CS364A: Problem Set #4

Due in class on Thursday, November 20, 2008

Instructions: Same as previous problem sets.

Problem 16

(a) (5 points) *Algorithmic Game Theory*, Exercise 19.9.

(b) (5 points) *Algorithmic Game Theory*, Exercise 19.10.

Problem 17

(a) (10 points) *Algorithmic Game Theory*, Exercise 19.13.

(b) (5 points) In this part and the next, we consider abstract finite game with n players with finite strategy sets $S_1, \ldots, S_n$. Each player has a payoff function π_i mapping outcomes (elements of $S_1 \times \cdots \times S_n$) to real numbers. Recall that an exact potential function Φ for such a game is defined by the following property: for every outcome $\mathbf{s} \in S_1 \times \cdots \times S_n$, every player i, and every deviation $s'_i \in S_i$,

$$\pi_i(s'_i, s_{-i}) - \pi_i(s_i, s_{-i}) = \Phi(s'_i, s_{-i}) - \Phi(s_i, s_{-i}).$$

Prove that if such a game admits two exact potential functions Φ_1 and Φ_2, then Φ_1 and Φ_2 differ by a constant. (I.e., for some $c \in \mathcal{R}$, $\Phi_1(\mathbf{s}) = \Phi_2(\mathbf{s}) + c$ for every outcome $\mathbf{s}$ of the game.)

(c) (10 points) Prove that a finite game admits an exact potential function if and only if for every two outcomes $\mathbf{s^1}$ and $\mathbf{s^2}$ that differ in two players' choices (say players i and j),

$$\left(\pi_i(s_i^2, s_{-i}^1) - \pi_i(\mathbf{s^1})\right) + \left(\pi_j(\mathbf{s}^2) - \pi_j(s_i^2, s_{-i}^1)\right) = \left(\pi_j(s_j^2, s_{-j}^1) - \pi_j(\mathbf{s^1})\right) + \left(\pi_i(\mathbf{s}^2) - \pi_i(s_j^2, s_{-j}^1)\right).$$

Problem 18

(15 points) *Algorithmic Game Theory*, Exercise 19.12.

Problem 19

(a) (7 points) Consider an atomic selfish routing game in which all players have the same source vertex and sink vertex (and each controls one unit of flow). Assume that edge cost functions are nondecreasing, but do not assume that they are affine. Prove that a (pure-strategy) Nash equilibrium (i.e., an equilibrium flow) can be computed in polynomial time.

[Hint: Remember the potential function. You can assume without proof that the minimum-cost flow problem can be solved in polynomial time. If you haven't seen the min-cost flow problem before, you can read about it in any book on "combinatorial optimization". Be sure to discuss the issue of fractional vs. integral flows, and explain how (or if) you use the hypothesis that edge cost functions are nondecreasing.]

(b) (7 points) Prove that in an atomic selfish routing network of parallel links, every equilibrium flow minimizes the potential function.

(c) (6 points) Show by example that (b) does not hold in general networks, even when all players have a common source and sink vertex.

Problem 20

(a) (7 points) Recall the scheduling games of Problem #15, where each job j has a processing time p_j and selfishly schedules itself on one of m machines to minimize its completion time. Assume that every job can be feasibly scheduled on any machine (in the notation of Problem #15, S_j is the set of all machines for each j). *Unlike* in Problem #15, you should assume that all jobs on a common machine complete at the same time. So if jobs with processing times 1, 3, and 5 are scheduled together, all of them will complete at time 9. As before, the global objective is to minimize the makespan (the maximum completion time).

Suppose we run best-response dynamics long enough, starting from an arbitrary initial outcome, so that every job has had at least one opportunity to deviate. Prove that the makespan of the resulting outcome is at most twice that of an optimal solution.

[Hint: note that $\sum_j p_j/m$ and $\max_j p_j$ are both legitimate lower bounds on the minimum-possible makespan.]

(b) (8 points) Consider an atomic selfish routing network with affine cost functions. Suppose that each player runs a no-regret algorithm to select a route in each of T time steps, resulting in a sequence of flows $f^1, \ldots, f^T$. (Recall from lecture or Chapter 4 of the AGT book that a no-regret algorithm produces a sequence of actions whose expected average cost is at most an $o(1)$ additive term more than the average cost that would have been incurred by any *fixed* action.)

Modify the proof from lecture (that upper bounded the POA in this context) to show that, for every $\epsilon > 0$ and $T = T(\epsilon)$ sufficiently large, the expected cost

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{e\in E} c_e(f_e^t)f_e^t$$

is at most $[(3+\sqrt{5})/2] + \epsilon$ times the cost of an optimal flow.