ expresses P iff

$$P = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m \text{ such that } (x, y) \in P'\}.$$

Clearly, if P' expresses P, then $\max\{c^T x : x \in P\} = \max\{c^T x : (x, y) \in P'\}$. In other words, to maximize $c^T x$ over P, one might as well maximize $c^T x$ over P'. Now, if P' is polynomial sized, then the latter problem is polynomial time solvable. Does this approach work? (as in, do the additional variables added, bring down the number of constraints?) We will see an example where it does help before we move ahead to the question of whether it helps in the case of the TSP polytope.

The Parity Polytope

$$PP = \text{conv}\{x \in \{0, 1\}^n : x \text{ has an odd number of 1's}\}.$$

It can be checked that the PP has exponential many facets (constraints), namely:

$$\sum_{i \in S} x_i - \sum_{i \in S^c} x_i = |S| - 1, \quad \forall S \subseteq [n], |S| \text{ even},$$
$$0 \leq x_i \leq 1$$

Howevwer, the following simple observation shows that there exists a polynomial sized polytope PP' that expresses PP.

$$PP = \text{conv}\left\{\bigcup_{k\text{ – odd}} \text{conv}\left\{x \in \{0, 1\}^n : |x| = k\right\}\right\},$$

where $|x|$ denotes the number of 1's in x. In other words, $x \in PP$ iff x can be written as $x = \sum_{k\text{ - odd}} \alpha_k y_k$ where $\alpha_k \in [0, 1]$ such that $\sum \alpha_k = 1$ and $y_k \in C_k$ where $C_k = \text{conv}\{x \in \{0, 1\}^n : |x| = k\}$. It can be checked that $C_k = \{x \in [0, 1]^n : \sum x_i = k\}$. This helps us write another polytope PP' that expresses PP, namely:

$$\sum_{k \text{ } odd} \alpha_k = 1$$
$$x_i = \sum_{k \text{ } odd} z_{ik}, \qquad \forall i \in [n]$$
$$\sum_{i \in [n]} z_{ik} = k\alpha_k, \qquad \forall \text{ odd } k$$
$$\alpha_i \geq 0, \qquad \forall \text{ odd } k$$

In the formulation above, the vector $z_k/\alpha_k = (z_{1k}/\alpha_k, z_{2k}/\alpha_k, \ldots, z_{nk}, \alpha_k)$ plays the role of the vector y_k in the intuition explained earlier.Hence, PP' (which has only polynomiall many variables and constraints) expresses PP. Hence, increasing the number of variables has helped us reduce the size of the polytope expressing PP.

Does this technique work in general? Given a polytope $P = \{x : Cx \leq d\}$, where size of C is large, what is the minimum number of new variables we need to add such that the polytope $P' = \{(x, y) : C'x + D'y \leq d'\}$ expresses P, where C' and D' are small in size? To answer this, we first need a convenient representation of a polytope given by a linear program.

Definition 6(a).5. *Let $P = \{x \in \mathbb{R}^n : Cx \leq d\}$ be a polytope. Wlog. assume that all the constraints are linearly independent. Let M and N denote the number of facets (equivalently the number of constraints) and the number of vertices respectively of the polytope P. We will construct a $M \times N$ non-negative matrix S, which we will call the Slack Matrix*[6]*, which has a row for each constraint "$\langle c_j, x\rangle \leq d_j$" and a column for each vertex u_i, with entries:*

$$S_{ij} = d_j - \langle c_j, u_i \rangle$$

In other words,

$$S = \begin{pmatrix} d \cdot \overline{1} & -C \end{pmatrix} \begin{pmatrix} 1 & 1 & \dots & 1 \\ u_1 & u_2 & \dots & u_N \end{pmatrix}.$$

The (i, j)-th entry of the matrix holds the slack that vertex u_i witnesses for the constraint "$\langle c_j, x\rangle \leq d$". A 0 entry in the matrix means the vertices is tight wrt. that constraint. Since the c_j's are linearly independent, for every constraint there will be atleast M vertices which are tight. Since S can be written as $F' \cdot V'$ where F' is a $(M \times (n+1))$-matrix and V' is a $((n+1) \times N)$-matrix, the rank of S is at most $(n+1)$. However, observe that both F' and V' have negative entries. Can we write $S = F \cdot V$ where F and V are respectively $(M \times m)$ and $(m \times N)$ *non-negative* matrices. The smallest such m is called the positive rank of S, denoted by m^*. More precisely,

$$m^*(S) = \min \{m : [S]_{M\times N} = [F]_{M\times m}.[V]_{m\times N}, \text{where } F \text{ and } V \text{ are non-negative matrices}\}$$

Yannakakis observed the following connection between m^* and the minimum number of variables and variables over all LP's that express P.

Theorem 6(a).6. *Let P be a polytope and m^* be the positive rank of the it's slack matrix. The minimum number of variables plus constraints over all LP's that express P is $\Theta(n+m^*)$.*

We will see the proof of this theorem, and it's connection to communication complexity in the next lecture.

References

[BN07] Liad Blumrosen and Noam Nisan. *Combinatorial auctions*. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, eds., *Algorithmic Game Theory*, chapter 11, pages 267–300. Cambridge University Press, 2007.

[Yan91] Mihalis Yannakakis. *Expressing combinatorial optimization problems by linear programs.* J. Computer and System Sciences, 43(3):441–466, 1991. (Preliminary Version in *20th STOC*, 1988). doi:10.1016/0022-0000(91)90024-Y.

[6]Note that this matrix could be too large to be written down, and we use it here for analysis purposes only