CS364A: Problem Set #4

Due to the TAs by noon on Friday, November 22, 2013

Instructions:

(0) We'll grade this assignment out of a total of 75 points; if you earn more than 75 points on it, the extra points will be treated as extra credit.

(1) Form a group of at most 3 students and solve as many of the following problems as you can. You should turn in only one write-up for your entire group.

(2) Turn in your solutions directly to one of the TAs (Kostas or Okke). Please type your solutions if possible and feel free to use the LaTeX template provided on the course home page. Email your solutions to `cs364a-aut1314-submissions@cs.stanford.edu`. If you prefer to hand-write your solutions, you can give it to one of the TAs in person.

(3) If you don't solve a problem to completion, write up what you've got: partial proofs, lemmas, high-level ideas, counterexamples, and so on.

(4) Except where otherwise noted, you may refer to your course notes, and to the textbooks and research papers listed on the course Web page *only.* You cannot refer to textbooks, handouts, or research papers that are not listed on the course home page. If you do use any approved sources, make you sure you cite them appropriately, and make sure that all your words are your own.

(5) You can discuss the problems verbally at a high level with other groups. And of course, you are encouraged to contact the course staff (via Piazza or office hours) for additional help.

(6) No late assignments will be accepted.

Problem 22

(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 from Lecture 13. 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 23

This problem develops some theory about potential games, which were introduced in Lecture 13. We consider an 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 a 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 $s'_i \in S_i$,

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

(a) (8 points) A *team game* is a game in which all players have the same payoff function: $\pi_1(\mathbf{s}) = \cdots = \pi_n(\mathbf{s})$ for every outcome $\mathbf{s}$. In a *dummy game*, the payoff of every player i is independent of its strategy: $\pi_i(s_i, \mathbf{s}_{-i}) = \pi_i(s'_i, \mathbf{s}_{-i})$ for every $\mathbf{s}_{-i}$ and every $s_i, s'_i \in S_i$.

Prove that a game with payoffs $\pi_1, \ldots, \pi_n$ is a potential game (i.e., admits a potential function Φ) if and only if it is the sum of a team game $\pi_1^t, \ldots, \pi_n^t$ and a dummy game $\pi_1^d, \ldots, \pi_n^d$ (i.e., $\pi_i(\mathbf{s}) = \pi_i^t(\mathbf{s}) + \pi_i^d(\mathbf{s})$ for every i and $\mathbf{s}$).

(b) (4 points) Prove that if a game admits two potential functions Φ_1 and Φ_2, then Φ_1 and Φ_2 differ by a constant. That is, for some $c \in \mathcal{R}$, $\Phi_1(\mathbf{s}) = \Phi_2(\mathbf{s}) + c$ for every outcome $\mathbf{s}$ of the game.

Thus, it is well defined to speak of "the potential function maximizer" of a potential game.

(c) (8 points) Prove that a finite game admits a 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),

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

Problem 24

Recall from Lecture 13 that a *congestion game* is like an atomic selfish routing game except we drop the assumption that strategies represent paths in a network. That is, there is a ground set E (previously, the edges), and each $e \in E$ has a cost function c_e. Each player i has a strategy set S_i, and each strategy $s_i \in S_i$ is a subset of E (previously, a path). In an outcome $\mathbf{s}$, if x_e players are using a strategy that contains e, then player i's cost is $\sum_{e \in s_i} c_e(x_e)$. We effectively proved in lecture that every congestion game is a potential game. In this problem we prove the converse.

Two games G_1 and G_2 are *isomorphic* if: (i) they have the same number k of players; (ii) for each i, there is a bijection f_i from the strategies A_i of player i in G_1 to the strategies B_i of player i in G_2; and (iii) these bijections preserve payoffs, so that $\pi_i^1(s_1, \ldots, s_n) = \pi_i^2(f_1(s_1), \ldots, f_n(s_n))$ for every player i and outcome $s_1, \ldots, s_n$ of G_1. (Here π^1 and π^2 are the payoff functions of G_1 and G_2, respectively.)

(a) (7 points) Prove that every team game (see Problem 22) is isomorphic to a congestion game.

(b) (7 points) Prove that every dummy game (see Problem 22) is isomorphic to a congestion game.

(c) (6 points) Prove that every potential game is isomorphic to a congestion game.

Problem 25

(10 points) *Algorithmic Game Theory*, Exercise 19.14.

Problem 26

(a) (10 points) *Algorithmic Game Theory*, Exercise 19.16(b). Proving that these games are "valid utility games" just means proving that they satisfy the same 3 properties as the location games discussed in Lecture 14: surplus is at least the sum of players' payoffs; the payoff of a player is at least the surplus increase caused by the presence of its location[1]; and surplus is a submodular function of the set of chosen locations.

(b) (5 points) Prove that the games in part (a) are potential games.

Problem 27

(10 points) *Algorithmic Game Theory*, Exercise 19.17(b).

Problem 28

Consider n identical machines and m selfish jobs (the players). Each job j has a processing time p_j. Once jobs have chosen machines, the jobs on each machine are processed serially from shortest to longest. (You can assume that the p_j's are distinct.) For example, if jobs with processing times 1, 3, and 5 are scheduled on a common machine, then they will complete at times 1, 4, and 9, respectively. The following questions concern the game in which players choose machines in order to minimize their completion times, and the objective function of minimizing the sum $\sum_{j=1}^{m} C_j$ of the jobs' completion times.

(a) (5 points) Define the *rank* R_j of job j in a schedule as the number of jobs on j's machine with processing time at least p_j (including j itself). For example, if jobs with processing times 1, 3, and 5 are scheduled on a common machine, then they have ranks 3, 2, and 1, respectively.

Prove that in these scheduling games, the objective function value of an outcome can also be written as $\sum_{j=1}^{m} p_j R_j$.

(b) (5 points) Prove that the following algorithm produces an optimal outcome: (i) sort the jobs from largest to smallest; (ii) for $i = 1, 2, \ldots, m$, assign the ith job in this ordering to machine $i \bmod m$ (where machine 0 means machine m).

(c) (15 points) Prove that for every such scheduling game, the expected objective function value of every coarse correlated equilibrium is at most twice that of an optimal outcome.

[Hint: In Lecture 14, the (λ, μ)-smoothness condition was required for all pairs $\mathbf{s}, \mathbf{s}^*$ of outcomes. Weaken the definition so that this condition only needs to hold for some optimal outcome $\mathbf{s}^*$ and all outcomes $\mathbf{s}$. Observe that the POA of coarse correlated equilibria remains at most $\frac{\lambda}{1-\mu}$ assuming only this weaker condition (with the same proof as before). Prove that these scheduling games satisfy this weaker condition with $\lambda = 2$ and $\mu = 0$.]

Problem 29

Recall the set-up for online regret-minimization: there is a fixed set A of actions; each day $t = 1, \ldots, T$ you pick an action $a^t \in A$ (possibly from a probability distribution) based only on information from previous days; and then a cost vector $c^t : A \to [0, 1]$ is unveiled. The goal is to design a (randomized) algorithm that, for every sequence of cost vectors, has small expected average regret. [Recall that the (average, per time-step) regret is the difference between your average cost ($\frac{1}{T}\sum_{t=1}^{T} c^t(a^t)$) and that of the best fixed action ($\min_{a \in A} \frac{1}{T}\sum_{t=1}^{T} c^t(a)$).]

(a) (5 points) Suppose that, every day, you pick the action that performed best in the past (i.e., that minimizes the cumulative cost $\sum_{s=1}^{t-1} c^s(a)$ over $a \in A$). Show that, in the worst case, the average regret of this algorithm is $\Omega(1)$ as $T \to \infty$.

[1] In the location games in lecture, this held with equality. You should check that the weaker property here is sufficient for the POA bound of $\frac{1}{2}$.

(b) (5 points) Let's consider the following randomized pre-processing step: independently for each action a, initialize the starting cumulative cost to a geometric random variable $-X_a$ with parameter ϵ (i.e., to the number of coin flips needed until you get "heads", assuming that the probability of "heads" is ϵ). Then, every day, you pick the action that minimizes the perturbed cumulative cost $-X_a + \sum_{s=1}^{t-1} c^s(a)$ over $a \in A$.

First prove that, for each day t, with probability at least $1 - \epsilon$, the smallest perturbed cumulative cost of an action prior to day t is at least 1 less than the second-smallest perturbed cumulative cost of an action prior to day t.

(c) (5 points) As a thought experiment, consider the (unimplementable) algorithm that, every day, picks the action that minimizes the perturbed cumulative cost $-X_a + \sum_{s=1}^{t} c^s(a)$ over $a \in A$, *taking into account the current day's cost vector*. Prove that the average regret of this algorithm is at most $(\max_a X_a)/T$.

(d) (5 points) Prove that $\mathbf{E}[\max_a X_a] = O(\epsilon^{-1} \log n)$, where n is the number of actions.

(e) (5 points) Use (c) and (d) to prove that, for a suitable choice of ϵ, the algorithm in (b) has expected average regret $O(\sqrt{\frac{\log n}{T}})$, just like the multiplicative weights algorithm covered in class. (Make any assumptions you want about how ties between actions are broken.)