CS364A: Problem Set #2

Due in class on Thursday, November 18, 2004

Instructions:

(1) You are expected to attempt all of the following problems, except for those labeled "extra credit". If you cannot solve a problem to completion, explain your progress and where you got stuck.

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

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

Problem 1

Go to `http://theory.stanford.edu/~tim/f04/projects.html` and pick a paper-reading project—or design your own—and email the instructor with your choice, along with a project partner, if applicable. Topics are available on a first-come, first-serve basis, and you should make your choice by Monday, November 8. More detailed instructions are available at `http://theory.stanford.edu/~tim/f04/instr.html`.

Problem 2

In lecture on October 26, we briefly discussed a load-balancing game. This problem considers the following version of the game. There are n players, where player j has 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. The global objective function is the *makespan*—the total weight on the most heavily loaded machine. We will call the class of all such games the *basic KP model*.

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 *mixed strategy* for player j is a probability distribution $\{p_i^j\}_{i=1}^m$ over the machines. Players are assumed to randomize independently; for example, given a mixed strategy p^j for each player j, the probability that all players are assigned to the first machine is precisely $\prod_{j=1}^n p_1^j$. Given a mixed strategy p^j for each player j, let $L^j(p^1, p^2, \ldots, p^n)$ denote the total weight on the machine to which player j is assigned. Note this is a random variable. Let $E[L^j(p^1, p^2, \ldots, p^n)]$ denote its expectation. A *mixed-strategy Nash equilibrium* is a collection $\{p^j\}_{j=1}^n$ of mixed strategies so that for each player j and each mixed strategy $\hat{p}^j$,

$$E[L^j(p^1, \ldots, p^j, \ldots, p^n)] \leq E[L^j(p^1, \ldots, \hat{p}^j, \ldots, p^n)];$$

in words, no player can decrease the expected load of its machine by picking a different mixed strategy.

(a) Prove that a pure-strategy Nash equilibrium always exists in the basic KP model. [Problem 4(b) implies this result, so if you solve 4(b) feel free to leave this one blank.]

(b) A result from class implies that the makespan of a pure-strategy Nash equilibrium in the basic KP model is at most twice that of an optimal solution. Prove a matching lower bound. In other words, exhibit an instance (or a family of instances) in which the makespan of a pure-strategy Nash equilibrium can be arbitrarily close to twice as large as the smallest-possible makespan.

(c) Suppose there are two machines and two players who both have unit weight. Describe the Nash equilibria (both pure- and mixed-strategy).

[Hint: there are three.]

(d) Prove that once mixed-strategy Nash equilibria are taken into consideration, the price of anarchy in the basic KP model is not upper-bounded by a constant.

[Hint: if you need inspiration, look at Chapter 3 of *Randomized Algorithms*, the textbook by Motwani & Raghavan.]

[Hint: you may use without proof the following fact: there are positive constants c_1 and c_2 such that, if $n \geq 1$ balls are thrown independently and uniformly at random into n bins, then the probability that some bin receives at least $\frac{c_1 \ln n}{\ln \ln n}$ balls is at least c_2.]

Problem 3

In this problem we will consider the *restricted KP model*. Everything is the same as in the previous problem, except each player j additionally 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 basic KP model is the special case where S_j is the set of all machines for every player j.

In this problem we will consider only pure-strategy Nash equilibria, which are defined analogously to the previous problem. How much larger can the makespan of such an equilibrium be, relative to that of an optimal assignment of players to eligible machines? Prove the strongest upper and lower bounds that you can. Such bounds can be constant or functions of the number of machines and/or players.

Problem 4

In this problem we will investigate the convergence of best-response dynamics in variants of the KP model. Recall that "best-response dynamics" means that you begin at an arbitrary assignment, and as long as the current assignment is not a (pure-strategy) Nash equilibrium, an arbitrary player who can benefit by deviating is allowed to switch machines. A deviating player is assumed to switch to its best option (e.g., in the basic KP model, a machine with the lightest load).

(a) Consider a variant of the basic KP model where all players have the same (unit) weight but players can have different cost functions. Formally, each player j incurs a cost $c_i^j(k)$ on machine i if it is among k players assigned to i. Assume that for each fixed j and i, $c_i^j(k)$ is nondecreasing in k.

Solve one of the following four problems for full credit. You will receive extra credit for each problem after the first that you solve.

(i) Prove that best-response dynamics need not converge.

(ii) Prove that, despite (i), a pure-strategy Nash equilibrium always exists.

(iii) Give a polynomial-time algorithm for computing a pure-strategy Nash equilibrium.

(iv) Show that if there are only two machines, then best-response dynamics converges to a pure-strategy Nash equilibrium.

(b) Now return to the basic KP model, where all players have the same cost function but can have different weights. Consider restricting best-response dynamics so that at each step, if there are several players who want to deviate, then only the player with largest weight is allowed to deviate (ties are broken arbitrarily). Prove that this process converges to a pure-strategy Nash equilibrium after at most n iterations.

(c) **(Extra credit)** Does (unrestricted) best-response dynamics converge to a pure-strategy Nash equilibrium in the basic KP model in polynomial time?