CS364A: Problem Set #2

Due in class on Tuesday, November 7, 2006

Instructions:

(1) This is a long and challenging problem set, and you are not expected to solve all of the problems to completion. You are, however, expected to think hard about all of them. Give complete solutions for as many as you can; for the others, explain your progress and where you got stuck. (If you've spent, say, 12 hours on the problem set and are sick of it, then you should just turn it what you have.)

(2) You may refer to your course notes, general references (e.g., textbooks), and material on the course Web site, but *not* to additional specific research papers.

(3) Collaboration on this homework is actively encouraged. However, your write-up must be your own, and you must list the names of your collaborators on the front page.

(4) Grades will be assigned on a plus/check/minus scale.

Problem 0

By Tuesday, October 31st, pick a project topic from the list at `http://theory.stanford.edu/~tim/f06/projects.html`, or come up with your own topic. Email the instructor with your choice. Note that topics are available on a *first-come, first-serve basis*, with at most one student (or team of students) from the class working on a single project topic.

Problem 1

Consider a set Ω of outcomes and n players, where player i has a private real-valued valuation $v_i(o)$ for each outcome $o \in \Omega$. Suppose the function $f : \Omega \rightarrow \mathcal{R}$ has the form

$$f(o) = c(o) + \sum_{i=1}^{n} w_i v_i(o),$$

where c is a publicly known function of the outcome, and where each w_i is a nonnegative, public, player-specific weight. Such a function is called an *affine maximizer*.

(a) Show that for every affine maximizer objective function, there is a truthful mechanism that optimizes it.

(Hint: modify the VCG mechanism. Don't worry about computational issues or individual rationality.)

(b) (Easy.) Argue that maximizing surplus in the presence of costs (as discussed in lecture) reduces to optimizing an affine maximizer.

(c) Argue that the VCG mechanism can be implemented in polynomial time for the special case of the fixed-tree multicast problem.

(d) Give a concrete example showing that the VCG mechanism is not budget-balanced in the special case of the fixed-tree multicast problem. (A detail: recall that the VCG mechanism is defined up to the choice of the "pivot term". Restrict attention to pivot terms which ensure individual rationality.)

Problem 2

In this problem we return to the winner determination problem and the design of truthful, polynomial-time, approximate combinatorial auctions. Recall that a valuation v_i is *subadditive* if $v_i(T_1) + v_i(T_2) \geq v_i(T_1 \cup T_2)$ for every pair T_1, T_2 of disjoint subsets of goods. As usual, we assume that valuations are nondecreasing (so $T_1 \subseteq T_2$ implies that $v_i(T_1) \leq v_i(T_2)$) and that $v_i(\emptyset) = 0$.

(a) In the first three parts, we consider the winner determination problem with subadditive valuations (i.e., we don't worry about payments or incentive constraints, just surplus maximization). Fix a set S of goods and subadditive valuations $v_1, \ldots, v_n$. Call the problem *lopsided* if there is an optimal allocation in which at least half of the total surplus of the allocation is due to players that were allocated a bundle with at least $\sqrt{m}$ goods. (I.e., if $2\sum_{i \in A} v_i(T_i^*) \geq \sum_{i=1}^n v_i(T_i^*)$, where $\{T_i^*\}$ is the optimal allocation and A is the subset of bidders i with $|T_i^*| \geq \sqrt{m}$.)

Show that in a lopsided problem, there is an allocation that gives all of the goods to a single player and achieves an $\Omega(1/\sqrt{m})$ fraction of the maximum-possible surplus.

(b) Show that in a problem that is not lopsided, there is an allocation that gives at most one good to each player and achieves an $\Omega(1/\sqrt{m})$ fraction of the maximum-possible surplus.

(Hint: use subadditivity.)

(c) Give a polynomial-time $O(\sqrt{m})$-approximate winner determination algorithm for subadditive valuations.

(Hint: make use of a graph matching algorithm.)

(d) Give a polynomial-time, $O(\sqrt{m})$-approximate, truthful combinatorial auction for subadditive valuations.

(Hint: what happens to the VCG mechanism if you restrict it to optimize only over a subset of all possible outcomes?)

Problem 3

Suppose we are given a set $N = \{1, 2, \ldots, n\}$ of players and a nonnegative, nondecreasing cost function $C : 2^N \to \mathcal{R}^+$. Recall that a nonnegative vector $c_1, \ldots, c_n$ is *in the core* if $\sum_{i \in N} c_i = C(N)$ and $\sum_{i \in S} c_i \leq C(S)$ for every $S \subseteq N$. Recall that the *Shapley value of* i is the expected value of $C(S \cup \{i\}) - C(S)$, where the expectation is over the uniformly random ordering of the players, and S denotes the players preceding i in this ordering.

(a) Suppose that C is a submodular function in the sense of Problem 4 from HW #1. Show that the Shapley value of C is in the core.

(You can prove this directly or derive it from (b), below.)

(b) For a subset $S \subseteq N$ of players, let C^S denote the restriction of the cost function C to S. Define a cost-sharing method χ for C by, for every $S \subseteq N$ and $i \in S$, setting $\chi(i, S)$ equal to the Shapley value of i with respect to the set S of players and the cost function C^S.

Show that if C is submodular, then χ is budget-balanced and cross-monotonic. (Recall that χ is called *cross-monotonic* if for all $S \subseteq T \subseteq N$ and $i \in S$, $\chi(i, S) \geq \chi(i, T)$.)

(c) Show that for every fixed-tree multicast instance, the corresponding cost function is submodular. Show that the "Shapley cost-sharing method" from lecture (share each edge cost equally) coincides with the one defined by part (b) for this cost function.

(d) Prove that a cost function C admits a budget-balanced, cross-monotonic cost-sharing method only if it is subadditive (in the same sense as in Problem 2).

(e) **Extra credit:** The converse of (d) is false. A natural example is the following. Let $G = (V, E)$ be an undirected complete graph, with a root vertex $r \in V$ and a nonnegative cost on each edge. The players N are the non-root vertices $V \setminus \{r\}$. For a subset $S \subseteq N$ of players, let $C(S)$ denote the cost of a minimum-spanning tree in the subgraph induced by $S \cup \{r\}$. (As a warm-up, prove that C is subadditive but not necessarily submodular.) Show that there is a choice of the graph G that yields a cost function C that admits no budget-balanced, cross-monotonic cost-sharing method.

(Hint: one approach is to give an example of such a function with an empty core.)

Problem 4

Recall the definitions of cross-monotonic cost-sharing methods and the corresponding Moulin mechanisms from lecture.

(a) Recall digital goods auctions from lecture and Problem 1 of HW #1. There is a 4-competitive (in expectation) digital goods auction, called RSPE, that works as follows. First, bids are collected and then each bidder is randomly placed into one of two groups. Second, the auction uses the bids to compute the maximum-possible profits P_1^* and P_2^* (respectively) obtainable from the bidders in each of these groups using a fixed price (a different price can be used for each group). Third, the auction tries to extract P_1^* profit from the second group and P_2^* profit from the first group.

The third step of the RSPE auction motivates the following *profit extraction* problem for digital goods (unlimited supply, no costs): given a set of bidders and a revenue target P^*, design a truthful mechanism that extracts revenue P^* whenever this is possible via a fixed price. (If there is no fixed price that extracts revenue P^*, the mechanism is permitted to generate zero revenue.) First, show how to solve the profit extraction problem using a Moulin mechanism based on a cross-monotonic cost-sharing method. (Truthfulness of this mechanism follows from the next part of this problem.) Second, describe what the bid-independent thresholds of your profit extraction mechanism are (recall the characterization from Problem 1 in HW #1).

(b) Prove that every Moulin mechanism induced by a cross-monotonic cost-sharing method is truthful. Show by counterexample that the "cross-monotonic" hypothesis is in general necessary for truthfulness.

(c) A Moulin mechanism is called *groupstrategyproof (GSP)* if every coordinated set of false bids by a coalition decreases the utility of some player in the coalition (or has no effect on any of their utilities). This is a form of collusion-resistance, assuming that "side payments" between colluding players are not allowed. Formally, let $T \subseteq N$ be a coalition of players. Let b and b' be two bid vectors (indexed by N) for which $b_i = b'_i$ for all players $i \notin T$ outside the coalition. Assume that players of T bid truthfully in b ($b_i = v_i$ for all $i \in T$). Let (S, p) and (S', p') be the mechanism outcomes (winners and prices) given the bids b and b', respectively. The GSP condition requires that if $u_i(S', p') > u_i(S, p)$ for some player $i \in T$, then $u_j(S', p') < u_j(S, p)$ for some other player $j \in T$. (Here u_i denotes the usual quasi-linear utility function.) Note that truthfulness is simply the GSP condition for singleton coalitions.

Prove that every Moulin mechanism induced by a cross-monotonic cost-sharing method is GSP.

Problem 5

This problem considers the following scheduling game. There are n players, where player j has a nonnegative weight w_j. There are m identical machines. Each player picks one machine, and wants to minimize the total weight (aka *load*) on its machine. In addition, each player j has a restricted (non-empty) subset S_j of eligible machines that it can use—player j cannot be assigned to a machine outside of S_j. The global objective function is the *makespan*—the total weight on the most heavily loaded machine.

A *pure-strategy Nash equilibrium* is an assignment of players to machines so that no player has a unilateral incentive to deviate. In other words, in a pure-strategy Nash equilibrium, no player can switch machines and wind up on a less loaded machine than before.

(a) Prove that every such scheduling game has at least one pure-strategy Nash equilibrium.

(Hint: prove that "best-response dynamics"—iteratively allowing an unsatisfied player to switch to its favorite machine—converges, necessarily to a Nash equilibrium, in a finite number of iterations.)

(b) (Easy.) Consider a pure-strategy Nash equilibrium, and suppose that player j is scheduled on a machine with load L. What can you say about the loads of the other machines?

(c) Suppose that for all players j, the set S_j is all of the machines—so every machine is eligible for every player. What is the largest-possible price of anarchy—the ratio between the largest makespan of a pure-strategy Nash equilibrium and the minimum-possible makespan—in such a game? Prove the best upper and lower bounds that you can. Such bounds can be constant or functions of the number of machines and/or players.

(Hint: to prove an upper bound on the price of anarchy, make use of the following two lower bounds on the minimum-possible makespan: (1) the maximum weight of a player; (2) the sum of the players' weights divided by the number of machines.)

(d) What is the largest-possible price of anarchy when the sets S_j are arbitrary?

(Hint: for both the upper and lower bounds, you might initially restrict attention to instances with only unit-weight players.)