CPSC455b: Written Homework Assignment #2

Due in class on February 19, 2002

These exercises are drawn from the following papers (all obtainable from http://pantheon.yale.edu/~sz38/) and the related lectures and discussions in class:

- The Nisan-Ronen, Hershberger-Suri, and Roughgarden-Tardos papers in the Routing section
- The Ronen, Goldberg *et al.*, and Nisan-Ronen papers in the Auctions section

Problem 1 (20 points)

A mechanism is *group-strategyproof* if no group of agents can jointly gain an advantage by lying about their private types. More formally, let $t = (t^1, \ldots, t^n)$ be a true type vector and $S \subseteq \{1, \ldots, n\}$ be a strategizing group. Let $a_S = \{a^i\}_{i \in S}$ be the group strategy of S; so, for $i \in S$, $a^i \neq t^i$. Let $t_{\overline{S}} = \{t^i\}_{i \notin S}$ be the types of the agents that are not in the strategizing group. Then $o' = o(t_{\overline{S}}, a_S)$ and $q^i = p^i(t_{\overline{S}}, a_S)$ are the output and payment values that the mechanism computes when each $i \in S$ gives the input a^i and each $i \notin S$ gives the input t^i, while $o = o(t)$ and $p^i = p^i(t)$ are the output and payment values that it would have computed if all agents had supplied their true type values. The mechanism is group-strategyproof if, for all t, S, and a_S, either $u^i(o', t^i) = u^i(o, t^i)$ for all $i \in S$, or there is at least one $i \in S$ for which $u^i(o', t^i) < u^i(o, t^i)$. That is, if the group strategy a_S results in a strictly higher utility for at least one agent in S, it must also result in a strictly lower utility for at least one agent in S.

(a) (15 points) VCG mechanisms are strategyproof but not group-strategyproof. Convince yourself that this is true by examining the lowest-cost path mechanism studied in the Nisan-Ronen and Hershberger-Suri papers. Construct an infinite family of graphs, edge costs, and group strategies that show that this mechanism is not group-strategyproof.

* (b) (5 points) Comment on this definition of group-strategyproofness (which is one of the standard definitions that appear in the mechanism-design literature). Does it capture your intuition about what it means for a group to "jointly gain an advantage by lying," and, if not, why not?

Problem 2 (20 points)

(a) (10 points) Consider a network in which all of the latency functions are linear, *i.e.*, $l_e(f_e) = a_e \cdot f_e + b_e$, for each edge e. Show that these two facts follow from the Roughgarden-Tardos results:

- If $a_e = 0$ for all e, then the Nash and optimal flows coincide.
- If $b_e = 0$ for all e, then the Nash and optimal flows coincide.

(b) (10 points) Recall from Homework assignment #1 that one difference between the Roughgarden-Tardos formulation of selfish routing and real-world IP routing is that the former deals with "splittable" or "infinitely divisible" flows. An instance of the "finite, unsplittable" flow problem is similar to an instance of the splittable flow problem in that it consists of a graph G, a set of latency functions l_e (one for each edge $e \in G$), and agents $\{1, \ldots, k\}$, where agent i controls r_i units of flow; however, in the finite, unsplittable case, each agent must choose a single path along which to route all of the flow it controls. Show that the main bicriteria result of Roughgarden-Tardos (*i.e.*, that the cost of a Nash flow for (G, r, l) is at most the cost of an optimal flow for $(G, 2r, l)$) does not hold for finite, unsplittable flows.

Problem 3 (20 points)

(a) (10 points): In their paper "Computationally Feasible VCG Mechanisms," Nisan and Ronen define "feasibly dominant actions." Explain their assertion that "A dominant action is obviously feasibly dominant." Specifically, what form does the strategic knowledge function b^i have for an agent i that has a dominant strategy?

(b) (10 points): Show that the multicast transmission problem, as Nisan and Ronen define it in "Computationally Feasible VCG Mechanisms," satisfies their definition of a CMAP. Show that it also satisfies their definition of an NP-complete mechanism-design problem. (Note: They should really use the term "NP-hard mechanism-design problem," because the term "NP-Complete" is usually used to describe a decision problem, not an optimization problem.)

Problem 4 (20 points)

Consider Theorem 4.1 in Ronen's paper "On Approximating Optimal Auctions." He states that one can generalize this result to the case of k copies of the same item with unit demand by having the seller reject all but the k highest offers and then construct the optimal auction on the remaining agents. Prove that, as claimed, this generalization is 2-approximately optimal as well.

Problem 5 (20 points)

Consider the proof of Theorem 5.1 in "Competitive Auctions," by Goldberg *et al.* Note that the bid vectors on which the auction $\mathcal{A}_f$ performs badly contain only two distinct bid values. Are these the only bad cases? If so, prove that they are the only bad cases. If not, construct bad cases in which the bid vectors contain as many distinct bid values as possible.